



HAL
open science

Clinical decision support system for facial mimic rehabilitation

Tan-Nhu Nguyen

► **To cite this version:**

Tan-Nhu Nguyen. Clinical decision support system for facial mimic rehabilitation. Bioengineering. Université de Technologie de Compiègne, 2020. English. NNT : 2020COMP2590 . tel-03312827

HAL Id: tel-03312827

<https://theses.hal.science/tel-03312827>

Submitted on 2 Aug 2021

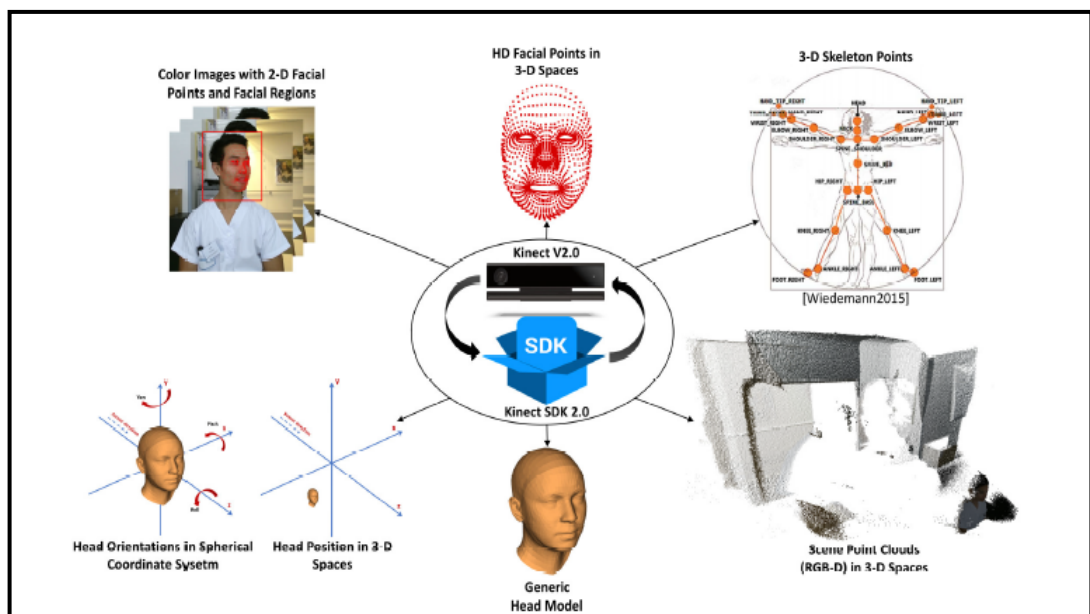
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Par Tan-Nhu NGUYEN

*Clinical decision support system for facial
 mimic rehabilitation*

Thèse présentée
 pour l'obtention du grade
 de Docteur de l'UTC



Soutenu le 19 novembre 2020

Spécialité : Biomécanique et Bio-ingénierie : Unité de
 Recherche Biomécanique et Bio-ingénierie (UMR-7338)

D2590



UNIVERSITY OF TECHNOLOGY OF COMPIÈGNE

DOCTORAL SCHOOL

Doctorate

Spécialité : Biomécanique et Bio-ingénierie

Biomechanics and Bioengineering

NGUYEN Tan-Nhu

CLINICAL DECISION SUPPORT SYSTEM FOR FACIAL MIMIC REHABILITATION

Thesis directed by Prof. DAO Tien-Tuan

Defended on 19/11/2020

Jury Members:

DAO Tien-Tuan, Professor, Biomechanics, Centrale Lille Institut (Director of the thesis)

VANDER SLOTEN Jos, Professor, Biomechanics, KU Leuven (Reporter)

PUDLO Philippe, Professor, Automatic-Biomechanics, Université de Valenciennes et du Hainaut-Cambrésis (Reporter)

BURDIN Valérie, Professor, Biomechanics, IMT Atlantique

MARQUE Catherine, Professor, Modeling of Physiological Signals, UTC-BMBI, Centre de recherche de Royallieu

HO BA THO Marie-Christine, Professor, Biomechanics, UTC-BMBI, Centre de recherche de Royallieu

DAKPÉ Stéphanie, M.D., Maxillofacial Surgery, CHU Amiens (Invited member)

Acknowledgement Integrity

This thesis could not be completed without important supports of my supervisor, Prof. Tien-Tuan Dao. I would like to give my thankfulness to his direction, advices, and motivation in researching and writing scientific papers. Moreover, his experts in biomechanics and computer science helped me in model and system developments.

I would like to send my best regards to the jury members for your interests in acceptance of evaluating my work. Your comments were valuable not only for improving the quality of my thesis and but also for developing my future careers.

I would like to appreciate the University of Technology of Compiègne and the Doctoral School for their administrative assistances. I also thank the Biomechanics and Bioengineering laboratory (BMBI – UMR CNRS 7338) for organizing professional working environments and facilitating good researching tools during my PhD period. Moreover, I wish to show my gratitude to the CHU Amiens Hospital, especially Dr. Stéphanie Dakpé, for supporting in clinical validations, medical data, and feedbacks for improving the system.

My work also received cooperation from my colleagues and my student. First, I regard Dr. Vi-Do Tran (Ho Chi Minh City University of Technology and Education) and Dr. Ho-Quang Nguyen (Thu Dau Mot University) for their helps in medical model reconstructions. I give my especial thank to Ms. Saba Ketbchi, a student from the University of Waterloo, for her co-development of the first serious game. Moreover, I would like to thank Dr. Vu-Linh Nguyen, Mr. Yonatan Carranza, Dr. Víctor Acosta Santamaria, Dr. Abbass Ballit, and Mr. Duc-Phong Nguyen for scientific discussion.

I am deeply grateful to my wife, Thi-Dung Tran, for her sacrifice in taking care of my parents and my son in Vietnam during my PhD period in France. I love you and our son.

This work was carried out and funded in the framework of the Labex MS2T. It was supported by the French Government, through the program “Investments for the future” managed by the national Agency for Research (Reference ANR-11-IDEX-004-02). I also acknowledged the Région Hauts-de-France for co-funding.

Abstract

Facial disorders negatively affect professional, social, and personal lives of involved patients. Thus, recovery of facial mimics into normal and symmetrical conditions allows these patients to improve their life qualities. Functional rehabilitation of facial disorders is an important clinical step to improve qualities of surgical interventions and drug therapies. However, facial mimic rehabilitation currently remains a major scientific, technological, and clinical challenge. Especially, conventional rehabilitation processes lack of quantitative and objective bio-feedbacks. Moreover, rehabilitation exercises just included long-term and repetitive actions. This makes patients less ambitious for completing their training programs. Besides, numerous modeling methods, interaction devices, and system architectures have been successfully employed in clinical applications, but they have not been successfully applied for facial mimic rehabilitation. Consequently, this thesis was conducted to complement these drawbacks by designing a clinical decision-support system for facial mimic rehabilitation. Especially, patient-specific models and serious games were integrated with the system for providing quantitative and objective bio-feedbacks and training motivations.

The thesis has six main contributions: (1) a novel real-time subject-specific head generation & animation systems, (2) a novel head-to-skull prediction process, (3) a muscle-oriented patient-specific facial paralysis grading system, (4) a novel serious game system for facial mimic rehabilitation, (5) a novel clinical decision-support system for facial mimic rehabilitation, and (6) a reference guide for developing real-time soft-tissues simulation systems.

This thesis opens new avenues for new research areas relating to automatic generation of patient specific head from visual sensor and internal structures using statistical shape modeling and real-time modeling and simulation for facial mimic rehabilitation.

Keywords: facial clinical-decision support systems, facial mimic rehabilitation, biomechanical head/face models, real-time head/face animations, serious games for facial mimic rehabilitation.

Table of Content

Acknowledgement Integrity	2
Abstract	3
Table of Contents	4
List of Figures	7
List of Tables.....	12
List of Abbreviations.....	13

Chapter 1: Introduction

1.1. Clinical context and needs.....	16
1.2. Objectives	20
1.3. Thesis organization.....	21

Chapter 2: State-of-the-Art

2.1. Fast generating and real-time animating subject-specific head models	23
2.2. Quantitative and objective muscle-oriented facial paralysis grading	25
2.3. Generating subject-specific skull models from surface head models.....	28
2.4. Serious games for facial mimic rehabilitation	30
2.5. The concept of system of systems (SoS) for healthcare system developments.....	32
2.6. Technology selection for the target system development.....	34
2.7. Conclusions.....	38

Chapter 3: Real-time computer vision system for tracking simultaneously subject-specific rigid head and non-rigid facial mimic movements

3.1. Materials and methods	39
3.1.1. System framework	39
3.1.2. Data acquisition subsystem.....	40
3.1.3. Subject-specific head generation subsystem.....	41
3.1.4. Subject-specific texture generation subsystem	43
3.1.5. Head animation subsystem	45
3.1.6. Mirror effect subsystem	46
3.1.7. System development technologies.....	48
3.1.8. Model accuracy and system performance analyses	48
3.2. Results.....	50
3.2.1. Model validation with MRI data in neutral position	50
3.2.2. Model validation with depth sensor data in different facial mimic positions.....	51
3.2.3. System reproducibility and repeatability test	52
3.2.4. Illumination effect analysis.....	53
3.2.5. Model generation time and visual framerates.....	53
3.2.6. A demonstration of system functionalities and animation.....	54
3.2.7. Rendering and Hausdorff distance outcomes of mirror effects	55
3.3. Discussion.....	56
3.4. Conclusions.....	62

Chapter 4: Statistic shape modeling approach for predicting subject-specific human skull from head surface

4.1. Material and methods	63
4.1.1. PLSR- and CBD-based head-to-skull generation workflow.....	63
4.1.2. Head-skull database reconstruction and transformation.....	64
4.1.3. Head-to-skull learning using the PLSR method	68
4.2. Results.....	74
4.2.1. Head and skull volumes	74

4.2.2. Hyperparameter tuning outcomes	75
4.2.3. Ten-fold cross-validation	77
4.2.4. Best and worst prediction cases	77
4.3. Discussion.....	79
4.4. Conclusions.....	81

Chapter 5: Kinect-driven patient-specific head, skull, and muscle network modelling for facial palsy patients

5.1. Material and methods	82
5.1.1. General modeling workflow	82
5.1.2. Kinect-to-head process	82
5.1.3. Head-to-skull process	83
5.1.4. Muscle network definition & generation process.....	85
5.1.5. Model validation on healthy and facial palsy subjects	86
5.2. Results.....	88
5.3. MRI-based model validation in the neutral position	88
5.4. Point cloud-based validation for different facial mimic positions.....	92
5.5. Discussion.....	94
5.6. Conclusions.....	97

Chapter 6: Serious games for facial mimic rehabilitations

6.1. Facial mimic rehabilitation exercises and serious game strategies.....	98
6.2. Material and methods	99
6.2.1. Serious game system framework	99
6.2.2. Game play workflow	100
6.2.3. Game play GUIs	100
6.2.4. Serious game 1: ‘Exploding bubbles’	101
6.2.4.1. Game level 1: ‘Smile to explode bubbles’	101
6.2.4.2. Game level 2: ‘Smile to explode left/right bubbles’	103
6.2.4.3. Game level 3: ‘Kiss to release bubbles’	104
6.2.5. Serious game 2: ‘Smile balancing’	106
6.2.5.1. Game level 1: ‘Smile to balance left & right balls’	106
6.2.5.2. Game level 2: ‘Smile to balance ball weights’	108
6.2.5.3. Game level 3: ‘Smile to lift balls’	109
6.3. Discussion and conclusions	110

Chapter 7: General discussion

7.1. Thesis overview	112
7.2. A novel real-time subject-specific head generation & animation system	114
7.3. A novel head-to-skull prediction process	115
7.4. A muscle-oriented patient-specific facial paralysis grading system.....	116
7.5. Novel serious games for facial mimic rehabilitation	116
7.6. A novel clinical decision-support system for facial mimic rehabilitation	117
7.7. A reference for developing real-time soft-tissue simulation systems.....	118
7.8. Current limitations	119

Chapter 8: Conclusions and perspective

8.1. Conclusions.....	121
8.2. Perspectives	122
8.3. Potential research recommendations	123

Publications

Journal articles.....	126
-----------------------	-----

Conferences	126
References	127
Appendices	
Appendix A: A systematic review of real-time medical simulations with soft-tissue deformation: computation approaches, interaction devices, system architectures, and clinical validations	
A1. Materials and methods	141
A1.1. Selection methodology	142
A1.2. Eligibility criteria	145
A1.3. Quality assessment.....	146
A2. Results	147
A2.1. Overall quality assessment analysis.....	147
A2.2. Computational approaches.....	148
A2.3. Model development approaches	154
A2.4. Model implementation approaches.....	158
A2.5. Interaction devices	159
A2.6. System architectures	162
A2.7. Clinical validations	164
A3. Discussion	165
A3.1. Computational approaches.....	165
A3.2. Interaction devices in real-time simulation systems.....	167
A3.3. Suitable execution schemes in simulation systems.....	168
A3.4. Clinical validations	169
A4. Conclusions	169
Appendix B: Cage-based deformation method	170
Appendix C: Graphic user interface (GUI) system	
C1. Overall system functions	171
C2. Overall system execution workflow and GUIs.....	171
C3. Biomechanical head rendering.....	174
C4. Serious game playing.....	177
C5. Mimic data & bio-feedback rendering.....	178
C6. Data capturing & recording	179
C7. System parameter setting	179
C8. User help visualizing	180
C9. Discussion and conclusions	181

List of Figures

Figure 1. Example cases of facial transplantation: (a) the first facial transplantation in the world in 2005 by Prof. B. Devauchelle (CHU Amiens, France) [14] , (b) the first facial transplantation in USA in 2008 by Dr. Maria Siemionow (Cleveland Clinic, USA) [18], and (c) the main with three faces in 2018 by Dr. Laurent Lantieri (Georges Pompidou Hospital Paris, France) [19].	17
Figure 2. Overall system development tasks: (a) subject-specific head generation and real-time animation, (b) skull model generation and muscle network definition, (c) serious game development, and (d) graphic user interface development.	21
Figure 3. Examples of simulated head/face models in literature: (a) the head model was not included skin deformations; (b) the head models were not fully simulated; (c) the head models were not subject-specific; (d) only mouth regions were animated; (e) non-subject-specific tongues were reconstructed from MRI images.	24
Figure 4. Examples of raw data and extracted facial features using different types of input interaction devices: (a) single cameras, (b) stereo cameras, (c) electromagnetic motion sensors, (d) laser scanners, and (e) RGB-D sensors.	25
Figure 5. Computer-aided non-clinical facial paralysis grading methods: (a) by analyzing asymmetries between left and right facial images, (b) by analyzing geometries of facial markers, (c) by analyzing 3-D surface difference between original models and mirrored models, (d) by analyzing internal blood flows.	26
Figure 6. Physics-based head/face modeling methods: (a) a muscle-based parameterized face model, (b) a statistics-based anthropometric face model, (c) a non-subject-specific FE-based face model without the skull layer, (d) & (e) subject-specific FE-based face model from MRI images, and (f) Mass-Spring-Damper-based face model for estimating external muscle forces from vertex displacements.	27
Figure 7. Statistics-based skull-to-head modeling methods: (a) face-skull relationship was trained through 52 landmarks on skull surfaces using PCA; (b) relationship between head and skull surface was represented through non-subject-specific 2-D tissue maps; (c) face-skull relationship was trained in different ROIs using PCA; skull-to-head relations were trained through different ROIs using PLSR; (e) tissue depth markers were used for measuring face-skull thickness using SSM.	29
Figure 8. Interaction tool usages in serious games for health up to 2013 [98].	31
Figure 9. Kinect-based serious games for (a) cognitive rehabilitation exercises, (b) virtual reality-based rehabilitation, (c) lower limb rehabilitation, (d) upper limb rehabilitation, (e) cerebral palsy rehabilitation, (f) home-based physical exercises.	31
Figure 10. Game engines for health serious game developments [98].	32
Figure 11. SoS types [154].	33
Figure 12. Data types supported by a Kinect V2.0 sensor using Kinect SDK 2.0.	36
Figure 13. SoS-based system framework of the clinical decision-support system for facial mimic rehabilitation.	37
Figure 14. The system framework of the developed real-time computer vision system for tracking simultaneously rigid head and non-rigid facial mimic movements.	40
Figure 15. Data acquired from the Kinect: (a) HD facial points, (b) the generic head model, (c) HD facial points matching with facial vertices, (d) MPEG-4 facial points, and (e) color images marked with facial regions and facial HD points.	41
Figure 16. Positional set up during the data acquisition procedure for subject-specific head generation.	42
Figure 17. Illustration of different processing steps of the subject-specific head generation procedure.	42

Figure 18. Positional set up during the data capture process for subject-specific texture generation.....	43
Figure 19. The subject-specific facial model (a) and its flattened surface (b).....	44
Figure 20. The deformed facial images before hole-filling in left (a), right (b), and front (c) views; and after hole-filling in left (d), right (e), and front (f) views.....	44
Figure 21. One-third on left (a), middle (b), and right(c) of the facial images and merging results: before boundary-removing (d), after boundary-removing (e), and texture in parameterized coordinates (f).....	45
Figure 22. Model left/right mirror effect.....	47
Figure 23. Texture left/right mirror effect.....	48
Figure 24. The transformation procedure for registering the MRI-based head model to Kinect-based head model on facial regions.....	49
Figure 25. Head reconstruction procedure from a depth sensor data: (a) scene point cloud, (b) nearest neighbor searching sphere, (c) head point-cloud and its outliers, (d) head point-cloud after outlier removal, (e) reconstructed head model from points, and (f) reconstructed head model after smoothing.....	50
Figure 26. The reconstructed head models from MRI sets in neutral position of the female subject #01 (a) and the male subject #02 (b).....	50
Figure 27. The generated head models from Kinect sensor and their Hausdorff distance maps in facial regions according to MRI-based models in neutral position of the female subject #01 (a) and the male subject #02 (b).....	51
Figure 28. The distributions of the distance maps between reconstructed MRI-based models and generated Kinect-based models of the female subject #01 (a) and the male subject #02 (b).....	51
Figure 29. Comparison between the point-cloud-based reconstructed head models and the generated head models of the male subject #02 in different facial mimics: (a) Facial images, (b) reconstructed models and generated models in the same coordinate systems, (c) generated facial models with textures, and (d) distance color maps in facial regions.....	52
Figure 30. Statistical results of errors in facial regions between the reconstructed head models and the generated head model in different facial mimic positions: neutral, smile, sounds ([e] and [u]).	52
Figure 31. The average differences from the reproducibility and repeatability test on the developed system.....	53
Figure 32. Head animations in different illumination conditions.....	53
Figure 33. Differences between two generated models using the HD facial point set and the MPEG-4 facial point set.....	54
Figure 34. Different levels of rendering qualities and their appropriate framerates.....	54
Figure 35. Different levels of rendering qualities and their appropriate framerates.....	55
Figure 36. Rigid head and non-rigid facial mimic movements without texture information (a) and with texture information (b).....	55
Figure 37. Textured face models in different rendering modes of the female facial palsy patient.....	56
Figure 38. Rendering and Hausdorff distance outcomes of the mirror effects. Note that the asymmetrical levels of the facial palsy subjects are much larger than ones of the healthy subjects.....	56
Figure 39. Schematic illustration of our proposed head-to-skull prediction workflow.....	64
Figure 40. Segmentation and surface reconstruction of the head and skull from CT images using 3D Slicer software.....	65

Figure 41. Post-processing procedure for outlier removal, head cutting, hole filling, surface reconstructing and skull cutting.....	65
Figure 42. Illustration of the head and skull model registration process: the first head/skull was registered into a global coordinate system and then all other heads/skulls were registered into the first head/skull.....	66
Figure 43. Illustration of the head/skull shapes generated from CT-based reconstructed heads/skulls.....	67
Figure 44. Head-skull surface sampling process.....	68
Figure 45. The skull model generation process from regressed skull shape: (1) registration, (2) initial deformation, and (3) refinement deformation.....	73
Figure 46. Volume distributions of all reconstructed subjects.....	74
Figure 47. Mean Hausdorff distances computed from the convergence analysis to determine the optimal number of head/skull feature points: the number of head/skull feature points was increased from 100 to 3100 features with the step-size of 100 points. Note that for each learning configurations, the mean Hausdorff distance was computed between the best regressed skull shape and its appropriate CT-based skull shape.....	75
Figure 48. RMS errors computed from the convergence analysis to determine the optimal number of components: the number of components was increased from 1 to 20 components with the step size of one component. The number of head/skull feature points were kept at the optimal value of 2300.....	76
Figure 49. Mean Hausdorff distances between the generated skull shapes and the CT-based skull shapes in different number of control points during the refinement process of skull deformation. For each number of control points, three randomly selected testing datasets were evaluated.....	76
Figure 50. Mean errors between the regressed outputs and the testing outputs from the 10-fold cross-validation. The mean errors were evaluated in different learning configurations with the optimal numbers of components and head/skull feature points.....	77
Figure 51. Comparisons of best and worst cases throughout 10-fold of cross-validations: (a) mean Hausdorff distances and (b) volume deviations between the regressed skull shapes and CT-based skull shapes.....	78
Figure 52. Comparison of the best and worst reconstructed skulls with the respective CT-based skulls from the 10-fold cross-validation: (a) best case and (b) worst case.....	78
Figure 53. The general workflow of model generations: (1) Kinect-to-head generation process, (2) head-to-skull generation process, and (3) muscle network definition & generation process.....	83
Figure 54. Outputs from the head-to-skull training procedure: (a) the reference head model (14,5420 vertices; 290,633 facets) with landmarks (left ear, right ear, forehead, top nose, and mouth center points), (b) the sampling surfaces (2,305 vertices, 4,606 facets) with sampling rays (2,305 rays), and (c) the head-to-skull model coefficient matrix (2,305 rows; 2,305 columns).....	84
Figure 55. Kinect-based head pre-processing and registration procedure. The Kinect-based head neck was pre-processed to keep only the head region and registered to the coordinate system of the reference head model.....	84
Figure 56. Kinect-based head-to-skull procedure. The Kinect-based head model was sampled to get head feature points, which was used to predict the skull shape using head-to-skull regressor. The generic skull was deformed to fit with the predicted skull shape.....	85
Figure 57. Muscle network definition: (a) selected muscles, (b) muscle insertion points on the head model, and (c) muscle attachment points on the skull model.....	86
Figure 58. Head and skull reconstruction procedures from MRI images.....	87

Figure 59. Validation procedure for Kinect-based head and skull models. The MRI-based head and skull models were transformed to the coordinate system of the Kinect-based head and skull models before drawing Hausdorff distance maps.	87
Figure 60. Perturbation positions of insertion and attachment points of the right risorius muscle on 6-mm-radius sphere: (a) 6-mm-radius radius sphere on the attachment point a, (b) the right risorius muscle based on randomly selected insertion and attachment points.	88
Figure 61. Kinect- and MRI-based reconstruction results for head, skull, skull shape, and muscle network of two healthy subjects and two facial palsy patients.	89
Figure 62. Hausdorff distance color maps between Kinect- and MRI-based head/skull models of healthy subjects and facial palsy patients.	89
Figure 63. Hausdorff distance distributions between Kinect- and MRI-based head/skull models of healthy subjects and facial palsy patients.	90
Figure 64. Validations results with point cloud-based models (color images, Kinect- Vs. point cloud-based head models, Hausdorff distance distributions, and Kinect-driven biomechanical models) in different facial mimics for two healthy subjects and two facial palsy patients: (a) 2-D color images, (b) Kinect- Vs. point cloud-based head models, (c) Hausdorff distance distributions in color maps, (d) Hausdorff distance distributions in boxplots, and (e) Kinect-driven biomechanical models.	93
Figure 65. Serious game system framework in the main system based on the concept of SoS.	99
Figure 66. Game play workflow of the serious game system.	100
Figure 67. The general game GUI.	101
Figure 68. The GUI of each game.	101
Figure 69. Interface of the “smile to explode bubbles” game: (a) game parameter interface and (b) game scene interface.	102
Figure 70. Game rule of the “smile to explode bubbles” game.	103
Figure 71. Interface of the “smile to exploded left/right bubbles” game: (a) game parameter interface and (b) game scene interface.	104
Figure 72. Game rule of the smile to exploded left/right bubbles” game.	104
Figure 73. Interface of the “kiss to release bubbles” game: (a) game parameter interface and (b) game scene interface.	105
Figure 74. Game rule of the “kiss to release bubbles” game.	106
Figure 75. Interface of the “smile to balance left & right balls” game: (a) game parameter interface and (b) game scene interface.	107
Figure 76. Game rule of the “smile to balance left & right balls” game.	107
Figure 77. Interface of the “smile to balance ball weights” game: (a) game parameter interface and (b) game scene interface.	108
Figure 78. Game rule of the “smile to balance ball weights” game.	109
Figure 79. Interface of the “smile to left balls” game: (a) game parameter interface and (b) game scene interface.	110
Figure 80. Game rule of the “smile to left balls” game.	110
Figure 81. Workflow of the selection process using PRISMA protocol for the performed systematic review.	144
Figure 82. Distribution of computational approaches (MD and MI) and associated techniques for MD approach in the literature.	149
Figure 83. Overview of all modeled soft tissues and different described behaviors for mesh-based studies.	149
Figure 84. Overview of all modeled soft tissues and different described behaviors for meshfree-based studies.	150

Figure 85. Overview of common computation strategies for mesh-based studies.	154
Figure 86. The distribution of using interaction devices in the chosen literature.....	160
Figure 87. The generic skull model and its cage: (a) generic skull model, (b) Generic skull cage, and (c) Generic skull model inside its cage.....	170
Figure 88. General execution workflow of the main system.....	173
Figure 89. Main interface of the GUI system	174
Figure 90. Submenus of the main interface: (a) file menus, (b) view menu, (c) biomechanical head rendering modes, (d) mirror effect rendering modes, (e) help menus, (f) mimic data rendering modes, and (g) biofeedback rendering modes.....	174
Figure 91. System working directory selections: (a) the “select working directory” command button, (b) the selecting/creating folder interface, (c) the structure of a working directory.	175
Figure 92. User supporting interfaces for generating new subject-specific models: (a) normal starting animation command, (b) user identification input interface, (c) folder structure of a new user directory, (d) user detection interface, (e) data capturing interface for model generations, and (f) data capturing interface for face texture generation.....	176
Figure 93. Rendering modes of biomechanical head models: (a) non-textured head model, (b) non-texture head model with internal structures, (c) textured head model, (d) textured head model with internal structures, and (e) skull model & muscle network.....	176
Figure 94. User supporting interfaces for fast aligning subject-specific models: (a) quick starting animation command, (b) user selection interface, (c) user detection interface and (d) data capturing interface for fast model alignments.	177
Figure 95. User supporting interfaces for opening static data: (a) open, first, previous, next, and last navigating buttons and (b) user & static data selection interface and for playing dynamic data: (c) play, pause, stop, and replay controlling buttons, (d) user & dynamic data selection interface, and (e) dynamic timing slider.	177
Figure 96. Serious game play interfaces: (a) serious game play command button, (b) game browser, and (c) game group interfaces.	178
Figure 97. Rendering modes of facial mimics: (a) facial mimic rendering modes, (b) high definition facial points, (c) high definition facial points with head model, (d) MPEG-4 facial points, (e) MPEG-4 facial points with head models, (f) head model with internal structures, (g) skull model with muscle network, and (h) distance color maps for left/right mirror effects.	178
Figure 98. Graphing modes of bio-feedbacks: (a) bio-feedback graphing modes, (b) MPEG-4 FAPs, (c) FACS AUs, (d) linear facial muscle lengths, (e) orbicularis horizontal and vertical muscle lengths, and (f) orbicularis muscle areas.....	179
Figure 99. Data capturing & recording interfaces: (a) data capturing & recording command buttons and (b) data organization structure of the captured & recorded data.....	179
Figure 100. System and muscle setting interfaces: (a) setting parameters for main system, (b) setting parameters for the biomechanical head rendering window, (c) setting parameters for the mimic analyzing window, (d) setting parameters for the biofeedback graphing window, (e) system & facial muscle setting command buttons, and (f) facial muscle setting interface. ...	180
Figure 101. User help interfaces: (a) MPEG-4 facial parameter definitions, (b) FACS Action Unit definitions, and (c) facial muscle definitions.	180

List of Tables

Table 1. Technological characteristics of the Microsoft Kinect V2.0 controlled by Kinect SDK 2.0.....	35
Table 2. Technology selections of the four system development aspects for developing the target system.	37
Table 3. Plausible ranges of head orientations and positions.....	55
Table 4. Comparison with existing head and facial mimic animation systems	58
Table 5. Muscle lengths of three healthy subjects and two facial palsy patients in neutral position.....	90
Table 6. Muscle strain values in smiling and [u]-pronouncing mimics of the two health subjects and two facial palsy patients.....	93
Table 7. Muscle length comparisons with reported values in related studies.....	95
Table 8. The search terms used for the systematic review process.....	142
Table 9. The number of included/excluded articles according to the selection procedure. ..	143
Table 10. The inclusion criteria for each search terminology.....	145
Table 11. Summary of the statistical results of the quality assessment procedure.....	148
Table 12. Classification of developed modelling methods for soft tissue deformations in real time: mesh-based techniques.....	150
Table 13. Classification of developed modelling methods for soft tissue deformations in real time: meshfree-based techniques.....	152
Table 14. Classification of developed modelling methods for soft tissue deformations in real time: combination-based techniques.....	154

List of Abbreviations

2-D	Two-Dimension
3-D	Three-Dimension
ANNs	Artificial Neural Networks
APIs	Application Programming Interfaces
AUs	Action Units
BEM	Boundary Element Method
BMI	Body Mass Index
CBD	Cage-Based Deformation
CDSSs	Computer-aided Decision-Support Systems
CGAL	Computational Geometry Algorithm Library
CMOS	Complementary Metal Oxide Semiconductor
CNN	Convolutional Neural Network
Covid-19	Corona Virus Disease 2019
CPD	Coherent Point Drift
CPU	Central Processing Unit
CT	Computed Tomography
DBF-FEM	Discontinuous Basic Function FEM
DICOM	Digital Imaging and Communications in Medicine
DoF	Degrees of Freedom
E+MD	Elastic-plus-Muscle-Distribution-based
EbE PCG-FEM	Element-by-element Precondition Conjugate Gradient FEM
eFace	clinician-graded electronic Facial Paralysis
EMG	Electromyography
EOMs	Extraocular Muscles
FaCE	Facial Clinometric Evaluation
FACS	Facial Action Coding System
FAPs	Facial Animation Parameters
FDI	Facial Disability Index
FE	Finite Element
FEM	Finite Element Method
FMR	Facial Motion Regression algorithm
fps	frames per second
GAN	Generative Adversarial Network
GPGPUs	General-Purpose Graphics Processing Units
GPU	Graphics Processing Unit
GUIs	Graphical User Interfaces
H1/2/3	Healthy subject 1/2/3
HD	High Definition
HDDs	Hard Disk Drives
HEML-FEM	Hyper-Elastic Mass Link for FEM
HIP	Haptic Interface Point
ICP	Iterative Closest Point
IDC	Inverse Dynamic Computation

KdTree	K-dimensional Tree
L/RB	Left/Right Buccinator
L/RCS	Left/Right Corrugator Supercilliary
L/RDAO	Left/Right Depressor Anguli Oris
L/RDLI	Left/Right Depressor Labii Inferioris
L/RFI	Left/Right Frontalis Inner
L/RFM	Left/Right Frontalis Major
L/RFO	Left/Right Frontalis Outer
L/RLAO	Left/Right Levator Anguli Oris
L/RLLS	Left/Right Levator Labii Superioris
L/RLLSAN	Left/Right Levator Labii Superioris Alaeque Nasi
L/RM	Left/Right Mentalis
L/RNa	Left/Right Nasalis
L/RO	Left/Right Orbicularis Oris Muscles
L/ROO	Left/Right Orbicularis Oculi
L/RR	Left/Right Risorius
L/RU	Left/Right Upperlip
L/RZm	Left/Right Zygomaticus Minor
L/RZM	Left/Right Zygomaticus Major
L-FEM	Linearized FEM
LSCDA	Least Square Canonical Dependency Analysis
LSD	Laplacian Surface Deformation
MD	Model Development
MI	Model Implementation
MJED-FEM	Multiplicative Jacobian Energy Decomposition FEM
MPEG-4	Moving Picture Experts Group 4
MR-FEM	Multi Rate FEM
MRI	Magnetic Resonance Imaging
MRM	Marquardt Radial basis Meshless method
MSD	Mass Spring Damper
MSM	Mass-Spring System Method
LSD	Laplacian Surface Deformation
MSR-FEM	Matrix System Reduction FEM
MSS-CS	Mass-Spring System with Corrective Springs
MTM	Mass Tensor Method
NL-FEM	Nonlinear FEM
ORM	Order Reduction Method
P1,2	Patient 1,2
PC	Personal Computer
PCA	Principal Component Analysis
PCL	Point Cloud Library
PCMFS	Point Collocation-based Method of Finite Spheres
PDEs	Partial Differential Equations
PEES	Personal Facial Expression Space method
PLSR	Partial Least Squares Regression

Pre-comp FEM	Precomputation-based FEM
Pre-Cond FEM	Precondition FEM
PSI	Particle Surface Interpolation
RBF	Radial Basic Functions
RBF	Radial Basic Functions
RGB	Red Green Blue
RGB-D	Red Green Blue-Depth
ROIs	Region of Interests
SAM	Statistical Analysis Method
SDK	Software Development Kit
SDM	Supervised Descent Method
SFGS	Sunnybrook Facial Grading System
SoS	System of Systems
SVD	Singular Value Decomposition
SVR	Support Vector Regression
TCIA	The Cancer Imaging Archive
TL	Total Lagrangian
TLED-FEM	Total Lagrangian Explicit Dynamic FEM
TL-FEM	Total Lagrangian FEM
TNN	Tan-Nhu Nguyen
ToF	Time-of-Flight
TPS	Thin-Plate-Spline
TSD	Thin-shell deformation
TSVE-Chainmail	Time-Saving Volume-Energy conserved ChainMail method
TTD	Tien-Tuan Dao
VRML	Virtual Reality Modeling Language
VRPN	Virtual Reality Peripheral Network
VTK	Visualization Tool Kit
XFEM	eXtended Finite Element Method

Chapter 1

Introduction

1.1. Clinical context and needs

Facial expressions are important to personal identity, race, and health [1], so facial paralysis negatively affects individual, social and professional life qualities of involved patients [2]. In fact, facial expressions are deformation results on facial skins caused by contraction and/or relaxation of facial muscles connecting between attachment points on the skull layer and insertion points on the skin layer [3–5]. These muscles are controlled by electrical signals generated from the seventh cranial nerve (CN VII) [6]. Dysfunction of these nerves may cause unwanted facial movements such as unnatural relaxation of eyelids, cheeks, mouth corners [7]. Moreover, patients with facial paralysis may not have symmetrical mimics when smiling, kissing, and sound pronouncing [8]. Patients can also have negative effects if they cannot express emotions using normal facial mimics [6]. Globally, approximately from 20 to 30 per 100,000 people are affected by facial nerve paralysis [9]. Facial paralysis might happen on different age ranges but more common to people between 31 and 60 years old and pregnant women [10]. Majority of facial palsy patients (~80-84%) could be recovered to their normal and symmetrical facial mimics while the others (~16-20%) were permanently affected by chronic facial paralysis [11,12].

Numerous facial palsy causes have been discovered including virus infections, trauma, tumor, strokes, and neonatal conditions [13]. Especially, one of the most particular causes was post-facial transplantations, and recovering processes were complex and needed long-term treatments [14–18]. In particular, the first facial transplantation (Figure 1a) was conducted on November 27, 2005 by Prof. Bernard Devauchelle (CHU Amiens-Picardie, France). Although the patient could have good sensitivity of heat and cold after 6 months of transplantations, the motor function was slowly recovered. She could close her mouth after 10 months of facial transplant and smile normally after 18 months of facial transplant. After 18 months of post-transplantation, the patient was satisfied with the aesthetic, functional, and psychological improvements [14]. The second case (Figure 1b) was the first face transplant in USA by Dr. Maria Siemionow (Cleveland Clinic, USA). The recovering rate of muscle motors was also slower than one of the aesthetic appearances. Excellent functional, psychological, and social outcomes of the patient were reported at 8 months after transplantation [18]. The third case (Figure 1c) was a man with three faces conducted by Dr. Laurent Lantieri (Georges Pompidou Hospital Paris, France). Although the patient could have acceptable aesthetics in three different appearances, his mouth could not be well controlled [19]. Since the first face transplant in 2005, total of 37 facial transplantations have been conducted throughout the world from 2005 to December of 2015. The age of patients were from 19 to 59 years old in which 81.1% of patients was men [16,20]. After transplantations, while the sensory nerves, which were responsible for hot, cold, light-touch, and painful sensations, could be recovered relatively fast (from 3 to 8 months of post-transplantations), the motor nerves, which help patients be able to form food boluses, swallow, and lip movements, needed much longer time to be improved. For example, the patients could only smile as late as 2 years and continued to improve their normal smiling ability for at least 8 years [16]. Especially, rehabilitation programs after face transplantations were vital to reducing the recovering time of motor functional outcomes [16].



Figure 1. Example cases of facial transplantation: (a) the first facial transplantation in the world in 2005 by Prof. B. Devauchelle (CHU Amiens, France) [14], (b) the first facial transplantation in USA in 2008 by Dr. Maria Siemionow (Cleveland Clinic, USA) [18], and (c) the man with three faces in 2018 by Dr. Laurent Lantieri (Georges Pompidou Hospital Paris, France) [19].

Facial mimic rehabilitation must be conducted in two clinical stages with both physical and psychological treatments [21]. **In the first stage**, facial paralysis degrees of facial palsy patients needed to be evaluated using clinical and/or non-clinical grading systems for personalizing a suitable rehabilitation program to each patient. *In clinical facial paralysis grading systems*, a subjective assessment procedure for facial muscle functionalities were conducted by clinicians and/or patients [22]. Numerous clinical grading scales have been applied including Sunnybrook Facial Grading System (SFGS) [23], clinician-graded electronic Facial Paralysis (eFace) [24], Facial Clinimetric Evaluation (FaCE) [25], and Facial Disability Index (FDI) [26]. The SFGS and eFace were designed for clinicians to evaluate symmetries and synkinesis pattern in static positions and dynamic movements [21,23,24]. The FaCE and FDI were developed for patients to self-report their physical and psychological conditions before and during the rehabilitation programs [21,23,24]. *In non-clinical facial paralysis grading systems*, quantitative and objective assessment methods were developed based on the support of computers for evaluating facial paralysis degrees from data acquired by electronic sensors [22]. **In the second stage**, patients were then moved to an individualized treatment program that contains five main components: (1) patient education, (2) soft-tissue mobilization, (3) functional retraining, (4) facial expression retraining, and (5) synkinesis management [21]. Note that, each patient was participated into all or some of these treatment components based on his/her facial paralysis degrees. *In patient education*, patients were educated on face anatomy: face nerves and facial muscles and their roles during facial expressions [21]. *In soft-tissue mobilization*, they might also be massaged on their affected sides by using soft-tissue mobilization techniques for improving mobilities and comfort on the affected facial muscles [27]. *In functional retraining*, the patients could also be trained for the basic oral competence referring to the maintenance of lip closure, ability of eating, drinking, and communicating [28]. *In facial expression retraining*, patients could be trained to conduct initial facial expressions such as smiling and sound-pronouncing with passive movements to improve their abilities of emotional expressions [29]. *In synkinesis management*, patients managed to control unwanted movements in relaxation positions [8].

Many challenges have been risen from all stages of facial mimic rehabilitation. **In the facial paralysis grading stage**, clinical facial paralysis grading systems are easy to use and required non expensive equipment for evaluating patients, but they are inaccurate and subjective. On the other hand, although the non-clinical facial paralysis grading systems are objective and more advantageous than the clinical ones, they have not been developed sufficiently for clinical applications [22]. In particular, most non-clinical grading systems were not accurate, non-real time, and needed expensive devices for first-time facilities. In fact, these systems should be easy to use, non-expensive, real-time, accurate, mobile, and multi-functional [22]. **In the treatment stage**, most treatment components require patients to meet their therapists directly.

This could be inconvenient for such a long-term procedure of facial mimic rehabilitation. Beside the physical aspect, the psychological aspect is also important to facial mimic rehabilitation [21]. Current rehabilitation treatments just included repetitive training exercises for intensifying specific movements of appropriate facial muscles [8]. Patients could be less motivative during these long-term and repetitive rehabilitation exercises. This could be especially true for pediatric facial palsy rehabilitation in which the strategies and delivery of treatments need to be more child-friendly and interesting [21,30]. Computer-based serious games coupled with interactive simulations for facial mimic rehabilitation should be one of the best solutions.

A computer-aided clinical decision-support system for facial mimic rehabilitation is necessary for complementing above drawbacks risen from the current facial mimic rehabilitation programs. An objective facial paralysis grading system with abilities of fast generating subject-specific models and real-time tracking facial mimics is the first requirement in this system. During external facial mimics, internal facial muscle activities and facial symmetries should also be computed to give external and internal bio-feedbacks for real-time facial paralysis grading. Moreover, serious games for facial mimic rehabilitation should be included in this system for providing positive psychological effects to facial palsy patients during their rehabilitation exercises. Last but not least, user-friendly, multifunctional, and interactive graphical user interfaces (GUIs) should be designed for effectively interacting between user commands and system functions. A modern and powerful system development approach is necessary for designing such complex interactive, and multifunctional interfaces. Overall, fulfilling all above requirements is a major scientific, technological, and clinical challenge.

Fast generating and real-time animating subject-specific head models are primarily required when developing a real-time clinical-decision support system for facial mimic rehabilitations. The simulated head model must be fast personalized to the target patient to well describe facial appearances in different facial mimics [31,32]. Moreover, head rigid and facial non-rigid movements must also be tracked and animated on the simulated model in real-time for analyzing dynamic facial mimics [33]. Facial textures also play an important role for increasing the subject-specific level of the model [34–36]. In literature, numerous computation methods have been proposed for simulating facial movements, but they hardly archived real-time framerates [36–52], full head animations [32,35–38,40–44,48,49,51–53], and subject-specific geometries [31,45,46,50,54,55]. Moreover, different types of interaction devices, including camera-based [37–40,42–53], stereo-camera based [54], electromagnetic-based [56], 3-D scanner-based [31,32], and red-green-blue-depth (RGB-D) camera-based [36,41] devices, have been employed for tracking face and head movements. However, facial features were only extracted from raw data of these interaction devices using complex 2-D/3-D computer vision methods, so real-time framerates and realistic animations could hardly be satisfied. Last but not least, facial textures were rarely included in simulated head/face models [31,32,40,41,44,52], or they had to be manually facilitated in other system platforms [34–36]. This hardly applied to fast generate face textures for new patients. Consequently, a system that can combine subject-specific model & texture generation and real-time model animations into a system execution scheme is original and challenging for being developed.

Quantitative and objective muscle-oriented facial paralysis grading is also novel and challenged for directly analyzing facial muscle activations during facial mimics. In literature, most computer-aided facial paralysis grading systems just analyzed 2-D/3-D external facial appearances/movements acquired by visual sensors [57–68]. These visual sensors could only acquire exteriors of the target objects, so internal structures could not be directly measured. Although CT/MRI scanners can capture both exteriors and interiors of the target objects as 2-D slices, 3-D model reconstructions and physics modeling are time consuming and need clinical experts. This is hardly applied for generating subject-specific models for new users and

computing internal muscle activations during real-time facial mimics. Numerous physics-based head/face modeling methods have been developed for computing facial movements on the skin layer from muscle activation, but they were not real-time and have not computed internal muscle actions from external facial movements [3–5,69–72]. Consequently, non-clinical muscle-oriented facial paralysis grading is original and essential for quantitative and objective evaluation of facial mimics. This function requires a novel modeling method that can fast generate patient-specific head, skull, and muscle network. Moreover, external and internal bio-feedbacks (e.g. face asymmetries and muscle strains) should be computed in real-time during facial movements.

Generating subject-specific skull models from surface head models is required for modeling subject-specific biomechanical head models when only external head surface models are available. Although skulls play important roles in forming face shapes during facial mimics [73–78], they were rarely included in physics-based face models [3,70–72] or approximated to target models using affine transforms [69]. This was because most employed visual sensors can only measure exterior structures/appearances of target objects. In literature, skull models are often reconstructed from medical imaging like MRI/CT techniques. However, reconstruction procedure including segmentation, voxelization, and mesh generation is time consuming even in automatic procedures [79,80]. On the other hand, head surface models could be fast reconstructed using visual sensors such as single images [51], laser-scanners[32,53], and Microsoft Kinect sensors [35], but these sensors cannot acquire internal structures. Consequently, a method that can generate internal skull structures based on head surface models is essential. Currently, no studies have been developed for resolving this issue. Some studies just tried to estimate face models from given skull models in forensic facial reconstructions [81–89]. In these studies, several statistics-based methods have been successfully applied for learning skull-to-face relationships on face-skull datasets reconstructed from MRI/CT images, but they have not been applied for learning head-to-skull relations. Consequently, head-to-skull relation should be studied using statistics-based methods so that internal skull structures can be predicted based on head surface models. Subject-specific skull structures will also help increase accuracies of muscle-oriented facial paralysis grading systems.

Serious games for facial mimic rehabilitation are also recommended for providing training motivations during rehabilitation exercises. Serious games were shown positive effects for improving treatment/rehabilitation results [90–92]. Although numerous serious games have been developed for different rehabilitation applications such as upper-limb stroke rehabilitation [93,94], trauma bones and soft-tissue injuries [95], traumatic brain injuries [96], and home-based stroke rehabilitation [97], no serious games have been developed for facial mimic rehabilitation. This was because of the lacks of tracking devices (interaction tools) and modeling method that can generate and animate subject-specific head model in real-time. Nowadays, numerous high-level tracking devices such as Wii Peripheral, Kinect, Camera, and BioFeedBack have been used for serious games [98], but they have rarely been applied for facial applications. Moreover, many game engines have been available for game developments in various platforms such as Windows, Linex, iOS, and Android [99]. These multifunctional game engines could help reduce time-to-market and increase system reliability in game developments. However, these game engines are specified for entertainment purposes, so accurate physical simulations cannot be supported. A flexible game engine that has ability of combining graphical, physical, and user interface packages into a single system might be essential for medical serious game developments. Moreover, most game characters were crafted by game developers using graphical design tools, so these game characters could not be subject-specific. In fact, subject-specific models are important to treatment/rehabilitation outcomes [100]. Consequently, serious games coupled with subject-specific character models are original and desirable for facial mimic rehabilitation.

The concept of System of Systems (SoS) for healthcare system developments is novel and necessary for developing such complex and multifunctional computer-aided decision-support

systems in which interactions between living body systems and related technological systems need to be well-managed [101]. However, many challenges have been risen when applying this novel concept for developing computer-aided healthcare systems [102]. These challenges related to computational approaches, interaction devices, system frameworks, and clinical validations [103]. For computational approaches, in SoS-based systems, computational methods in each sub-system need to be optimized according to requirements of the main system purpose. About interaction devices, most healthcare systems require multiple types of data to be acquired in real-time, so stable data transmission between interaction devices and the main system are important to maintain good working condition for the main system. About system frameworks, all sub-systems need to be executed parallelly in multiple hardware platforms or multiple computational threads, so main systems must well manage interactions among sub-systems for keeping stable working conditions and reducing latency among sub-systems. For clinical validations, multi-level verification and validation procedures must be conducted for all sub-systems and main systems in both short-term and long-term evaluation campaigns. A standard medical software development and maintenance processes [104] should also be followed for covering these requirements.

Overall, facial palsy negatively affects both personal and professional lives of involved patients. Although facial mimic rehabilitation can effectively help recover these mimics into their normal and symmetrical movements, many challenges and drawbacks have been risen from all stages of recovering processes. A computer-aided clinical decision-support system coupled with quantitative & objective bio-feedbacks and serious games for facial mimic rehabilitation should be one of the best solutions. However, development of such a real-time and multifunctional system is a major scientific, clinical, and technological challenge.

1.2. Objectives

This thesis was conducted to answer two research questions: (1) how to help patients improve both psychological and physical effects of conventional facial mimic rehabilitation programs and (2) how to help clinicians supervise the rehabilitation progresses. To answer these questions, a computer-aided clinical decision support system coupled with quantitative & objective bio-feedbacks and serious games for facial mimic rehabilitation was developed. In particular, a subject-specific meta head model including head, skull, and muscle network was generated from subject-specific data acquired from a Microsoft Kinect sensor. This meta model was animated according to facial movements to compute external and internal bio-feedbacks. Moreover, two serious games with three difficulty levels for each game were designed for motivating patients during their rehabilitation exercises. A graphic user interface was finally designed based on the SoS concept for managing complex interactions between users and system functions.

The system was developed in four system development tasks (Figure 2). **In the first system development task** (Figure 2a), I generated the subject/patient-specific head model and face texture of the target user from the subject/patient-specific data acquired from the Microsoft Kinect V2.0 [105]. After generation, his/her current facial mimics could be tracked and animated on his/her textured head model in real-time. **In the second system development task** (Figure 2b), his/her skull model was regressed based on the generated head model, and the muscle network between the head and skull models could also be estimated based on the pre-definition of facial muscle structures. Based on the subject-specific head, skull, and muscle network, muscle actions and other standard facial mimic parameters could be computed according to real-time head animations. **In the third system development task** (Figure 2c), the Kinect-driven biomechanical head model was used as one of main game objects. Two serious games with three different game level each were designed for training smiling and

kissing mimics. **In the final system development task** (Figure 2d), the GUI system was designed for interfacing between all system functions and user commands. All system development tasks were developed based on the IEC 62304:2006+A1:2015(E) standard of medical device software – software life-cycle processes [104].

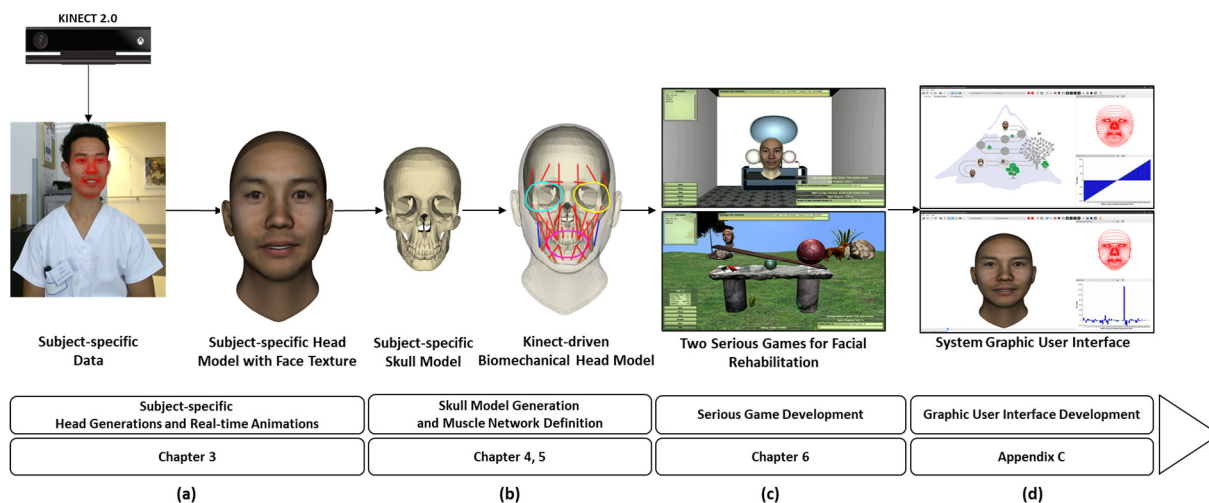


Figure 2. Overall system development tasks: (a) subject-specific head generation and real-time animation, (b) skull model generation and muscle network definition, (c) serious game development, and (d) graphic user interface development.

1.3. Thesis organization

This thesis is organized as the followings.

- **Chapter 1: Introduction.** In Chapter 1, I introduced clinical context and needs for developing a novel clinical decision-support system for facial mimic rehabilitation. Objectives and system development tasks were also presented.
- **Chapter 2: State-of-the-art.** In Chapter 2, all scientific, clinical, and technological challenges risen from the development of the target system are presented. Moreover, technology selections for developing the target system will also be proposed.
- **Chapter 3: Real-time Computer Vision System for Tracking Simultaneously Subject Specific Rigid Head and Non-rigid Facial Mimic Movement.** Realistic subject-specific head animation is primarily required for developing a computer-aided facial paralysis grading system. In Chapter 3, a complete procedure of generating and animating subject-specific head models with face texture will be explained in detail. Model validations in neutral & mimic positions and comparisons with related studies will also be reported.
- **Chapter 4: Statistical Shape Modeling Approach for Predicting Subject-Specific Human Skull from Head Surface.** Skull models also need to be subject-specific for estimating muscle actions during facial animations. In Chapter 4, a novel statistics-based head-to-skull training and regressing processes will be presented. Accuracy of the regressed skulls based on hyper parameter tuning and cross-validation procedures will also be reported.
- **Chapter 5: Kinect-driven Patient-specific Head, Skull, and Muscle Network Modelling for Facial Palsy Patients.** Muscle-oriented facial mimic evaluation is a direct and effective approach for facial paralysis grading systems. In Chapter 5, I will describe a complete procedure of generating patient-specific biomechanical head models from data acquired from the Kinect V2.0 sensor. Validation of model geometrical accuracy and muscle actions with healthy and facial palsy subjects will also be analyzed in this chapter.
- **Chapter 6: Serious Games for Facial Mimics Rehabilitations.** Serious games showed positive effects for motivating patients during their rehabilitation exercises. In Chapter

6, I will describe two serious game specifically designed for training smiling and kissing mimics using the Kinect-driven biomechanical head model.

- **Chapter 7: *General Discussion*.** In Chapter 7, an overview of the thesis will be first stated, and then main contributions of the thesis will also be discussed. Moreover, current system limitations will be indicated.
- **Chapter 8: *Conclusions and Perspectives*.** In Chapter 8, conclusions of the thesis contents will be first presented. Then, more system development tasks will be proposed for solving the current limitations. Moreover, potential research recommendations will be proposed.
- **Appendix A: *A systematic review of real-time medical simulations with soft-tissue deformation: computation approaches, interaction devices, system architectures, and clinical validations*.** Facial mimics are formed by complex soft-tissue deformations on facial skins, and developments of real-time soft-tissue simulation systems are clinical, scientific, and technological challenges. In Appendix A, I will present a systematic review that was conducted to understand how these systems have been developed in literature. This review was used as a reference for developing the target system and other real-time soft-tissue simulation systems.
- **Appendix B: *Cage-based deformation method*.** The cage-based deformation method is one of the well-known geometrical deformation for 3-D models. This method was effectively applied for subject-specific skull generation. It will be briefly presented in Appendix B.
- **Appendix C: *Graphic user interface (GUI) system*.** Graphic user interfaces are also important to interacting between user commands and system functions. In Appendix C, all system functions with their appropriate graphic user interfaces in the GUI system will be described in details.

Chapter 2:

State-of-the-Art

Developing the computer-aided clinical decision-support system for facial mimic rehabilitation is a scientific, technological, and clinical challenge. This challenge is risen from all system development tasks of the target system. First, fast generating and real-time animating subject-specific head models are primarily required, these functions have not been solved in literature due to limitations of computational approaches and interaction devices. Second, facial mimics are resultants of muscle activations on facial skins, so muscle actions should be directly estimated during external facial movements for facial paralysis grading. However, this issue has not been resolved in literature due to lack of internal structures being able to be acquired in real-time using visual sensors. Third, although skulls are important to forming facial appearances during facial mimics, they were rarely included in biomechanical head models. Even if they could be accurately reconstructed from CT/MRI images, the CT/MRI data acquisition and model reconstruction processes are expensive and time-consuming. Fourth, serious games are necessary for improving training motivation during rehabilitation exercises, but no serious games have been developed for facial mimic rehabilitation. This is because of the lacks of tracking devices and modeling methods that can capture subject-specific facial movements in real-time. Finally, interactive interfaces between users and system functions are also important to a clinical decision-support system. The concept of System of Systems (SoS) was proposed to manage this complex interaction, but the employment of this concept for healthcare system developments is also challenging. This chapter will review in details each of above challenges in literature. Moreover, developments of such a real-time medical system for facial mimic rehabilitation need compromising selections of four system development aspects: computational approaches, interaction devices, system frameworks, and clinical validation campaigns. Technology selection for developing the target system was also proposed based on a systematic review on those system development aspects.

2.1. Fast generating and real-time animating subject-specific head models

Fast generating and real-time animating subject-specific face movements are primarily required in a clinical-decision support system for facial mimic rehabilitation, especially for real-time facial paralysis grading. Many methods have been proposed for simulating these complex movements, but numerous challenges have not been solved thoroughly when trying to achieve simultaneously rigid head and non-rigid facial mimic animations in 3-D spaces and in a subject-specific manner with a real-time framerate [106]. Generally speaking, head rigid movements are composed of translations and rotations while non-rigid facial mimic movements illustrate scaling and deformation effects. While rigid animations are easily computed given rotation angles and translation vectors, the non-rigid cases are relatively difficult to simulate due to non-linear deformations of simulated subjects. Consequently, the non-rigid geometries often required much more computation costs than the rigid ones. In particular, to achieve realistic animations, computation speed of both rigid and non-rigid animations must reach real-time framerates. Note that the real-time rates are equal or larger than the graphic animation rate of 30 frames per second (fps) [107].

Numerous computational methods have been proposed for simulating head and facial expression animations, but they hardly accomplished real-time framerate and realistic animations. Examples of simulated head/face models are shown in Figure 3. Specifically, most studies could not actually simulate full rigid head and non-rigid facial animations in real-time [36–52]. Precisely, even though the framerates could be reached in real-time, head models did not include skin deformations [53] (Figure 3a), full head models [53] (Figure 3b), subject-specific models [54,55] (Figure 3c), and fully facial animations [56] (Figure 3d). Moreover, in these studies the procedure of generating subject-specific models was not integrated into an

animation system, or the animated models were not subject-specific. Additionally, most studies just simulated non-rigid animations of facial regions, but back-head regions were discarded or lacked [32,35–38,40–44,48,49,51–53] (Figure 3b). In fact, although full head models could be simulated, they were not subject-specific [31,45,46,50,54,55] (Figure 3c). Finally, internal structures such as tongues and teeth were also included into head models, but they were reconstructed from Magnetic Resonance Imaging (MRI) images and cannot be fast personalized to a new user [39] (Figure 3e).

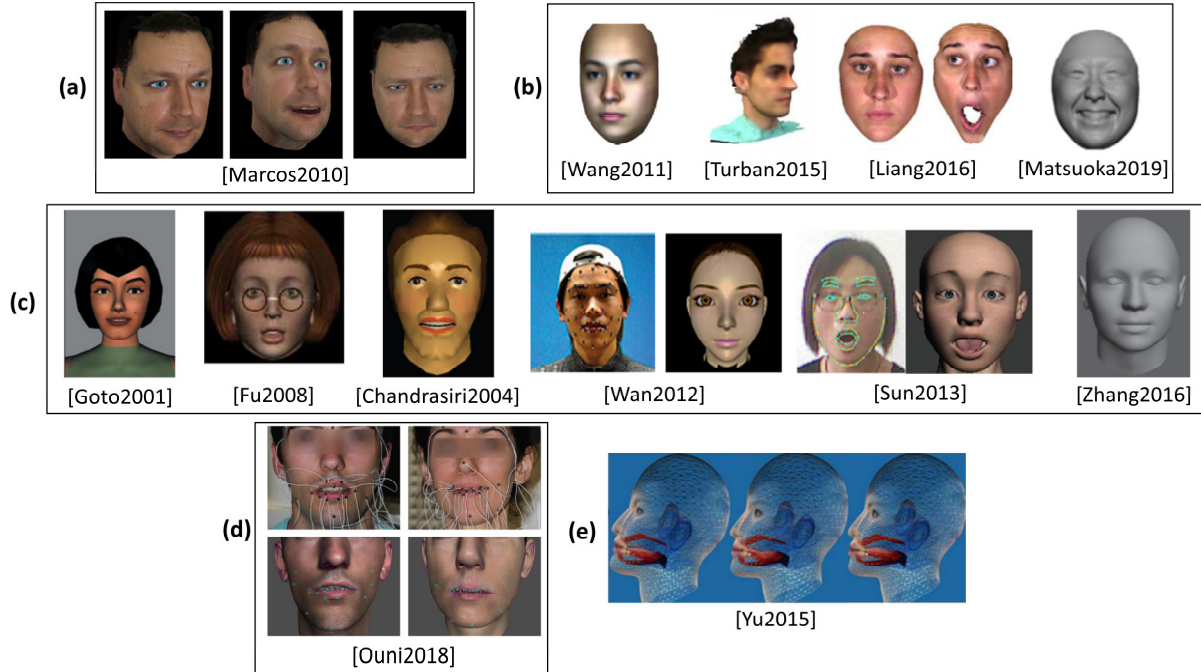


Figure 3. Examples of simulated head/face models in literature: (a) the head model was not included skin deformations; (b) the head models were not fully simulated; (c) the head models were not subject-specific; (d) only mouth regions were animated; (e) non-subject-specific tongues were reconstructed from MRI images.

Input interaction devices also have their significant contributions to both system accuracy and system framerates [103]. Based on specific types of input data acquired from the input devices, studies must choose appropriate reconstruction methods for generating subject-specific models. During animations, subject-specific data must be detected and tracked to animate the subject-specific models, so the data acquisition bandwidths of input devices also notably affect system framerates. Current interaction devices can be classified into single camera-based, stereo-camera based, electromagnetic-based, 3-D scanner-based, and red-green-blue-depth (RGB-D) camera-based devices. Examples of raw data and their extracted facial features using different types of input interaction devices are shown in Figure 4. Single cameras were mostly employed in animation systems due to fast data acquisition speeds, low-costs, and widespread popularities [37–40,42–53]. **However, due to lack of 3-D data from a 2-D single image, more processing procedures were needed to reconstruct 3-D data from 2-D data.** Consequently, **most studies using single cameras hardly achieve real-time framerates** (Figure 4a). Stereo cameras were also be utilized in reconstructing and animating human heads in 3-D spaces [54]. Due to large computation cost on processing 2-D images, 3-D motion estimations were simplified by using facial markers. **However, long set-up time and the limited number of facial markers made facial-marker-based systems hard to increase animation quality and implement to new users in a subject-specific manner** (Figure 4b). Electromagnetic sensors could also be used as facial markers to capture facial motions [56]. **Despite of high motion accuracy and data acquisition bandwidth, the sensor’s wires and the limited number of sensor channels were the main drawbacks when using this type of sensors** (Figure 4c). Additionally, single-camera-based and stereo-camera-based systems were negatively affected

by illumination conditions. Being able to acquire 3-D data directly with high-accuracy and without depending on light conditions, 3-D laser scanners were also used [31,32] for modeling facial mimics, **but they lacked of texture information** (Figure 4d). In addition, RGB-D cameras can acquired both 3-D point clouds and texture images [36,41]. **For facial applications, the Kinect and Kinect Software Development Kit (SDK) are popularly used. Despite of the Kinect's suitability in facial applications [108], most studies just used them for facial recognition applications [109–115].** Some studies also used the Kinect for reconstructing facial expressions [35,36], **but they just generated subject-specific facial models and could not animate these models in real-time** (Figure 4e). Even though the framerates could be reached in real-time, the animated model was not subject-specific [55]. Although texture mapping was also important to increase the subject-specific level of simulated models, not all studies included textures on their generated models [31,32,40,41,44,52]. **Even the textures were integrated, their generation procedures were not included in the system procedures [34–36].** A classical texture mapping procedure was proposed by Lee et al., 2000 [34]. However, **no studies have taken advantages of high-level data from Kinect camera for animating subject-specific rigid head and non-rigid facial mimic movements with texture information.**

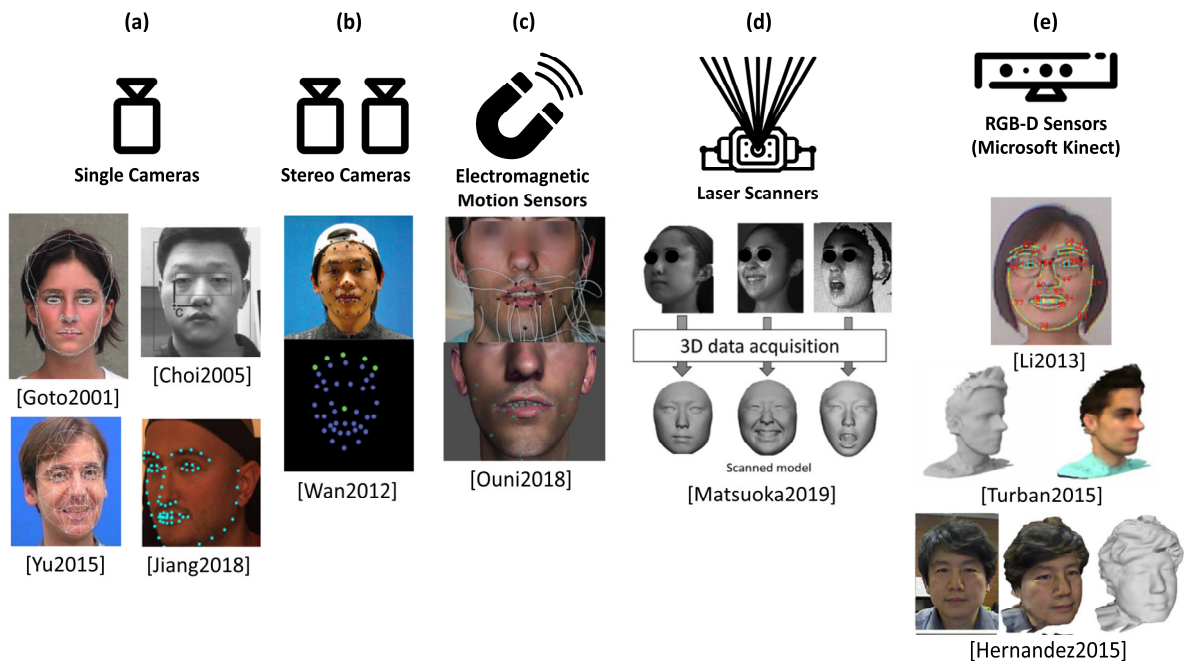


Figure 4. Examples of raw data and extracted facial features using different types of input interaction devices: (a) single cameras, (b) stereo cameras, (c) electromagnetic motion sensors, (d) laser scanners, and (e) RGB-D sensors.

Overall, subject-specific head model generations coupled with real-time animations are first required for developing a clinical-decision support system for facial mimic rehabilitation, especially for facial paralysis grading functions. However, these tasks are rarely accomplished in literature due to high computational cost for computing non-rigid facial animation. **These challenges should be solved with supports of high-level visual sensors (e.g. the Microsoft Kinect V2.0 sensor).**

2.2. Quantitative and objective muscle-oriented facial paralysis grading

Numerous non-clinical facial grading systems, which are quantitative and objective, have been developed, but they did not directly analyze facial muscle contraction due to lacks of patient-specific internal head anatomic structures (Figure 5). In these grading systems, facial paralysis degrees were mostly measured through evaluating geometrical asymmetries

between left and right facial appearances and/or movements computed from 2-D images and/or 3-D point clouds [57,61–68,116,117]. For instances, intensity differences between left and right facial images could be used for grading facial paralysis in facial expressions [57] (Figure 5a). Especially, geometrical motions of facial features/landmarks were mostly used. In particular, the symmetries of locations and displacements of facial features were used to evaluate the unilateral peripheral facial paralysis [65,66] (Figure 5b). Besides, landmark displacements, inter-landmark distances, landmark bounding areas, landmark motion velocities could also be employed for evaluating degrees of facial palsy [66,67,117] (Figure 5b). Moreover, facial feature movements could be combined with facial textures in appropriate region of interests for estimating basic facial action units (AUs) defined in the Facial Action Coding System (FACS) [118]. Facial palsy levels could be evaluated through analyzing the computed AUs in time serial data [61]. 3-D surface differences between 3-D face models and their horizontally mirrored models could also be used for computing facial asymmetries [60,119] (Figure 5c). Other than that, internal blood flows measured by laser contrast imaging technique in different facial regions could be used for quantifying the severity of facial palsy patients [120] (Figure 5d). Overall, external geometrical factors of facial appearances and/or movements have mostly been taken advantages for computing degrees of facial palsy. Actually, facial mimics are deformation results of facial muscle activation on the facial skin [3–5], so facial muscle activation should be directly analyzed. However, most visual sensors such as cameras, laser scanners, and infrared sensors can only capture exterior shapes/textures of subjects. Although some interior scanning devices such as sonar scanners and Computed Tomography (CT)/MRI scanners can accurately acquire 2-D slices of internal structures, 3-D reconstruction procedures are slow and need numerous manual processing [121]. Consequently, a modeling method that can take advantages of real-time exterior facial mimics for predicting in-vivo internal facial muscle actions is essential for facial paralysis grading applications, and the prediction process should be based on the external-internal relationship trained by accurate CT/MRI-based 3-D data.

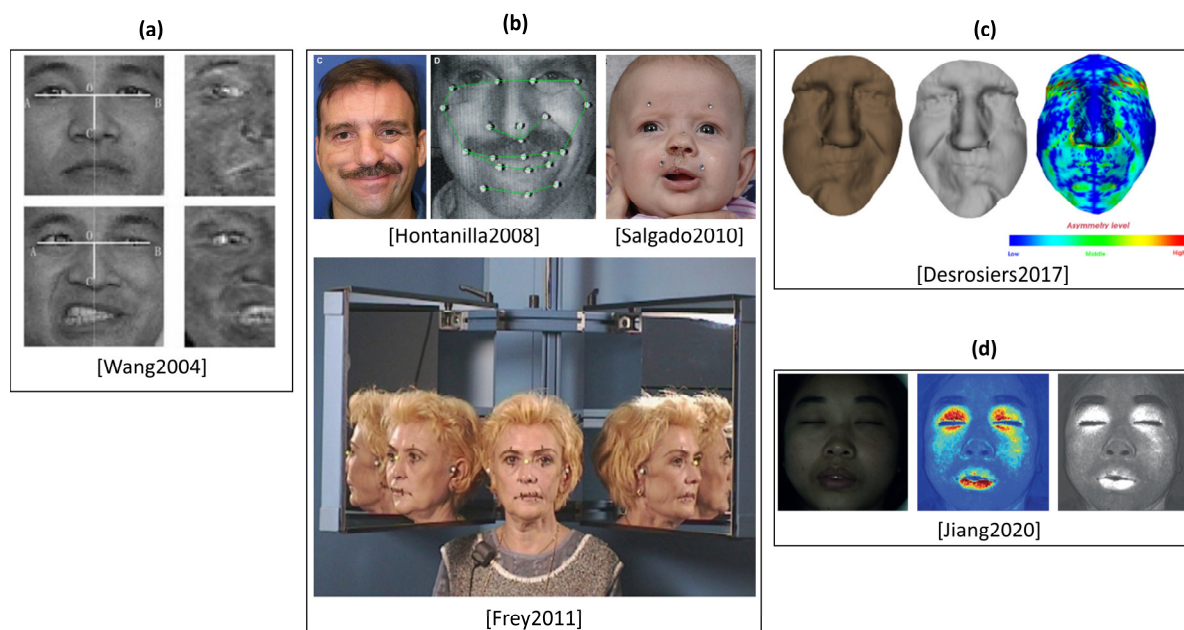


Figure 5. Computer-aided non-clinical facial paralysis grading methods: (a) by analyzing asymmetries between left and right facial images, (b) by analyzing geometries of facial markers, (c) by analyzing 3-D surface difference between original models and mirrored models, (d) by analyzing internal blood flows.

Physics-based head/face modeling methods have been widely developed in literature, but they mostly estimated facial animations from muscle activations and have not extracted internal muscle actions from external facial mimics (Figure 6). For instances, a muscle-based parameterized face model was introduced by King et al., 2005 [69]. The face model

including movable lips and a tongue was modeled as B-spline surfaces. Facial mimics were generated by moving control points of the B-spline models in lip regions (Figure 6a). Moreover, a constrained 3-D active appearance modeling methods was applied for modelling 3-D anthropometric-muscle-based face model. This model could be deformed to fit with different face-types and facial expressions by adjusting the motivated muscle actuators and statistics-based anthropometric controls [70,71] (Figure 6b). The finite element (FE) modeling method was also employed for modeling different types of facial muscles. From muscle activations, a FE-based face model including multiple layers could be deformed to generate realistic facial expressions on skin layers [3] (Figure 6c). However, although skull structures had major contributions to facial mimics [73–77], they were not included in the simulated models [3,70,71] or just approximated using affine transforms [69]. Subject-specific biomechanical head models with heads, skulls, and muscle networks were reconstructed from MRI images and simulated using the FE method in the studies of Fan et al., 2017 [4] and Dao et al., 2018 [5] (Figure 6d&e). Note that computational cost is very expensive for such complex models. In particular, MRI data acquisition, 3-D model reconstruction, and FE model generation procedures were time consuming and needed much clinical expertise, so the whole subject-specific model generation could not be fast applied to new subjects. Recently, few studies have also tried to estimate muscle feedback from exterior facial movements. Particularly, using the Mass-Spring-Damper (MSD) method for modeling a 3-D face model, external muscle forces affected on facial skins could be estimated from 3-D facial vertex displacements in the study of Erkoç et al., 2018 [72] (Figure 6f). However, the face model in this study did not include skull layers, and internal facial muscle actions, were not directly computed. Moreover, modeling subject-specific face models from 2-D images were not accurate, and facial features were not automatically extracted from 2-D images. In fact, extracting and tracking facial muscle activation need full subject-specific biomechanical head model including head, skull, and muscle network cooperating with real-time tracking facial features in 3-D spaces.

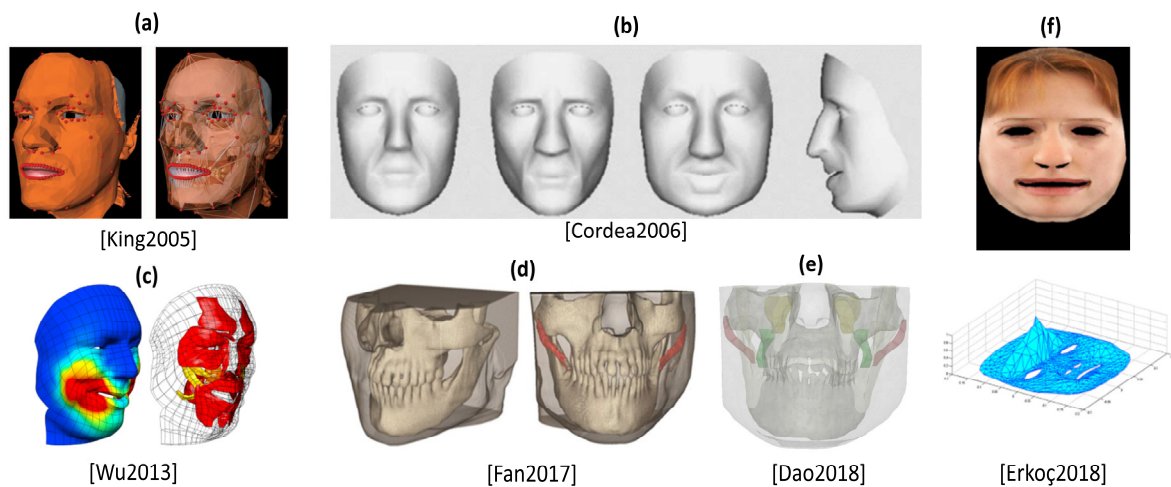


Figure 6. *Physics-based head/face modeling methods: (a) a muscle-based parameterized face model, (b) a statistics-based anthropometric face model, (c) a non-subject-specific FE-based face model without the skull layer, (d) & (e) subject-specific FE-based face model from MRI images, and (f) Mass-Spring-Damper-based face model for estimating external muscle forces from vertex displacements.*

Overall, non-clinical muscle-oriented facial paralysis grading is original and essential for quantitative and objective evaluations of facial mimics. **These systems required a novel modeling method that can fast generate patient-specific head, skull, and muscle network.** Moreover, **based on the patient-specific biomechanical head models, external and internal bio-feedbacks (e.g. face asymmetries and muscle strains) should be computed in real-time during facial movements.**

2.3. Generating subject-specific skull models from surface head models

Human skull is an important body structure for jaw movement and facial mimic simulations. In particular, this structure highly affects the accuracy of the model development and simulation outcomes [73–76]. In fact, an accurate skull model can generate most realistic animations based on physical deformations and interactions among biological structures inside a head model [77]. Note that the internal structures of a head model are composed of a skin layer, a fat-and-muscle layer, and underlying bone structures [74]. Skull is one of these important structures in head. This structure supports the facial structures [76] and protects the brain. This structure also allows generating facial mimics movements based on muscle contractions [73]. Thus, the accuracy of the skull model is of great important to determine both geometrical and mechanical characteristics of skin and muscle behaviors [74]. Moreover, the shape of mandible structure highly affects the jaw movements in facial mimics [78]. Consequently, to realistically simulate jaw movements and facial mimic deformation, skull models must be accurately developed in a subject-specific manner.

While surface head models can be easily reconstructed by portable sensors such as 3-D scanners or 2-D cameras, internal skull models are hard and time-consuming to be reconstructed. In particular, 3-D head models can be accurately reconstructed using single images [51], laser-scanners [32,53], and Microsoft Kinect sensors [35]. However, due to the technological limitations of these sensors, skull models were rarely added inside these head models. From biomechanical point of view, skull models are often reconstructed from medical imaging like MRI/CT. However, the reconstruction procedure including segmentation, reconstruction, and meshing is time-consuming even with some automatic procedures [79,80]. Moreover, the acquisitions of MRI/CT images are very slow, expensive, and harmful. In fact, medical imaging techniques are not suitable for real-time computer-aided simulation systems in which a new subject-specific skull model needs to be generated in runtime. In the literature, only one study tried to estimate and generate skull models from outside head models, but the estimated skulls were incomplete [74]. The authors developed a facial animation system in which the skull model was used to attach the skin through the muscle layers. The skull model was approximated to fit with the head model using affine transformations. However, the accuracy of the generated skull model was not validated according to the ground-truth data.

Besides, head-skull relationship has been commonly studied in forensic facial reconstructions from given skull model, but they have not been applied for predicting skull models from given head models. 3-D facial/head model of unknown people is reconstructed from the remaining skull. Numerous methods have been used to learn the skull-to-face relations, as illustrated in Figure 7. Claes et al., 2008 [81] applied the Principal Component Analysis (PCA) to learn relationship between skull and facial surfaces through 52 artificial landmarks manually put on both skulls and facial regions of 118 face-skull datasets reconstructed from CT images. A Thin-Plate-Spline (TPS) facial model was deformed to new facial landmarks' positions computed from the trained skull-to-face model (Figure 7a). Liang et al., 2009 [82] also used the PCA to train head-skull surface relations with artificial landmarks as in the study of Claes et al., 2008 [81], but the Radial Basic Functions (RBF) were used to deform the face model. However, these studies did not study directly skull-face thickness distributions, and only facial region was analyzed (Figure 7a). A tissue-map technique was then presented by Yuru et al., 2008 [83]. In this study, one head and skull dataset reconstructed from CT images was unwrapped to the 2-D map. The relationship between head and skull surfaces was represented as distances between head and skull surfaces on the 2-D image called a tissue map. As a result, given a new skull surface model, the tissue map could be used to infer the new face surface model. However, the tissue map was not representative to a population and could not be used to predict new face model in a subject or patient specific manner (Figure 7b). The PCA method was also applied by Zhang et al., 2010 [84], but each of 25 face-skull datasets

reconstructed from CT images was divided into 5 region of interests (ROIs) for training task. In this study, only the facial regions were estimated, and the merging outcome of the ROIs was not smooth (Figure 7c). A larger dataset (>200 subjects) from CT images with different ages, regions, and genders was used for training the PCA model, and the TPS modeling method was also used for deforming a generic facial model to fit the estimated data from the PCA model [85], [86]. Also using the TPS for deforming the generic facial model to the target facial landmarks, but Duan et al., 2015 [87] applied the Partial Least Squares Regression (PLSR) method for learning face-skull surfaces in 5 facial ROIs in each of 96 pairs of skulls and faces (Figure 7d). Moreover, the Least Square Canonical Dependency Analysis (LSCDA) could also be combined with the PCA method to select the most dependency of shape parameters of skulls and skins [88]. However, all above studies did not directly analyze the distributions of skull-face thicknesses. Madsen et al., 2018, [89] used tissue depth markers placed on skull surfaces to measure skull-face thickness in 9 head-skull datasets reconstructed from MRI images. In this study, a statistical shape model (SSM) was used to select the most appropriate head models over 300 3-D head datasets given a skull and its tissue-depth data (Figure 7e).

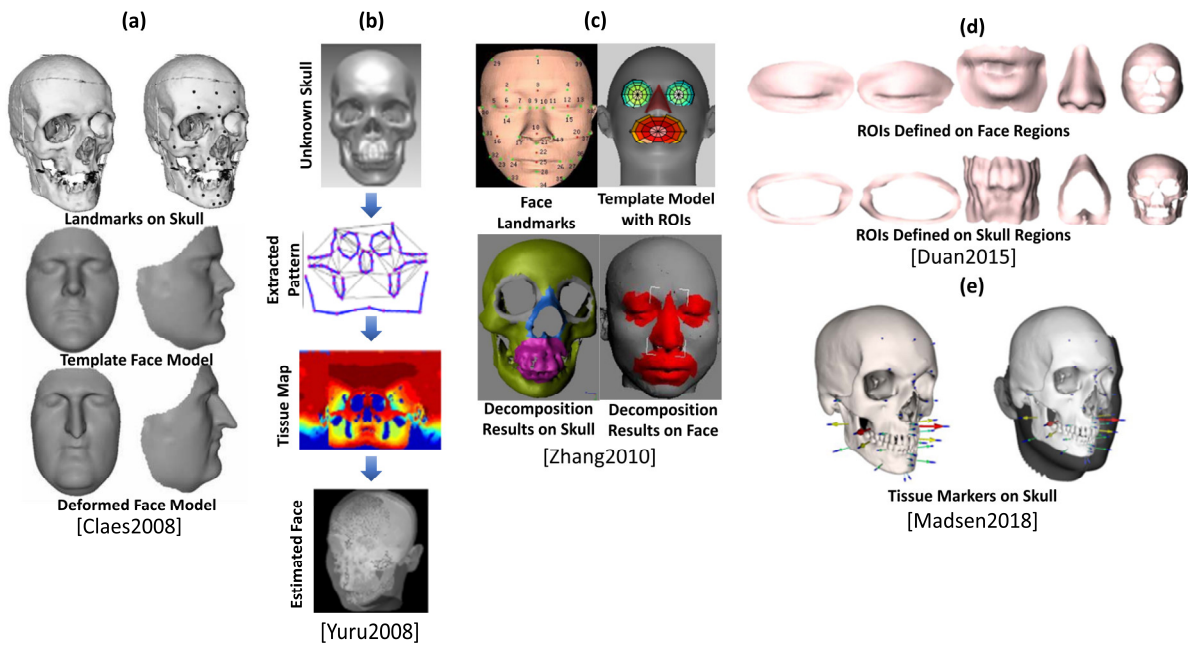


Figure 7. Statistics-based skull-to-head modeling methods: (a) face-skull relationship was trained through 52 landmarks on skull surfaces using PCA; (b) relationship between head and skull surface was represented through non-subject-specific 2-D tissue maps; (c) face-skull relationship was trained in different ROIs using PCA; skull-to-head relations were trained through different ROIs using PLSR; (e) tissue depth markers were used for measuring face-skull thickness using SSM.

Several statistics-based methods have been applied for learning face-skull relationship on a number of face-skull datasets reconstructed from MRI/CT images. The numbers of datasets range from 25 to 331. Artificial facial landmarks are often placed on facial and skull surfaces to acquire head-skull surface features. PCA methods have been often used to reduce the data dimension by using orthogonal transformation between possibly correlated variables [122,123]. Thus, computed first principal components are assumed to represent the highest variability in the initial dataset. However, in real applications, this assumption is not always satisfied. It is important to note that the PCA methods are more suitable when applying on datasets with highly correlated variables. Particular attention should be paid on the data standardization at the initial step. Another limitation of the PCA methods relates to the unphysical negative behavior for some applications (e.g. astronomy) [124]. To overcome the drawback of the possible dispersion between principal components and outcome variables, the PLSR could be used as an alternative approach [125,126]. This approach allows identifying principal components related to both original variables and outcomes. Thus, this dimension reduction

strategy is supervised by the outcomes ensuring the best possible covariance between the original variables and outcomes. Note that non-negative matrix factorization (NMF) could be used to overcome the drawback of PCA methods related to the unphysical negative behavior [127,128]. Moreover, Body Mass Indices (BMI) values can also be integrated as descriptors to generalize the predicted outputs. **In particular, the skull-to-face relation has been well-studied, but the head-to-skull relation has not been well analyzed yet. Thus, the distribution of head-skull thicknesses remains unclear. Moreover, model hyperparameter (optimal number of feature points, appropriate learning configurations and associate parameters) tuning has been rarely done in the literature.**

Overall, patient-specific biomechanical head modeling also requires patient-specific skull structures. Skulls can be accurately reconstructed from MRI/CT images, but the reconstruction processes are slowed and require a lot of clinical experts. On the other hand, external head surface models can be fast reconstructed using visual sensors (e.g. the Microsoft Kinect sensor), internal skull structures cannot be acquired using these types of sensors. Consequently, **head-to-skull relationship should be studied so that internal skull structures can be predicted based on the external head surface model. The PLSR is the most suitable over the other statistics-based methods for this type of issue.** This PLSR-based model should be applied in the patient-specific biomechanical head generation for muscle-oriented facial paralysis grading.

2.4. Serious games for facial mimic rehabilitation

Facial mimic rehabilitation must contain both physiology and psychology treatments, and serious games are particularly suitable for these requirements [21]. In fact, serious games were defined as computer-based games specifically designed with challenged goals, scoring strategies, interesting scenes, and real-time interactions. Moreover, skills, knowledge, and attitude that users were achieved during playing serious games can be applied in the real world [129]. Based on types of players, serious games were classified into patient and non-patient-oriented games. About non-patient-oriented games, these serious games could be used for health and wellness, professional training, and non-professional training. About patient-oriented games, these serious games could be designed for health monitoring, clinical detection, treatment, rehabilitation, and self-care education [130]. Each serious game has five main components: rules (gameplay), challenge, interaction, and goal (explicit: game objectives and implicit: skills, knowledge, experiences) [130]. Many review studies have been done for the topics of treatment and rehabilitation serious games, and they mostly concluded that serious games were shown positive effects on chronic, stroke, and traumatic diseases [90–92]. So, numerous serious games have been designed for treatment and rehabilitation for patients. We could find multiple applications of serious games in upper-limb stroke rehabilitation [93,94], trauma bones and soft-tissue injuries [95], traumatic brain injuries [96], home-based stroke rehabilitation [97], neuropsychological assessment, training and rehabilitation [131], and orthopedic trauma surgery [132]. Especially, serious games were particularly effective for rehabilitations and treatments of older adults [133] and pediatric patients [134]. Serious games could also provide positive cognitive effects by generating interesting 3-D virtual reality scenes for motivating primary school children from fearful procedures from the clinical treatment procedures [135]. Moreover, it could increase adherence and compliance to exercises for elderly [133].

Although numerous serious games have been designed for treatment and rehabilitation in various applications, no serious games have been developed for facial mimic rehabilitations. This might be caused by lacks of tracking devices (interaction tools) and modeling methods that can generate and animate patient-specific biomechanical head/face models in real-time. Numerous interaction tools have been used in serious games

for health applications (Figure 8). Based on the review of Wattanasoontorn et al., 2013 [98], computer mouse were mostly used in serious games for controlling game objects (50.93% in Figure 2). Some high-level tracking devices such as Wii Peripheral, Kinect, Camera, and BioFeedBack, etc., were rarely employ in serious game systems. Actually, these real-time tracking devices, which can help game systems acquire high-level motion data, have revolutionized the way users play serious games [136]. Among them, the Microsoft Kinect sensor have been popularly used for rehabilitation serious games, such as cognitive rehabilitation exercises [137] (Figure 9a), virtual reality-based rehabilitation [138] (Figure 9b), lower limb rehabilitation [139] (Figure 9c), upper limb rehabilitation [140] (Figure 9d), cerebral palsy rehabilitation [141] (Figure 9e), home-based physical exercises [142,143] (Figure 9f), etc.. However, most Kinect-based serious game systems took advantages of Kinect’s kinematic data for interacting between users and game objects. Although the Kinect sensor coupled with Kinect SDKs can offered other types of high-level data such as high-definition facial points [60], no serious games have been taken advantages of these types of data for interacting with users. Certainly, developments of novel serious games for facial mimic rehabilitation can be benefited from the Kinect’s facial point data.



Figure 8. Interaction tool usages in serious games for health up to 2013 [98].

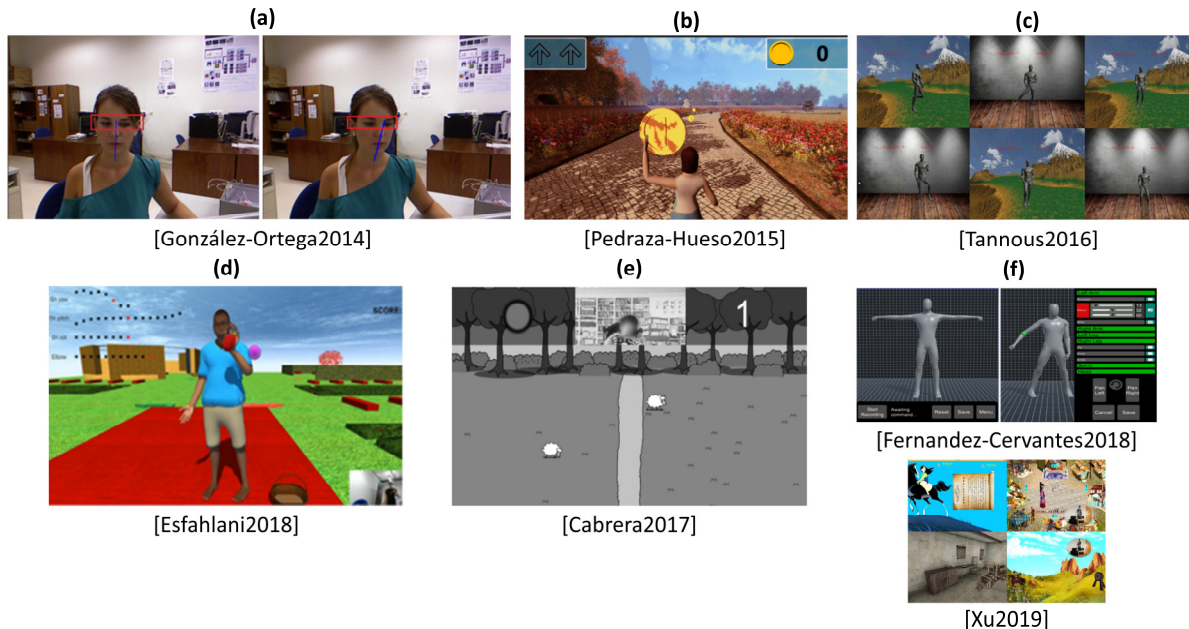


Figure 9. Kinect-based serious games for (a) cognitive rehabilitation exercises, (b) virtual reality-based rehabilitation, (c) lower limb rehabilitation, (d) upper limb rehabilitation, (e) cerebral palsy rehabilitation, (f) home-based physical exercises.

Beside the interaction tools, numerous game engines have been available for serious game developments (Figure 10). Most game engines can be used to develop games for multiple operating systems such as Windows, Linux, iOS, and Android [99]. This allows us to develop games for both static and mobile devices for both clinical and home-based applications. All game engines can also support scripting in high-level programming languages, such as C#,

Python, etc. [99], so time-to-market will be significantly decreased. Some game engines, such as Unity [144], Blender [145], and Unreal [146], support interactive physical user interfaces (GUIs) so that game developers can fast design their game scenes and characters and receive immediately the graphic rendering results. Moreover, multiple engines such as physics, GUIs, and sounds, were also integrated in some game engines (Unity [144], XNA [147], Unreal [146], Blender [145], etc.) with user-friendly interfacing functions, so game developers can fast design their games with realistic graphical, sound, and physical effects. However, most game engines were designed for developing entertainment games. **Currently, no game engines were specialized for medical applications, especially for health serious games. Moreover, accurate physics simulation engines for biomechanical simulations such as SOFA [148], PhysX [149], Bullet [150], etc., can acceptably simulate soft-tissue behaviors, but they cannot support lively graphical effects like in game engines.** Consequently, integration of biomechanical physics engines into general game engines might be essential for rehabilitation serious game developments. A flexible game engine that can allow developers to integrate accurate physics engines into it might be essential for medical serious game developments. **Object-Oriented Graphics Rendering Engine (OGRE3D) [151] can be a good candidate.** This 3-D game engine can support flexible mesh management, mesh animation, scene management, scene graphic hierarchy, graphic rendering, and other high-level graphic processing techniques. Moreover, because this game engine is an open-source library, it is easily integrated with other packages, such as networking, collision detection, physics, sound, and input control, etc., supporting for serious game developments. Additionally, automatic subject-specific model generations are rarely supported by serious game engines/tools. Developers usually craft their character models from graphical design tools such as Blender [145] and MeshLab [152], and these models are then imported to game engines for rendering and animating. Therefore, subject-specific model generations are time consuming and need crafting experts. In fact, subject-specific models are important to treatment/rehabilitation procedures [100]. **A serious game system with the ability of generating subject-specific game objects should be necessary for improving rehabilitation procedures.**

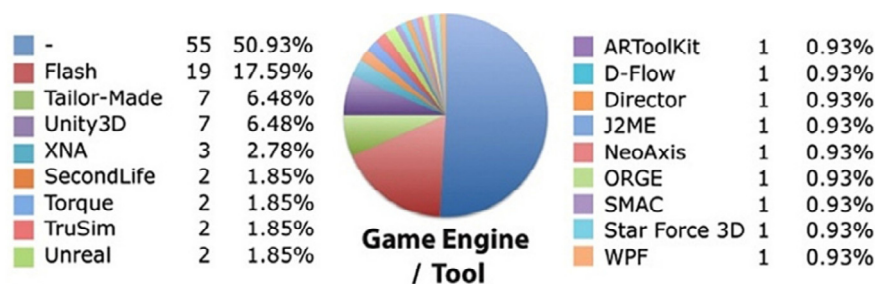


Figure 10. Game engines for health serious game developments [98].

Overall, psychological effects are also important to facial mimic rehabilitation, and serious games are one of the best tools for motivating patients during their rehabilitation exercises. However, although several tracking devices and game engines/tools have been developed for serious game developments, no serious games have been developed for facial mimic rehabilitation. Additionally, subject-specific game objects should also be used for improving cognitive and physical rehabilitation outcomes. Consequently, **serious games coupled with a patient-specific head model is novel and desirable for facial mimic rehabilitation.**

2.5. The concept of System of Systems (SoS) for healthcare system developments

System of systems (SoS) is defined as ‘a collection of task-oriented or dedicated systems that pool their resources and capabilities together to create a new, more complex system which offers more functionality and performance than simply the sum of the constituent systems.’

[101]. This concept is particularly suitable for system developments in which reference frames, though processes, quantitative analysis, tools, and design methods are not complete [101].

The SoS has five outperforming characteristics over the other types of system including (1) anatomy, (2) belonging, (3) connectivity, (4) diversity, and (5) emergence [153]. For the anatomy property, although subsystems inside a SoS system has multiple functions for general purposes, some of their functions are optimized for the unique single purpose when integrated into a whole system. For the belonging property, the subsystems can be worked as separate systems as staying outside the main system. For the connectivity property, the SoS systems can also support dynamic connectivity among all available subsystems and new subsystems. Every subsystem also has ability of creating/deleting connections with other subsystems for integrating/separating into/from the main system. For the diversity property, the main system can have more functions that cannot be possessed by any subsystems when standing alone. For the emergence property, the SoS systems can support predicting and analyzing emergent, especially undesirable, behaviors, appeared from the main system and subsystems.

Based on interrelations among subsystems inside the main system, the SoS systems are classified into four types: (1) virtual, (2) collaborative, (3) acknowledged, and (4) directed systems (Figure 11) [154]. In virtual SoS systems, connections among subsystems are temporary in necessary conditions for executing central purposes, and they do not have a central management authority. In collaborative SoS systems, no central units are designed for managing or guiding subsystems, and they have to self-collaborate together to achieve the central purpose of the main system. In acknowledge SoS systems, a central unit is designed for managing central purpose of the main system, but it cannot have complete authority over the subsystems. In particular, in this type of SoS, subsystems maintain their independent ownership, objectives, funding, and development approaches, but they cannot have feedbacks to the central manager. In directed SoS systems, they also have a central unit for managing and guiding central purposes of the main system, and their subsystems can receive/send commands from/to the central unit. With these advantages and flexible structures, the concept of SoS has been applied in multiple disciplines whose system developments require complex interactions among independent subsystems.

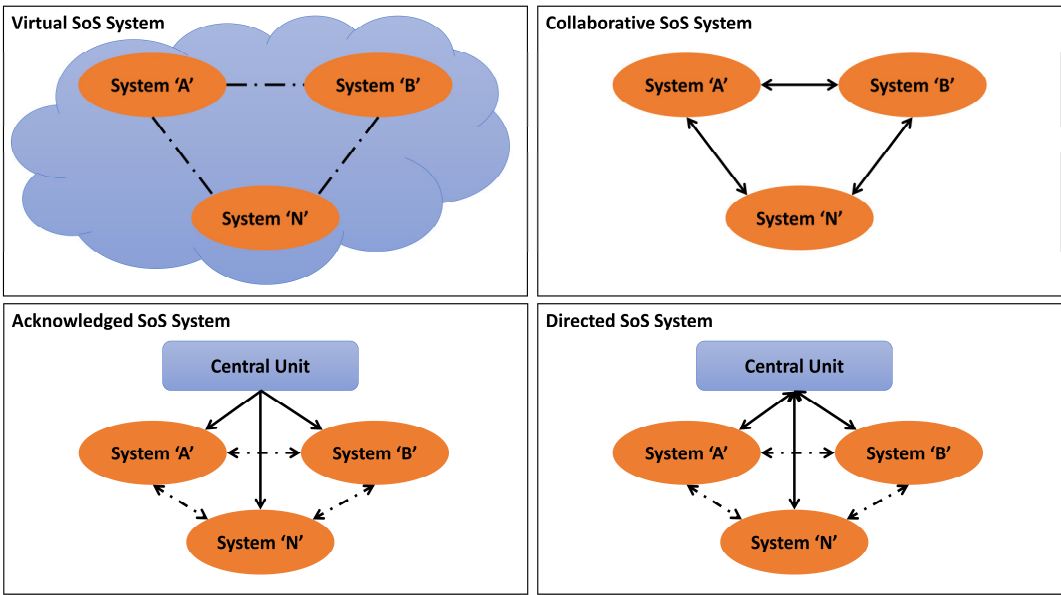


Figure 11. SoS types [154].

Although the concept of SoS has numerous advantages, some challenges have been risen for applying this concept to develop systems in specific disciplines [155]. (1) In an SoS, each subsystem has its own authority about stakeholders, users, business processes, and development approaches, so most SoS cannot include a unique authority for all subsystems. (2) Because there is no unique authority for controlling all subsystems, so leadership in the multiple organizational environments of an SoS is challenging to declare for providing coherence and direction among subsystems. (3) Implementing a system into a new SoS is also challenged because the system may not have the same interfaces, data types, and contexts as the SoS. (4) Capabilities and requirements of a SoS may or may not be compatible with ones of its subsystems, so more interfaces need to be designed so that the SoS can meet the needs of the subsystems. (5) Interdependencies among subsystems may lead to unexpected or unpredictable behaviors in a SoS. (6) Because the SoS comprises of large and diverse subsystems, traditional end-to-end testing, validating, and learning procedures are costly. Consequently, the validation data of SoS are mostly computed from modeling, simulating, and analyzing. (7) The concept of SoS is relatively new, so the general principles of SoS that can be applied for developing systems in different areas might be difficult to be declared.

The above challenges are also risen when applying the concept of SoS for developing computer-aided healthcare systems. Most causality healthcare systems include measurement, recognition, estimation, and evolution subsystems [102]. In the measurement subsystem, multiple types of sensors, such as cameras, infrared sensors, EMGs, ultrasonic sensors, etc., could be requested simultaneously in real-time during system executions. Consequently, the main system should resolve the reliability of data transmission to keep stable working conditions. Moreover, each type of sensor has its own driver for interfacing with other subsystems, so data acquisition interfaces should be designed for controlling these device drivers. Other subsystems should still work if data are not available from these sensors. All subsystem needs to be executed parallelly in different hardware systems or a single multi-threaded hardware system. Data communications among multiple hardware configurations or computation threads should be fast enough to reduce latency among subsystems. Moreover, **programming frameworks are necessary for reducing time-to-market during SoS-based system development procedures.** System verifications and validations should also be conducted for all subsystems and the whole system should be conducted in short- and long-term processes. Last but not least, a standard medical software development and maintenance processes and activities should be followed to cover all above requirements risen when developing systems based on the concept of SoS.

2.6. Technology selections for the target system development

Facial mimics are formed by soft-tissue deformations on facial skins, so realistic simulation of facial mimics needs real-time and accurate simulation of soft-tissues. Moreover, developments of realistic soft-tissue simulation systems for medical applications require compromising selections of all system development aspects including computational approaches, interaction devices, system architectures, and clinical validations [103]. **A systematic review was also conducted relating to real-time medical simulations of soft-tissue deformation: computational approaches, interaction devices, system architectures, and clinical validation.** The review provides useful information to characterize how real-time medical simulation systems with soft tissue deformations have been developed. By clearly analyzing advantages and drawbacks in each system development aspect for real-time simulation of soft-tissues from all related studies in literature, this review was used to provide technological selections for developing the target system. Details of this review were presented in Appendix A.

For the computational approach, hybrid modelling methods were the most powerful computational approaches for realistically simulating soft-tissue deformations. The combination of MTM, FEM, and Pre-Comp FEM [156] or the combination of BEM, MSM, and a particle surface interpolation algorithm [157] can be some of the best candidates for these hybrid modelling methods. SOFA framework [158] with abilities of modularly and flexibly integrating multiple of existing/new modeling methods on the same target model can be suitable for developing new hybrid modelling methods. **However, computation cost for modeling methods could be reduced if suitable interaction devices were selected** [103].

For the interaction devices, input interaction devices significantly contribute to the whole system accuracy and execution time. **Some multifunctional and well-calibrated sensors such as the Microsoft Kinect can tracked multiple types of user data.** The Microsoft Kinect V2.0 [159] with the support of Kinect SDK 2.0 [160] can help developers acquired from low-level to high-level user data with high resolution in real-time, which cannot be found in other types of sensors. Table 1 lists some technological characteristics of Kinect V2.0 [159] with the support of Kinect SDK 2.0 [160]. Figure 12 also illustrates all data types being able to be acquired by a Kinect V2.0 controlled by Kinect SDK 2.0 in real-time. In particular, using the time-of-flight (ToF) infrared sensors, the Kinect can acquire depth values of scenes inside the $[70 \times 60]$ -degree FoV with the resolution of 512×424 pixels. The acquired color images have the resolution of $1,920 \times 1,080$ pixels. The depth and color images can be acquired up to 30 FPS within the plausible ranges from 0.5 m to 4.5 m. Especially, the Kinect SDK 2.0 can also support a high-resolution head model (2,582 vertices \times 5,160 facets) whose facial vertices are fitted with the 1,347 tracked high-definition facial points. **Consequently, no computation costs will be consumed for real-time soft-tissue simulations if the Kinect-based head model and HD facial points are employed for tracking facial mimics.** Moreover, these HD facial points can be easily mapped to color image spaces, so face texture can be fast and automatically generated. Because of all above benefits, the Microsoft Kinect V2.0 [159] coupled with Kinect SDK 2.0 [160] will be recommended as the input interaction device of the main system.

Table 1. *Technological characteristics of the Microsoft Kinect V2.0 controlled by Kinect SDK 2.0.*

Parameters	Values
Infrared (IR) camera resolution	512 \times 424 pixels
RGB camera resolution	1,920 \times 1,080 pixels
Field of view (FoV)	70 \times 60 degrees
Framerate	30 frame per second
Measuring plausible range	From 0.5 to 4.5 m
Generic head model size	2,582 vertices; 5,160 facets
Number of HD facial points	1,347 points
Kinematic skeletons	6 simultaneous 25-joint skeletons

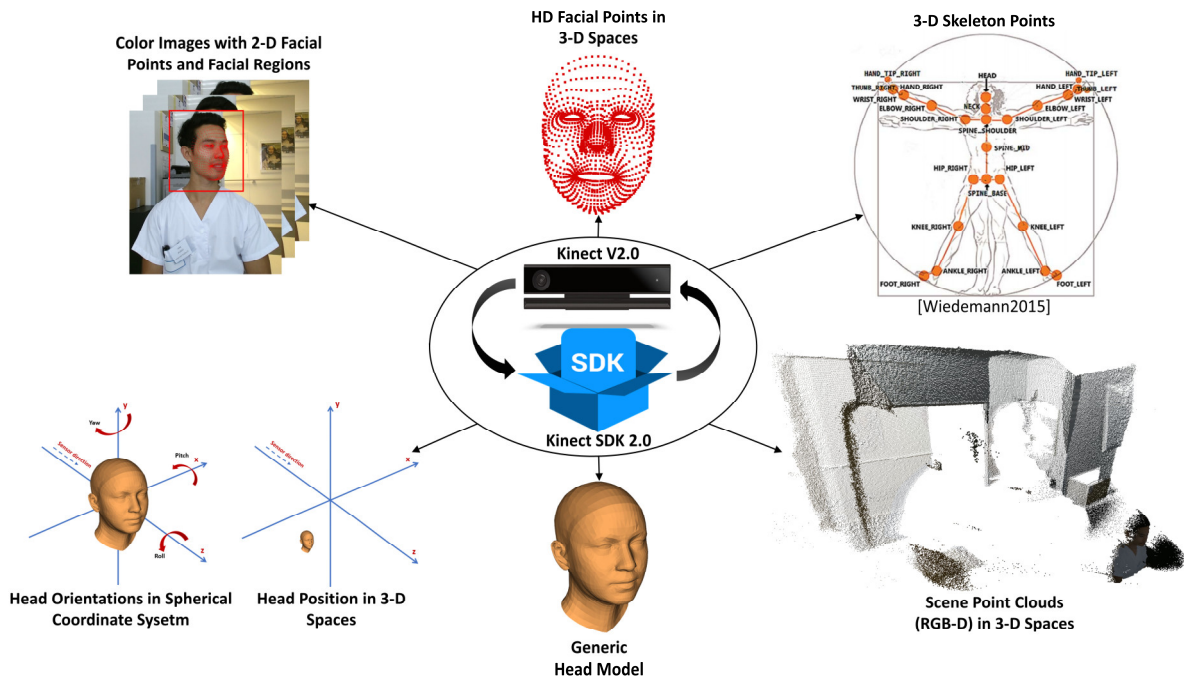


Figure 12. Data types supported by a Kinect V2.0 sensor using Kinect SDK 2.0.

For the system architectures, the multithreaded system execution scheme is outperformed compared with the distributed one when designing systems based on the concept of SoS. This scheme can take advantages of multicore CPUs with high-speed data transmission through internal buses. Subsystems in the main system should be designed to execute parallelly on multiple threads so that they can have both independent and complex interaction during executions as required in the concept of SoS. **The directed structures of the SoS concept (Figure 11) was used for designing the target system framework**, which is shown in Figure 13. In particular, the main system includes four systems: (1) model computation system for acquiring subject-specific data and generating & animating subject-specific models, (2) serious game system for executing game rules, (3) graphic user interface system for interfacing between users and system functions and rendering graphics, and (4) central managing system for diagnosing and synchronizing all system functions. Details of each system will be presented in the following chapters/appendices. A flexible system programming framework, such as Qt [161], can help multithreaded system programming be easier and faster. Beside interactive and cross-platform graphic user interfaces (GUIs), this framework can support numerous application program interfaces (API) for multithread programming. Communications among threads can be easily declared and managed using the signals and slots mechanisms, which make Qt [161] more advantageous than other frameworks. For real-time graphical rendering, VTK [162] can be one of the best suitable visual frameworks for integrating with Qt [161]. This framework can support 3-D model rendering, 3-D mesh modeling, image processing, and 2-D/3-D plotting in high-level object-oriented programming interfaces. Because the main system needs to be optimized from low-level to high-level programming classes, the C++ programming language is employed for system developments. The target system will be run on Microsoft Windows, as it is currently the most popular and user-friendly operating system.

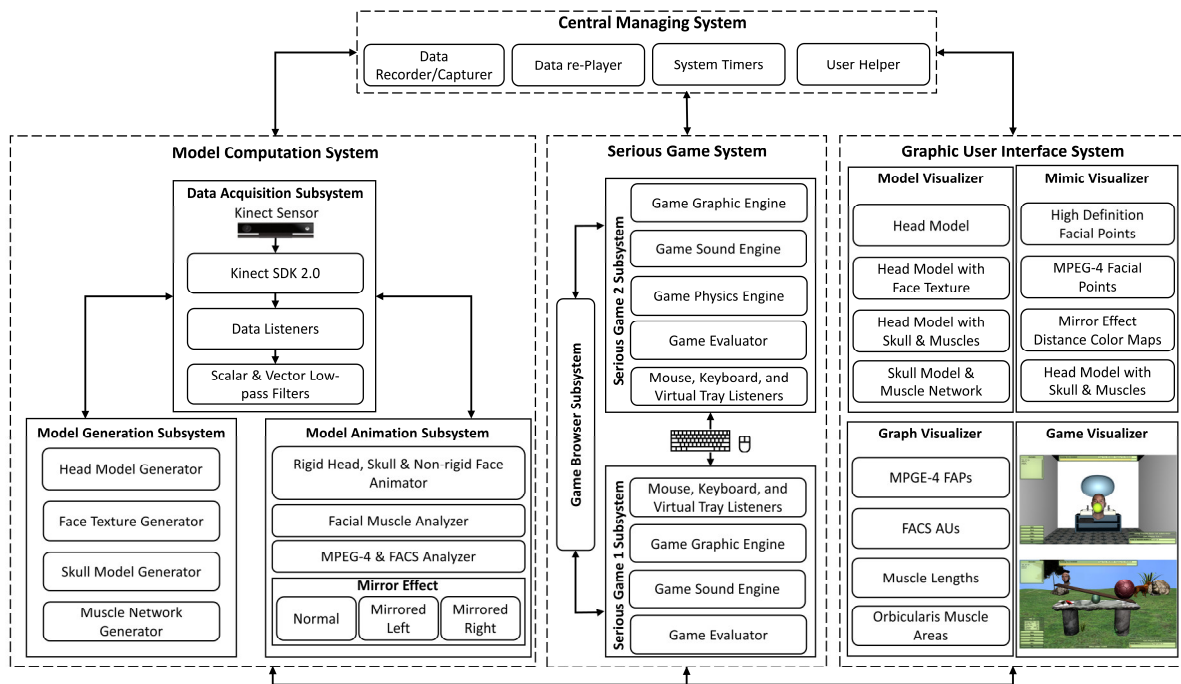


Figure 13. SoS-based system framework of the clinical decision-support system for facial mimic rehabilitation.

For the clinical validation, the target system should also be validated in a multi-level validation procedure including geometrical validation, model behavior validations, and user acceptability/safety validations so that it can be translated into clinical routine practices. The subject-specific biomechanical head/face models should be compared with MRI/CT- or point cloud-reconstructed models in geometrical validations. Facial muscle actions during facial mimics can be validated with other related studies. User acceptability/safety validations will be conducted on system graphic user interfaces and system reliabilities in both long-term and short-term evaluation campaigns. Hand or built-in outcome questionnaires can be used for these validations. Moreover, a standard process of medical software development and maintenance should be applied to improve the safety and reliability of the developed medical software system. In this project, all system development tasks were completed based on the IEC 62304:2006+A1:2015(E) standard of medical device software – software life-cycle processes [104].

Overall, all selections throughout the four system development aspects (computational approaches, interaction devices, system architectures, and clinical validations) for developing the target system are listed in

Table 2.

Table 2. Technology selections of the four system development aspects for developing the target system.

System development aspects	Selected technologies
Computational approach	Directly use the Kinect-based HD facial points
Interaction devices	
• Input interaction devices	Microsoft Kinect V2.0, PC Mouse, Keyboard
• Output interaction devices	PC Screen
System architectures	
• Programming framework	Qt 5.12.5 [161]
• Input device SDK	Microsoft Kinect SDK 2.0 [160]
• Graphic rendering framework	VTK 7.1.1 [162]
• Serious game engine	Ogre3D [151]
• Sound engine	irrKlang [163]

System development aspects	Selected technologies
<ul style="list-style-type: none"> • Physics engine • Programming language • Target platform 	Bullet real-time [150] C++ [164] Microsoft Windows with Visual Studio C++ 2015
Clinical validations	
<ul style="list-style-type: none"> • Geometrical validation • Model behavior validation • User acceptability/safety validation • Standard medical software development and maintenance processes 	Comparisons with CT/MRI-reconstructed model Comparisons with related studies Questionnaires & Surveys on patients & clinical experts IEC 62304:2006+A1:2015(E) [104]

2.7. Conclusions

Development of a clinical decision-support system for facial mimic rehabilitation is a scientific, technological, and clinical challenge. This chapter analyzed in details this challenge in literature throughout all system development tasks. Base on this analysis, no studies have been successfully integrated subject-specific head generations and real-time head animations in a single system execution scheme. Although facial muscle should be directly analyzed for facial paralysis grading, most external facial movements were evaluated. Moreover, serious games have their positive contributions to rehabilitation outcomes, no serious games have been developed for facial mimic rehabilitation. **Consequently, a system with ability of fast generating & real-time animating subject-specific head, evaluating facial muscle actions during facial mimics, serious game playing for facial mimic rehabilitation is essential and original in literature and medical applications.** Last but not least, successful development of such a multifunctional, real-time, and complex clinical system needs compromising selections of all system development aspects: computational approaches, interaction devices, system frameworks, and clinical validations. A systematical review of these aspects is therefore required for the compromising selections.

Chapter 3:

Real-time Computer Vision System for Tracking Simultaneously Subject Specific Rigid Head and Non-rigid Facial Mimic Movement

Head and facial mimic animations play a vital role for facial analysis applications. Numerous studies have been trying to simulate these animations. However, they hardly achieved both rigid head and non-rigid facial mimic animations in a subject-specific manner with real-time framerates. Consequently, a real-time computer vision system for tracking simultaneously rigid head and non-rigid facial mimic movements was proposed. The system was developed using the system of system approach. A data acquisition subsystem was implemented using a contactless Kinect sensor. A subject-specific model generation subsystem was designed to create the geometrical model from the Kinect sensor without texture information. A subject-specific texture generation subsystem was designed for enhancing the reality of the generated model with texture information. A head animation subsystem with graphical user interfaces was also developed. Moreover, the system was also integrated with a mirror effect subsystem for measuring asymmetries during facial movements. Model accuracy and system performances were analyzed. The comparison with MRI-based model shows a very good accuracy level (distance deviation of ~ 1 mm in neutral position and an error range of [2-3mm] for different facial mimic positions) for the generated model from our system. Moreover, the system speed can be optimized to reach a high framerate (up to 60 fps) during different head and facial mimic animations. In the following chapters, an internal skull model and a muscle network will be estimated from the subject-specific head model for computing real-time muscle actions during facial animations. The animated head model will also be the interactive game object for the serious game system. The model & texture mirror effects can support for real-time facial paralysis grading.

3.1. Materials and methods

3.1.1. System framework

The concept of system of systems [165] was applied in our head and facial mimic animation system. In this concept, a system is composed of multiple subsystems that can be executed independently and optimized to achieve a unique optimization according to a system development target. The software architecture of the proposed system, shown in Figure 14, includes data acquisition, subject-specific head generation, texture generation, head animation, and GUI subsystems. They are parallelized to be executed in internal system threads and targeted at achieving both real-time framerates and acceptable accuracies for head and facial mimic animations.

Specifically, in the data acquisition subsystem, the Kinect sensor 2.0 was used and controlled by the Kinect SDK 2.0. The acquired data comprise of HD facial points, head orientations, head positions, and color images. A subject-specific head model was generated from a template head model, supported by the Kinect SDK, using the subject-specific facial points in the subject-specific head generation subsystem. A facial texture image and texture coordinates were also created from the color images marked with facial points in the texture generation subsystem. In the head animation subsystem, the generated model was then transformed to current head orientations and current head positions. The non-rigid animations of facial regions were formed by replacing facial vertices by HD facial points. The left/right facial region of the animated head model was also mirrored to the right/left face region using left/right mirror affecter in the mirror effect subsystem. In this subsystem, the distance color map between the normal face and the left/right-mirrored face was also computed using the Hausdorff distance metric [166]. Finally, the animated head model and the generated texture were rendered in the GUI

subsystem. With the GUI subsystem, users could also control all system procedures, change system parameters, and capture current subject-specific data.

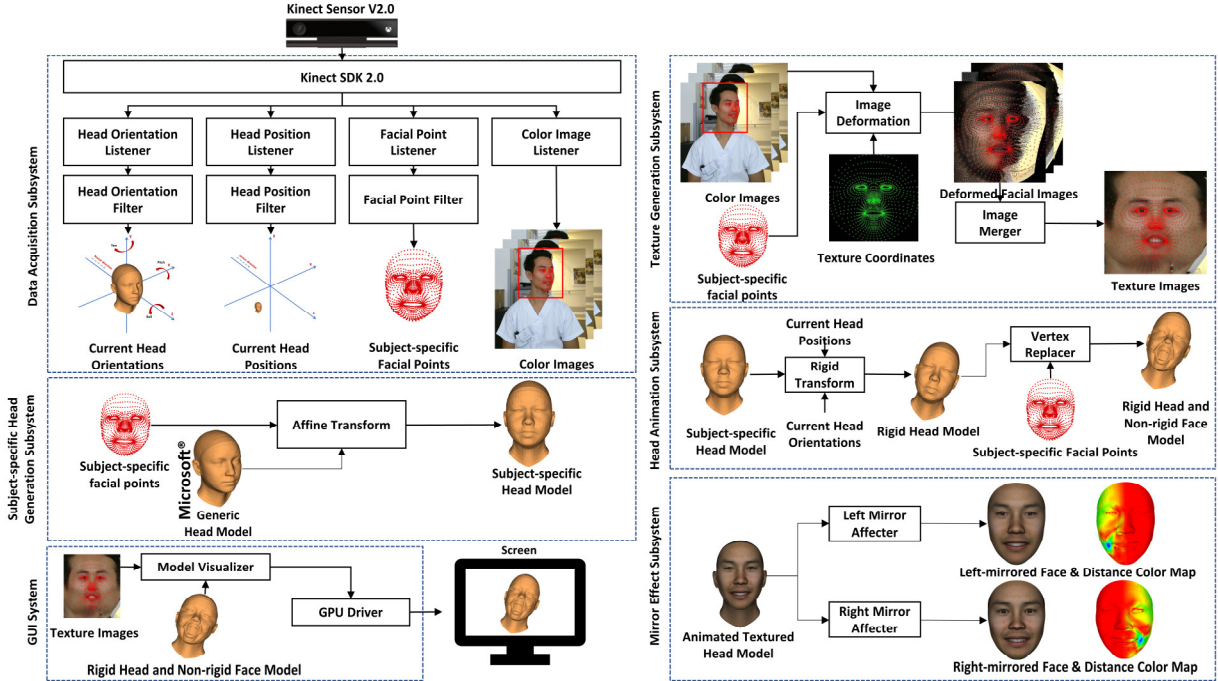


Figure 14. The system framework of the developed real-time computer vision system for tracking simultaneously rigid head and non-rigid facial mimic movements.

3.1.2. Data acquisition subsystem

A data acquisition interface was programmed using system thread to continuously capture available data from the Kinect SDK. This thread was composed of HD facial point, head orientation, head position, and color image listeners. In fact, the Kinect sensor can simultaneously detect and tracked six users inside its field of vision (FOV), but only one user can be supported for high-definition facial analysis. Consequently, only the nearest user was selected for further facial analysis. Specifically, the nearest user was chosen based on the minimum value among distances between head positions and the Kinect sensor. The detected and tracked facial points included 1,347 points in both 3-D spaces and 2-D color image spaces. Specially, the MPEG-4 facial point set [167] including 35 points in 3-D spaces was also supported by the Kinect SDK. To remove noises from the HD facial points (the MPEG-4 facial points), low-pass filters were implemented for each 3-D point in HD (MPEG-4) facial point sets, and cut-off frequencies could be adjusted by users. Additionally, head orientations and head positions were also acquired and filtered in the data acquisition subsystem. The current head positions were tracked in 3-D Euclidean space ($\mathbf{P}_c(x_c, y_c, z_c)$), and the current head orientations were tracked in quaternion-coordinates ($\mathbf{I}_q(x_q, y_q, z_q, w_q)$), which were converted into the pitch, yaw, and roll rotation angles in Euclidean space ($\mathbf{I}_c(pitch_c, yaw_c, roll_c)$) using the following equations:

$$pitch_c = \sin^{-1} 2(w_q y_q + z_q x_q) \quad (1)$$

$$yaw_c = \tan^{-1} \frac{1-2(y_q^2+z_q^2)}{2(w_q z_q+x_q y_q)} \quad (2)$$

$$roll_c = \tan^{-1} \frac{2(w_q x_q+y_q z_q)}{1-2(x_q^2+y_q^2)} \quad (3)$$

where $pitch_c$, yaw_c , and $roll_c$ were the rotation angles (radian) around x -, y -, and z -axis in the Euclidean space. For texture generation, color HD (1920×1080) images were also

captured during tracking facial and head data. In the HD images, rectangles contained facial regions and facial points in 2-D image spaces were automatically generated by the Kinect SDK. Finally, the Kinect SDK also supports a template head model whose facial vertices are matched with the HD and MPEG-4 facial points [167]. Figure 15 shows examples of data types and the template head model supported by the Kinect SDK.

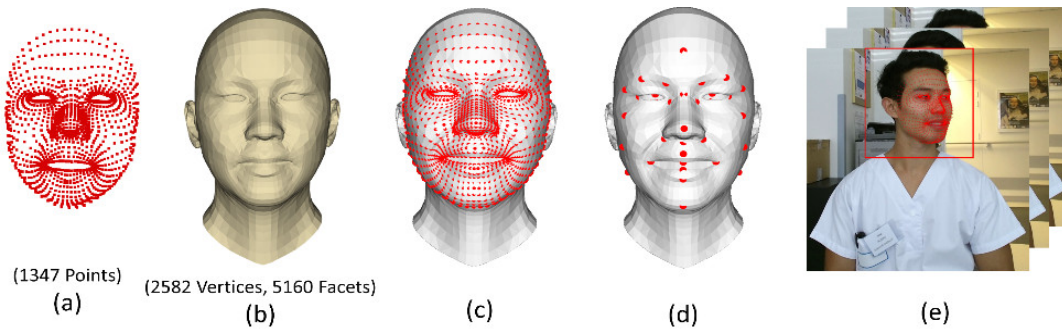


Figure 15. Data acquired from the Kinect: (a) HD facial points, (b) the generic head model, (c) HD facial points matching with facial vertices, (d) MPEG-4 facial points, and (e) color images marked with facial regions and facial HD points

Although multiple types of data can be acquired by the Kinect sensor in real-time, to optimize the data acquisition bandwidth for appropriate subsystems, a data acquisition strategy was proposed. For model and texture generations, all types of data (HD facial points, head orientations, head positions, HD color images) were acquired. In this case, the highest data acquisition framerate of the Kinect was just 30 fps. However, for model animations, only HD facial points, head orientations, and head positions were needed, so the data acquisition bandwidth could be accelerated up to 60 fps. Particularly, when any type of data had not been available by the Kinect SDK, the system was able to continue to acquire other types of data without waiting for this type of data.

3.1.3. Subject-specific head generation subsystem

This subsystem aimed to generate subject-specific head geometrical model from the HD facial points. In the data collection stage, head orientations and positions of the nearest user were tracked to instruct the user to move his/her head into required directions. Figure 16 shows the instruction procedures to get the left, right, up, and front views of a user in the GUI subsystem. Specifically, left, right, up, and front views of the tracked user were needed, and the Kinect SDK would decide the needed view. Note that the instructing head model was in mirrored direction with the current head model in yaw rotation (Figure 16). According to the requirements from the Kinect SDK, suitable guiding images were displayed. Moreover, to help the user control his/her head movements, a template head model was controlled to rotate according to his/her current head orientations in real-time.

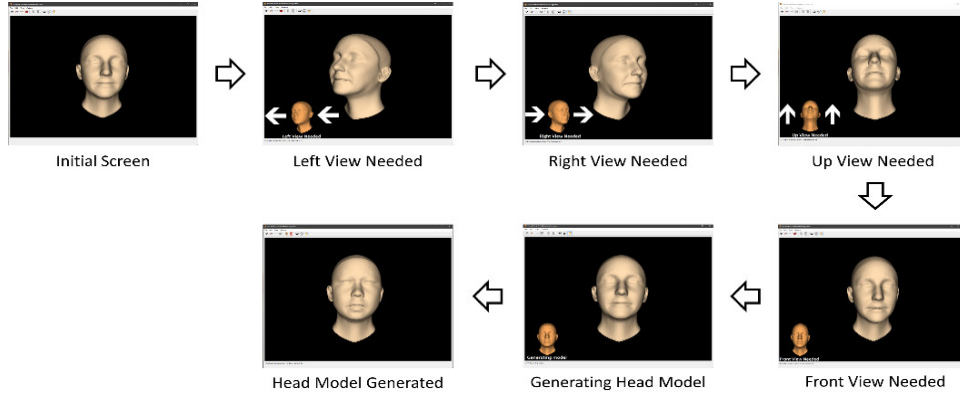


Figure 16. Positional set up during the data acquisition procedure for subject-specific head generation.

In the subject-specific model generation stage (Figure 17), an affine transformation matrix from the generic facial vertices ($F_G[1347 \times 3]$) to the subject-specific facial points ($F_S[1347 \times 3]$) was estimated. Note that the head model composes of back head vertices and facial vertices. The coherent point drift (CPD) registration algorithm was used to estimate the best suitable affine transform matrix [168]. Then, estimated affine transform matrix was applied to the template head vertices ($H_G[2582 \times 3]$) to form the affined head vertices ($H_A[2582 \times 3]$). Finally, the subject-specific head vertices were created by replacing the facial vertices with the subject-specific facial points. As a result, the boundary vertices of the back-head vertex set are relatively fitted with ones of the facial vertex set. For rendering on graphic systems, the facet structure of the subject-specific model was kept the same as one of the template model. It is important to note that the use of the Kinect sensor does not allow head back information to be acquired directly for a subject or patient specific model. Thus, we applied affine transform to the generic model for deducing the best possible back side information for a specific subject. Note that only facial vertices were replaced, then the use of affine transform allows the back-head information to be deformed and used directly on the final generated model. The quality of fitting process depends on the number of points in the registration point sets used when applying the CPD method. Consequently, discontinuities on the boundary vertices appear when the density of facial vertices was decreased.

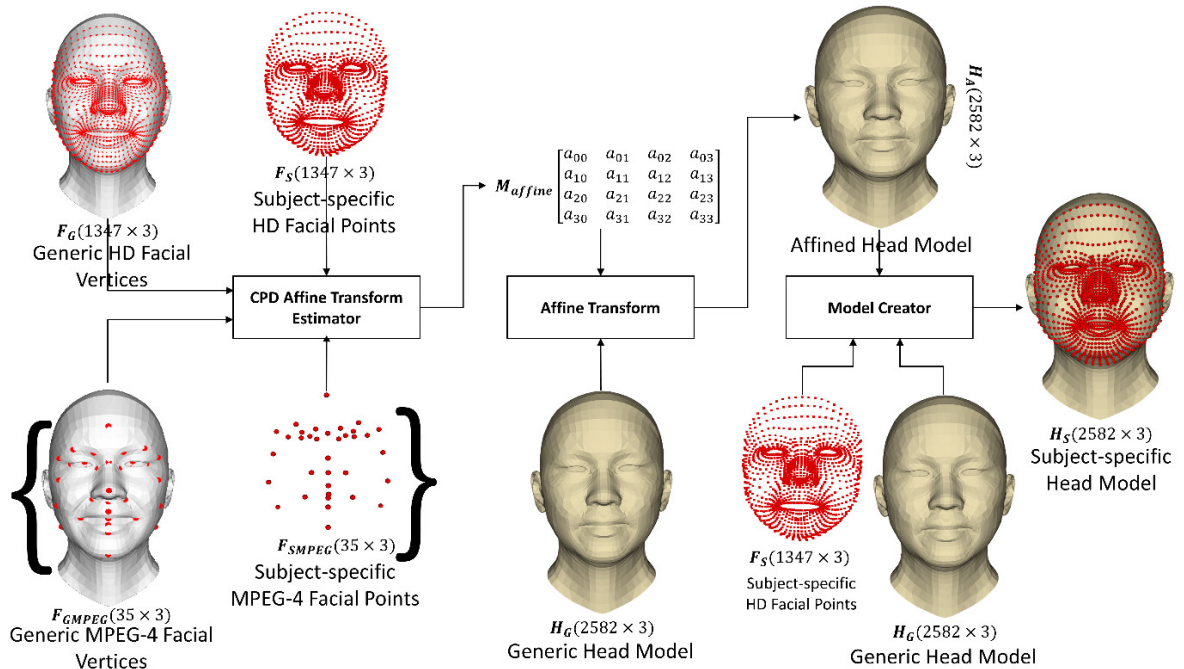


Figure 17. Illustration of different processing steps of the subject-specific head generation procedure.

In this transformation procedure, the processing was mostly costed at estimating the affine transform with CPD. The computation time of this step was significantly reduced when decreasing the density of the facial pointset. An alternative facial point sets was selected as Moving Picture Experts Group (MPEG)-4 facial standard [167]. Especially, the Kinect SDK can also support tracking 35 MPEG-4 facial points on a user face. However, when decreasing the density of facial point sets, subject-specific information would also be reduced, and discontinuities on the boundary vertices between the back head and facial regions appear. In the GUI subsystem, users can choose between the high-detail facial points and MPEG-4 facial points for generating their head models.

3.1.4. Subject-specific texture generation subsystem

In this subsystem, the texture image and texture coordinates of the facial region were generated for rendering in the GUI subsystem.

First, a data capture process was developed (Figure 18). Left, right, and front views of the user’s face were needed. Because the details of each view were highly affected by capturing angles, to optimize the facial details the tracked user was guided to rotate his/her head yaw angles to pre-defined values of -20° , 20° , and 0° according to left, right, and front views. Note that the instructing head model was in the same direction with the current head model in yaw rotation. The -20° , 20° , and 0° angles were chosen experimentally so that the compromise between capturing details and stability of HD facial points could be achieved. Once reaching appropriate yaw values, his/her color images were captured automatically. Especially, the Kinect SDK also supports getting facial rectangles and HD facial points in 2-D image spaces. The facial rectangles could be as region-of-interests (ROIs) for cropping facial regions inside the left, right, and front images. Second, reference texture coordinates in 2-D spaces were generated. Subject-specific facial model from the generated head model was flattened into a 2-D circle surface (Figure 19). The boundary of the facial surface was computed and then boundary vertices were mapped onto a parameterized circle in 2-D spaces. The circle had the center at $(0.5, 0.5)$ and the radius of 0.5. The remaining vertices were mapped to the circle region using the harmonic parameterization method [169].

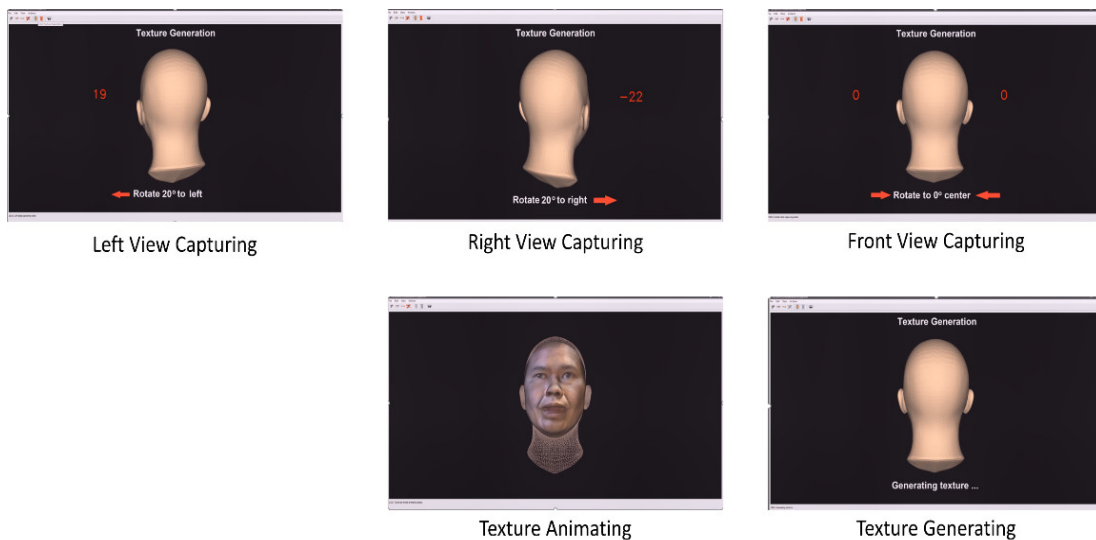


Figure 18. Positional set up during the data capture process for subject-specific texture generation.

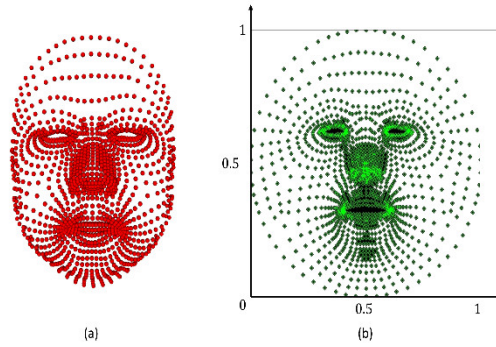


Figure 19. *The subject-specific facial model (a) and its flattened surface (b).*

When the reference frame was established, left, right, and front facial images were then deformed to this reference using the moving least square image deformation method [170] (Figure 20). To deform images to the parameterized space, the flattened HD facial points were chosen as the target control points for these deformations. Note that a hole-filling process was also performed. In fact, colors of black pixels were estimated using the conditional nearest K-search in 2-D spaces. For each black pixel in each deformed image, its ten neighbor pixels that having color data were collected. The average color value of the collected pixels was the color of the current black pixel. Finally, all deformed facial images were merged into a single texture image. One-third on left, middle, and right of the left, front, and right of the facial images were firstly cropped in ROIs. Then, they were merged together to form a single image (Figure 21). To remove boundary effects appeared in left-front and front-right image pairs, the Laplacian blending technique [171] was used. Precisely, the one-third images were first decomposed into multiple levels of resolutions. At each level, each pair of images was merged. The merged images at all decomposing levels were reconstructed to a color image that was the final texture image.

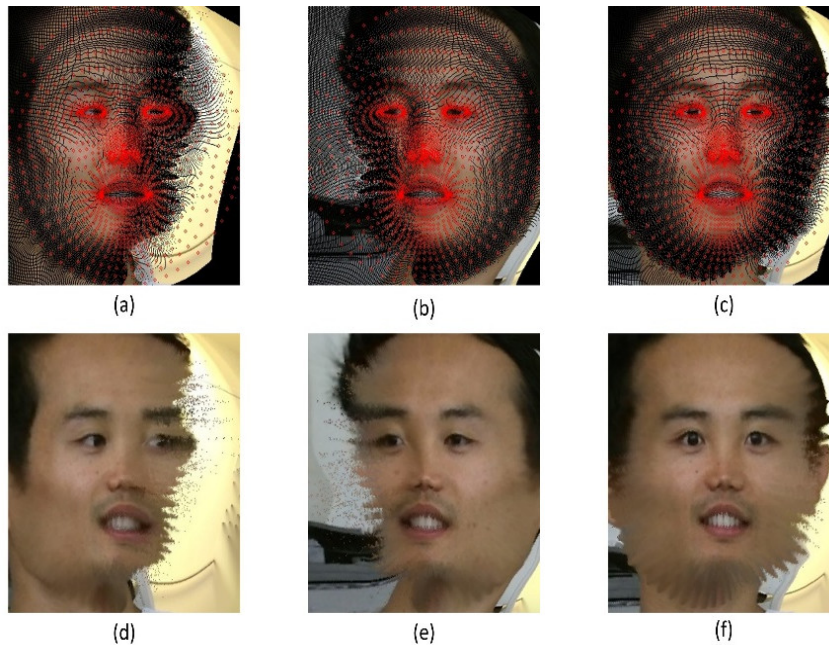


Figure 20. *The deformed facial images before hole-filling in left (a), right (b), and front (c) views; and after hole-filling in left (d), right (e), and front (f) views.*

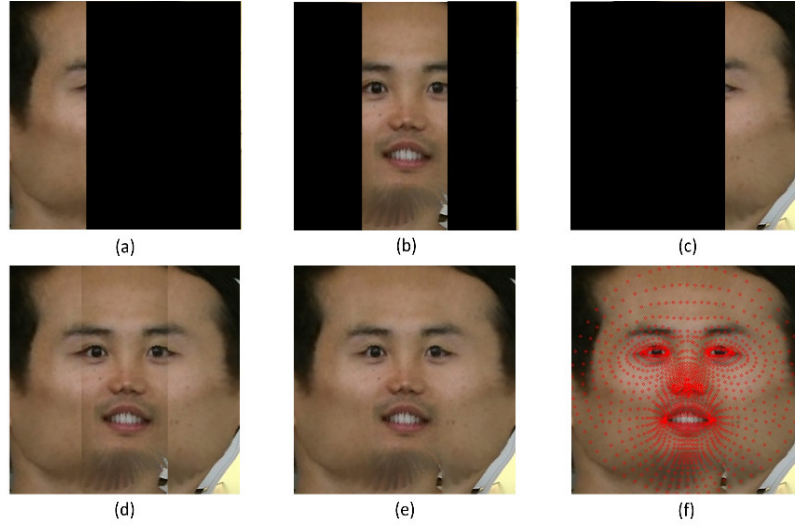


Figure 21. One-third on left (a), middle (b), and right(c) of the facial images and merging results: before boundary-removing (d), after boundary-removing (e), and texture in parameterized coordinates (f).

3.1.5. Head animation subsystem

The output data of the subject-specific head generation subsystem were the subject-specific head model (including head vertices ($\mathbf{H}_s[2582 \times 3]$), facial vertices ($\mathbf{F}_s[1347 \times 3]$), and back head vertices ($\mathbf{B}_s[1297 \times 3]$)), last head position ($\mathbf{P}_l(x_l, y_l, z_l)$), and last head orientation ($\mathbf{I}_l(\text{pitch}_l, \text{yaw}_l, \text{roll}_l)$). Based on the current data of subject-specific facial points ($\mathbf{F}_c[1347 \times 3]$), head orientations ($\mathbf{I}_c(\text{pitch}_c, \text{yaw}_c, \text{roll}_c)$), and head positions ($\mathbf{P}_c(x_c, y_c, z_c)$) from the data acquisition subsystem, the objective of the head animation subsystem was to add both rigid and non-rigid animations to the subject-specific head model in real-time. Note that (x_l, y_l, z_l) and (x_c, y_c, z_c) are the x-, y-, and z- coordinates of the last position and the current position, respectively, of the nearest user's head in the global Euclidean coordinate system. The head positions were acquired from the data acquisition subsystem (Figure 13). $(\text{pitch}_l, \text{yaw}_l, \text{roll}_l)$ and $(\text{pitch}_c, \text{yaw}_c, \text{roll}_c)$ are the pitch, yaw, and roll angles of the last orientation and the current orientation, respectively, of the user's head, and they were also acquired from the data acquisition subsystem.

For the rigid head animations, rigid transforms including translations and rotations of the subject-specific head vertices were computed. Firstly, head orientation differences ($\mathbf{I}_d(\text{pitch}_d, \text{yaw}_d, \text{roll}_d)$) and head position differences ($\mathbf{P}_d(x_d, y_d, z_d)$) were calculated using the following equations:

$$\mathbf{I}_d(\text{pitch}_d, \text{yaw}_d, \text{roll}_d) = \mathbf{I}_c - \mathbf{I}_l = (\text{pitch}_c - \text{pitch}_l, \text{yaw}_c - \text{yaw}_l, \text{roll}_c - \text{roll}_l) \quad (4)$$

$$\mathbf{P}_d(x_d, y_d, z_d) = \mathbf{P}_c - \mathbf{P}_l = (x_c - x_l, y_c - y_l, z_c - z_l) \quad (5)$$

In which, $(\text{pitch}_d, \text{yaw}_d, \text{roll}_d)$ are the pitch, yaw, and roll different angles between the current orientation and the last orientation of the user's head. (x_d, y_d, z_d) are the x-, y-, and z-different coordinates between the current position and the last position of the user's head.

Then, rotation matrices around x-, y-, and z- axis were estimated based on these differences. These rotation matrices were computed using the following equations:

$$\mathbf{R}_x(\text{pitch}_d) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \text{pitch}_d & -\sin \text{pitch}_d \\ 0 & \sin \text{pitch}_d & \cos \text{pitch}_d \end{bmatrix} \quad (6)$$

$$\mathbf{R}_y(\text{yaw}_d) = \begin{bmatrix} \cos \text{yaw}_d & 0 & \sin \text{yaw}_d \\ 0 & 1 & 0 \\ -\sin \text{yaw}_d & 0 & \cos \text{yaw}_d \end{bmatrix} \quad (7)$$

$$\mathbf{R}_z(\text{roll}_d) = \begin{bmatrix} \cos \text{roll}_d & -\sin \text{roll}_d & 0 \\ \sin \text{roll}_d & \cos \text{roll}_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Moreover, original translation vectors ($\mathbf{T}_o[2582 \times 3]$) and current translation vectors ($\mathbf{T}_c[2582 \times 3]$) that translate the head vertices to the original coordinate and back to the current position were defined as follows:

$$\mathbf{T}_o = \begin{bmatrix} x_l & y_l & z_l \\ \vdots & \vdots & \vdots \\ x_l & y_l & z_l \end{bmatrix} \quad (9)$$

$$\mathbf{T}_c = \begin{bmatrix} x_d & y_d & z_d \\ \vdots & \vdots & \vdots \\ x_d & y_d & z_d \end{bmatrix} \quad (10)$$

Finally, the rigid-transformed vertices of the head model ($\mathbf{H}_c[2582 \times 3]$) were computed based on the following equation:

$$\mathbf{H}_c^T = [\mathbf{R}_z \mathbf{R}_y \mathbf{R}_x (\mathbf{H}_s^T - \mathbf{T}_o^T)] + \mathbf{T}_o^T + \mathbf{T}_c^T \quad (11)$$

Once having the rigid head vertices (\mathbf{H}_c), non-rigid facial animations were performed by replacing the facial vertices by the current facial points (\mathbf{F}_c) from the data acquisition subsystem. With this strategy, the computation costs was significantly reduced due to just computing the rigid transformations of the subject-specific head model.

3.1.6. Mirror effect subsystem

In the mirror effects, the left/right face region of a user was mirrored to his/her right/left face region, so facial movements on the left/right face side would be the same as ones on the right/left face side after left/right mirrored. Face symmetries in different mimics could then be measured by compared the mirrored face model with the original face model using the Hausdorff distance metric [166]. The left/right mirror effect included model and texture left/right mirror effects.

The procedure of model left/right mirror effect is shown in Figure 22. Inputs of the model mirror effects were the left/right face model and left/right mirror plane features. The left/right plane features were selected as bottom face point, top face point, and top nose point on the left/right face model. These feature points were used for representing the left mirror plane ($a_l x + b_l y + c_l z + d_l = 0$) and the right mirror plane ($a_r x + b_r y + c_r z + d_r = 0$). The Householder reflection method [172] was employed to form the left/right reflecting transform matrix from the left/right mirror plane. The left mirror transform matrix was formed as

$$\begin{bmatrix} 1 - 2a_l^2 & -2a_l b_l & -2a_l c_l & -2a_l d_l \\ -2a_l b_l & 1 - 2b_l^2 & -2b_l c_l & -2b_l d_l \\ -2a_l c_l & -2b_l c_l & 1 - 2c_l^2 & -2c_l d_l \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

The right mirror transform matrix was formed as

$$\begin{bmatrix} 1 - 2a_r^2 & -2a_r b_r & -2a_r c_r & -2a_r d_r \\ -2a_r b_r & 1 - 2b_r^2 & -2b_r c_r & -2b_r d_r \\ -2a_r c_r & -2b_r c_r & 1 - 2c_r^2 & -2c_r d_r \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

in which the $a_{l/r}$, $b_{l/r}$, $c_{l/r}$, and $d_{l/r}$ are plane parameters computed from the left/right mirror plane features. In the left mirror effect, the left face vertices were transformed using the left reflecting transform matrix to form the reflected left face vertices. The left and reflected left face vertices were replaced to the vertices of the generic face model to form the left-mirrored face model. In the right mirror effect, the right face vertices were transformed using the right reflecting transform matrix to form the reflected right face vertices. The right and reflected right face vertices were replaced to the vertices of the generic face model to form the right-mirrored face model.

The procedure of left/right texture mirror effect is shown in Figure 23. The input of this procedure is the texture image generated from the subject-specific texture generation subsystem. First, the texture image was horizontally flipped. In the left texture mirror effect, the right side of the flipped texture image was merged to the left side of the texture image to form the left-mirrored texture image. In the right mirror texture effect, the left side of the flipped texture image was merged to the right side of the texture image to form the right-mirrored texture image. The texture coordinates were kept the same as ones in the normal textures. Finally, left/right asymmetries during facial mimics were computed as Hausdorff distances between the original animated and left/right-mirrored face models in real-time. This objective and quantitative asymmetry could be used as real-time bio-feedback for non-clinical facial paralysis grading.

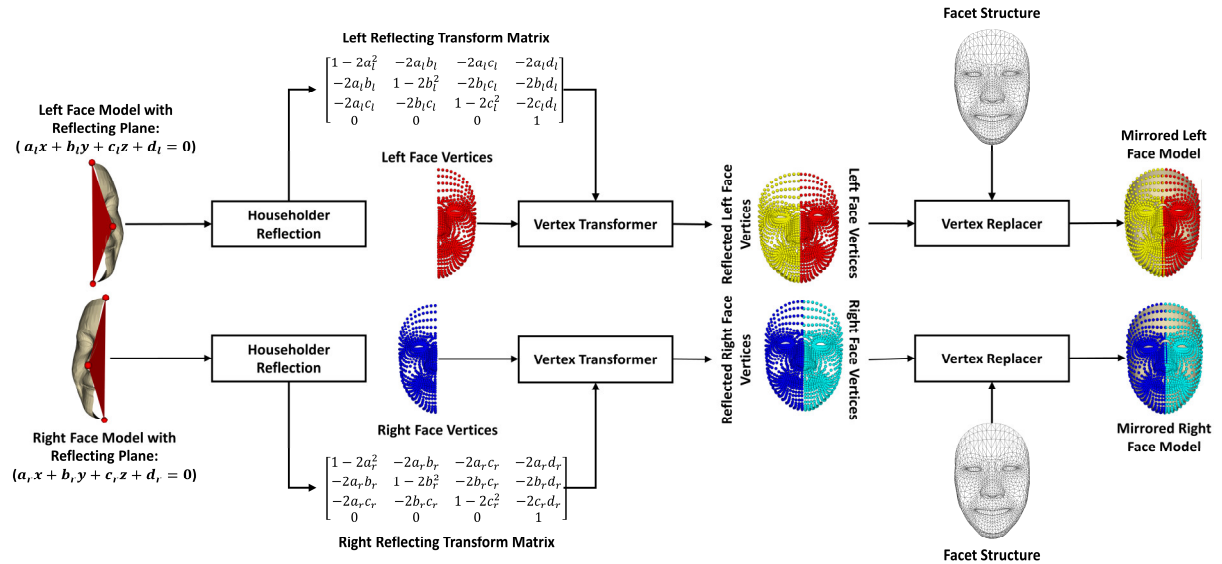


Figure 22. Model left/right mirror effect.

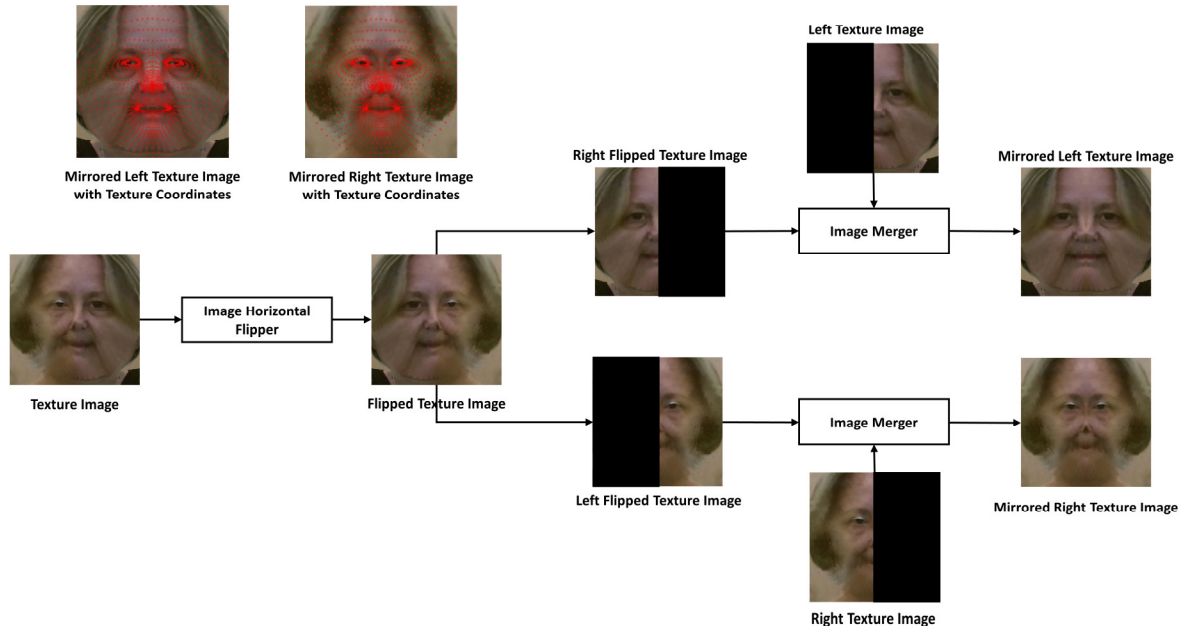


Figure 23. *Texture left/right mirror effect.*

3.1.7. System development technologies

The system was designed based on the hardware configuration of Intel® Core™ i7-3720QM CPU @ 3.5GHz 16GB DDRAM with the graphic card NVIDIA Quadro K2000M 745MHz 2GBytes. The system was programmed in Visual Studio C++ 2015. The system framework and GUIs were supported by the Qt 4.7.0 C++ platform [173]. Models and textures were rendered in the graphic card using VTK 7.1.1 [162].

3.1.8. Model accuracy and system performance analyses

The accuracy of the generated subject-specific head model was evaluated using magnetic resonance imaging (MRI) data of the same subject on the facial regions. MRI data were acquired on two healthy subjects (one female and one male) in neutral position at the University Hospital Center of Amiens (Amiens, France). Moreover, two female facial palsy patients with facial asymmetries were asked to verify the mirror effect function of the system. All subjects have signed an informed consent agreement before participating into the data acquisition process. The protocol was approved by local ethics committee (n°2011-A00532-39). Then, acquired MRI images were processed by using 3-D Slicer open-source software [174] to reconstruct the ground-truth geometrical models. The developed system was used on the same subjects in neutral position to generate subject-specific head models. Then, the comparison between each generated Kinect-based model and MRI-based model in facial regions of the same subject was performed using Hausdorff distance metrics [166]. To compare the reconstructed MRI-based head model must be transformed to the same coordinate system with the generated Kinect-based head model. The registration procedure is described in Figure 24. Only facial regions of the two head models were used for estimating the transform matrix. First, anatomical landmarks on both Kinect-based and MRI-based face models were manually selected. The rigid transform matrix for transforming the MRI-based landmarks to the Kinect-based landmarks was then estimated using the singular value decomposition (SVD) rigid registration method [175]. After applying the SVD rigid transform matrix to the MRI-based face model, the MRI-based face model was relatively on the same coordinate with the Kinect-based face model. However, because of errors during the manual landmark selection, the two face models were not well registered. Consequently, the iterative closest point (ICP) registration method [175] was employed for estimating the best rigid transform so that the rigid differences between the

two facial models were minimized. After registration, Hausdorff distances between the two facial regions were computed. Distance map was also visualized.

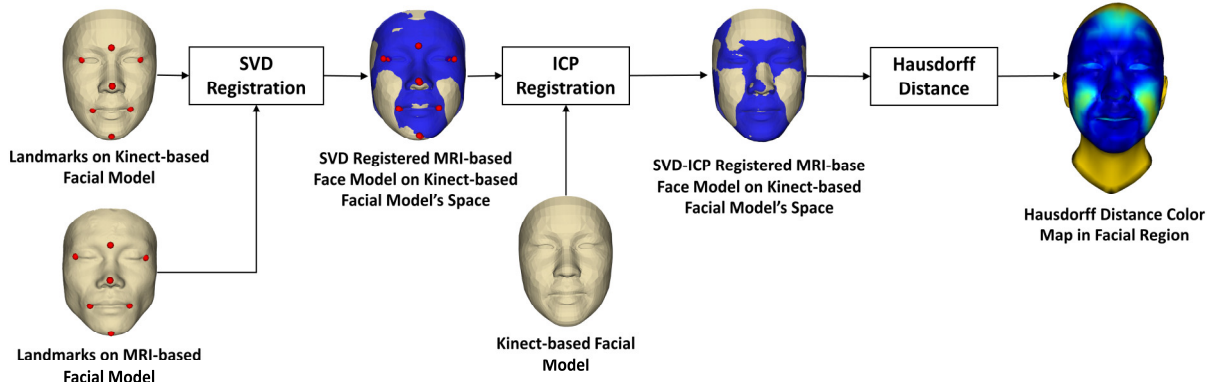


Figure 24. The transformation procedure for registering the MRI-based head model to Kinect-based head model on facial regions.

In addition, model accuracy was also assessed for different facial mimics using subject-specific models reconstructed with the depth data from RGB-D sensor. The reconstruction procedure is shown in Figure 25. A 3-D point cloud of a user in the neutral facial mimic was first computed. In this stage, when a depth image was available in the Kinect SDK, we converted the depth values to a 3-D point cloud (Figure 25a) using an available conversion function from the Kinect SDK. The head point-cloud (Figure 25c) was then searched using Kd-Tree nearest neighbor radius search algorithm [176]. The center of the radius-based search was the current head position ($\mathbf{P}_c(x_c, y_c, z_c)$) of the nearest user acquired from the Kinect SDK. The radius value ($\sim 170\text{ mm}$) was experimentally selected so that all head points were in the searching sphere (Figure 25b). As shown in Figure 25c, the selected head point-cloud still has some outliers after the radius-based search. To remove these outliers, a statistical outlier removal algorithm was used [176]. Currently, the filtered head point cloud (Figure 25d) was ready for surface reconstructions. The Poisson surface reconstruction Delaunay method [177] was applied to generate a surface triangle mesh (Figure 25e) from the filtered head point-cloud. Finally, the Laplacian smoothing method [178] was applied to increase the smoothing level of the reconstructed head surface from points (Figure 25f). The point-cloud-based reconstruction procedure was repeated to other point-clouds captured during pronunciations of different facial mimic movements like smiles and sounds ([e], [u]). Note that the captured point-clouds and the generated head models were in the same coordinate systems, so the reconstructed head models were automatically registered with the generated head models. Moreover, the generated head models do not include subject-specific hairs and ears. Thus, only facial regions in the generated head models were compared with ones in the reconstructed head model. For each vertex in the generated facial models we find its projected point on the reconstructed head surfaces. The projecting directions were from the head position to the vertices of the generated facial model. Distances between the facial vertices and the projected points were computed and analyzed.

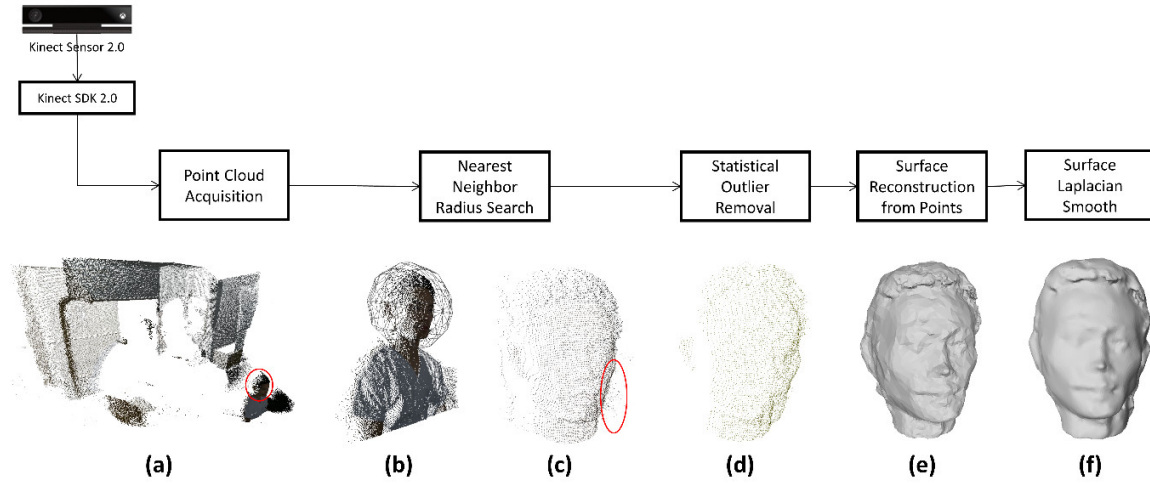


Figure 25. Head reconstruction procedure from a depth sensor data: (a) scene point cloud, (b) nearest neighbor searching sphere, (c) head point-cloud and its outliers, (d) head point-cloud after outlier removal, (e) reconstructed head model from points, and (f) reconstructed head model after smoothing.

Reproducibility and repeatability tests were conducted to evaluate the system stability. The system had controlled by 3 different operators to generate the specific models. Moreover, each operator has repeated the acquisition task in 10 times. Distance map and statistical information were used for comparison purpose. Furthermore, effects of illuminations on the system's behaviors were quantified. Precisely, both model and texture generation procedures were executed on very low, low, average, high, and very high light conditions, and their results are analyzed. Finally, model generation time and visual framerates were analyzed.

3.2. Results

3.2.1. Model validation with MRI data in neutral position

The reconstructed models from MRI sets of two subjects in neutral position are shown in Figure 26. Their respective head models were also generated with the developed system (Figure 27). The deviation distance maps between reconstructed MRI-based models and generated Kinect-based models are also illustrated in Figure 26. The distance distributions of the two distance maps are shown in Figure 28. The median errors of the two volunteers are approximately 1 mm. Note that the variability patterns of the two distance maps are different. The first variability is larger than the second one due to the appearance of deeper dimples in the face of the volunteer 1 than ones of the volunteer 2. Moreover, outliers in the two distributions show that the back regions of the generated head model have large errors due to lacks hairs, shoulders, and user-specific ears in these models.

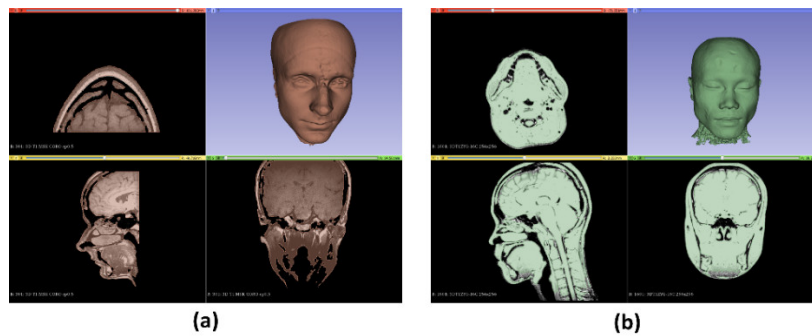


Figure 26. The reconstructed head models from MRI sets in neutral position of the female subject #01 (a) and the male subject #02 (b).

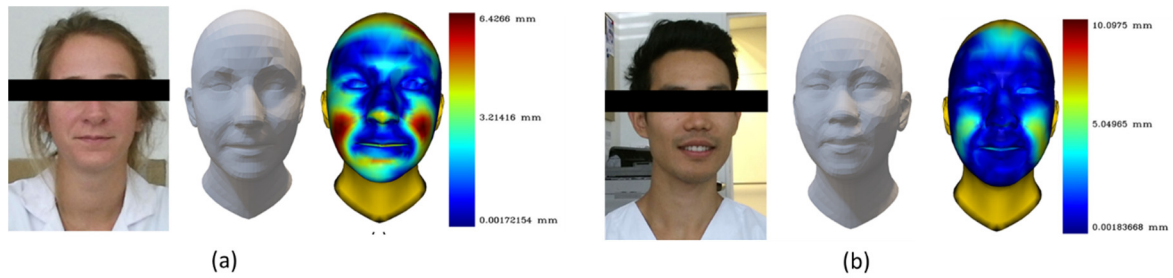


Figure 27. The generated head models from Kinect sensor and their Hausdorff distance maps in facial regions according to MRI-based models in neutral position of the female subject #01 (a) and the male subject #02 (b).

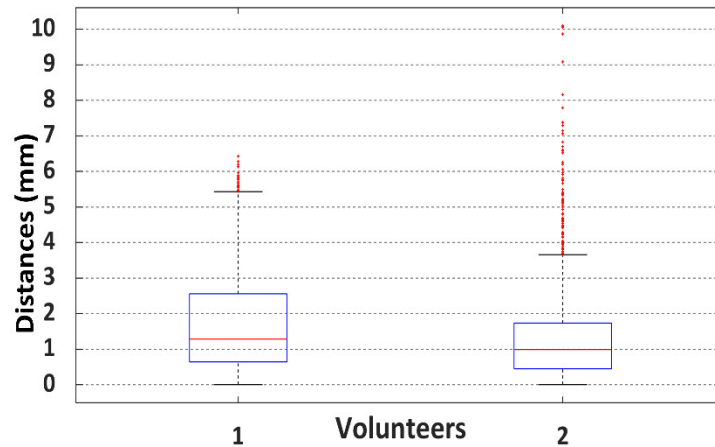


Figure 28. The distributions of the distance maps between reconstructed MRI-based models and generated Kinect-based models of the female subject #01 (a) and the male subject #02 (b).

3.2.2. Model validation with depth sensor data in different facial mimic positions

The comparison between generated models and models reconstructed from depth sensor of the male subject #02 is shown in Figure 29 and Figure 30 for different facial mimic positions. Overall, the median errors in different facial mimics are around 2 mm (from 1.8 mm to 2.2 mm). However, the number of outliers is different. In the neutral facial mimic, the median error is lowest (1.8 mm), and the number of outliers is smallest. When smiling and pronouncing sounds (e.g. [e] and [u]) the number of outliers gradually increases.

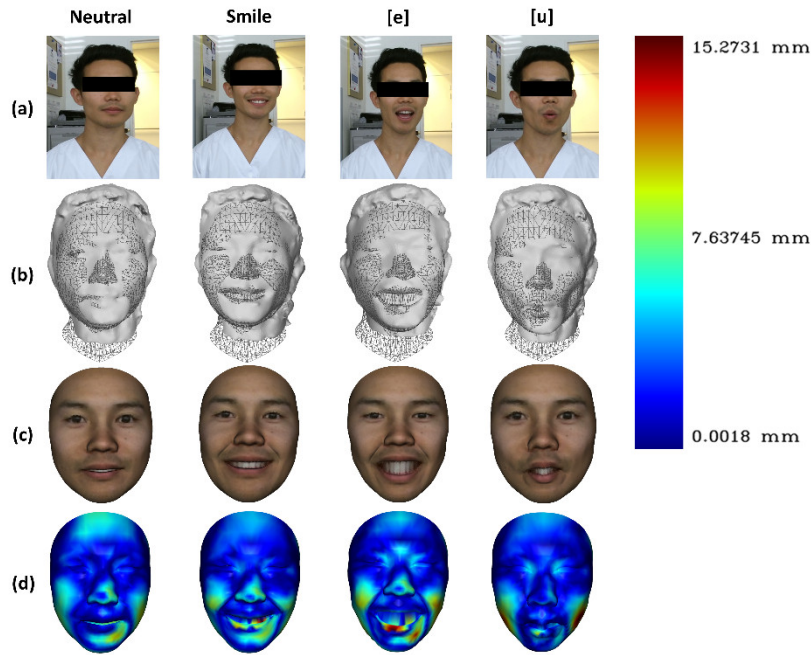


Figure 29. Comparison between the point-cloud-based reconstructed head models and the generated head models of the male subject #02 in different facial mimics: (a) Facial images, (b) reconstructed models and generated models in the same coordinate systems, (c) generated facial models with textures, and (d) distance color maps in facial regions.

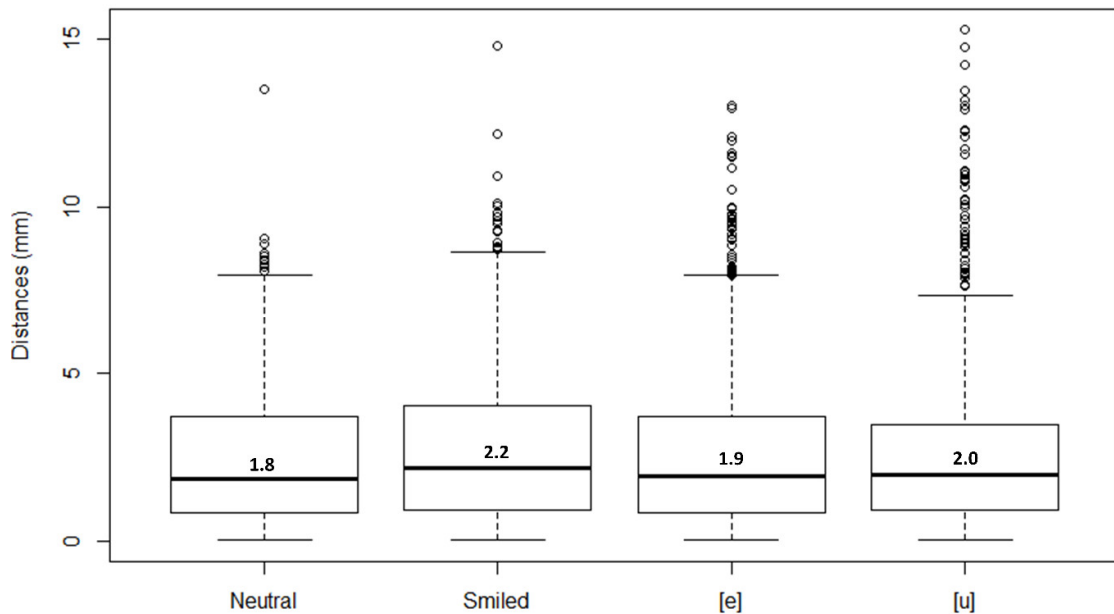
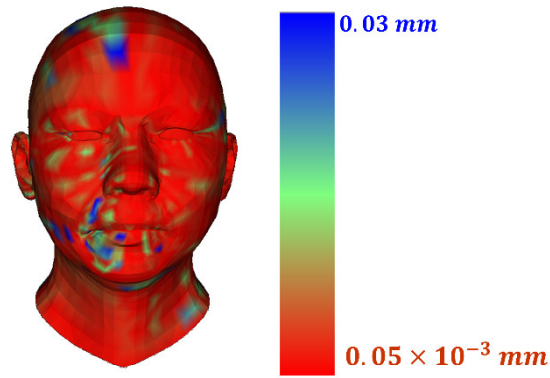


Figure 30. Statistical results of errors in facial regions between the reconstructed head models and the generated head model in different facial mimic positions: neutral, smile, sounds ([e] and [u]).

3.2.3. System reproducibility and repeatability test

The reproducibility and repeatability test show that the minimum errors are between $0.05 \times 10^{-3} \text{ mm}$ and $0.03 \times 10^{-2} \text{ mm}$, and the maximum errors are between 0.03 mm and 0.14 mm . Note that some errors appeared due to the animations of user faces during the acquisition states (Figure 31).



Error range (10 times, 3 testing subjects, 3 operators):
 min = $[0.05 \times 10^{-3} \ 0.03 \times 10^{-2}]$ mm, max = $[0.03 \ 0.14]$ mm

Figure 31. The average differences from the reproducibility and repeatability test on the developed system.

3.2.4. Illumination effect analysis

Because of high dependences of RGB cameras on light conditions, texture images were highly affected by illuminations (Figure 32). However, the geometrical animations were rarely affected, so the subject-specific head model was well stabled in wide ranges of light-levels in the non-texture animation mode. Specially, if the texture image had generated in advanced, the head model with texture could be animated naturally in very low light conditions.

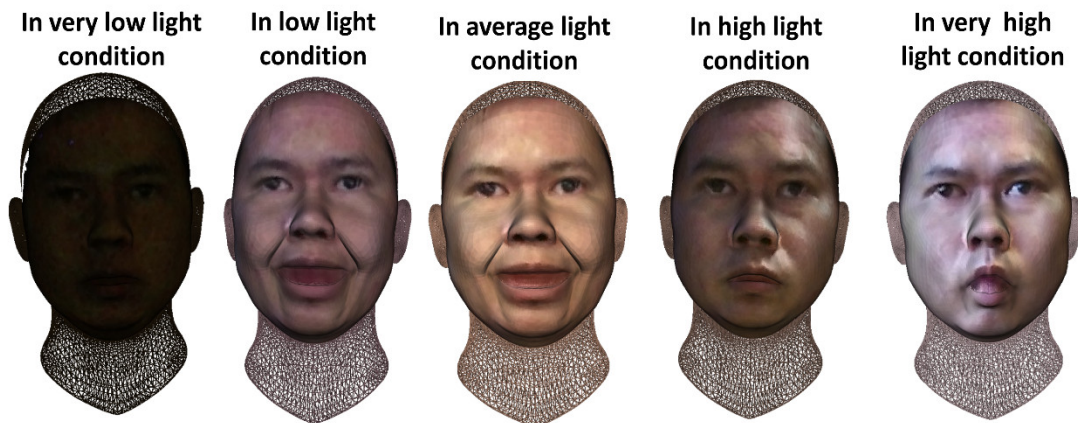


Figure 32. Head animations in different illumination conditions.

3.2.5. Model generation time and visual framerates

Model generation time are highly depended on the number of input facial points. If the high-definition facial datasets were used, the generating time was about 9.7 ± 0.3 . Alternatively, in the case of using the MPEG-4 facial datasets for generating the head model, the generating time was just about 0.0562 ± 0.005 . However, the user-specific level of the back-head region was lower when using the MPEG-4 datasets. Figure 33 shows differences between the two head models generated using the high-definition facial points and MPEG-4 facial points from the Kinect. The illustration shows that differences appear mostly on the back-head surface, but in the facial surface the differences are nearly reached to zeros.

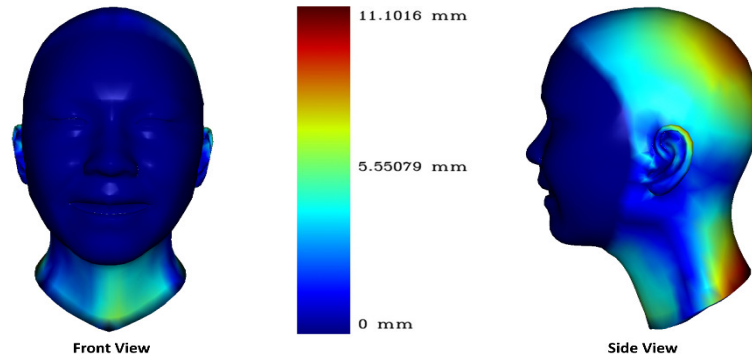


Figure 33. Differences between two generated models using the HD facial point set and the MPEG-4 facial point set.

Model animation framerates are also affected by rendering qualities in the GUI subsystem. To increase the rendering quality, the loop subdivision method was employed [179]. Figure 34 shows three levels of rendering qualities and their appropriate framerates. If the animated model kept its original quality, the system framerate could be up to 60 fps. When doubling or quadrupling the number of model vertices, the framerates were also decreased to 35 fps and 11 fps respectively. The rendering qualities can be controlled by users in the GUI subsystem. Both real-time framerates and high-rendering quality can be achieved if more powerful hardware configuration will be used.

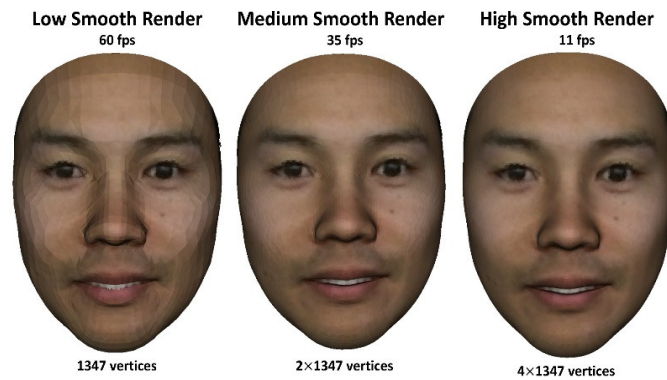


Figure 34. Different levels of rendering qualities and their appropriate framerates.

3.2.6. A demonstration of system functionalities and animations

Figure 35 shows the current execution workflow of the developed system. There are three execution modes. For the first use of the system, the generating time of a subject-specific head model is 9.7 ± 0.3 s. After generating the model, the system automatically changes to the mode of model animations without texture (Figure 36a). At this state, the system can reach a framerate up to 60 fps. When texture information is required, the generation time is 29.5 ± 0.2 s. After the texture generation, the system will render the generated texture image into the animated geometrical model with the framerate up to 60 fps (Figure 36b). For the next use of the system, the subject-specific data have been saved, so the user does not have to generate his/her model and texture. The model animation mode with or without texture can be directly selected for animating the rigid head and non-rigid facial mimic movements in real-time. Note that plausible ranges of head orientations and positions during animations are reported in Table 3.

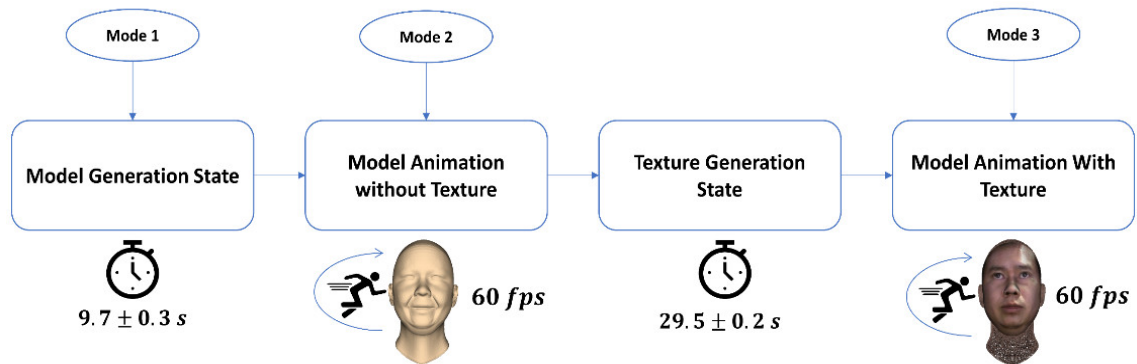


Figure 35. Different levels of rendering qualities and their appropriate framerates.

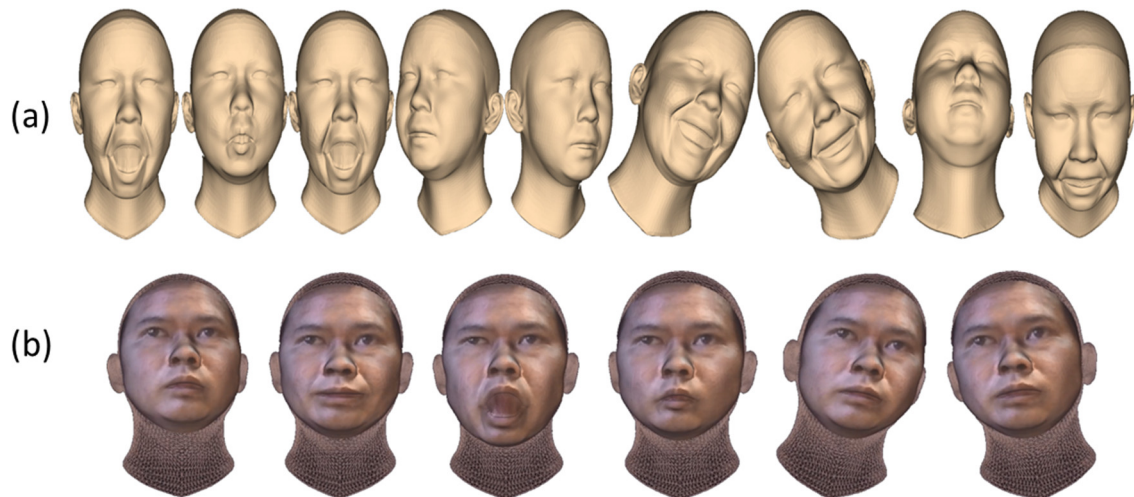


Figure 36. Rigid head and non-rigid facial mimic movements without texture information (a) and with texture information (b).

Table 3. Plausible ranges of head orientations and positions.

Parameters	Plausible Ranges [<i>min</i> ; <i>max</i>]
Yaw Angles ($^{\circ}$)	$[-47.76^{\circ}; 52.01^{\circ}]$
Roll Angles ($^{\circ}$)	$[-89.79^{\circ}; 85.88^{\circ}]$
Pitch Angles ($^{\circ}$)	$[-51.40^{\circ}; 46.17^{\circ}]$
z-coordinates	[0.59m; 4.58m]
FOV ($^{\circ}$) (H / V)	$70^{\circ} / 60^{\circ}$
Framerate (fps)	60 fps

3.2.7. Rendering and Hausdorff distance outcomes of mirror effects

Once having the left/right mirrored face models, texture images, and texture coordinates, the user could select one of the rendering modes: left-mirrored rendering, right-mirrored rendering, texture left-mirrored rendering, and texture right-mirrored rendering. In left/right-mirrored rendering mode, left/right-mirrored face model and left/right-mirrored texture image were rendered in the GUI system. In texture left/right-mirrored rendering mode, original animated face model and left/right-mirrored texture image were rendered in the GUI system. All rendering modes of a female facial palsy patient were illustrated in Figure 37. The left/right mirrored face models were also compared with the animated face models using the Hausdorff distance metrics. The rendering and Hausdorff distance outcomes of two healthy subjects and two facial palsy patients are shown in Figure 38. For healthy subjects, there were no significant

differences between the original faces/textures and the left/right-mirrored faces/textures. However, for facial palsy patients, the texture left/right-mirrored rendering was more beautiful than the normal rendering. Moreover, note that the mean Hausdorff distances between the left/right-mirrored face models and the animated face models of healthy subjects were smaller than ones of the facial palsy patients, as the patients could not have symmetries in their facial mimics. Especially, face symmetries could not be perfectly existed even in the healthy subjects.

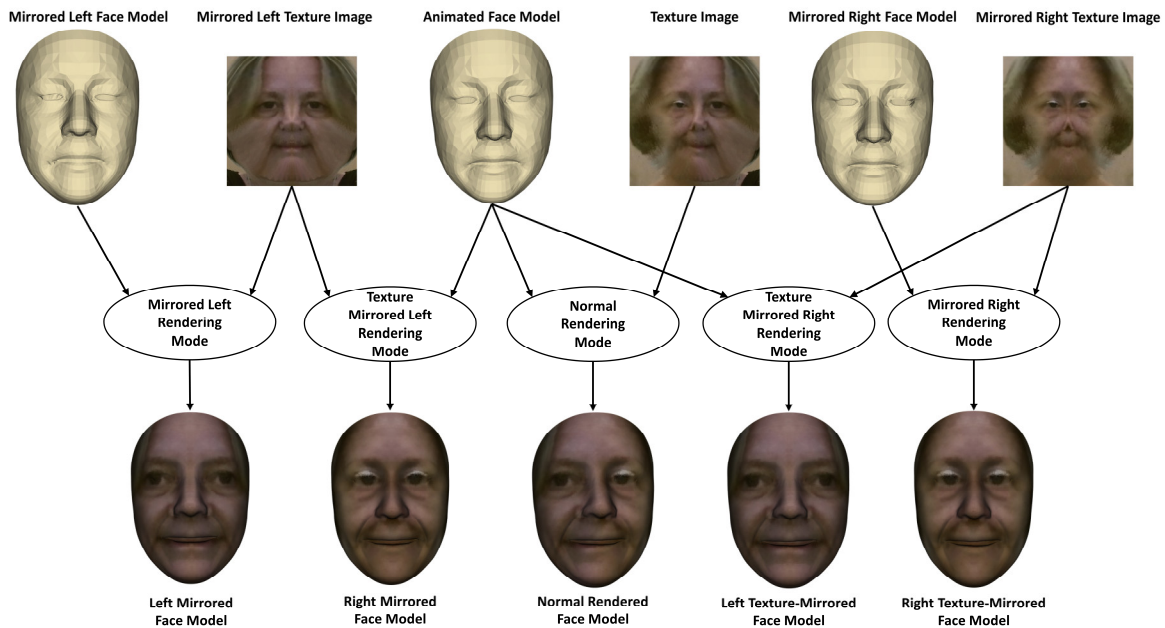


Figure 37. Textured face models in different rendering modes of the female facial palsy patient.

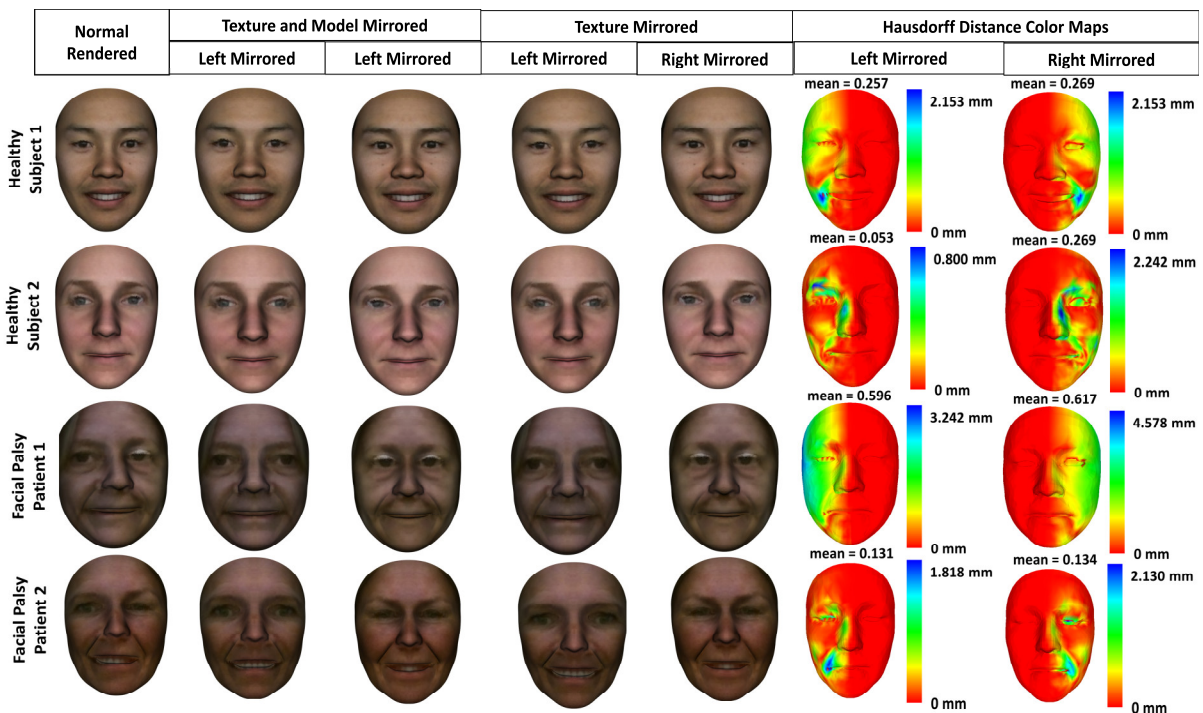


Figure 38. Rendering and Hausdorff distance outcomes of the mirror effects. Note that the asymmetrical levels of the facial palsy subjects are much larger than ones of the healthy subjects.

3.3. Discussion

To develop a computer vision system for facial rehabilitation with real time feedback, the animations of the rigid head and non-rigid facial mimic movements need to be executed and tracked in real-time condition. Many research works have been investigated on this challenging

topic [31,36–41,43–56]. However, it is still very challenging when tracking simultaneously rigid head and non-rigid facial mimics in 3-D spaces and in a subject-specific manner with a real-time framerate. To solve this challenge, the present study proposed a novel computer vision system using a contactless Kinect sensor, a system of systems approach and fast computational procedures. System of systems engineering strategy allows us to integrate different autonomous systems (e.g. data acquisition, head model generation, texture generation, head and facial animation, mirror effect, and GUI) into a larger collaborative system to achieve a unique set of tasks related to the facial mimics monitoring and rehabilitation. Furthermore, a multi-level evaluation process was also performed on the developed system to show a good accuracy level and high performance of our system in the real use cases.

Several head and facial mimic animation systems have been developed by using different computational methods and input devices to simulate head and/or facial mimic animations. These systems can hardly achieve both real-time framerates and subject-specific animations. Table 4 lists these existing animation systems. It is important to note that the achievement of real-time framerates remains a challenge. Moreover, the achievement of subject-specific model generations and animations adds more complexity and computational resources. Using single cameras, the very low framerates were from 5.52×10^{-3} fps to 12.65×10^{-3} fps in [37,43,51] because of their targets at generating subject-specific models rather than animating them. Others single-camera-based studies could only reach relatively slow framerates from 2 fps to 27 fps because of large computational costs for detecting and tracking facial points and deforming 3-D models from 2-D images. The laser scanners were only used for generating high-details 3-D model, so they focused on improving graphical qualities rather than generation speeds [31]. The Kinect sensors (version 1.0) could also be employed for reconstructing facial mimics, but the framerate was only 2.85 fps because of long processing time of low-level point cloud registrations and mesh generation [41]. Some existing systems could achieve real-time framerates, but they lacked of subject-specific models, realistic animations, and full animation parts. Using single cameras, only one study could accomplish a real-time framerate of 100 fps [53]. However, this animation system had some drawbacks. The high-detail subject-specific with texture head model was generated offline by another laser scanner-based system, so there was no subject-specific model generation process. Moreover, head animations were performed in a game engine using the pose linear deformation technique, so skin deformations could not be included in the animated models. Additionally, because facial features were detected and tracked in 2-D color images, this system could not work in very low illumination conditions. Some studies could also use stereo cameras to animate head/facial model in real-time. Wan et al., 2012 [54] used a stereo camera to detect and track facial markers in 3-D spaces. Using geodesic distance-based radial basic function (RBF) interpolation method, the movements of facial markers were transferred to deformations of a head model. Although the system could achieve a real-time framerate of 31.65 fps, the animated model was not subject-specific. Moreover, detections of facial makers were based on color image processing, so they were highly affected by low illumination conditions. Specially, the number of facial makers being able to put on the user face was limited. Ouni et al., 2018 [56] employed electromagnetic sensors as facial markers to track movements of a user's lips. Although the system framerate was relatively high, up to 140 fps, only lip regions were simulated. Moreover, the wires and limited number of sensors made the system difficult to facilitate on other users. The Kinect sensors (version 1.0) could also be used in a real-time head animation system [55]. In this system the framerate was 30 fps, but the head model was again not subject-specific. Facial points were detected on 2-D RGB images and converted to 3-D spaces using depth images, so the system could not work in very low illumination environments. According to these systems, our system can support a full subject-specific rigid head and non-rigid facial mimic animations with texture information in real-time. Our achieved system framerate (60 fps) can be faster than those of most studies using single cameras, laser scanners, stereo cameras, and the Kinect sensors. Although some studies whose framerates were higher than of our system, our rigid head and

non-rigid facial animations are subject-specific. In particular, the balance between graphical rendering qualities and system framerate can be controlled by users from the GUI subsystems to adapt with current hardware configurations. Compared with the studies of Marcos et al., 2008 [53] and Ouni et al., 2018 [56], although their framerates were up to 100 FPS, our non-rigid facial animations were more realistic. As shown in Figure 36, the animated head models can express skin deformations around mouth and chin regions during pronunciations of sounds ([o], [pɹ]) and smiles. This could not be done in [53]. Moreover, full head animations including rigid head and non-rigid facial mimics movements are simulated in our system while only mouth regions could be animated in [56]. The Kinect sensor (version 1.0) was used by Li et al., 2013 to capture 3-D motions of facial features, and they animated a 3D avatar in the 3ds Max software [55]. Although the framerate was 30 fps, the avatar was not subject specific. Because facial features were captured and tracked based on RGB color image sequences, the tracking stability and quality were negatively affected by illumination conditions. Again, skin deformations were not included in avatar's animations. Excepting laser-scanner-based and point-cloud-based non-real-time reconstructing systems, all above studies were mainly based on facial points detected from RGB images to animate facial models, so their animation qualities were negatively affected by illumination conditions. Our system can work well in different light conditions. As shown in the Figure 32, even in extremely low illuminations, head animations can be well performed. Light conditions just affect to the texture quality; consequently, if the texture was available in advanced, the system could work in very low light conditions. Furthermore, numerous systems just animated non-subject-specific models. Even though the subject-specific models were used, most studies could not integrate both model personalization and animation into system functions; therefore, the system abilities cannot easily apply to new users. In our system, personalization procedures can automatically instruct new users to capture their personal data for generating subject-specific head models and facial textures. Last but not least, rarely studies conducted a multi-level system evaluation process to determine model accuracy, system reproducibility and repeatability, illumination effect and system performance. In fact, with an acceptable average error (~1 mm) and a real-time framerate (60 fps), our developed system could be applied for tracking and animating facial rehabilitation movements. Moreover, Hausdorff distances between left/right-mirrored face model and animated face model could be used for real-time face symmetry evaluations in facial paralysis grading systems.

Table 4. Comparison with existing head and facial mimic animation systems

Studies	Movements	Interaction Devices/Input data	Model generations	Framerate (fps)	Evaluations
This present system	Rigid head and non-rigid facial mimics (eye and mouth movements, skin deformations)	Kinect Sensor V2.0 (Face HD Points, color images, head orientations and positions)	Full subject-specific head models (9.7±3 s) and textures (29.5±2 s)	60	Multi-level process (visual assessment, model accuracy, system reproducibility and repeatability, illumination effect)
Yin et al., 2001 [37]	Non-rigid facial mimics (eye and mouth movements)	Single cameras (Single color images)	Subject-specific facial models and textures	12.65 × 10 ⁻³	Reproducibility and repeatability, visual assessments
Goto et al., 2001 [45]	Non-rigid facial mimics (eye and mouth movements)	Single cameras (Two orthogonal color images) Microphone (speech signals)	Full generic head models and textures	15	Visual assessments

Studies	Movements	Interaction Devices/Input data	Model generations	Framerate (fps)	Evaluations
Chandrasiri et al., 2004 [46]	Non-rigid facial mimics (mouth movements)	Webcam (video sequences) PC mouse and keyboard (texts)	Full generic head models and textures	12	Visual assessments, model accuracy, user-acceptability
Choi et al., 2005 [47]	Rigid head movements	Webcam (Video sequences) Microphone (speech signals)	Full generic head models with subject-specific textures	9	Visual assessments, user-acceptability validations
Zha et al., 2007 [49]	Non-rigid facial mimics (mouth movements)	Single cameras (Video sequences) Microphone (speech signals)	Full subject-specific head models and textures	5.78	Visual assessments
Liu et al., 2007 [48]	Non-rigid facial mimics (mouth movements)	Single cameras (front and half-profile images)	Subject-specific facial models and textures	6.67	Visual assessments
Fu et al., 2008 [50]	Rigid head and non-rigid facial mimics (mouth movements)	Webcam (Live videos) Keyboard (texts) Microphone (speech signals)	Full generic head models and textures	27	Visual assessments, reproducibility and repeatability.
Marcos et al., 2010 [53]	Rigid head and non-rigid facial mimics (eye and head movements)	Webcam (video sequences) Microphone (speech signals)	Subject-specific facial models and textures	100	Visual assessments, system accuracy, user-acceptability
Wang et al., 2011 [51]	Non-rigid facial mimics	Single cameras (single facial color images)	Subject-specific facial models with textures	3.33×10^{-3}	Visual assessments, system accuracy
Wan et al., 2012 [54]	Non-rigid facial mimics (mouth movements)	Vicon optical motion capture (stereo image-pairs) Facial markers	Full generic head models and textures	31.65	Visual assessments, hyperparameter tuning
Song et al., 2012 [52]	Non-rigid facial mimics (Full facial movements)	Single cameras (Single images)	Subject-specific facial models without textures	1.69	Visual assessments, system accuracy
Li et al., 2013 [55]	Rigid head and non-rigid facial mimics (eye and mouth movements)	Kinect sensors V1.0 (Single facial color images, depths)	Full generic head models and textures	30	Visual assessments
Luo et al., 2014 [38]	Non-rigid facial mimics (mouth movements)	Ordinary camera (Single color images)	Partial subject-specific facial models with textures	22	System accuracy, visual assessments
Yu et al., 2015 [39]	Rigid and non-rigid facial mimics (mouth movements)	Single cameras (Video sequences) Microphone (speech signals) Keyboards (texts)	Subject-specific facial models and textures Generic tongue models	19.6	Visual assessments, User-acceptability validations

Studies	Movements	Interaction Devices/Input data	Model generations	Framerate (fps)	Evaluations
Hernandez et al., 2015 [36]	No movements	The PrimeSense camera (RGB-D images)	Partial subject-specific head models and textures	10	System accuracy, visual assessments
Liang et al., 2016 [40]	Non-rigid facial animations	Single cameras (Single color images)	Full subject-specific facial models without textures	4.76	System accuracy, visual assessments
Zhang et al., 2016 [31]	Non-rigid facial mimics (mouth movements)	Laser scanners (3D point clouds)	Full generic head models with textures	75.18×10^{-3}	System accuracy, visual assessments
Zhan e al., 2017 [41]	No movements	Kinect sensors V1.0	Partial subject-specific head models without textures	2.85	Visual assessments
Ouni et al., 2018 [56]	Non-rigid facial mimics (mouth movements)	Articulography AG501 – Electromagnetic sensors (3D or 5D coordinates of facial markers)	Full generic head models and textures	140	System accuracy, visual assessments
Jiang et al., 2018 [43]	Non-rigid facial mimics (skin deformations)	Single cameras (Single color images)	Subject-specific facial models without textures	5.52×10^{-3}	System accuracy, visual assessments
Dou et al., 2018 [44]	Non-rigid facial mimics (mouth movements)	Single cameras (Multi-view color images)	Subject-specific facial models without textures	2	System accuracy, visual assessments

In addition to the comparison with MRI data in neutral position, the comparison with depth sensor data showed an error range of [2-3mm] for different facial mimic positions, which is larger than the error (~1mm) with MRI-based analysis in neutral position. This could be explained by the fact that the facial surface curvatures are not similar in different positions. Particularly, the surface curvature in the neutral position is less concave than that in the other mimic positions. In fact, the concave levels are getting deeper when being in neutral, smiles, and sounds ([e] and [u]) respectively. Moreover, the face HD points from Kinect cannot animate concave surfaces effectively, so more errors will appear in deeper concave surface curvatures. Due to the lack of available MRI data, depth sensor data was used. This was proven to be suitable for reconstructing 3-D subject-specific head models with acceptable accuracy [36]. In particular, Hernandez *et al.*, 2015, used a RGB-D PrimeSense [180] camera to reconstructed a 3-D head model with texture from depth and color images acquired in left and right views with different facial mimics [36]. The reconstructed head model was compared with the model reconstructed from laser scanners and showed relatively high accuracy (average error = 1.33 mm). In the present paper, we use directly 3-D point cloud computed from the Kinect SDK 2.0 with the Kinect 2.0 instead of non-standard computed point clouds from depth images of a RGB-D sensor as in [36]. This source of data is more stable and accurate than one from non-standard reconstruction procedures. Moreover, the Kinect sensor V2.0 has better accuracy and resolution than the PrimeSense sensor in different working conditions [181]. Consequently, we designed a procedure to reconstruct head models in different facial mimics from the Kinect point-clouds, and the reconstructed head model could be used to validate the generated head models in different facial mimics.

The selection of the appropriate interaction device is crucial for the accuracy and performance of a computer vision system dedicated for head and facial mimic animations. It is interesting to note that single cameras were mostly used in existing systems [37–40,42–53]. However, because 3-D data is not available in 2-D image sequences, additional computational cost will be needed for reconstructing the 3D data from 2-D data. Moreover, 3-D deformation models cost also more computational resources. In fact, most animation systems using this device could hardly achieve real-time framerates [37–40,43–52]. The use of stereo cameras is also an option [54]. For using this device, a facial marker set should be defined. The 3-D motions of facial markers were reconstructed from horizontal differences between their left and right images captured by left and right cameras. However, the number of facial markers is limited, so local animations could not be estimated realistically. Additionally, the system accuracy was highly affected by the resolutions of stereo-cameras and illuminations of working environments. Finally, a long setup time will be needed for applying facial markers into new users. So, robustness, mobility, and graphic rendering quality are main drawbacks of the animation systems using this device. By using electromagnetic sensors as facial markers, 3-D facial motions could be tracked fast and directly [56]. However, the limited number of input channels in a system was the main drawback when using this type of sensors. Other devices such as 3-D scanner could be used but this required much more computational cost for point cloud processing [31,32]. To overcome these limitations, RGB-D cameras can be used to acquire both 3-D point clouds and texture images [37,49]. Specially, with system development toolkits (SDKs) supported for appropriate types of RGB-D cameras, even more high-level information could be available rather than 3-D point clouds and color images. Various types of RGB-D cameras are presented on markets such as Asus Xtion PRO (1.0, 2.0) and Microsoft Kinect (1.0, 2.0), and some SDKs supported for controlling these sensors are OpenNI and Microsoft Kinect SDKs (1.0, 2.0). This present study confirmed also that the use of Kinect sensor 2.0 is well suitable to develop a real-time computer vision system for rigid head and non-rigid facial mimic movements in a subject-specific manner. However, fast computational algorithm needs to be implemented for non-rigid deformations. In our system, the coherent point drift (CPD) registration algorithm was selected for extracting affine transform from the non-rigid transform. In fact, based on the correspondence probability between two data sets, the CPD try to maximize the likelihood to choose the best rigid and non-rigid transform. This algorithm was very robust with noise, outliers, and even missing points.

One of the limitations of our system relates to the lacking of subject-specific information such as hairs, ears, teeth, and tongues, and irises of eyes. Further improvement of our system will integrate this information. Moreover, regarding the model accuracy analysis, the comparison with MRI-based model in neutral position was performed. Thus, even if the model generation was assessed in different facial mimic using depth sensor data, more MRI data in different mimic positions [182] will be acquired to enhance the evaluation outcomes. Furthermore, the ground truth data in a more natural position (sitting or standing) will be also needed for enhancing this evaluation because of the difference in postural set-ups of the subjects in Kinect-based (sitting or standing) environment and MRI-based (supine) environment. Finally, only limited number of subjects was tested. Thus, the evaluation of our system with a larger subject/patient cohort will enhance the findings of our present study. In particular, the accuracy evaluation of our subject- or patient specific head generation process will be performed on facial palsy patients with complex facial deformity patterns. It is expected that more advanced processing procedures should be developed to scope with theses deformities. One of potential approaches for improving the 3D face reconstruction problem relates to the deep learning, which has been recently developed in the literature [183–186]. This approach with complex and advanced methods and algorithms (e.g. Convolutional Neural Network (CNN), Generative Adversarial Network (GAN)) allows enhancing 3-D shape and texture reconstruction with limited information (2-D image, multiple views of 2-D image).

3.4. Conclusions

This chapter presents a novel computer vision system for tracking simultaneously subject-specific rigid head and non-rigid facial mimic movements in real time. The system framerate can be optimized to get up to 60 fps. Thus, subject-specific head model with texture information could be generated and tracked in real time conditions. Moreover, left/right face asymmetries could be graded in real-time using the left/right mirror effects. A multi-level (model accuracy, visual assessment, system reproducibility and repeatability, system illumination effect, system speed and performance) evaluation process were performed. In the following chapters, an internal skull and a muscle network will be estimated from the subject-specific head model for computing real-time muscle actions during facial mimics. The content of this chapter was published in the Computer Method and Program in Biomedicine journal (Q1, IF@2019 = 3.632) [187].

Chapter 4:

Statistical Shape Modeling Approach for Predicting Subject-Specific Human Skull from Head Surface

Human skull is an important body structure for jaw movement and facial mimic simulations. Surface head can be reconstructed using 3-D scanners in a straightforward way. However, internal skull is challenging to be generated when only external information is available. Very few studies in the literature focused on the skull generation from outside head information, especially in a subject-specific manner with a complete skull. Consequently, a novel process for predicting a subject-specific skull with full details from a given head surface using a statistical shape modeling approach was developed. PLSR-based method was used. A CT image database of 209 subjects (genders: 160 males and 49 females; ages: 34-88 years old) was used for learning head-to-skull relationship. The CT images were collected from the Cancer Imaging Archive (TCIA) [188]. Heads and skulls were reconstructed from CT images to extract head/skull feature points, head/skull feature distances, head-skull thickness, and head/skull volume descriptors for the learning process. A hyperparameter tuning process was performed to determine the optimal numbers of head/skull feature points, PLSR components, deformation control points and appropriate learning strategies for our learning problem. Two learning strategies (point-to-thickness with/without volume descriptor and distance-to-thickness with/without volume descriptor) were proposed. Moreover, a 10-fold cross-validation procedure was conducted to evaluate the accuracy of the proposed learning strategies. Finally, the best and worst reconstructed skulls were analyzed based on the best learning strategy with its optimal parameters. The optimal number of head/skull feature points and deformation control points are 2,300 and 1,300 points respectively. The optimal number of PLSR components ranges from 4 to 8 for all learning configurations. Cross-validation showed that grand means and standard deviations of the point-to-thickness, point-to-thickness with volumes, distance-to-thickness, and distance-to-thickness with volumes learning configurations are 2.48 ± 0.27 mm, 2.46 ± 0.19 mm, 2.46 ± 0.15 mm, and 2.48 ± 0.22 mm respectively. Thus, the distance-to-thickness is the best learning configuration for our head-to-skull prediction problem. Moreover, the mean Hausdorff distances are 2.09 ± 0.15 mm and 2.64 ± 0.26 mm for the best and worst predicted skull respectively. Overall, a novel head-to-skull prediction process based on the PLSR method was developed and evaluated. This process allows, for the first time, predicting 3-D subject-specific human skulls from head surface information with a very good accuracy level. In the following chapter, this head-to-skull process will be applied for generating a skull model and a muscle network from the Kinect-based subject-specific head model. Muscle lengths can also be computed based on the estimated biomechanical head model in real-time.

4.1. Materials and methods

4.1.1. PLSR- and CBD-based head-to-skull generation workflow

A head-to-skull prediction workflow was developed and shown in Figure 39. The workflow included two main stages: head-skull thickness distribution learning and skull generation. In the learning stage, the input data were head and skull models reconstructed from 209 head-neck CT image-sets. All head-skull models were then post-processed to keep only the skull and head regions. Next, the processed head-skull datasets were rigidly transformed to the coordinate system of the first head-skull dataset based on markers manually picked on the head models. Then, point-to-thickness, distance-to-thickness, head volume, skull volume and head-skull volume descriptors were estimated for the learning process using the PLSR method. At the end of this stage, a model coefficient matrix of the PLSR model was computed. In the skull generation stage, a subject-specific skull shape was regressed from a new head surface using the trained coefficient matrix. A generic skull was finally deformed with the cage-based

deformation (CBD) method so that its shape was fitted with the regressed skull shape to form the generated skull. Precisely, the generic skull was covered with a cage. All vertices of the generated cage were the control points of the skull. Thus, the skull vertices were parameterized into the positions of control points so that they can move relatively according to the movements of the control points. In this study, we used the mean value coordinates [189] as the parameters of the skull vertices.

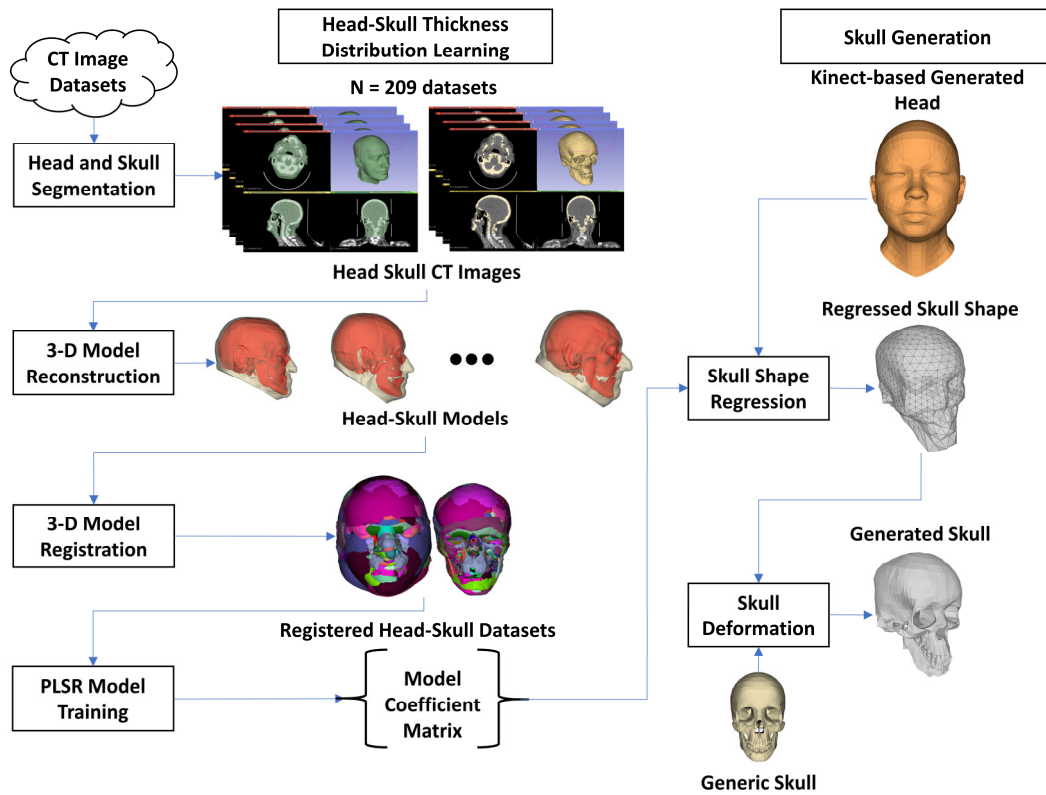


Figure 39. Schematic illustration of our proposed head-to-skull prediction workflow.

4.1.2. Head-skull database construction and transformation

A public head-neck CT image database was retrieved and downloaded from The Cancer Imaging Archive (TCIA) [188]. The database is composed of DICOM [512×512] CT images of 209 patients carefully selected from Canadian 298 patients of a head-and-neck cancer treatment project [190]. The CT images were selected to ensure both head and neck regions were fully available. The patient population age ranges from 34 to 88 years old with 160 males and 49 females.

CT images of head and skull structures were segmented using a thresholding method with 3D Slicer software [191] (Figure 40). Appropriate threshold values were selected for hard and soft tissues within the head. The threshold was selected by using the pixel density values for soft tissue (i.e. envelop of the head) and cortical bone (i.e. skull structure) [192]. Then, the marching cube algorithm was applied for surface reconstruction [193]. Finally, the generated head and skull models were smoothed to output the final head and skull models using the Laplacian smoothing technique [194]. A post-processing procedure was performed using MeshLab software for outlier removal, head cutting, hole filling, surface reconstructing and skull cutting (Figure 41). Even if the thresholds were carefully selected, the reconstructed head model still had some internal outliers (Figure 41a). To determine the internal vertices, ambient occlusion values were computed for each vertex in the head model. In the ambient occlusion method, each vertex was given a number of view directions and computed how many times this vertex was visible from these directions. As a result, the head vertices were classified based on their

ambient occlusion values. An ambient occlusion threshold was selected manually so that only internal vertices were included, and the selected vertices and their related facets were removed from the head model (Figure 41b). After outlier removal, neck vertices and facets (Figure 41c) were cut out manually using the rectangular facet selection tool in MeshLab (Figure 41d). The cutting edge was parallel to the jaw of the skull model and displaced from the skull model with a distance relatively equal to head-skull thicknesses at the chin region. After cutting, the head model was re-meshed to an isometric and closed model (Figure 41e). The Poisson-disk sampling method [195] was first applied for uniformly sampling the head model with the number of sample of 100,000. The sampled point cloud was then reconstructed using the ball pivoting surface reconstruction method [196]. All holes in the reconstructed surface were closed. The re-meshing procedure was repeated one more time to isomerize the facets of the filled holes (Figure 41f). At the final step, cervical vertebrae were manually selected and removed using facet selection and facet deletion tools in MeshLab (Figure 41h). Finally, head volume, skull volume and head-skull volume were computed using available function (i.e. FT CGAL::Polygon_mesh_processing::volume) of the CGAL library [197].

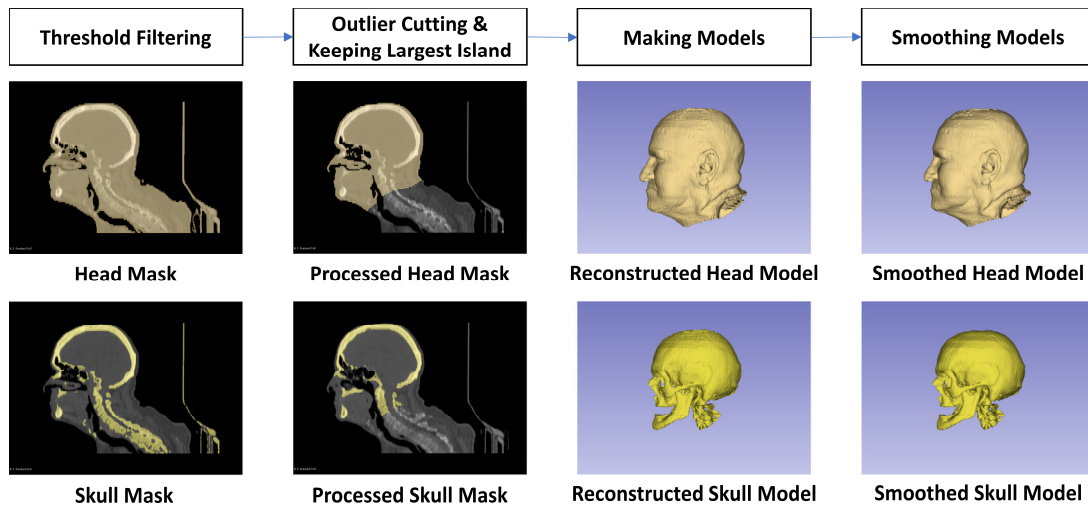


Figure 40. Segmentation and surface reconstruction of the head and skull from CT images using 3D Slicer software.

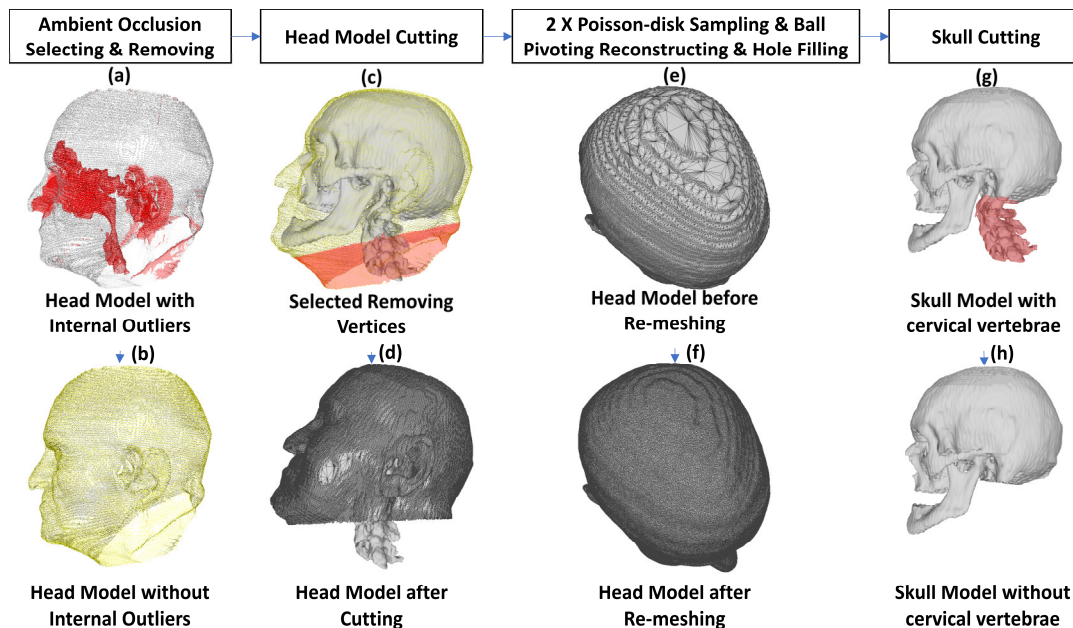


Figure 41. Post-processing procedure for outlier removal, head cutting, hole filling, surface reconstructing and skull cutting.

Before applying the regression model, all head and skull shapes were registered to the same coordinate system (Figure 42). The first head-skull dataset was first translated to the origin of the global coordinate system. This means that the centroid of the first head-skull dataset was moved to the origin of the global coordinate system. Note that orientation axes of the first model do not change. Moreover, because the first dataset was chosen as the reference of all datasets, its local coordinate system becomes the global coordinate system of all datasets. Then, the SVD rigid registration method [198] and the ICP algorithm [175] were applied for each other dataset to transform it into the first head-skull dataset. Note that anatomical landmarks (left and right ears, forehead center, top nose, and mouth center) were added in each head model as reference points for the SVD registration process. Thus, the head model was only best aligned in facial regions. To optimally aligned all head regions, the ICP registration process was then applied on all vertices of the head model. The number of head vertices of all datasets was normalized to about 100,000 points in the post-processing procedure (Figure 41). Finally, the SVD and ICP registration matrices estimated from the head model were applied to the related skull model. As a result, by using both SVD and ICP registration methods, the differences among head-skull datasets were minimized. In fact, the accuracy of the registration process was also optimized to get the best normalization result. Moreover, any holes in the heads and skulls were closed by using a shape generation procedure (Figure 43). The convex hull of each head/skull (Figure 43c) was first generated so that it covered all head/skull vertices (Figure 43b). The convex hull was then re-meshed using the isotropic surface remeshing method [199] with the target edge length equal to the average edge length of the convex hull (Figure 43d). All vertices of the re-meshed convex hull were projected to the nearest vertices of the head/skull model based on the KdTree-Flann nearest searching method in Point Cloud Library (PCL) [200]. After projected, the head/skull convex hull was not isometric (Figure 43e), so it was isometrically re-meshed to form a head/skull shape (Figure 43f) with the target edge length equal to the average edge length of the projected head/skull convex hull.

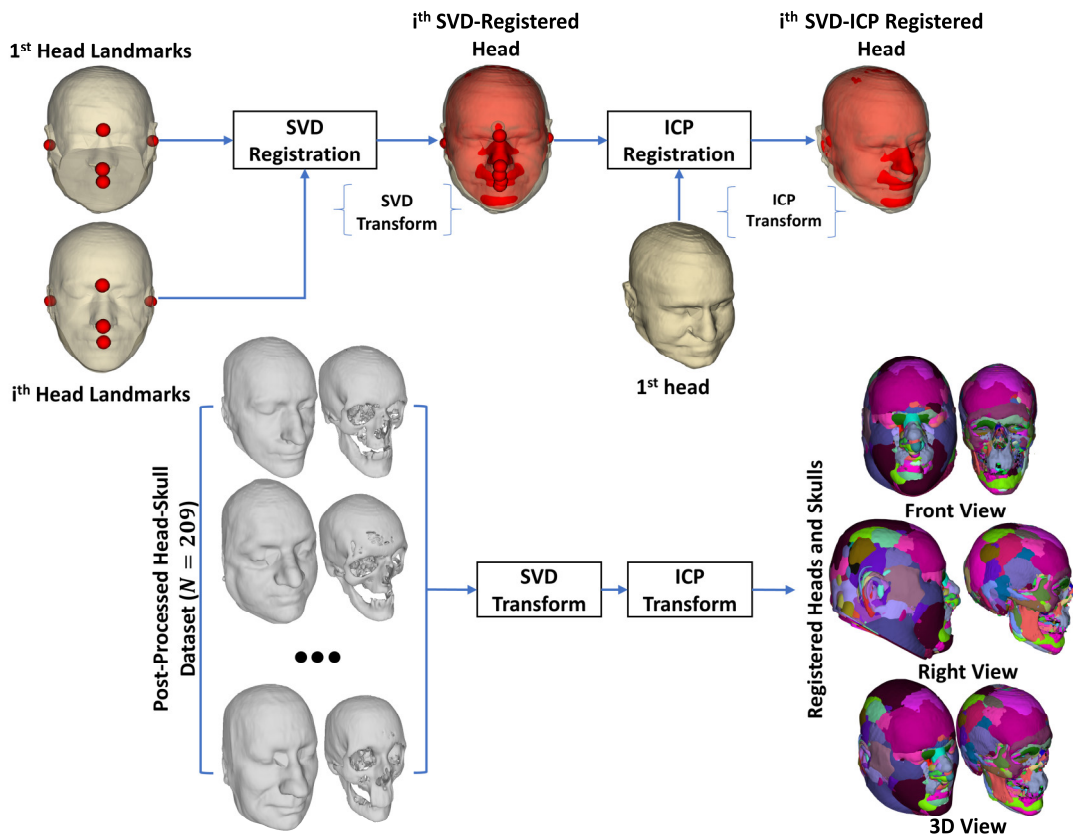


Figure 42. Illustration of the head and skull model registration process: the first head/skull was registered into a global coordinate system and then all other heads/skulls were registered into the first head/skull.

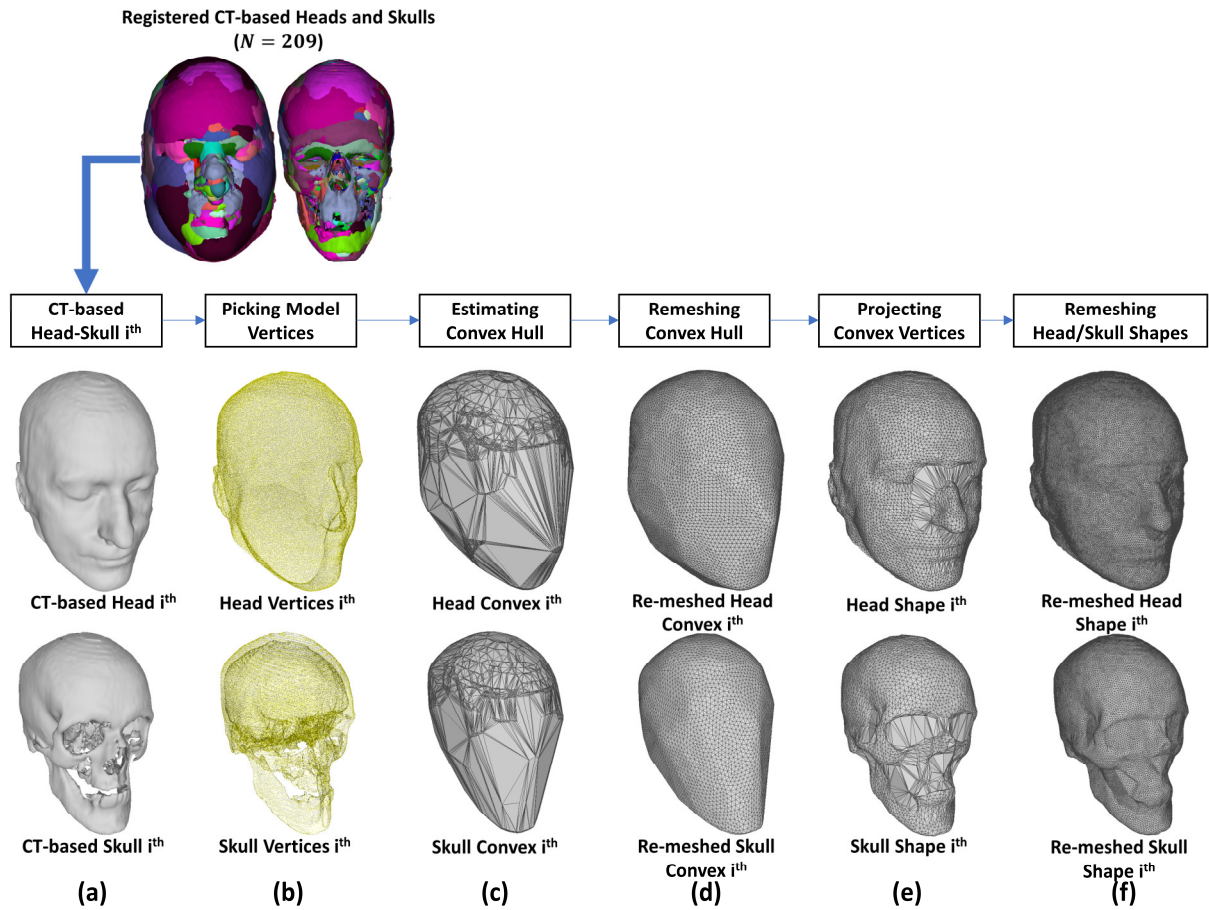


Figure 43. Illustration of the head/skull shapes generated from CT-based reconstructed heads/skulls.

In addition, due to different numbers of vertices and facets for each head/skull model, a sampling process was established to achieve a unified number of feature points for all head/skull datasets (Figure 44). The sampling rays were first generated from a sampling surface (Figure 44b). The sampling rays started from the center of the sampling surface and have directions that go from this center to the vertices of the sampling surface (Figure 44c). The sampling rays were also normalized to have the unit length. In particular, a convex hull of the first skull shape was estimated, and then this convex hull was isometrically re-meshed with the adjustable edge length to form the sampling surface. The number of vertices of the sampling surface can be controlled by adjusting the edge length during the isometric re-meshing process. Because all head-skull datasets were registered to the same coordinate system of the first head-skull dataset, the sampling surface was by default on the same coordinate system with all registered head-skull datasets. Intersected points between the sampling rays and the facets of the head/skull models were the head/skull feature points.

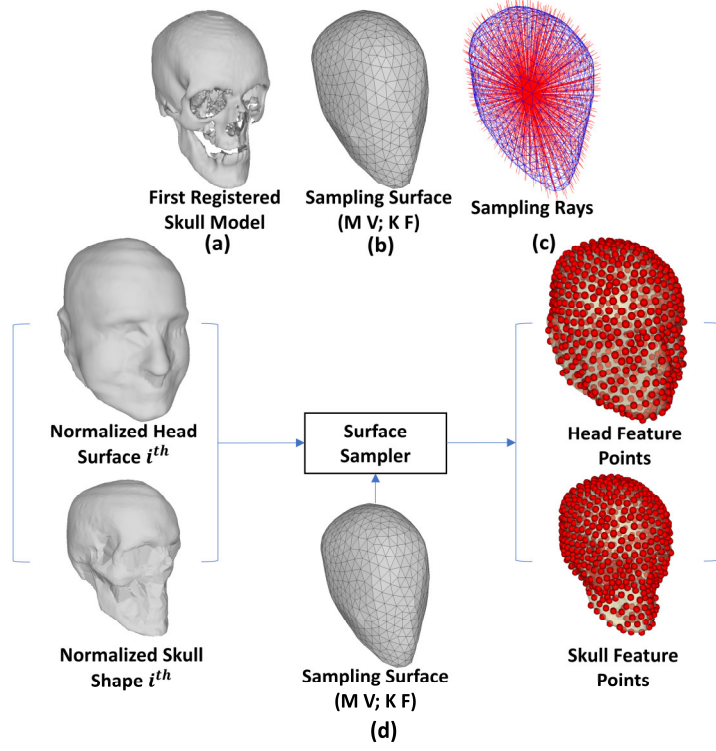


Figure 44. Head-skull surface sampling process.

4.1.3. Head-to-skull learning using the PLSR method

Two learning strategies were proposed for predicting skull shape from head shape. The first strategy was to study the relationship between the positions of head feature points and the head-skull thickness values at these feature points. The second strategy was to study the relationship between the distances from head feature points to the head center and the head-skull thickness values at these feature points. Additionally, each learning strategy also included two configurations in which the training input data were integrated with/without volume descriptor (head volumes).

After the head and skull shape sampling, the head and skull feature points of each dataset were stored in the head and skull feature-point matrices. We also had the sampling ray directions from the sampling surface. The head (\mathbf{V}_H^i) and skull (\mathbf{V}_S^i) feature-point matrices, the sampling surface vertices (\mathbf{V}_{SS}), the sampling center (\mathbf{v}_{SC}), the directions (\mathbf{V}_{SD}) and inversed directions (\mathbf{V}_{ISD}) of the sampling rays of the i^{th} dataset are defined as follows:

$$\mathbf{V}_H^i [M \times 3] = \begin{bmatrix} \mathbf{v}_1^H \\ \vdots \\ \mathbf{v}_M^H \end{bmatrix} = \begin{bmatrix} x_{i1}^H & y_{i1}^H & z_{i1}^H \\ \vdots & \vdots & \vdots \\ x_{iM}^H & y_{iM}^H & z_{iM}^H \end{bmatrix}, i = 1, 2, \dots, N \quad (14)$$

$$\mathbf{V}_S^i [M \times 3] = \begin{bmatrix} \mathbf{v}_1^S \\ \vdots \\ \mathbf{v}_M^S \end{bmatrix} = \begin{bmatrix} x_{i1}^S & y_{i1}^S & z_{i1}^S \\ \vdots & \vdots & \vdots \\ x_{iM}^S & y_{iM}^S & z_{iM}^S \end{bmatrix}, i = 1, 2, \dots, N \quad (15)$$

$$\mathbf{V}_{SS} [M \times 3] = \begin{bmatrix} \mathbf{v}_1^{SS} \\ \vdots \\ \mathbf{v}_M^{SS} \end{bmatrix} = \begin{bmatrix} x_1^{SS} & y_1^{SS} & z_1^{SS} \\ \vdots & \vdots & \vdots \\ x_M^{SS} & y_M^{SS} & z_M^{SS} \end{bmatrix} \quad (16)$$

$$\mathbf{v}_{SC} = \left(\frac{\sum_1^M x_i^{SS}}{M}, \frac{\sum_1^M y_i^{SS}}{M}, \frac{\sum_1^M z_i^{SS}}{M} \right) \quad (17)$$

$$\mathbf{V}_{SD}[M \times 3] = \begin{bmatrix} \mathbf{v}_1^{SD} \\ \vdots \\ \mathbf{v}_M^{SD} \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{v}_1^{SS} - \mathbf{v}_{SC}}{\|\mathbf{v}_1^{SS} - \mathbf{v}_{SC}\|} \\ \vdots \\ \frac{\mathbf{v}_M^{SS} - \mathbf{v}_{SC}}{\|\mathbf{v}_M^{SS} - \mathbf{v}_{SC}\|} \end{bmatrix} \quad (18)$$

$$\mathbf{V}_{ISD}[M \times 3] = \begin{bmatrix} \mathbf{v}_1^{ISD} \\ \vdots \\ \mathbf{v}_M^{ISD} \end{bmatrix} = -\mathbf{V}_{SD} \quad (19)$$

in which, $N = 209$ is the number of datasets; M is the number of sampling rays (the number of vertices of the sampling surface); \mathbf{v}_j^H , ($j = 1, 2, \dots, M$), is the j^{th} head feature point of the i^{th} head-skull dataset; \mathbf{v}_j^S , ($j = 1, 2, \dots, M$), is the j^{th} skull feature point of the i^{th} head-skull dataset; \mathbf{v}_j^{SS} , ($j = 1, 2, \dots, M$), is the j^{th} sampling surface (SS) vertex; \mathbf{v}_j^{SD} , ($j = 1, 2, \dots, M$), is the j^{th} sampling direction (SD); \mathbf{v}_j^{ISD} , ($j = 1, 2, \dots, M$), is the j^{th} inversed sampling direction (ISD); and x , y , and z are the x -, y -, z -coordinate values in 3-D spaces.

4.1.3.1. Strategy #1: point-to-thickness relationship learning

In this strategy, the relationship between the positions of head feature points and their respective head-skull thickness values were analyzed. In the i^{th} head-skull dataset, the thickness values (d_{i1}^{HS}) were distances between head and skull feature points. These distances of the i^{th} head-skull dataset (\mathbf{D}_{HS}^i) were computed using the following equation:

$$\mathbf{D}_{HS}^i = \begin{bmatrix} d_{i1}^{HS} \\ \vdots \\ d_{iM}^{HS} \end{bmatrix} = \begin{bmatrix} \|\mathbf{v}_1^H - \mathbf{v}_1^S\| \\ \vdots \\ \|\mathbf{v}_M^H - \mathbf{v}_M^S\| \end{bmatrix} \quad (20)$$

in which, d_{ij}^{HS} , ($j = 1, 2, \dots, M$), is the head-skull (HS) thickness value at the j^{th} head/skull feature point of the i^{th} head-skull dataset ($i = 1, 2, \dots, N$).

The head volume descriptor (\mathbf{C}) is defined as follows:

$$\mathbf{C} = \begin{bmatrix} c_1^{Volume} \\ \vdots \\ c_N^{Volume} \end{bmatrix} \quad (21)$$

in which, c_i^{Volume} , ($i = 1, 2, \dots, N$), is the head volume of the i^{th} head-skull dataset ($i = 1, 2, \dots, N$).

The predictor variable (\mathbf{x}^i) of the i^{th} dataset and the predictor matrix (\mathbf{X}) of all training dataset are defined as follows:

$$\{\mathbf{x}^i = (x_{i1}^H, y_{i1}^H, z_{i1}^H, \dots, x_{iM}^H, y_{iM}^H, z_{iM}^H, c_i^{Volume}) \mid i = 1, 2, \dots, N\} \quad (22)$$

$$\mathbf{X}[N \times L] = \begin{bmatrix} x_{11}^H & y_{11}^H & z_{11}^H & \dots & x_{1M}^H & y_{1M}^H & z_{1M}^H & c_1^{Volume} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ x_{N1}^H & y_{N1}^H & z_{N1}^H & \dots & x_{NM}^H & y_{NM}^H & z_{NM}^H & c_N^{Volume} \end{bmatrix}, \quad (23)$$

where $L = 3M + 1$. Note that the predictor matrix could be used with or without volume descriptor depending on the selected learning configuration.

In equations (22) and (23), x_{ij}^H, y_{ij}^H , and z_{ij}^H are the x-, y-, and z-axis values of the j^{th} ($j = 1, 2, \dots, M$) head feature point coordinate of the i^{th} dataset, which was formed in Eq. (14).

The response variable (\mathbf{y}^i) of the i^{th} dataset and the response matrix (\mathbf{Y}) of all training datasets are defined as follows:

$$\{\mathbf{y}^i = (d_{i1}^{HS}, d_{i2}^{HS}, \dots, d_{iM}^{HS}) \mid i = 1, 2, \dots, N\} \quad (24)$$

$$\mathbf{Y}[N \times M] = \begin{bmatrix} d_{11}^{HS} & d_{12}^{HS} & \dots & d_{1M}^{HS} \\ d_{21}^{HS} & d_{22}^{HS} & \dots & d_{2M}^{HS} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1}^{HS} & d_{N2}^{HS} & \dots & d_{NM}^{HS} \end{bmatrix} \quad (25)$$

In equations (24) and (25), d_{ij}^{HS} is the Euclidean distance between the j^{th} head feature point and the j^{th} skull feature point of the i^{th} dataset.

4.1.3.2. Strategy #2: distance-to-thickness relationship learning

In this strategy, the relationship between the distances from the head feature points to the head center and their respective head-skull thickness values of these feature points were analyzed. The distances (\mathbf{D}_H^i) between the head feature points and the head center in the i^{th} dataset were defined as follows:

$$\mathbf{D}_H^i = \begin{bmatrix} d_{i1}^H \\ \vdots \\ d_{iM}^H \end{bmatrix} = \begin{bmatrix} \|\mathbf{v}_1^H - \mathbf{v}_{SC}\| \\ \vdots \\ \|\mathbf{v}_M^H - \mathbf{v}_{SC}\| \end{bmatrix} \quad (26)$$

in which, d_{ij}^H , ($j = 1, 2, \dots, M$), is the distance from the j^{th} head feature point to the center of the i^{th} head model ($i = 1, 2, \dots, N$).

The predictor variable (\mathbf{x}^i) of the i^{th} dataset and the predictor matrix (\mathbf{X}) of all training dataset are defined as follows:

$$\{\mathbf{x}^i = (d_{i1}^H, d_{i2}^H, \dots, d_{iM}^H, c_i^{\text{Volume}}) \mid i = 1, 2, \dots, N\} \quad (27)$$

$$\mathbf{X}[N \times L] = \begin{bmatrix} d_{11}^H & d_{12}^H & \dots & d_{1M}^H & c_1^{\text{Volume}} \\ d_{21}^H & d_{22}^H & \dots & d_{2M}^H & c_2^{\text{Volume}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ d_{N1}^H & d_{N2}^H & \dots & d_{NM}^H & c_N^{\text{Volume}} \end{bmatrix}, \quad (28)$$

where $L = M + 1$. Note also that the predictor matrix could be used with or without volume descriptor depending on the selected learning configuration. The response variable (\mathbf{y}^i) of the i^{th} dataset and the response matrix (\mathbf{Y}) of all training datasets were the same as defined by the equations (27) and (28).

4.1.3.3. PLSR and improved PLSR algorithm

A multivariate linear model is defined as the following equation:

$$\mathbf{y} = \mathbf{x}\mathbf{B} + \boldsymbol{\varepsilon}, \quad (29)$$

in which, $\mathbf{y}[1 \times M]$ is the response variable, $\mathbf{x}[1 \times L]$ is the predictor variable, $\mathbf{B}[L \times M]$ is the model coefficient matrix, and $\boldsymbol{\varepsilon}[1 \times M]$ is the model error. Additionally, M is the number of response variables (the number of head/skull feature points in each dataset), and L is the number of predictor variables. Note that the predictor variables are positions of head feature points with or without volume descriptor or distances from head feature points to the head center with or without volume descriptor. The response variables were the head-skull thickness values.

The objective of the training stage was to estimate the model coefficient matrix $\mathbf{B}[L \times M]$ so that the response variables were fitted with the response training datasets ($\mathbf{Y}[N \times M]$) given the predictor variables in the predictor training datasets ($\mathbf{X}[N \times L]$). Eq. (29) will become the following equation:

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \boldsymbol{\varepsilon} \quad (30)$$

Using the PLSR method, the predictor datasets and the response datasets were decomposed into the following matrices:

$$\mathbf{X} = \mathbf{T}\mathbf{P}' + \mathbf{E} = \sum \mathbf{t}_h \mathbf{p}'_h + \mathbf{E} \quad (31)$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}' + \mathbf{F} = \sum \mathbf{u}_h \mathbf{q}'_h + \mathbf{F} \quad (32)$$

where \mathbf{T} and \mathbf{P}' are the scoring and loading matrices of \mathbf{X} . \mathbf{U} and \mathbf{Q}' are the scoring and loading matrices of \mathbf{Y} , respectively. \mathbf{X} with the rank r_x can be decomposed into r_x components \mathbf{p}'_i ($i = 1, 2, \dots, r_x$) with rank 1 matrices, and \mathbf{Y} with the rank r_y can also be decomposed into r_y components \mathbf{q}'_i ($i = 1, 2, \dots, r_y$) with rank 1 matrices. If $i = 1, 2, \dots, \alpha$ ($\alpha < r_x$, $\alpha < r_y$), the error matrix \mathbf{E} (between \mathbf{X} and \mathbf{X}_α) and \mathbf{F} (between \mathbf{Y} and \mathbf{Y}_α) appears. Actually, first few components can be enough for accounting the variances of the predictor and response datasets.

$$\mathbf{X}_\alpha = \mathbf{t}_1 \mathbf{p}'_1 + \mathbf{t}_2 \mathbf{p}'_2 + \dots + \mathbf{t}_\alpha \mathbf{p}'_\alpha \quad (33)$$

$$\mathbf{Y}_\alpha = \mathbf{u}_1 \mathbf{q}'_1 + \mathbf{u}_2 \mathbf{q}'_2 + \dots + \mathbf{u}_\alpha \mathbf{q}'_\alpha \quad (34)$$

The \mathbf{t}_h , \mathbf{p}'_h , \mathbf{u}_h , and \mathbf{q}'_h can be estimated based on the iterative partial least squares (NIPALS) [201] or kernel algorithms [202]. Moreover, after computing all necessary components for \mathbf{X} and \mathbf{Y} , the inner relation between \mathbf{X} and \mathbf{Y} were also regressed using a linear relation: $\hat{\mathbf{u}}_h = \mathbf{b}_h \mathbf{t}_h$. The model coefficient matrix \mathbf{b}_h is chosen so that the covariance between \mathbf{X} and \mathbf{Y} can be described as much as possible. In this study, the model coefficient \mathbf{B} in Eq. (30) was calculated using the improved kernel algorithm [203]. Compared with the classical kernel-based PLSR method [202], the improved kernel algorithm is faster and more efficient for large-size datasets. The estimated model coefficient matrix \mathbf{B} can well describe the variance of both \mathbf{X} and \mathbf{Y} and the covariance between \mathbf{X} and \mathbf{Y} throughout the training datasets. Therefore, the trained model coefficient matrix \mathbf{B} can well describe the variations of head-skull thicknesses at appropriate head/skull features throughout the varieties of the input head/skull data. As a result, given the predictor variables $\mathbf{x}[1 \times L]$, the response variables $\mathbf{y}[1 \times M]$ (thickness values of head features) could be regressed based on the model coefficient matrix \mathbf{B} using Eq. (30). Then, the positions the skull feature points could be computed using the following equation:

$$\mathbf{V}_S = \mathbf{V}_H + \mathbf{y}^T \mathbf{V}_{ISD} \quad (35)$$

where V_H is the positions of the new head features as Eq. (14), V_{ISD} is the inversed directions of the sampling rays as Eq. (19). The regressed skull shape was finally formed with the vertices as the regressed skull feature points and the facets of the sampling surface.

Regarding the skull reconstruction process, a generic skull (6,122 vertices, 9,537 facets), provided by Free3D [204], was deformed to fit its covering shape with the regressed skull shape to form the final predicted skull (Figure 45). The generation procedure includes three main steps: (1) registration, (2) initial deformation, and (3) refinement. First, the generic skull was registered to the regressed skull shape using the SVD [198], ICP [175], and coherent point drift (CPD) [168] registration methods. In particular, the generic skull model was initially registered to the centroid of the new head model using the SVD registration method with manually picked markers on both generic skull and head models. Then, the skull shape of the registered generic skull model was generated based on the skull shape generation procedure in Figure 43. This skull shape was registered to be best fitted with the regressed skull shape predicted from the new head model using both ICP and CPD rigid registration method. The ICP first centralized the SVD-registered generic skull shape inside the regressed skull shape. However, because of different vertex numbers, the two skull shapes were not optimally fitted. Consequently, the CPD rigid registration method was then used to compute the optimal rigid transformation. As a result, the SVD-registered generic skull was transformed using the optimal ICP-CPD rigid transform matrix to be best fitted with the regressed skull shape inside the new head model. Second, the registered skull was parameterized using mean value coordinates [205] before deforming to the regressed skull shape. In particular, the registered generic skull was parameterized using the cage with 14 vertices. The number of cage vertices was chosen so that only the general shape of the skull model could be affected. Using mean coordinate matrix, we could deform the generic skull model by moving cage vertices. The cage of the registered generic skull model was then initially deformed so that the difference between the regressed skull shape and the registered generic skull shape was minimized. After initial deformation, the deformed skull model was relatively fitted with the regressed skull shape. However, some detail regions were not perfectly fitted such as forehead, eye, chin, and teeth regions. Third, the parameterization and deformation were repeated with more control points. In particular, the deformed skull model was again parameterized using the mean value coordinates [205] with a larger number of cage vertices. The deformed skull cage was deformed so that the deformed skull shape was optimally fitted with the regressed skull shape. Note that the number of cage vertices in the refinement step affected to the final accuracy of the final predicted skull. We experimentally chose the number of cage vertices in this step as 1,300 vertices, which could give us the best fitted predicted skull model.

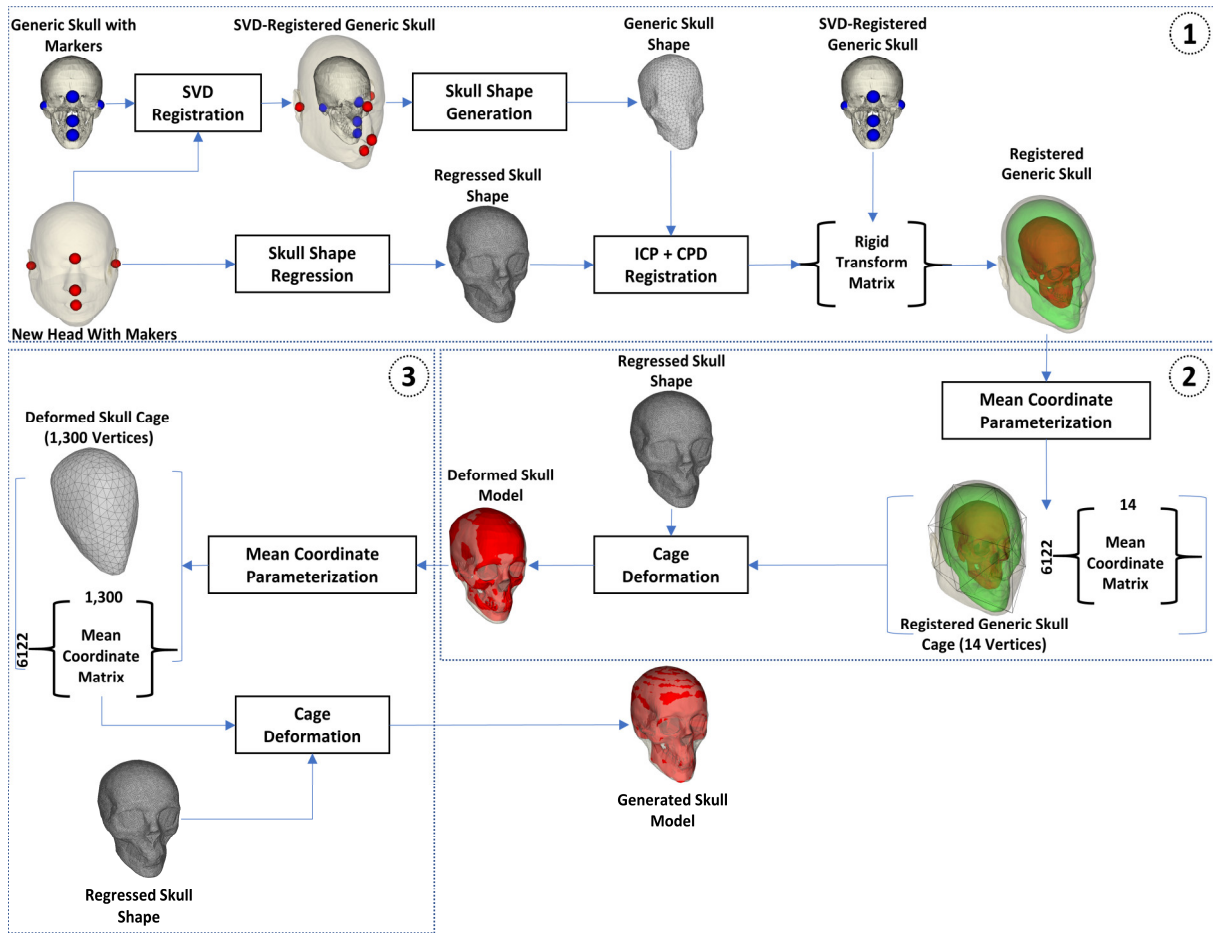


Figure 45. The skull model generation process from regressed skull shape: (1) registration, (2) initial deformation, and (3) refinement deformation.

4.1.3.4. Hyperparameter tuning and cross-validation

A convergence study was conducted to determine the optimal number of head/skull feature points, the optimal number of PLSR components for each learning strategy and the optimal number of control points for skull reconstruction process. Throughout the convergence analysis, the full database was divided into training head-skull datasets (70%, $N_{Training} = 146$) and testing datasets (30%, $N_{Testing} = 63$). To determine the optimal number of head/skull feature points, the number of components in the PLSR model was initially kept at 10, and the number of head/skull feature points was continuously increased from 100 to 3,100 with the step size of 100 points. For each number of feature points, the best regressed skull shape was compared with its appropriate CT-based skull shape using the Hausdorff distance criteria [206]. Precisely, we trained the PLSR model in the training datasets and regressed the skull shapes based on the head shapes in the testing datasets. For each regressed skull shape, we computed the mean absolute difference between the regressed outputs and the testing outputs. The best regressed skull shape was chosen so that its mean difference was minimum throughout the testing datasets. To determine the optimal number of components, the number of feature points was kept optimal, and the number of components was increased from 1 to 20 components with the step size of one component. The regressed outputs were compared with the testing outputs using the absolute difference criteria. To determine the optimal number of control points for skull reconstruction process, three randomly selected CT-based skull shapes from the testing dataset were used for skull reconstructions. The number of control points in the refinement process was then continuously increased from 100 to 3,100 with the step size of 100 points. For each number of control points, the reconstructed skull shapes were compared with the CT-based skull shapes using the Hausdorff distance criteria. All these convergence analyses were conducted with the threshold of 5%. Note that the threshold means the absolute difference based on Hausdorff

distance metrics between the current error (i.e. between the predicted skull shape and the CT-based skull shape) and the previous error within the iterative convergence analysis algorithm.

After having the optimal numbers of components and head/skull feature points, the PLSR models with four learning configurations (point-to-thickness, point-to-thickness with volume descriptor, distance-to-thickness, and distance-to-thickness with volume descriptor) were cross-validated to determine their accuracy. A 10-fold cross-validation process was performed. Mean errors were estimated and reported. Note that all learning configurations were cross-validated using the same protocol for selecting the best configuration. With the best learning configuration and its optimal parameters, the best and worst regressed outputs among all testing datasets were selected to reconstruct the appropriate skulls using the optimal number of control points.

The comparison between the regressed skull shape and CT-based skull shape was evaluated using different metrics including Hausdorff distance and volume deviation. Moreover, the comparison between the predicted skull (generated in Figure 45) and the CT-based skull was evaluated using only the Hausdorff distance criteria for evaluating similarity of internal structures. All training, regressing, and reconstructing procedures were executed on a mobile workstation system with the hardware configuration of Intel® Xeon® E-2176M CPU @ 2.7GHz 64 bits, 12 cores, 32GB DDRAM. All procedures were developed in Microsoft Visual Studio C++ 2015.

4.2. Results

4.2.1. Head and skull volume

The distributions of head, skull, head-skull volumes computed from head/skull shapes are shown in Figure 46. The average head volume is 3,987 cm³, the average skull volume is 2,607 cm³, and the average head-skull difference volume is 1,379 cm³. Note that the order of magnitude of the average head volume is in agreement with the mass properties of adult human head [195].

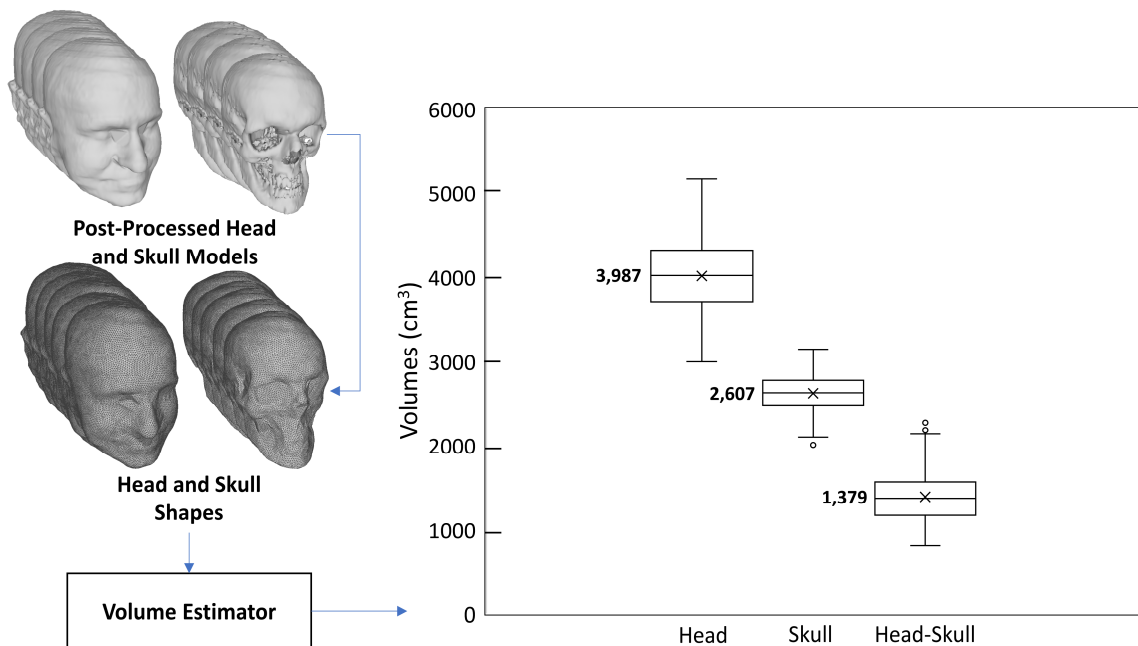


Figure 46. Volume distributions of all reconstructed subjects.

4.2.2. Hyperparameter tuning outcomes

The mean Hausdorff distances between the best regressed skull shapes and their CT-based skull shapes for all learning configurations (point-to-thickness, point-to-thickness with volume descriptor, distance-to-thickness, and distance-to-thickness with volume descriptor) according to different number of head/skull feature points are shown in Figure 47. Overall, the mean distances of all configurations significantly decrease when the number of head/skull feature points increases from 100 to 500 points. These distances decrease slowly and become stable when the number of feature points is from over 500 to 3,100 points. The average mean distances of four learning configurations gradually decrease and reach the minimum value at 1.08 mm when the number of feature points is 2,300 points. From this number, the average distance stops decreasing.

By using the optimal number of head/skull feature points as 2,300, we trained and tested our PLRS models to determine the optimal number of components. The root mean square errors of all tested learning configurations are shown in Figure 48. The optimal number of components for the point-to-thickness, point-to-thickness with volume descriptor, distance-to-thickness, and distance-to-thickness with volume descriptor are 8, 7, 5, and 4 respectively. Note that the optimal number of components decreases when the number of predictor variables decreases. Moreover, in the point-to-thickness configuration, the learning strategy with head volume descriptor has smaller optimal number of components than ones without the head volume descriptor.

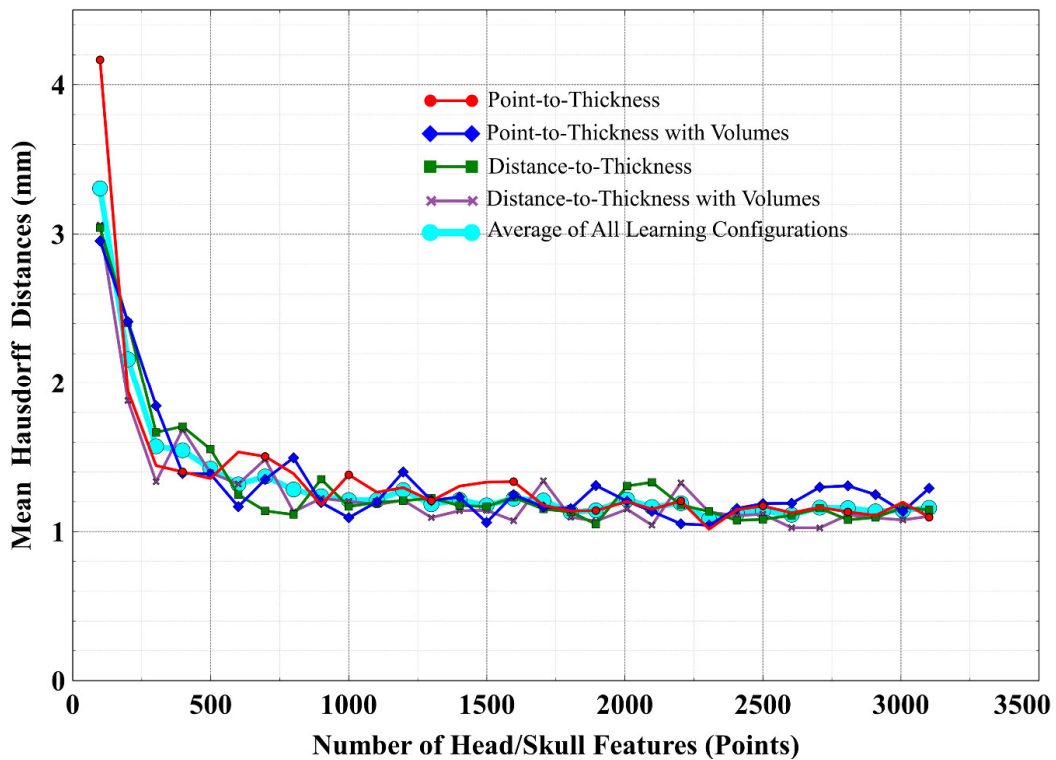


Figure 47. Mean Hausdorff distances computed from the convergence analysis to determine the optimal number of head/skull feature points: the number of head/skull feature points was increased from 100 to 3100 features with the step-size of 100 points. Note that for each learning configurations, the mean Hausdorff distance was computed between the best regressed skull shape and its appropriate CT-based skull shape.

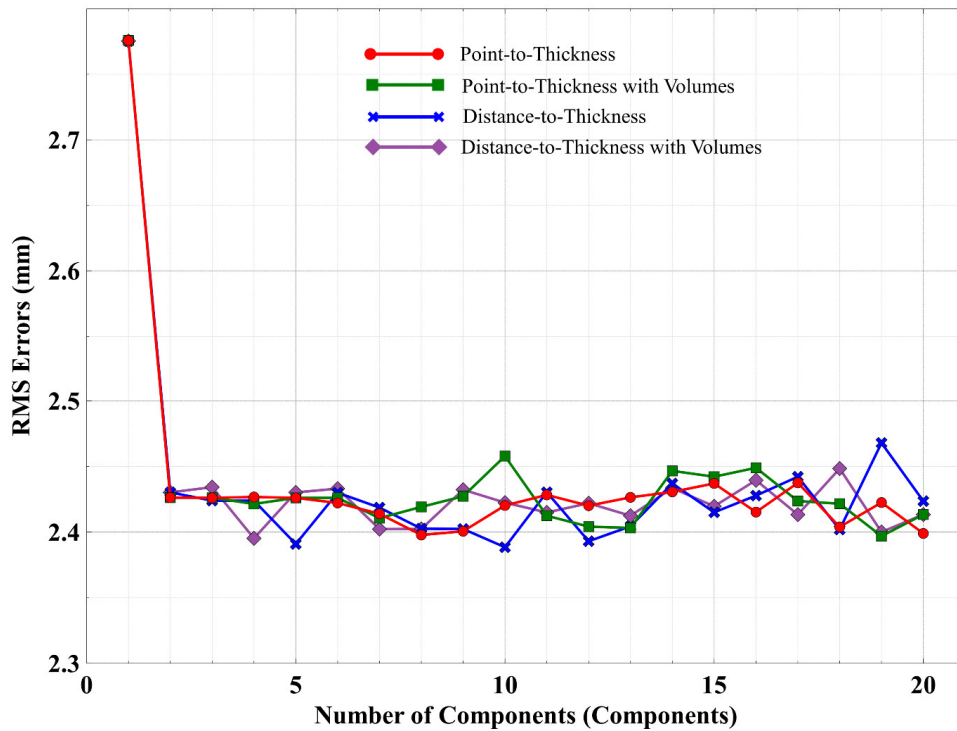


Figure 48. RMS errors computed from the convergence analysis to determine the optimal number of components: the number of components was increased from 1 to 20 components with the step size of one component. The number of head/skull feature points were kept at the optimal value of 2300.

The outcomes of the convergence analysis to determine the optimal number of control points for the skull reconstruction process in the refinement step are shown in Figure 49. Overall, when the number of control points increases from 100 to 3,100 points, the average differences of the three randomly selected testing datasets decrease from 2.5 mm to reach the first minimal value of 1.8 mm at 1,300 points. From 1,300 points to 3,100 points, the average errors fluctuate around 1.9 mm. Consequently, according to this analysis, we choose the optimal number of control points as 1,300 points.

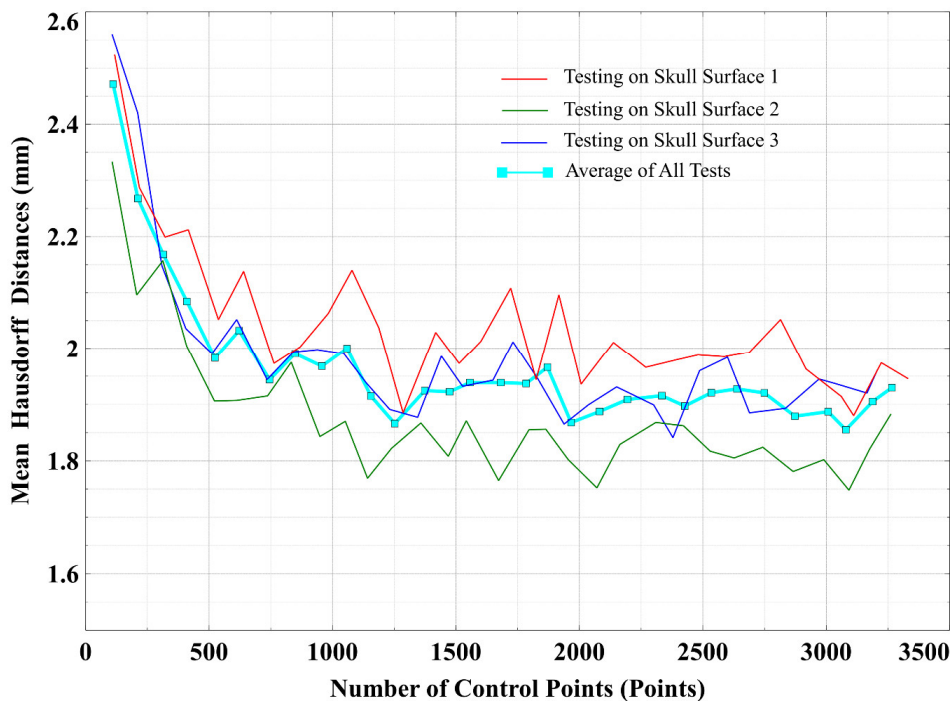


Figure 49. Mean Hausdorff distances between the generated skull shapes and the CT-based skull shapes in different number of control points during the refinement process of skull deformation. For each number of control points, three randomly selected testing datasets were evaluated.

4.2.3. Ten-fold cross-validation

The outcomes of the 10-fold cross-validation using optimal parameter values for each learning configuration are shown in Figure 50. All the regressed outputs of each learning strategy were relatively stable in each iteration of the cross-validation process. Grand mean and standard deviations of mean differences in the point-to-thickness, point-to-thickness with volumes, distance-to-thickness, and distance-to-thickness with volume learning configurations throughout 10 times of cross-validations are 2.48 ± 0.27 mm, 2.46 ± 0.19 mm, 2.46 ± 0.15 mm, and 2.48 ± 0.22 mm respectively. Thus, the distance-to-thickness is the best learning configuration for our head-to-skull predicting problem. Note that the training and testing time for each run of this optimal learning configuration is $9m4s \pm 10s$.

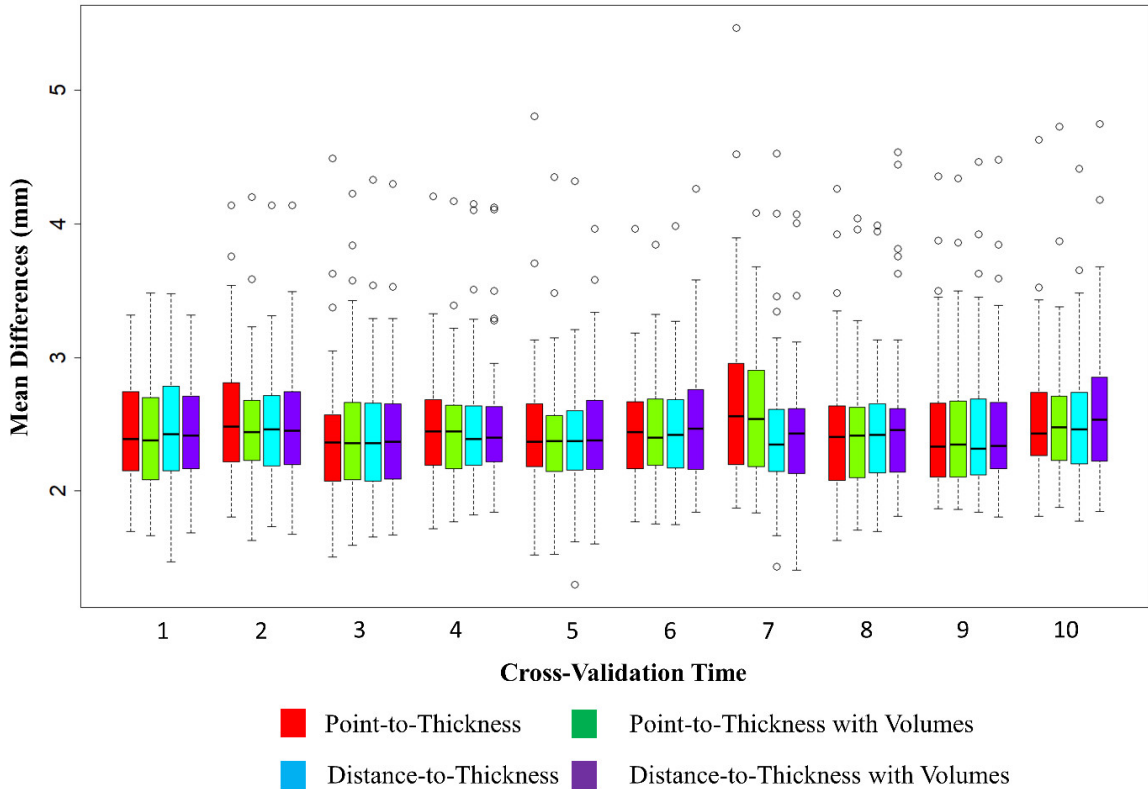


Figure 50. Mean errors between the regressed outputs and the testing outputs from the 10-fold cross-validation. The mean errors were evaluated in different learning configurations with the optimal numbers of components and head/skull feature points.

4.2.4. Best and worst prediction cases

The best and worst regressed skull shapes from the 10-fold cross-validation by using the optimal set of hyperparameters and the best learning configuration obtained from previous steps were analyzed. The regressed skull shapes were compared with their respective CT-based skull shapes using the Hausdorff distance and volume metrics. The comparison results are shown in Figure 51. The mean Hausdorff distances are 1.2 ± 0.14 mm and 3.1 ± 0.29 mm for the best and worst cases respectively. The volume deviations are 25.3 ± 26.4 cm³ ($1 \pm 1\%$) and 222.5 ± 56.9 cm³ ($8.7 \pm 1.8\%$) for the best and worst cases respectively. When reconstructing skulls from the best and worst regressed skull shapes (Figure 52), the mean Hausdorff distances are 2.09 ± 0.15 mm and 2.64 ± 0.26 mm for the best and worst cases respectively. In particular, the mean Hausdorff distances in muscle attachment regions are 1.68 ± 0.36 mm and 1.98 ± 0.36 mm for the best and worst cases respectively. The muscle attachment regions were defined based on the standard definitions of facial muscles in MPEG-4 [207], facial action coding system (FACS) [206], and facial anatomical landmark [208]. Based on these standards, we defined muscle attachment regions by manually selecting skull vertices in the skull models (both predicted and CT-based models). Note that the skull reconstruction time from skull shapes is $40.27 \pm 3.06s$.

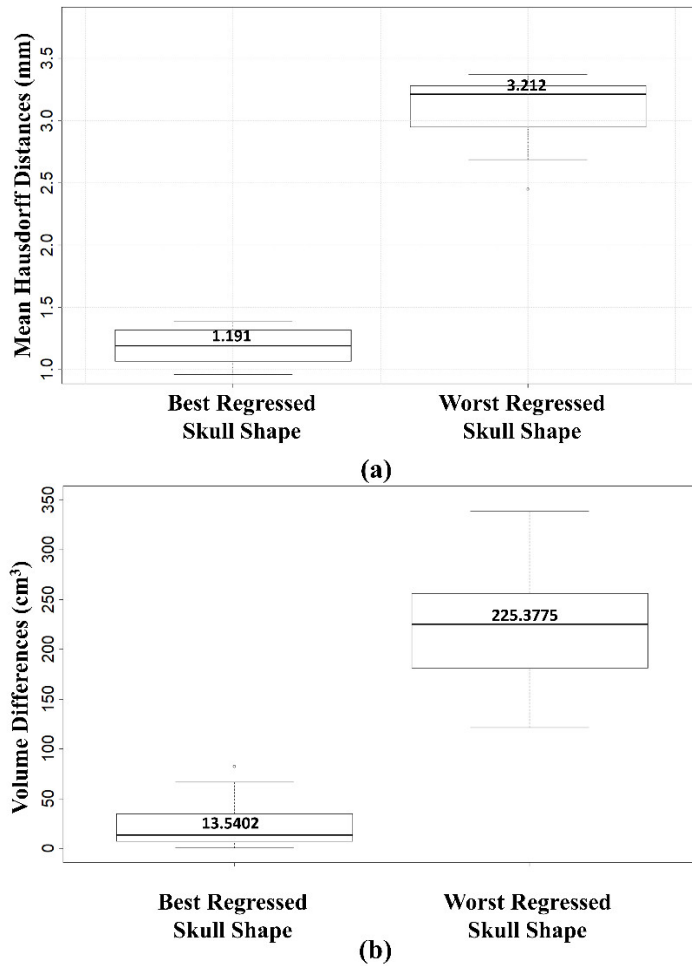


Figure 51. Comparisons of best and worst cases throughout 10-fold of cross-validations: (a) mean Hausdorff distances and (b) volume deviations between the regressed skull shapes and CT-based skull shapes.

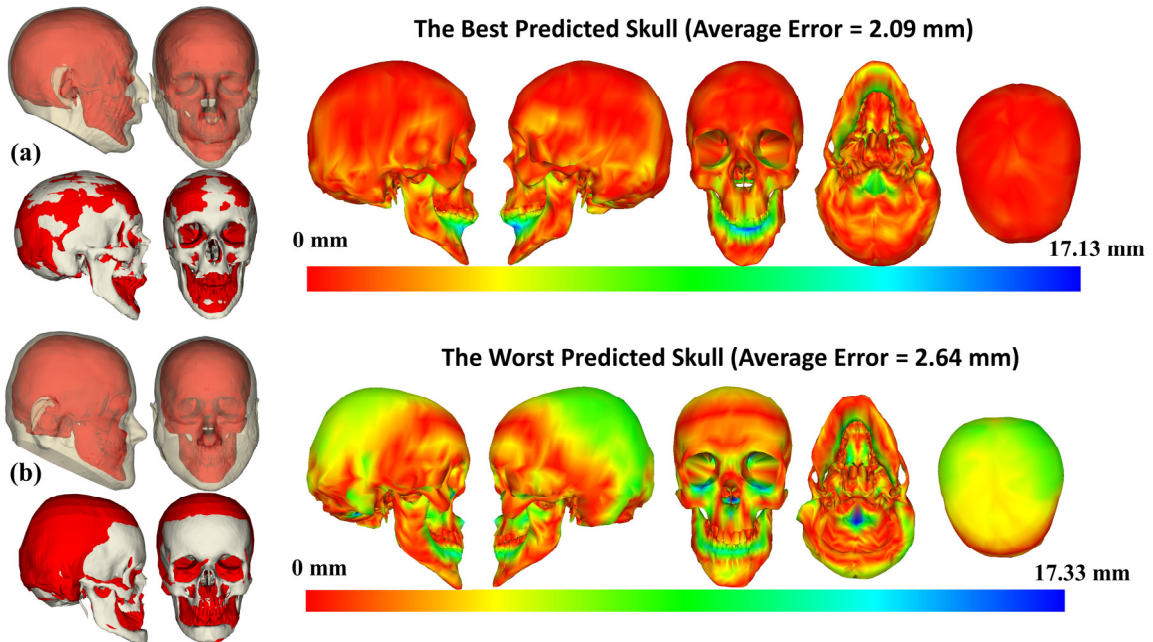


Figure 52. Comparison of the best and worst reconstructed skulls with the respective CT-based skulls from the 10-fold cross-validation: (a) best case and (b) worst case.

4.3. Discussion

Three-dimension skull is commonly reconstructed in computer-aided vision system to perform jaw movements and facial expression/mimic animations. Thus, the skull reconstruction affects directly the accuracy of such system. The use of medical imaging techniques like CT or MRI could lead to a very accurate skull reconstruction. However, these techniques are not suitable for a real-time computer-aided vision system in which only visual sensors such as 2-D cameras [51], 3-D scanners [32,53], and Kinect sensors [35] are available. Consequently, skull generation from head becomes an attractive and challenging issue. In this present paper, we proposed a novel process for predicting 3D skull from head using a statistical shape modeling approach. Our approach showed a high level of approximation with a mean Hausdorff distance ranging from 2.09 ± 0.15 mm to 2.64 ± 0.26 mm between statistics-based regressed skulls and their respective CT-based reconstructed skulls. In particular, the mean Hausdorff distances were from 1.68 ± 0.36 mm to 1.98 ± 0.36 mm in muscle attachment regions. It is important to note that our proposed skull reconstruction from head is a complex process with many processing steps ranging from medical image processing, geometrical transformation and regression. All these steps contribute into the reported errors. In particular, manual interventions (e.g. head and skull cutting) were performed during medical imaging processing and geometrical transformations. Thus, further study should be investigated to provide an automatic procedure for these interventions to reduce possibly the errors. Another possible improvement could be the increase in samples used in the training process to get better regression outcomes.

In the literature, only one study tried to estimate skull from head, but the generated skull was partial or lacked of skull details [74]. In fact, this study approximated the skull's forehead and jaws by just down scaling and down sampling a head model. The affine transformation was used to deform an available skull to be approximately inside a new head. However, this method was not validated with a ground truth skull. Moreover, the affine transforms were not enough for describing nonlinear deformations and surface variances of the head-skull relationship. Furthermore, skull-to-head relationship was commonly found in other studies for forensic facial reconstruction [81–89]. Compared with all above studies, our study regressed head-skull relationship with different learning strategies including point-to-thickness (with/without head volumes) and distance-to-thickness (with/without head volumes). Moreover, the improved PLSR method [203] was used in our study because it was more suitable for training large size of predictor and response variables than the classical PLRS method [201], which mostly used in the previous studies. Additionally, a hyperparameter tuning process was conducted leading to more robust prediction outcomes with optimal number of head/skull feature points and PLSR components. In particular, we applied the cage-based deformation method to deform a generic skull to fit with the regressed skull shape. The cage-based deformation method was more suitable than the free-form deformation method [209] for deforming models with complex internal structures like skulls.

In addition, in our prediction problem, the number of our training datasets ($N = 146$) was smaller than the number of predictor variables ($N < L$). The used PLSR method [201] is suitable for such problem in which $N < L$. Moreover, a hyperparameter tuning process was conducted to determine the optimal number of learning parameters of the PLSR model, which is rarely performed in the literature for the head-skull relationship learning. This process is very time consuming but this belongs to one of the best practices to be necessarily performed when developing statistical regression model. In particular, identified number of head/skull feature points fits perfectly with the number of head vertices available in visual camera like Kinect (2297 vertices in the head region without neck) [187]. This leads to the potential use of the proposed head-to-skull process in the Kinect-based computer vision system. However, it is expected that the accuracy of the regressed skull models will be affected by using different qualities of input head models, especially acquired from visual sensors like Kinect camera.

Moreover, additional data processing step to reconstruct head model from facial point cloud should be developed [187] before applying our skull generation process. Furthermore, the learning process of our study was based on a public database of Canadians with ages ranging from 34 to 88 years old. Thus, the application of our present outcomes on facial palsy patients will be challenging because of different races and especially the lack of patient data with facial deformities in the learning process. In fact, our database and the learning process need to be enhanced (e.g. more French patient data, patient group of less than 34-year-old, patients with facial deformities) when integrating the present approach on our Kinect-based computer vision system [187] for facial palsy patients.

Regarding the skull-to-head or head-to-skull relationships, because the number of predictor variables is often more than the number of training datasets, the dimension reduction methods like the Principle Component Analysis (PCA) [81,84–86,88] and the PLSR [87] methods were mostly investigated. While the PCA is an unsupervised pattern recognition technique aiming to analyze variances of predictor variables without considering the response variables, the PLSR is a supervised technique aiming to analyze intrinsic relationships between the variance of predictor variables and variance of response variables. Consequently, fewer components in the PLSR method are needed for fitting the response variables than in the case of PCA method [201,210]. In particular, in this present study, the improved PLSR method proposed by Dayal *et al.*, 1997 [203] was used. This method is faster and more efficient than the classical PLSR when analyzing a large number of predictor and response variables. Moreover, instead of directly studying skull-face thickness distributions, most studies tried to learn positions of landmarks manually marked on face and skull regions [81,84,85]. Thus, faces and skulls could also be divided into different ROIs, and their vertices can be used as the predictor and response variables to train the PCA or PLSR models [87]. However, the direct use of positions of face-skull markers/ROI vertices as the predictor and response variables increases significantly the complexity and accuracy of the trained models, especially when the number of head/face feature points is large. In this present study, our best learning strategy (i.e. distance-to-thickness learning configuration) has a minimal number of predictors and response variables. In particular, the vector values of head/skull feature points were transferred to the scalar values, so the dimensions of predictor and response variables were significantly decreased. Consequently, this learning configuration is suitable for analyzing large number of feature points. Furthermore, the use of manual sampling approach limits the number of feature points and datasets (38 points on 25 datasets in [84], 52 points on 118 datasets in [81], 78 points on 200 datasets in [85]). In this present study, the proposed automatic sampling procedure is more suitable for acquiring a large number of feature points on a large number of datasets (2300 points on 209 head-skull datasets). Finally, in the most studied works in the literature, back regions were always lacked in learning head-skull relationship. Our present study learns the head-to-skull relationship with full head/skull shapes. Consequently, our approach could be used for learning both head-to-skull and skull-to-head relationships in a straightforward way.

Three-dimension geometrical shape learning has become an attractive research field with many applications like 3-D object segmentation, 3-D shape correspondence or 3-D shape prediction. There are two common approaches to perform shape learning. The first one relates to classical statistical approach like the PLSR used in this present study. The second one deals with machine learning approach [211–213]. At the moment, with the progress of new efficient and robust machine learning algorithms, 3-D shape learning could be efficiently achieved. However, this requires specific machine learning libraries like TensorFlow or Keras. Thus, the integration of this approach into a real-time computer-vision system requires specific coding implementation specification. In the present study, a classical statistical approach was selected due to its potential integration into our computer vision system. However, although the used PLSR method is particularly suitable for our head-to-skull prediction problem, this method also has some drawbacks. For example, the variance of predictor variables and response variables is

only computed throughout the training datasets over different shapes. The inter-relation among input/output variables in each dataset is not taken into consideration. Thus, further studies will be investigated to incorporate this important behavior into the prediction model. For this purpose, a new class of deep learning approach [214], named geometric deep learning, should be adopted.

One of the limitations of this present study relates to the fact that CT image database was used. It is well known that medical imaging data is commonly acquired in supine position leading to shape artefacts due to gravity effect [215]. Thus, a postural transformation should be investigated in the future to correct the shape before using the reconstructed head/skull shapes for learning purpose. Moreover, a larger database focusing on the age group of less than 35 years old will complete our study. Finally, the increase of the number of data sets will improve the accuracy of statistical regression capacity for our head-to-skull prediction problem.

4.4. Conclusions

A novel head-to-skull prediction process based on PLSR method was developed and evaluated in chapter. The distance-to-thickness learning configuration was identified as the best learning strategy for our prediction problem. A hyperparameter tuning was performed to select the optimal set of learning parameters. The developed process allows to predict 3D human skull from head surface information with a very good accuracy level (i.e. the mean Hausdorff distances range from 2.0937 ± 0.1549 mm to 2.6371 ± 0.2569 mm). In the following chapter, the head-to-skull prediction process will be integrated in the subject-specific head generation & animation system [187] (presented in Chapter 3) to model a patient-specific biomechanical head model including head, skull, and muscle network. Based on this biomechanical head structure and face animations, facial muscle lengths can be estimated during facial mimics in real-time. The content of this chapter was published on the Medical & Biological Engineering & Computing Journal (Q2, IF@2019=2.022) [216].

Chapter 5:

Kinect-driven Patient-specific Head, Skull, and Muscle Network Modelling for Facial Palsy Patients

Facial palsy negatively affects both professional and personal life qualities of involved patients. Conventional facial mimic rehabilitation can recover these facial mimics into their normal and symmetrical movements & appearances. However, there is a lack of objective, quantitative, and in-vivo bio-feedbacks, especially internal muscle action behaviours for personalizing rehabilitation programs and diagnosing recovering progresses. Consequently, this chapter proposed a novel patient-specific modelling method for generating a full patient specific head model from a visual sensor and then computing external and internal bio-feedbacks for clinical decision making. In particular, patient-specific head surface model was generated and animated from high-definition facial points acquired from the Kinect sensor. Then, a skull model was predicted from the head surface model based on a statistics-based head-to-skull shape model. A muscle network was also defined from the head and skull models for computing muscle strains during facial mimics. Three healthy subjects and two facial palsy patients were selected for validating the proposed method. In neutral positions, MRI-based head and skull models were compared with Kinect-driven head and skull models. In mimic positions, infrared depth-based head models in smiling and [u]-pronouncing mimics were compared with appropriate animated Kinect-driven head models. The Hausdorff distance metric was used for comparison purpose. Moreover, computed muscle length and strain were compared to experimental and literature data. As a result, in neutral positions, the best mean error was 1.91 mm for the head models and 3.21 mm for the skull models. In mimic positions, these errors were 2.02 mm in smiling mimics and 2.00 mm in [u]-pronouncing mimics for the head models. Facial muscles exhibited shortened and elongated contraction behaviors for smiling and pronunciation of sound [u] respectively. Extracted muscle features (i.e. muscle length and strain) are in agreement with experimental and literature data. In this project, the Kinect-driven muscle features were used for muscle analysis during facial movements.

5.1. Materials and methods

5.1.1. General modeling workflow

The general modeling workflow of different model generations is shown in Figure 53. The workflow includes (1) Kinect-to-head, (2) head-to-skull, and (3) muscle network definition & generation process. The Kinect-to-head process used a Kinect-driven data to generate the patient specific geometrical head model with texture. The head-to-skull applied a statistical shape modeling approach to generate the patient specific skull from the patient specific geometrical head model. Finally, the muscle network definition and generation processes allow important facial muscles to be generated and then associated information such as muscle lengths and strains were estimated and tracked.

5.1.2. Kinect-to-head process

In this process, the subject-specific head model was generated and animated with texture using subject-specific data acquired from the Kinect V2.0 sensor. I describe briefly this process here, please refer to Chapter 3 for more detailed information. This process comprises of data acquisition, subject-specific head generation, texture generation, and head animation sub-processes. In the data acquisition sub-process, the Kinect sensor was controlled by data acquisition interface to acquire multiple types of subject-specific data including current 3-D head orientations, current 3-D head positions, 3-D HD facial points (1,347 points), and color images in real-time. Moreover, 2-D HD facial points and facial pixel regions could also be extracted in color image spaces. In the subject-specific head generation sub-process, a template head model (2,582 vertices; 5,160 facets) supported from Kinect SDK 2.0 was deformed so that the facial vertices was relative fitted with the HD facial points, and then the facial vertices were

replaced by the HD facial points to form the generated head model. In the texture generation sub-process, we captured left, right, and center images of the user before deforming and merging them into a single flatten space with the control points as the 2-D HD facial points to form his/her texture image. The texture coordinates were the projected points of 3-D HD facial points onto a projection plane. In the head animation sub-process, the generated head model was transformed to the current head orientation and the current head position acquired from the Kinect sensor to provide rigid animations. The facial vertices were then replaced by the HD facial points to provide non-rigid animations. As a result, we animated the textured head model according to current facial mimics with the framerate of 60 fps and the acceptable accuracy.

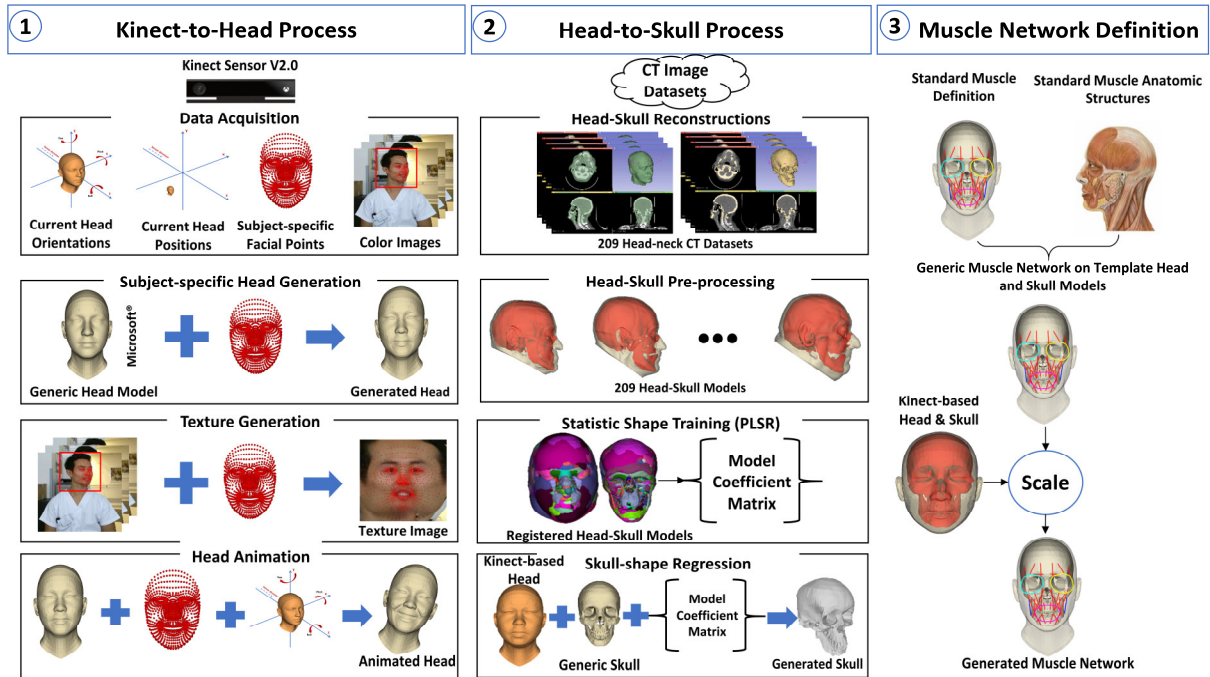


Figure 53. The general workflow of model generations: (1) Kinect-to-head generation process, (2) head-to-skull generation process, and (3) muscle network definition & generation process.

5.1.3. Head-to-skull process

In this process, the subject-specific skull model was predicted from the Kinect-based head model. In Chapter 4, relationship between head shapes and skull shapes was trained using PLSR method with the improved kernel algorithm [203]. In particular, 209 head-skull datasets were reconstructed from 209 head-neck CT image datasets. The reconstructed head-skull models were then pre-processed to obtain only head regions. The head-skull models were registered to a reference coordinate system before sampled to get head-skull feature points. Then, the PLSR-based shape model was trained using the head & skull feature points to achieve a PLSR model coefficient matrix. This coefficient matrix could be used to predict a new skull shape given a new CT-based head surface model. Finally, a generic skull model was deformed so that its shape was fitted with the regressed skull shape to form the generated skull model.

Outputs from the previous head-to-skull training procedure [216] (Chapter 4), shown in Figure 54, include the reference head model with pre-defined landmarks (left & right ears, forehead, top-nose, and mouth center) (Figure 54a), the sampling surface (Figure 54b), and the head-to-skull PLSR-model coefficient matrix (Figure 54c). *The Kinect-based head model was first registered to the same coordinate system of the reference head model before used for predicting the skull model.* The registration procedure was illustrated in Figure 55. First, the neck region in the Kinect-based head was removed to keep only the head region. The head model without neck was then transformed to the reference head model based on pre-defined landmarks in the two models using SVD rigid registration method [198]. For optimizing registration errors due

to the landmark selections, the ICP algorithm [175] was applied on all vertices of both the SVD-registered head model and the reference head model. As a result, the SVD-ICP-registered head model was optimally on the same coordinate system of the reference head model. *After registration, the SVD-ICP Kinect-based head model was used to predict the skull model.* The prediction procedure was shown in Figure 56. The registered head model was sampled to get head feature points by a surface sampler. The sampling rays have starting points as the centroid of the sampling surface and directions as from the starting points to the vertices of the sampling surface. The number of sampling rays was chosen as the optimal value after the hyperparameter tuning process in the head-to-skull training procedure [216]. The head features were intersections between the sampling rays and their nearest facets on the head model. The head feature points were then inputted to the head-to-skull regressor to predict the skull feature points using the PLSR coefficient matrix [216]. The regressed skull shape has vertices (2,305 vertices) as the predicted skull feature points and facets (4,606 facets) as the facets of the sampling surface. The generated skull model was formed by deforming the generic skull model (6,112 vertices; 9,537 facets) so that its shape was optimally fitted with the regressed skull shape using the cage-based deformation method [216]. After generated, the generated skull model was registered back to the original position of the Kinect-based head model after the Kinect-to-head process in Figure 53. During real-time head animations, the skull model was moved according to the rigid movements of the animated head model.

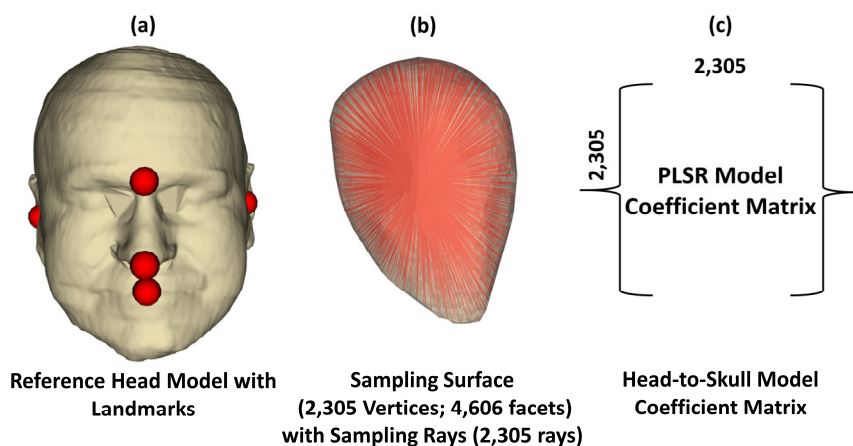


Figure 54. Outputs from the head-to-skull training procedure: (a) the reference head model (14,5420 vertices; 290,633 facets) with landmarks (left ear, right ear, forehead, top nose, and mouth center points), (b) the sampling surfaces (2,305 vertices, 4,606 facets) with sampling rays (2,305 rays), and (c) the head-to-skull model coefficient matrix (2,305 rows; 2,305 columns)

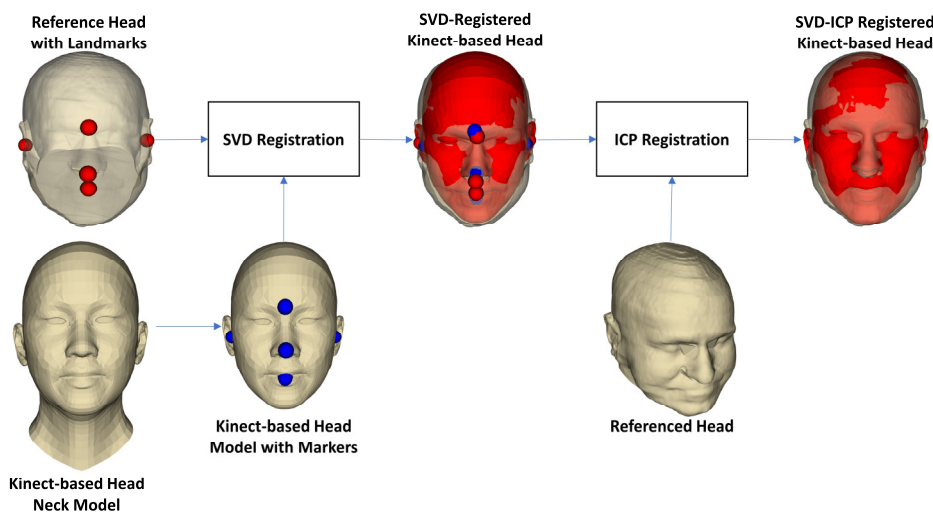


Figure 55. Kinect-based head pre-processing and registration procedure. The Kinect-based head neck was pre-processed to keep only the head region and registered to the coordinate system of the reference head model.

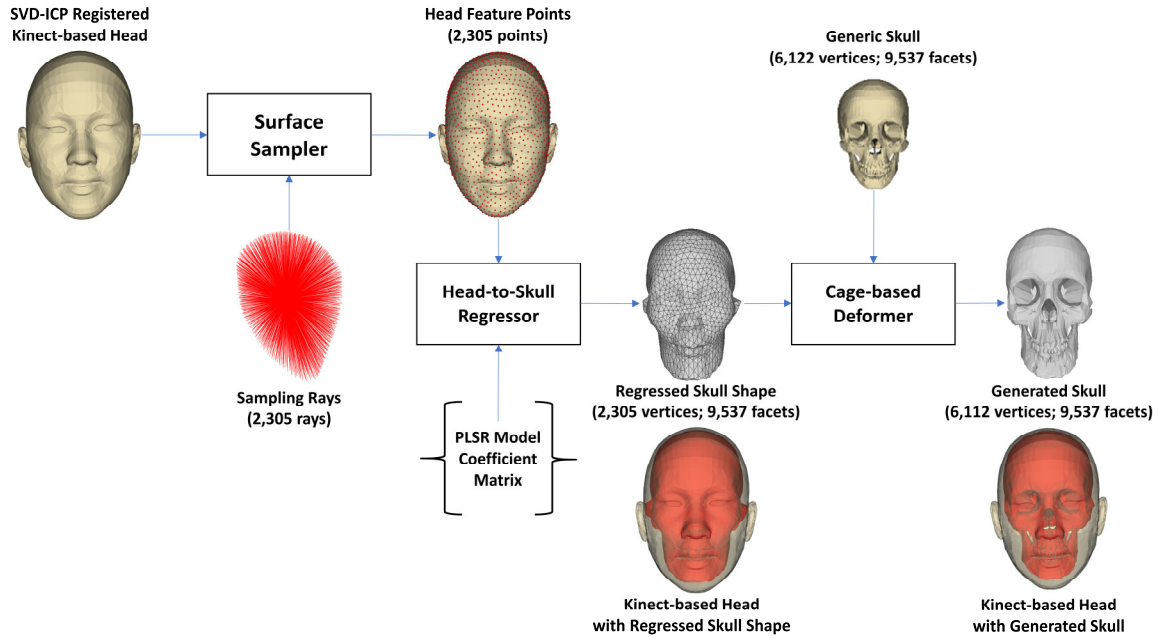


Figure 56. Kinect-based head-to-skull procedure. The Kinect-based head model was sampled to get head feature points, which was used to predict the skull shape using head-to-skull regressor. The generic skull was deformed to fit with the predicted skull shape.

5.1.4. Muscle network definition & generation process

Based on the generated head and skull models, facial muscles were generated. A network of the following muscles based on facial anatomy [4,5,217,218] was defined in both sides (left(L) and right(R)): *orbicularis oculi* (L/ROO), *orbicularis oris muscles* (L/RO), *frontalis inner* (L/RFI), *frontalis major* (L/RFM), *frontalis outer* (L/RFO), *corrugator supercilliary* (L/RCS), *nasalis* (L/RNa), *upperlip* (L/RU), *zygomaticus minor* (L/RZm), *zygomaticus major* (L/RZM), *risorius* (L/RR), *depressor anguli oris* (L/RDAO), *mentalis* (L/RM), *levator labii superioris* (L/RLLS), *levator labii superioris alaeque nasi* (L/RLLSAN), *levator anguli oris* (L/RLAO), *depressor labii inferioris* (L/RDLI), and *buccinator* (L/RB)) (Figure 57a). The insertion points were defined using the vertexes in Kinect-based head model, MPEG-4 facial feature points (FPs) [207] and face anatomy on healthy subjects [208]. The attachment points were defined by vertexes on the generated skull model (Figure 57b, Figure 57c). The positions of the attachment points were first manually defined in the generic skull model based on facial anatomy of healthy subjects [208] and then deformed to patient specific skull model (Figure 57c).

In addition to the muscle line geometrical representation, associated muscle features such as muscle length and strain were estimated and tracked. Regarding the muscle length, Euclidean distance metric between insertion and attachment points was computed. In particular, horizontal/vertical lengths of orbicularis muscles were also computed. Moreover, muscle strain of each muscle was computed as the relative difference change during a dynamic movement (e.g. smiling).

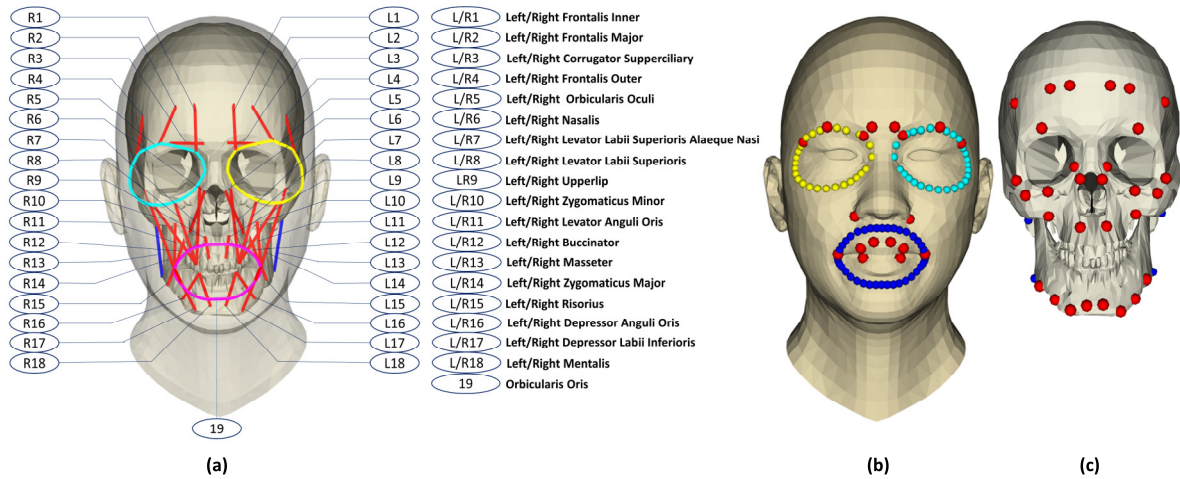


Figure 57. Muscle network definition: (a) selected muscles, (b) muscle insertion points on the head model, and (c) muscle attachment points on the skull model.

5.1.5. Model validation on healthy and facial palsy subjects

The proposed modeling workflow was validated on three healthy subjects (H1, H2, and H3) (2 males and 1 female) and two facial palsy patients (P1 and P2) (2 females) at the University Hospital Center of Amiens (CHU Amiens, France). Each subject had signed an informed consent agreement before participating into the data acquisition procedures. The protocol was approved by the local ethics committee (n°2011-A00532-39).

Regarding the head and skull models, MRI-based geometries were compared to the related generated geometries for each healthy subject and facial palsy patient. To reconstruct models from MRI images, different slice and mesh processing tools in 3-D Slicer [219] and MeshLab [220] were used. The reconstruction procedure is shown in Figure 58. The head and skull image slices (Figure 58a, Figure 58e) were first segmented to head and skull regions. The head segments were selected based on the pixel values of soft-tissue in MRI images so that all soft-tissue regions were selected in the head label (Figure 58b) using the threshold tool in 3-D Slicer. In MRI images bone structures are challenging to be segmented because pixel values in bone structures are relative similar to ones in empty regions. For each MRI slice, we first select both soft-tissue and bone regions using the level tracing tool in 3-D Slicer. The bone structures (Figure 58f) were formed by subtracting the selected regions from the head segment using the logic operator tool. After labelled, the head and skull models were reconstructed using the marching cube algorithm [193]. The reconstructed models were then smooth using Laplacian smoothing technique [194] (Figure 58c, Figure 58g). The head and skull models were finally post-processed to get only the external head and skull regions (Figure 58d, Figure 58h) using the ambient occlusion and vertex quality selection tools in MeshLab.

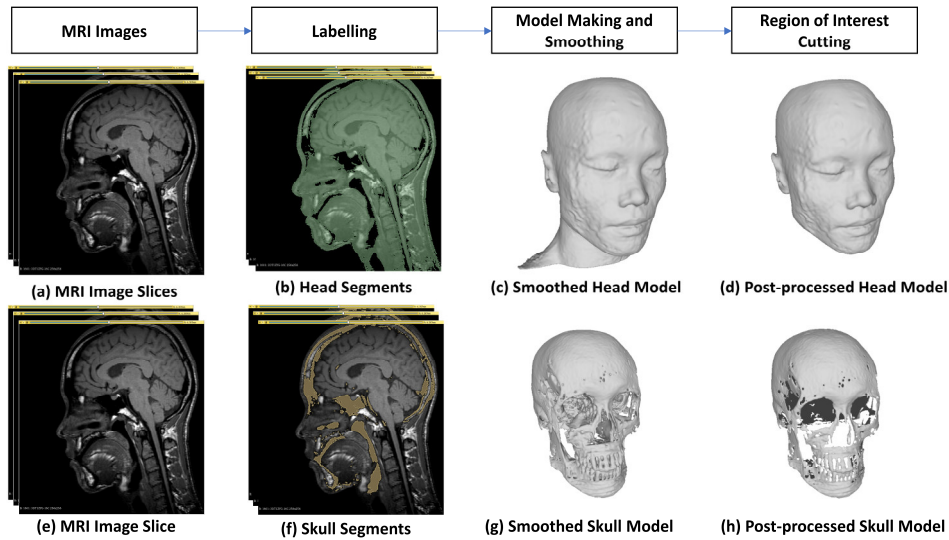


Figure 58. Head and skull reconstruction procedures from MRI images

The MRI-based head and skull models were registered to the coordinate system of the Kinect-based head and skull models based on their facial regions before compared in Hausdorff distance metrics [206]. The registration procedure was presented in our Kinect-to-head study [187]. In particular, the MRI-based face model were first registered to the Kinect-based face model based on manually selected landmarks on left & right eyes, forehead, top-nose, and mouth center using the SVD rigid registration method [198]. Registration errors due to manual landmark selections were optimally reduced using the ICP algorithm [175] based on all vertices of the face models. The estimated SVD-ICP transform matrix was used to transform the MRI-based head and skull models to the coordinate system of the Kinect-based head/skull models (Figure 59). Note that because internal structures of MRI-based skulls were hard to be fully reconstructed, their skull shapes were used for validations with the Kinect-based skull shapes. The skull shape generation procedure from skull models was presented in our head-to-skull study [216]. Because our MRI data of normal and facial palsy subjects were not all fully captured the head regions, only regions of interest were kept for validations using the Hausdorff distance metrics (Figure 59).

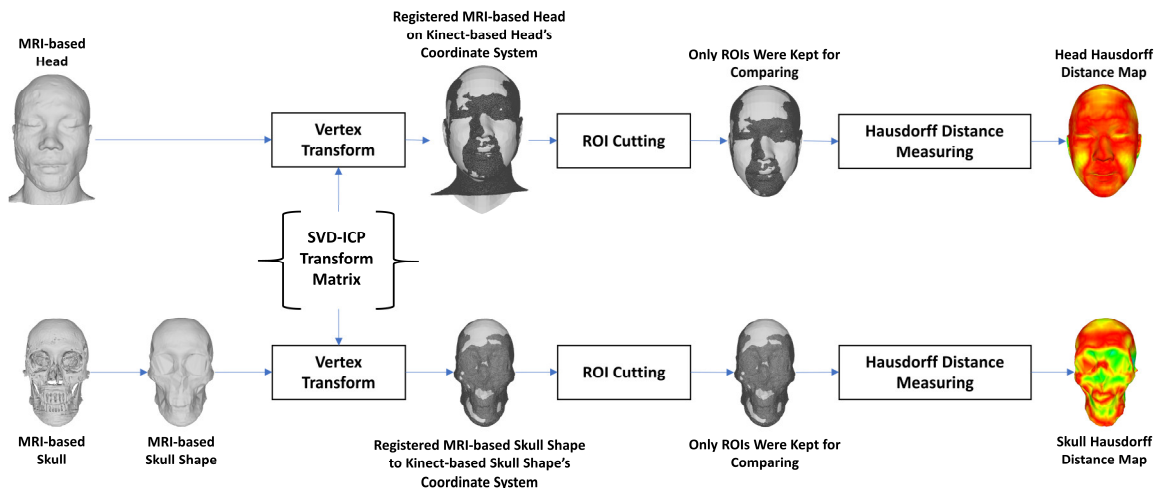


Figure 59. Validation procedure for Kinect-based head and skull models. The MRI-based head and skull models were transformed to the coordinate system of the Kinect-based head and skull models before drawing Hausdorff distance maps.

The Kinect-based head models were also validated with the animated head models reconstructed from the point cloud data acquired from the Kinect infrared sensor. Note that the RGB-D data acquired from Kinect sensors could be used to reconstruct facial models with

acceptable accuracy for facial analysis applications [108]. The reconstruction procedure was presented in details in Chapter 3. Because 3-D RGB-D point clouds and HD facial points were captured by the same Kinect sensor, the reconstructed head models from point clouds were in the same coordinate system with the Kinect-based head models. Consequently, Hausdorff distances could be directly computed without registrations.

Finally, the uncertainty of the definition of facial muscle insertion and attachment points due to manual manipulation was computed using 6-mm-radius spheres (Figure 60a) in a 10-fold validation [215]. In particular, the uncertainty spheres were centered at insertion/attachment points. For each time of validation, the actual positions of muscle points were randomly selected on their appropriate sphere surfaces and muscle lengths were also computed based on the selected positions (Figure 60b). The average muscle length of each muscle and its standard deviation were calculated after a 10-fold computation.

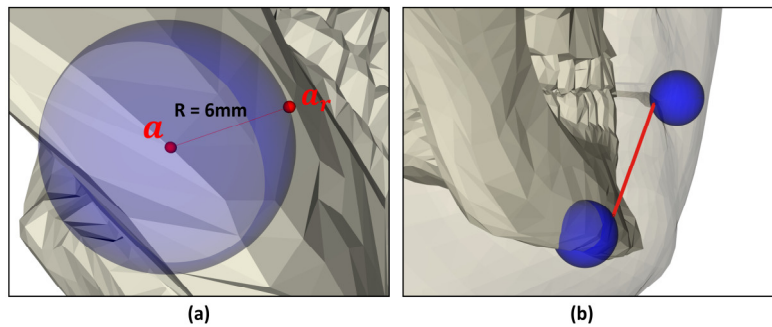


Figure 60. Perturbation positions of insertion and attachment points of the right risorius muscle on 6-mm-radius sphere: (a) 6-mm-radius radius sphere on the attachment point a , (b) the right risorius muscle based on randomly selected insertion and attachment points.

All modelling and validating procedures were executed on a mobile workstation system with the hardware configuration of Intel® Xeon® E-2176M CPU @ 2.7GHz 64 bits, 12 cores, 32GB DDRAM and developed in Microsoft Visual Studio C++ 2015.

5.2. Results

5.2.1. MRI-based model validation in the neutral position

Reconstruction and generation outcomes of the proposed modeling workflow are shown in Figure 61. Hausdorff distances of Kinect-based heads Vs. MRI-based heads and Kinect-based skulls Vs. MRI-based skulls were computed and illustrated in Figure 62. Additionally, the distance distributions are presented in Figure 63. For head comparison (Figure 63), the best mean errors are 1.91 mm and 1.98 mm for P2 and H2 respectively. Mean error of H1 (2.12 mm) is larger than ones of the H2 (1.98 mm) and P2 (1.91 mm). For skull comparison (Figure 63), the smallest mean error (3.12 mm) is in H1, and most errors are in the upper skull region. Only compared on the frontal skull regions, mean errors of H2 (4.97 mm) and P2 (4.32 mm) are larger than one of H1. The mean error of P2 is the largest (13.9 mm) in 4 subjects. Overall, the accuracy of Kinect-based skull models depends on the accuracy of the Kinect-based head models. Additionally, accuracies in facial head and frontal skull regions are better than one in back head and skull regions, especially in the muscle insertion and attachment regions (Figure 62).

Regarding the muscle features estimated in neutral position, muscle lengths were depicted in Table 5. Values were reported in average and standard deviation under the uncertainty effect of the manual manipulation for muscle definition. The overall length ranges from 21.53 ± 2.93 mm to 63.28 ± 2.99 mm. The minimal length is 21.53 ± 2.93 mm for the muscle LU. The maximal length is 63.28 ± 2.99 mm for the muscle RZM.

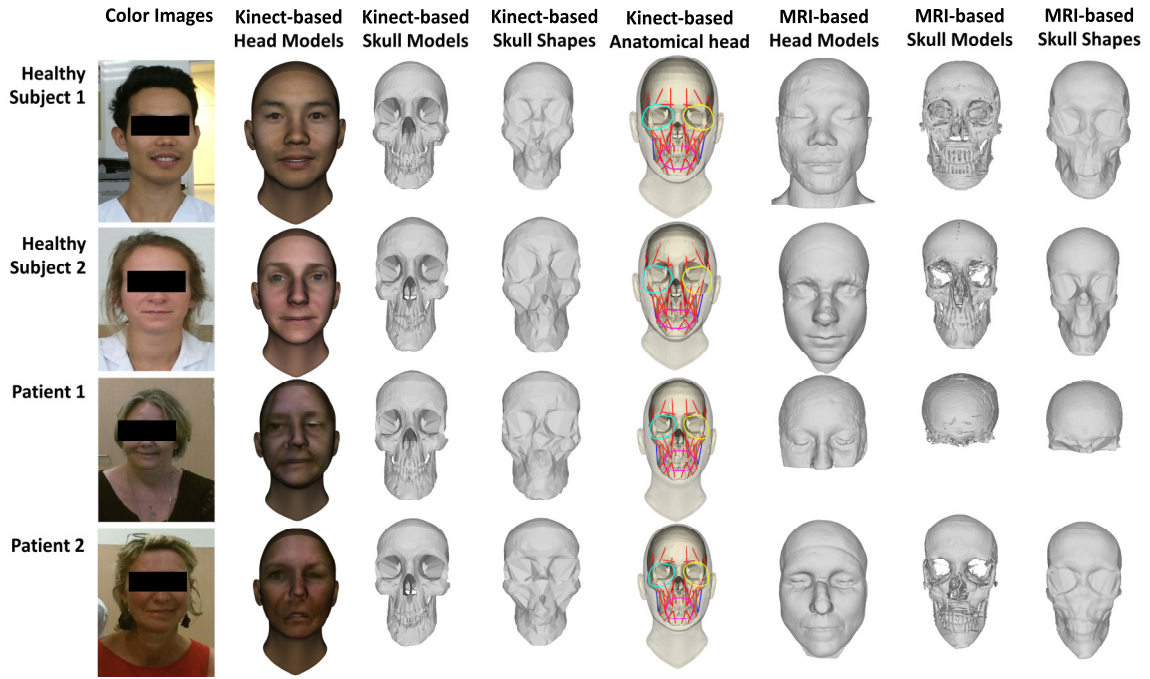


Figure 61. Kinect- and MRI-based reconstruction results for head, skull, skull shape, and muscle network of two healthy subjects and two facial palsy patients.

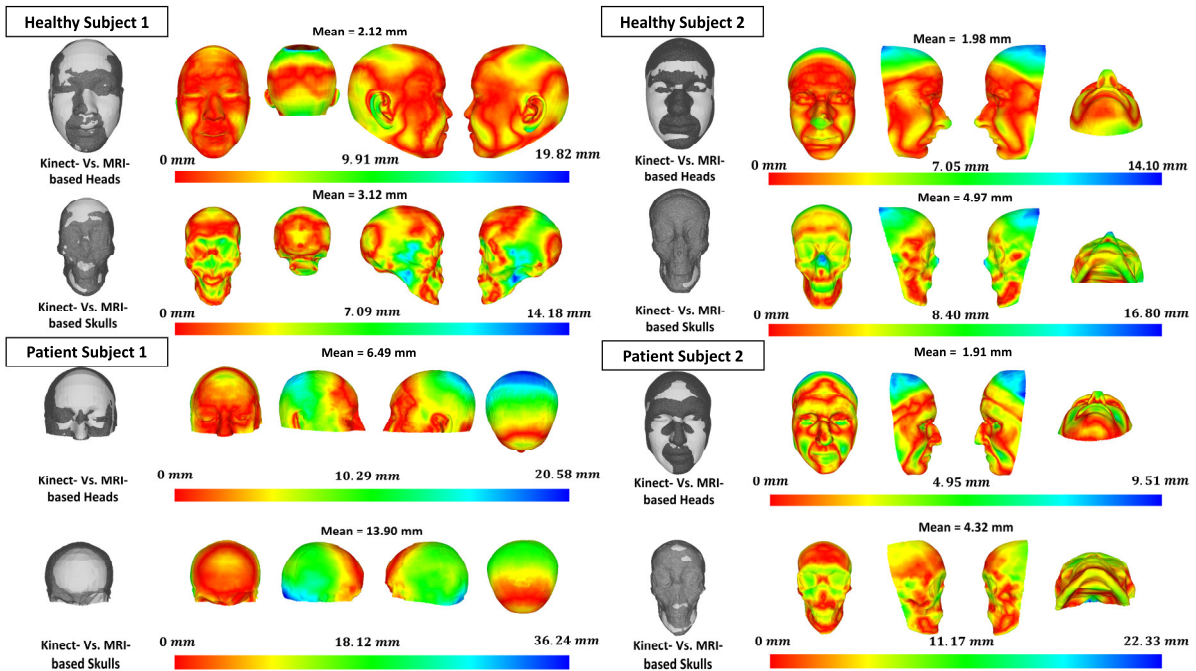


Figure 62. Hausdorff distance color maps between Kinect- and MRI-based head/skull models of healthy subjects and facial palsy patients.

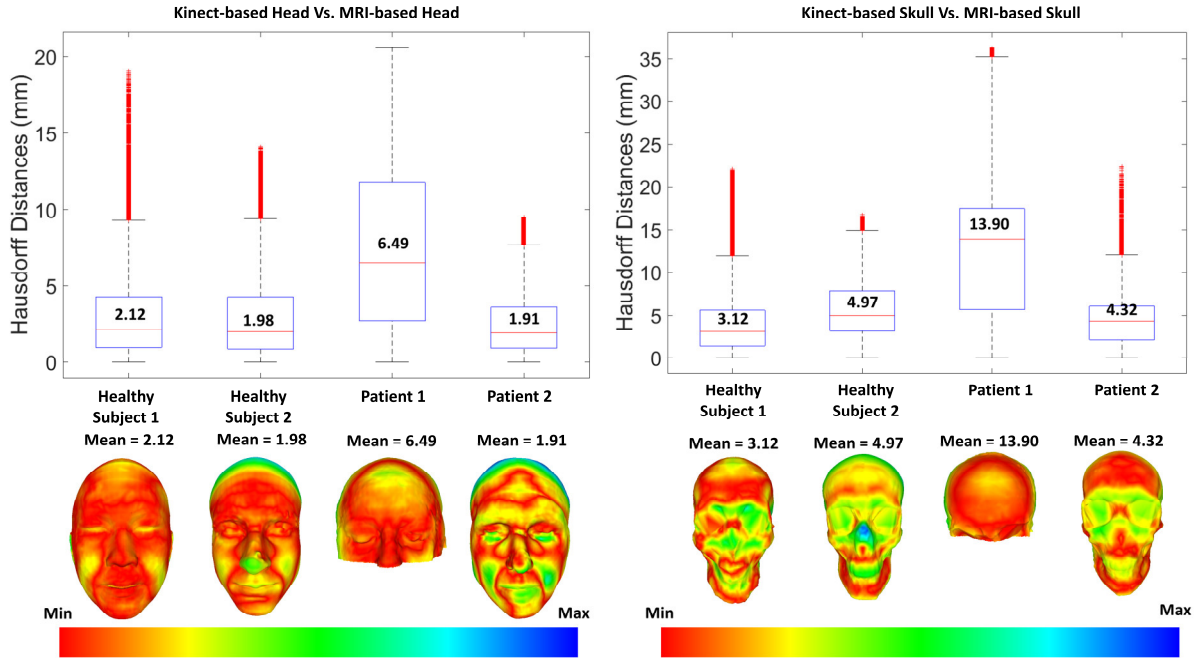


Figure 63. Hausdorff distance distributions between Kinect- and MRI-based head/skull models of healthy subjects and facial palsy patients.

Table 5. Muscle lengths of three healthy subjects and two facial palsy patients in neutral position.

Left/Right	Muscle Types	Muscle IDs	Action Line Lengths of Facial Muscles in Neutral Position ($\bar{x} \pm SD$ mm)				
			Healthy Subject 1 (H1)	Healthy Subject 2 (H2)	Healthy Subject 3 (H3)	Patient 1 (P1)	Patient 2 (P2)
Left	Frontalis Inner	LFI	31.22 \pm 1.89	42.97 \pm 1.87	28.48 \pm 2.06	31.77 \pm 1.94	32.96 \pm 1.77
Right		RFI	33.1 \pm 3.28	43.59 \pm 3.28	32.14 \pm 3.33	33.4 \pm 3.23	35.76 \pm 3.22
Left	Frontalis Major	LFM	28.54 \pm 2.16	41.56 \pm 1.92	28.21 \pm 2.13	29.5 \pm 1.94	29.58 \pm 1.89
Right		RFM	28.54 \pm 1.47	37.16 \pm 1.48	31.48 \pm 1.47	28.62 \pm 1.68	31.54 \pm 1.41
Left	Frontalis Outer	LFO	27.81 \pm 1.97	41.05 \pm 1.73	25.06 \pm 1.81	29.42 \pm 1.88	30.71 \pm 1.76
Right		RFO	30.87 \pm 1.74	37.12 \pm 1.67	34.34 \pm 1.73	31.6 \pm 1.67	34.73 \pm 1.97
Left	Corrugator Superciliary	LCS	26.99 \pm 1.91	26.29 \pm 2.17	27.25 \pm 1.84	24.49 \pm 2.32	25.1 \pm 2.05
Right		RCS	27.05 \pm 2.1	30.44 \pm 1.9	27.23 \pm 2.11	26.61 \pm 1.73	24.71 \pm 1.88
Left	Nasalis	LNa	30.81 \pm 2.22	34.88 \pm 2.23	23.54 \pm 2.29	26.03 \pm 2.62	28.78 \pm 2.33
Right		RNa	31.38 \pm 1.8	33.76 \pm 1.92	23.17 \pm 1.8	26.33 \pm 1.98	28.78 \pm 1.89
Left	Upperlip	LU	25.36 \pm 3.18	25.96 \pm 2.51	22.25 \pm 3.44	21.53 \pm 2.93	23.81 \pm 2.83
Right		RU	25.2 \pm 2.87	26.51 \pm 2.38	24.92 \pm 3.13	22.33 \pm 2.9	24.74 \pm 2.67
Left	Zygomaticus Minor	LZm	52.2 \pm 3.06	54.37 \pm 3.22	47.36 \pm 3.06	46.23 \pm 3.26	51.24 \pm 3.17
Right		RZm	54.42 \pm 1.91	53.18 \pm 2	54.7 \pm 1.89	48.91 \pm 2.21	53.98 \pm 2.09
Left	Left Zygomaticus Major	LZM	59.12 \pm 2.66	62.35 \pm 2.48	53.3 \pm 2.76	53.59 \pm 2.54	60.24 \pm 2.5
Right		RZM	60.93 \pm 2.96	59.31 \pm 2.98	61.36 \pm 2.94	56.78 \pm 2.97	63.28 \pm 2.99
Left	Risorius	LR	34.7 \pm 3.09	29.88 \pm 3.18	35.34 \pm 2.94	32.91 \pm 3.17	31.4 \pm 3.11
Right		RR	37.75 \pm 3.18	34.83 \pm 3.19	41.89 \pm 3.27	36.48 \pm 3.31	35.25 \pm 3.28
Left	Depressor Anguli Oris	LDAO	36.2 \pm 1.5	32.12 \pm 1.62	41.07 \pm 1.8	33.85 \pm 1.35	30.65 \pm 1.28
Right		RDAO	33.56 \pm 2.54	28.35 \pm 2.51	34.69 \pm 2.7	28.69 \pm 2.72	25.58 \pm 2.67
Left	Mentalis	LMe	28.41 \pm 1.59	23.86 \pm 1.44	36.91 \pm 1.53	29.45 \pm 1.54	26.58 \pm 1.57
Right		RMe	29.93 \pm 3.22	25.05 \pm 3.33	35.13 \pm 2.81	29.96 \pm 3.28	26.71 \pm 3.25
Left	Levator Labii Superioris	LLLS	47.43 \pm 1.41	50.67 \pm 1.39	42.67 \pm 1.4	41.33 \pm 1.32	47.12 \pm 1.4
Right		RLLS	49.76 \pm 2.07	49.31 \pm 1.94	47.98 \pm 2.17	43.76 \pm 2.09	49.5 \pm 2.01

Left/Right	Muscle Types	Muscle IDs	Action Line Lengths of Facial Muscles in Neutral Position ($x \pm SD$ mm)				
			Healthy Subject 1 (H1)	Healthy Subject 2 (H2)	Healthy Subject 3 (H3)	Patient 1 (P1)	Patient 2 (P2)
Left	Levator Labii Superioris Alaeque Nasi	LLSAN	59.96 \pm 2.61	62.54 \pm 2.12	54.18 \pm 2.6	51.84 \pm 2.01	57.49 \pm 2.22
Right		RLLSAN	61.62 \pm 1.94	62.28 \pm 1.97	56.71 \pm 1.87	53.24 \pm 1.83	59.1 \pm 1.89
Left	Levator Anguli Oris	LLAO	35.05 \pm 1.7	36.97 \pm 1.88	31.45 \pm 1.82	31.11 \pm 1.93	36.7 \pm 1.73
Right		RLAO	35.66 \pm 2.41	34.61 \pm 2.19	35.16 \pm 2.6	32.97 \pm 2.2	38.73 \pm 2.15
Left	Depressor Labii Inferioris	LDLI	36.64 \pm 2.92	30.62 \pm 3.23	39.55 \pm 2.64	35.1 \pm 3.34	33.21 \pm 3.12
Right		RDLI	35.86 \pm 2.48	31.24 \pm 2.58	41.27 \pm 2.4	35.04 \pm 2.55	33.87 \pm 2.63
Left	Buccinator	LB	59.05 \pm 3.23	55.38 \pm 3.16	56.23 \pm 3.23	50.68 \pm 3.12	56.03 \pm 3.18
Right		RB	57.18 \pm 1.09	52.77 \pm 1.17	56.7 \pm 1.13	49.84 \pm 1.42	55.35 \pm 1.32
Left	Masseter	LMa	45.16 \pm 2.42	45.5 \pm 2.39	44.08 \pm 2.43	41.06 \pm 2.39	43.82 \pm 2.4
Right		RMa	46.27 \pm 2.41	43.58 \pm 2.5	44.17 \pm 2.4	41.61 \pm 2.53	42.69 \pm 2.51
Vertical Left	Orbicularis Oculi	VLOO	42.47 \pm 2.85	41.3 \pm 2.84	40.99 \pm 2.79	35.98 \pm 2.91	40.37 \pm 2.91
Vertical Right		VROO	43.16 \pm 2.01	41.47 \pm 2.09	41.39 \pm 2.2	37.49 \pm 2.01	41.14 \pm 2.02
Vertical	Orbicularis Oris	VOO	38.09 \pm 2.64	30.82 \pm 2.63	33.47 \pm 2.66	32.56 \pm 2.63	34.85 \pm 2.68
Horizontal Left	Orbicularis Oculi	HLOO	59.38 \pm 1.66	54.35 \pm 1.84	60.21 \pm 1.73	49.59 \pm 1.96	53.02 \pm 1.77
Horizontal Right		HROO	59.73 \pm 1.19	54.9 \pm 1.2	60.4 \pm 1.13	49.98 \pm 1.25	53.75 \pm 1.16
Horizontal	Orbicularis Oris	HOO	59.17 \pm 2.02	57.93 \pm 2.1	55.18 \pm 2.06	49.25 \pm 2.45	48.36 \pm 2.2

5.2.2. Point cloud-based validation for different facial mimic positions

The comparison outcomes are presented in Figure 64. In this figure, 2-D color images (Figure 64a, Figure 64f) of two healthy subjects and two facial palsy patients are shown. Related Kinect-driven biomechanical models were generated according to appropriate facial mimics based on the HD facial points (Figure 64e, Figure 64j). Moreover, the animated head models were also reconstructed from the captured 3-D RGB-D point clouds (Figure 64b, Figure 64g). The Kinect-driven head models were also rendered in the same coordinate system with point cloud-based head models (Figure 64b, Figure 64g). Hausdorff distance distributions between the Kinect-driven head models and point cloud-based head models were illustrated (Figure 64c & Figure 64h and Figure 64d & Figure 64i). It is interesting to note that the facial palsy patients do not have geometrical symmetries while smiling and [u]-pronouncing compared with the healthy subjects. In particular, while smiling, the patient 1 (Figure 64.3a) cannot open her mouth as normally as H1 and H3 (Figure 64.1a, Figure 64.2a). Although P2 (Figure 64.4a) can open her mouth more widely than the patient 1 (Figure 64.3a), her right-hand mouth cannot smile as widely as the left-hand mouth due to malfunctions of her right zygomaticus minor and major muscles [118]. While [u]-pronouncing, P1 can only control her left zygomaticus minor and major muscles, so her right-hand mouth is not symmetrical with her left-hand mouth (Figure 64.3f). The patient 2 (Figure 64.4f) can do better than P1 (Figure 64.3f), but less symmetrical than H1 and H3 (Figure 64.1f, Figure 64.2f).

Based on error distributions between the Kinect-driven head models and point cloud-based head models (Figure 64c, Figure 64d, Figure 64h, Figure 64i), mean errors in smiling mimics are smaller than those in [u]-pronouncing mimics for each subject. For instances, the smiling mean errors of P1 and P2 (2.35 mm and 2.67 mm respectively) are smaller than the [u]-pronouncing mean errors of P1 and P2 (2.77 mm and 3.26 mm respectively). Moreover, the smiling mean errors of H1 and H3 (2.08 mm and 2.02 mm) are also smaller than the [u]-pronouncing mean errors of H1 and H3 (2.0 mm and 2.64 mm). In each facial mimic, mean errors of healthy subjects are usually smaller than ones of the facial palsy patients. For examples, the smiling mean errors of H1 and H3 (2.08 mm and 2.02 mm respectively) are smaller than ones of P1 and P2 (2.35 mm and 2.67 mm respectively). Moreover, the [u]-pronouncing mean errors of H1 and H3 (2.0 mm and 2.64 mm) are also smaller than ones of P1 and P2 (2.77 mm and 3.26 mm). For healthy subjects, the minimum mean error is 2.00 mm, and the maximum error is 2.64 mm. For facial palsy patients, the minimum error is 2.35 mm, and the maximum error is 3.26 mm.

The muscle strains estimated during facial mimic positions are reported in Table 6. When performing the smiling mimics, all subjects have horizontal elongations in their OOs (16.57%, 20.56%, 9.44%, and 19.93% for H1, H3, P1, and P2 respectively). Especially, P1 has the smallest elongations of 9.44%. Moreover, the strain values of L/RZms and L/RZMs are all negative, but the shortened ranges between the left and right muscles are different. For instances, in P1 the shortened ranges of LZm and LZM (0.40% and 6.76%) are smaller than ones of RZm and RZM (3.12% and 9.53%). When performing the [u]-pronouncing mimics, all subjects have horizontal shortenings in their OOs (-15.72%, -14.48%, -3.65%, and -19.70% for H1, H3, P1, and P2 respectively). Especially, P1 also has the smallest shortened range of 3.65%. Moreover, the strain values of L/RZms and L/RZMs are all positive, but their elongated ranges are not symmetrical between left and right muscles. For examples, in P2 the elongated ranges of LZm and LZM (4.59% and 10.02%) are not the same as ones of RZm and RZM (6.11% and 9.21%). In fact, these values can also illustrate asymmetries between left and right muscle actions.

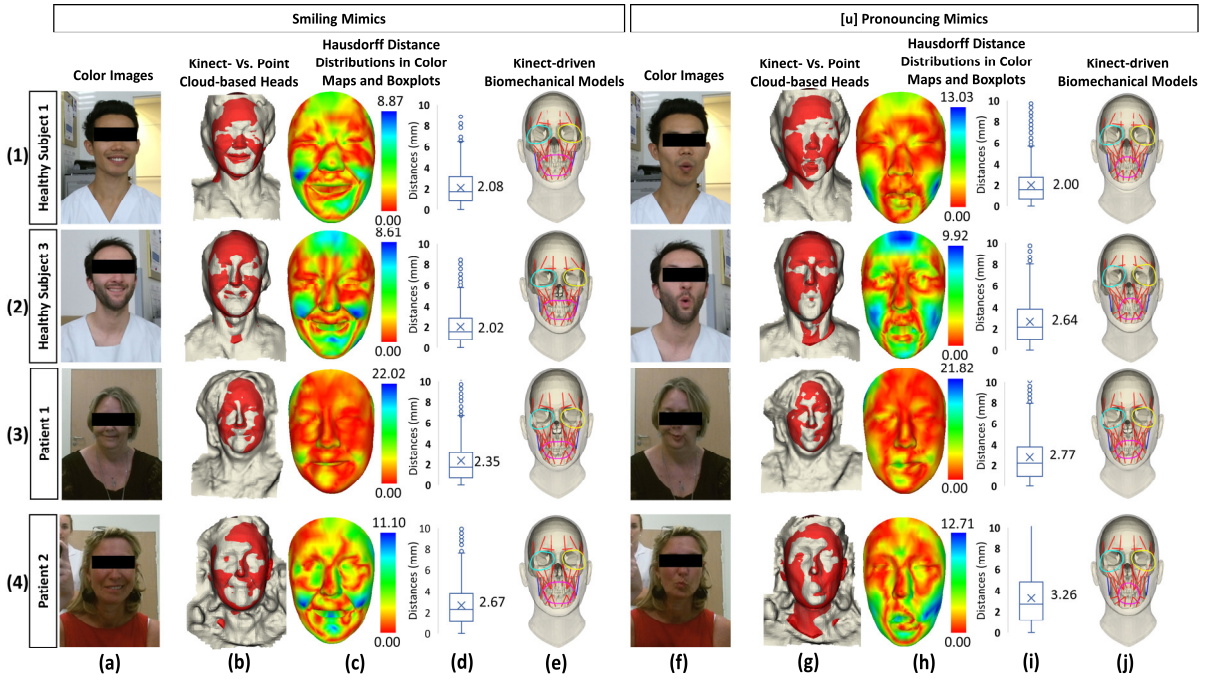


Figure 64. Validations results with point cloud-based models (color images, Kinect- Vs. point cloud-based head models, Hausdorff distance distributions, and Kinect-driven biomechanical models) in different facial mimics for two healthy subjects and two facial palsy patients: (a) 2-D color images, (b) Kinect- Vs. point cloud-based head models, (c) Hausdorff distance distributions in color maps, (d) Hausdorff distance distributions in boxplots, and (e) Kinect-driven biomechanical models.

Table 6. Muscle strain values in smiling and [u]-pronouncing mimics of the two health subjects and two facial palsy patients.

Muscle IDs	Muscle Strains in Positions (%)							
	Smile				[u]			
	H1	H3	P1	P2	H1	H3	P1	P2
LFI	20.62	1.78	2.67	-0.04	14.66	-9.26	1.36	-2.18
RFI	18.28	0.83	0.22	-1.83	12.20	-8.21	0.58	-7.90
LFM	12.78	-3.02	0.51	0.45	6.17	-12.33	-2.60	0.80
RFM	20.77	-0.45	1.51	1.03	14.19	-10.32	0.63	-0.91
LFO	10.38	2.19	2.96	1.58	1.63	-12.65	-0.93	-2.46
RFO	14.50	-1.38	-1.29	-2.24	7.60	-11.23	-3.37	-6.79
LCS	0.88	-5.03	-2.93	-2.48	-0.80	-3.04	-3.47	-2.89
RCS	9.50	3.15	3.12	4.98	7.73	2.78	3.05	1.68
LNa	2.28	-9.38	1.07	-5.60	15.68	14.33	2.51	8.75
RNa	-0.19	-9.32	-1.45	-9.23	11.37	10.46	1.77	4.39
LU	-12.02	-13.30	-7.52	-14.65	16.14	20.18	3.52	10.99
RU	-8.28	-13.14	-9.60	-17.35	21.66	19.42	3.12	10.49
LZm	-5.97	-9.93	-0.40	-7.23	8.94	8.59	2.84	4.59
RZm	-2.24	-9.93	-3.12	-9.81	13.46	7.68	3.54	6.11
LZM	-13.59	-21.32	-6.76	-11.82	10.76	17.70	-1.28	10.02
LZM	-9.13	-19.72	-9.53	-16.70	16.02	14.88	0.46	9.21
LR	7.30	3.55	0.98	-7.35	8.36	-12.23	4.87	-4.69
RR	6.96	-3.09	-0.90	-4.59	15.59	-3.57	5.51	9.70
LDAO	31.30	23.72	9.35	17.17	1.02	-22.67	4.19	-20.19
RDAO	31.20	29.20	18.59	37.20	-1.45	-22.55	6.88	-9.66
LMe	11.97	-7.79	2.92	-7.76	4.59	-32.65	0.64	-18.04

Muscle IDs	Muscle Strains in Positions (%)							
	Smile				[u]			
	H1	H3	P1	P2	H1	H3	P1	P2
RMe	1.70	-12.00	1.34	-8.00	0.20	-30.92	-2.33	-10.23
LLS	-7.56	-12.76	-2.33	-8.86	6.05	5.44	-0.31	2.02
RLLS	-7.31	-13.85	-5.89	-12.73	7.19	4.22	-0.58	1.88
LLLSAN	-0.69	-6.13	0.55	-4.08	5.51	2.62	0.21	1.24
RLLSAN	-1.99	-8.12	-1.88	-7.58	4.92	1.44	-0.66	-0.78
LLAO	-21.19	-28.03	-10.11	-16.02	12.05	25.79	-2.35	14.92
RLAO	-18.66	-29.46	-14.44	-24.61	18.80	23.93	0.67	12.93
LDLI	-5.57	-16.37	-0.67	-15.92	9.92	-17.24	4.44	-2.06
RDLI	0.30	-14.96	-2.31	-14.38	19.10	-7.43	3.55	10.18
LB	-12.58	-13.36	-5.51	-10.30	1.89	6.77	-1.71	4.33
RB	-9.22	-12.04	-5.05	-11.99	7.59	10.28	2.83	7.85
LMa	1.83	-0.99	-1.22	-1.08	1.83	-0.99	-1.22	-1.08
RMa	1.22	-1.39	-1.77	-1.73	1.22	-1.39	-1.77	-1.73
VLOO	-8.66	-8.66	-1.20	-7.04	2.09	11.66	-0.83	-3.19
VROO	-10.01	-11.99	-4.76	-11.95	0.84	9.04	-2.57	-2.61
VOO	18.62	21.71	-1.78	18.12	18.08	37.51	4.67	16.88
HLOO	-1.79	-2.34	-1.85	-2.22	-0.97	-0.83	-1.70	-1.63
HROO	-2.23	-2.45	-1.69	-2.99	-1.47	-0.90	-1.21	-2.25
HOO	16.57	20.56	9.44	19.93	-15.72	-14.48	-3.65	-19.70

5.3. Discussion

This study, for the first time, presents a biomechanical head modelling method for generating patient specific head, skull and muscle network from only HD facial points acquired by the visual Kinect V2.0 sensor. Computer-aided facial paralysis grading systems are important and necessary for quantitative and objective facial paralysis measurements before and during facial mimic rehabilitation [22]. Most developed computer-aided grading systems just analyzed external 2-D/3-D motions from facial appearances/movements extracted from visual 2-D imaging and/or 3-D point cloud data [57–68]. These exterior data could also be face image intensities [57], facial feature displacements [65,66], inter-feature distances, bounding areas, feature velocities, and feature symmetries from center lines [66,67,117]. These raw face appearances/movements needed large computation cost to be converted to meaningful information such as AUs from FACS [118] using 2-D/3-D computer vision-based methods [61]. In fact, AUs from FACS were mainly defined based on facial muscle actions [118]. Moreover, these external motions were just the effects caused by muscle actions on skin layers [3]. Thus, a directly analyze of facial muscle features is of great clinical interest. This present study proposed a complete workflow to reconstruct patient specific head and skulls model. Then, muscle features (e.g. muscle length and strain) could be estimated and tracked in a straightforward manner for facial paralysis applications.

Most previous studies extracted facial features directly from data acquired from visual sensors using 2-D/3-D computer vision-based methods. For instances, active shape model (ASM) was deformed to fit with counters of face, eyes, and nose for estimating facial features in video sequences [61,62], and the supervised descent method (SDM) could also be used to track facial features in 2-D images [66]. 3-D facial features could also be computed by deforming a 3-D generic facial model to fit with facial markers on 2-D images [63]. Computational complexity was decreased by using facial markers for tracking facial features. Less computation time

allowed us to detect facial features simultaneously on multiple images captured at different views for reconstructing their 3-D motions in motion capture systems [66–68]. 3-D motions of facial features detected on 2-D images could be estimated by combining infrared sensors with single cameras [116,117]. However, although facial features could be accurately detected and tracked in 2-D and 3-D spaces in above studies, large computation time was costed on processing raw data acquired from visual sensors. Our approach based on high-level subject-specific data supported by the Kinect SDK 2.0 controlling the visual Kinect 2.0 sensor, so much less computational time was used for extracting facial features.

In addition, numerous physics-based head/face models have been developed, but they mostly estimate facial animations based on muscle actions and have not extracted internal muscle actions from external facial appearances [3–5,69–71]. Moreover, most of them were not subject-specific or lacked of subject-specific skull layers [3,69–71]. FE facial models also costed much computation time for computing muscle displacements from forces [4,5]. In this presented study, we could fast generate subject-specific head models based only on external HD facial points acquired from the Kinect V2 sensor. After generated, the patient specific head model could be animated in real-time in rigid manner using current head orientations & positions and in non-rigid manner using current HD facial points. Therefore, we could achieve real-time head animations without computational complexities [187]. Last but not least, most physics-based face models from previous studies were developed based on semi-automatic procedures from other 3-D processing tools, such as 3DSlicer, ScanIP, Abacus, etc., or other studies. This could be inconvenient for the case of fast generating subject-specific models of new users. However, as shown in Figure 9, our proposed modelling method could fast generate subject-specific biomechanical head model according with texture. With the current hardware configuration, the total of head, skull, and muscle network generation costed 17.16 ± 0.37 s without counting time of reading and saving data from and to hard disk drives (HDDs). After generations, the framerate of data acquisitions, head animations, skull rigid transformations, muscle network computations, and graphic rendering was 40 fps.

From clinical point of view, the knowledge of muscle features is of great important for optimizing the treatment planning. The proposed method allowed muscle length and strain to be estimated and tracked in real time and in a patient-specific manner. The computed muscle lengths in neutral facial mimics were comparable with reported values in related studies [4,5,218,221,222], as listed in Table 7. Moreover, in comparisons with other accurate FE-based facial models [4,5], the ZMs in smiling were shortened with the negative strain values of -6.82% in [4]. In our study, as shown in

Table 6, the muscle strains of L/RZMs are also negative for all subjects. In [pμ] and [o] pronouncing mimics, the strain values of ZMs were all positive (10.4% and 24% for [pμ] and [o] sounds in [4]; 22% for [o] sound in [5]). In

Table 6, the strain values of L/RZMs are also positive in [u]-pronouncing mimics for H1, H3, and P2. However, in previous studies, the left and right facial muscle actions were considered to be perfectly symmetrical, so they just reported values on one side [4,5,218,221,222]. In our study, strain values of all major types of facial muscles were computed independently on left and right sides, so asymmetries of muscle actions could be evaluated during facial mimics.

Table 7. Muscle length comparisons with reported values in related studies.

Muscle IDs	Action Line Lengths of Facial Muscles in Neutral Position (mm)									
	H1	H2	H3	P1	P2	Freilinger et al., 1987 [221]	Happak et al., 1997 [218]	Bernington et al., 1999 [222]	Fan et al., 2017 [4]	Dao et al., 2018 [5]
LFI	31.22 ±1.89	42.97 ±1.87	28.48 ±2.06	31.77 ±1.94	32.96 ±1.77	-	-	-	-	-
RFI	33.1 ±3.28	43.59 ±3.28	32.14 ±3.33	33.4 ±3.23	35.76 ±3.22	-	-	-	-	-
LFM	28.54 ±2.16	41.56 ±1.92	28.21 ±2.13	29.5 ±1.94	29.58 ±1.89	-	-	-	-	-
RFM	28.54 ±1.47	37.16 ±1.48	31.48 ±1.47	28.62 ±1.68	31.54 ±1.41	-	-	-	-	-
LFO	27.81 ±1.97	41.05 ±1.73	25.06 ±1.81	29.42 ±1.88	30.71 ±1.76	-	-	-	-	-

Muscle IDs	Action Line Lengths of Facial Muscles in Neutral Position (mm)									
	H1	H2	H3	P1	P2	Freilinger et al., 1987 [221]	Happak et al., 1997 [218]	Bernington et al., 1999 [222]	Fan et al., 2017 [4]	Dao et al., 2018 [5]
RFO	30.87 ±1.74	37.12 ±1.67	34.34 ±1.73	31.6 ±1.67	34.73 ±1.97	-	-	-	-	-
LCS	26.99 ±1.91	26.29 ±2.17	27.25 ±1.84	24.49 ±2.32	25.1 ±2.05	-	-	-	-	-
RCS	27.05 ±2.1	30.44 ±1.9	27.23 ±2.11	26.61 ±1.73	24.71 ±1.88	-	-	-	-	-
LNa	30.81 ±2.22	34.88 ±2.23	23.54 ±2.29	26.03 ±2.62	28.78 ±2.33	-	-	-	-	-
RNa	31.38 ±1.8	33.76 ±1.92	23.17 ±1.8	26.33 ±1.98	28.78 ±1.89	-	-	-	-	-
LU	25.36 ±3.18	25.96 ±2.51	22.25 ±3.44	21.53 ±2.93	23.81 ±2.83	-	-	-	-	-
RU	25.2 ±2.87	26.51 ±2.38	24.92 ±3.13	22.33 ±2.9	24.74 ±2.67	-	-	-	-	-
LZm	52.2 ±3.06	54.37 ±3.22	47.36 ±3.06	46.23 ±3.26	51.24 ±3.17	-	51.8±7.4	-	-	-
RZm	54.42 ±1.91	53.18 ±2	54.7 ±1.89	48.91 ±2.21	53.98 ±2.09	-	51.8±7.4	-	-	-
LZM	59.12 ±2.66	62.35 ±2.48	53.3 ±2.76	53.59 ±2.54	60.24 ±2.5	M: 70.67±6.32 F: 69.50±6.58	65.6±3.8	-	43.65	52
RZM	60.93 ±2.96	59.31 ±2.98	61.36 ±2.94	56.78 ±2.97	63.28 ±2.99	M: 70.67±6.32 F: 69.50±6.58	65.6±3.8	-	43.65	52
LR	34.7 ±3.09	29.88 ±3.18	35.34 ±2.94	32.91 ±3.17	31.4 ±3.11	-	-	-	-	-
RR	37.75 ±3.18	34.83 ±3.19	41.89 ±3.27	36.48 ±3.31	35.25 ±3.28	-	-	-	-	-
LDAO	36.2 ±1.5	32.12 ±1.62	41.07 ±1.8	33.85 ±1.35	30.65 ±1.28	M: 37.83±4.38 F: 38.33±8.02	48±5.1	-	-	-
RDAO	33.56 ±2.54	28.35 ±2.51	34.69 ±2.7	28.69 ±2.72	25.58 ±2.67	M: 37.83±4.38 F: 38.33±8.02	48±5.1	-	-	-
LMe	28.41 ±1.59	23.86 ±1.44	36.91 ±1.53	29.45 ±1.54	26.58 ±1.57	-	-	-	-	-
RMe	29.93 ±3.22	25.05 ±3.33	35.13 ±2.81	29.96 ±3.28	26.71 ±3.25	-	-	-	-	-
LLS	47.43 ±1.41	50.67 ±1.39	42.67 ±1.4	41.33 ±1.32	47.12 ±1.4	M: 33.67±4.13 F: 35.50±6.69	47±7.5	-	29.3	-
RLLS	49.76 ±2.07	49.31 ±1.94	47.98 ±2.17	43.76 ±2.09	49.5 ±2.01	M: 33.67±4.13 F: 35.50±6.69	47±7.5	-	29.3	-
LLLSAN	59.96 ±2.61	62.54 ±2.12	54.18 ±2.6	51.84 ±2.01	57.49 ±2.22	-	61.6±7.6	-	-	-
RLLSAN	61.62 ±1.94	62.28 ±1.97	56.71 ±1.87	53.24 ±1.83	59.1 ±1.89	-	61.6±7.6	-	-	-
LLAO	35.05 ±1.7	36.97 ±1.88	31.45 ±1.82	31.11 ±1.93	36.7 ±1.73	-	42±2.5	-	27.4	-
RLAO	35.66 ±2.41	34.61 ±2.19	35.16 ±2.6	32.97 ±2.2	38.73 ±2.15	-	42±2.5	-	27.4	-
LDLI	36.64 ±2.92	30.62 ±3.23	39.55 ±2.64	35.1 ±3.34	33.21 ±3.12	-	29±4.9	-	-	-
RDLI	35.86 ±2.48	31.24 ±2.58	41.27 ±2.4	35.04 ±2.55	33.87 ±2.63	-	29±4.9	-	-	-
LB	59.05 ±3.23	55.38 ±3.16	56.23 ±3.23	50.68 ±3.12	56.03 ±3.18	-	56±7.4	-	-	-
RB	57.18 ±1.09	52.77 ±1.17	56.7 ±1.13	49.84 ±1.42	55.35 ±1.32	-	56±7.4	-	-	-
LMa	45.16 ±2.42	45.5 ±2.39	44.08 ±2.43	41.06 ±2.39	43.82 ±2.4	-	-	M: 45.9±5.8 F: 39.1±8.2	-	-
RMa	46.27 ±2.41	43.58 ±2.5	44.17 ±2.4	41.61 ±2.53	42.69 ±2.51	-	-	M: 45.9±5.8 F: 39.1±8.2	-	-
VLOO	42.47 ±2.85	41.3 ±2.84	40.99 ±2.79	35.98 ±2.91	40.37 ±2.91	-	60±9.6	-	-	-
VROO	43.16 ±2.01	41.47 ±2.09	41.39 ±2.2	37.49 ±2.01	41.14 ±2.02	-	60±9.6	-	-	-
VOO	38.09 ±2.64	30.82 ±2.63	33.47 ±2.66	32.56 ±2.63	34.85 ±2.68	-	-	-	-	-
HLOO	59.38 ±1.66	54.35 ±1.84	60.21 ±1.73	49.59 ±1.96	53.02 ±1.77	-	65±5.6	-	-	-
HROO	59.73 ±1.19	54.9 ±1.2	60.4 ±1.13	49.98 ±1.25	53.75 ±1.16	-	65±5.6	-	-	-
HOO	59.17 ±2.02	57.93 ±2.1	55.18 ±2.06	49.25 ±2.45	48.36 ±2.2	-	-	-	-	-

Despite potential capacity for clinical applications, the proposed modeling method has some limitations. In the Kinect-to-head process, back-head regions, which are often covered by hair, were approximated with face regions using affine transforms, so the back-head regions were less patient-specific than the face regions. This affected to the accuracy of the generated skulls. Relationship between face regions and back-head regions will be studied to improve accuracy of the Kinect-based head and skull models. In the head-to-skull process, the PLSR-based head-to-skull coefficient matrix was trained with a head-skull dataset of 209 healthy subjects. A larger number of datasets, especially including facial palsy patients, needs to be developed to enhance the training process of the PLSR model to improve the prediction accuracy. In the muscle network definition & generation process, only muscle lengths and strains could be computed during facial mimics. Based on muscle strains, muscle forces and stresses will be computed using rigid multi-bodies dynamics [223] and fast soft-tissue deformation methods (e.g. Mass-Spring System with corrective springs (MSS-CS) [224]). Moreover, jaw movements have not been included in the Kinect-driven head models. These movements will be considered when more facial mimics are analyzed. In validation process, facial muscle actions were only analyzed in smiling and [u]-pronouncing mimics on three healthy subjects and two facial palsy patients. More muscle action units in FACS [225] will be analyzed on a larger number of validation datasets. Moreover, accurate facial muscles should be reconstructed from MRI/CT images in different facial mimics to validate the extracted muscle features.

5.4. Conclusions

This chapter, for the first time, presented a novel method for modelling patient-specific head, skull, and muscle network using only external data acquired from a visual Kinect V2 sensor. The proposed method was evaluated with MRI data and the obtained results showed a high level of accuracy. Moreover, estimated muscle features were also in agreement with experimental and literature data. In the main system, this novel modeling approach is implemented in real-time facial paralysis grading during facial movements. The content of this chapter was published in the Computer Method and Program in Biomedicine journal (Q1, IF@2019 = 3.632) [226].

Chapter 6

Serious Games for Facial Mimic Rehabilitations

Facial mimic rehabilitation must contain both physical and psychological treatments in long-term recovering procedures, and serious games are particularly suitable for psychologically motivating patients during their rehabilitation exercises. Although various serious games have been designed for treatment and rehabilitation applications, but no serious games have been developed for facial palsy patients. Consequently, this chapter presents a novel serious game system for facial mimic rehabilitation. The serious game system was developed and communicated with other systems in the main system using the concept of system-of-systems (SoS). The patient-specific Kinect-driven biomechanical head model, as presented in Chapter 5, was used as the main serious game object interacted with game environments according to different game rules. Two serious games with three difficulty levels each were developed for training smiling and kissing mimics. With a user-friendly graphic user interface system, clinicians could easily change game difficulty levels and analyze playing performances to propose suitable rehabilitation exercises for their patients. Moreover, while playing games, users could also receive bio-feedbacks of both facial mimic parameters and muscle strains. In perspective, clinical and user acceptability validations will be conducted on facial palsy patients and clinical experts in short-term and long-term evaluation campaigns. More serious games for other facial mimics will also be developed for training more types and behaviors of facial muscles.

6.1. Facial mimic rehabilitation exercises and serious game strategies

Facial mimic rehabilitation procedures included five main components: (1) patient education, (2) soft-tissue mobilization, (3) functional retraining, (4) facial expression retraining, and (5) synkinesis management [21]. In patient education, patients were educated about facial nerves & muscles and their roles during expressions [21]. In soft-tissue mobilization, patients were softly massaged on their affected sides using soft-tissue mobilization techniques for improving mobilities and comfort on the affected facial muscles [27]. In functional retraining, patients were trained for basic oral competence referring to maintenance of basic mouth functions [28]. In facial expression retraining, patients were trained to conduct initial facial expressions such as smiles, sound pronunciations with passive movements for expression their emotions using facial mimics [29]. In synkinesis management, patients were trained to control and manage unwanted movements in facial relaxation positions [8].

Facial expression retraining and synkinesis management were the most difficult and time-consuming facial mimic rehabilitation components [14,16]. Current rehabilitation treatments include repeated training exercises for intensifying specific movements of appropriate facial muscles [8]. They were classified into rehabilitation exercises specified for stretching and relaxation of facial muscles, symmetry of facial muscle movements, and synkinesis management and movement control [8]. Serious games in the serious game system were designed based on this classification. This system includes two games with three different difficulty levels for each game. The first game called “*exploding bubble games*” was dedicated for training stretching and relaxation of facial muscles. In this game, the user tried to horizontally move his/her mouth to up/down scale bubbles. The bubbles would be exploded/released when their diameters reached the upper/lower thresholding values. While playing the games, facial muscles responsible for controlling smiling/kissing mimics were stretched repetitively. The second game called “*smile balancing*” games was designed for training symmetries of facial muscle movements and synkinesis management & movement control. In this game, the user tried to smile as symmetrically as possible to balance a table. Moreover, he/she could also use his/her left/right smiling mimics to move a ball to balance the

table. While playing the games, the user could train to control symmetries and synkinesis in smiling and neutral mimics.

6.2. Material and methods

6.2.1. Serious game system framework

The system framework of the serious game system is shown in Figure 65. The serious game system communicates with other systems based on the concept of SoS. In particular, this system was designed for executing parallelly with *Model Computation System*, *Central Managing System*, and *Graphic User Interface System*. Data among these systems were transmitted through high speed internal buses and managed by *Central Managing System*, so fewer latencies existed between processing and visual systems. This system configuration allowed users to play serious games, compute bio-feedbacks, and visualize bio-feedbacks simultaneously in real-time. The *Serious Game System* received Kinect-driven biomechanical head model and texture image from *Model Computation System* as one of its game objects. *Serious Game System* comprised of *Game Browser Subsystem*, *Serious Game 1 Subsystem*, and *Serious Game 2 Subsystem*. *Game Browser Subsystem* was responsible for managing game play data and navigating among games. Each serious game subsystem included *Game Graphic Engine* for rendering game objects, *Game Sound Engine* for generating sound effects, *Game Evaluator* for scoring game players, and *Mouse & Keyboard & Virtual Tray Listeners* for receiving basic commands from players through PC mouse and keyboard. Moreover, in *Serious Game 2 Subsystem* was also integrated *Game Physics Engine* for simulating physical interactions among game objects. Animated models & bio-feedbacks from *Model Computation System* and game object animations from *Serious Game System* were visualized in *Graphic User Interface System*. These data were also recorded by *Central Managing System*.

The main system was designed based on the Qt 5.12.5 framework [161]. This programming framework was responsible for graphic user interface developments, data transmissions, and multi-threaded system communications. *Game Graphic Engine* was developed based on Ogre3D game engine [151] for graphical object managing and rendering. *Game Sound Engine* was supported by irrKlang sound engine [163]. Physical effects were simulated by Bullet real-time physics engine [150]. *Mimic Visualizer* and *Graph Visualizer* was designed using VTK [162]. The whole system was executed in the hardware configuration of Intel® Xeon® E-2176M CPU @ 2.7GHz 64 bits, 12 cores, 32GB DDRAM. All procedures were developed in Microsoft Visual Studio C++ 2015.

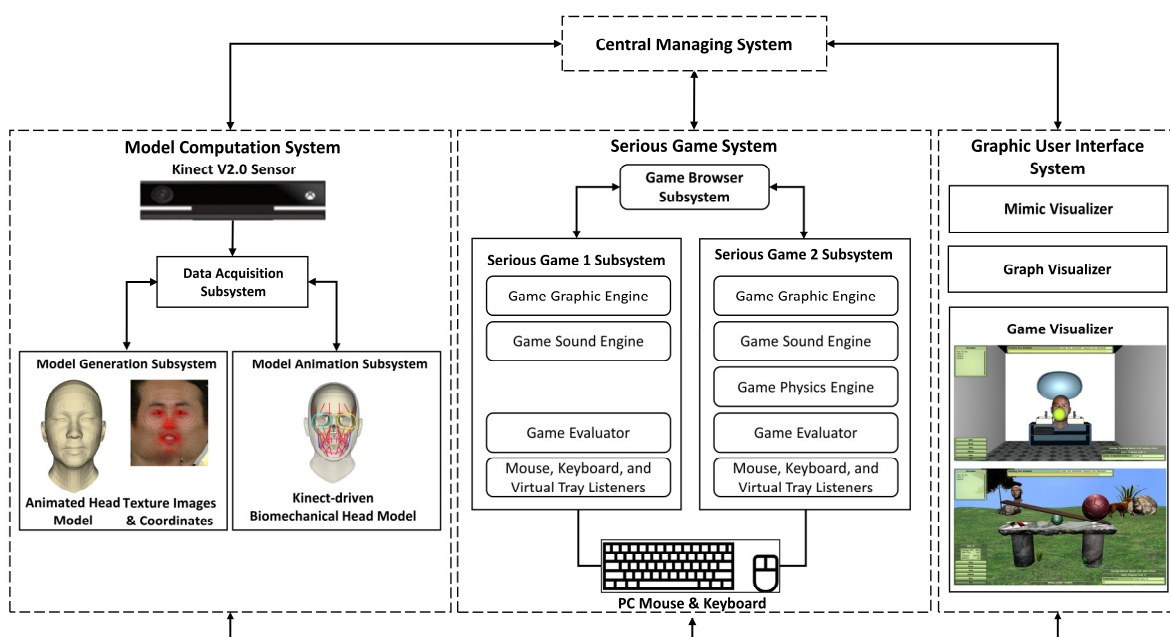


Figure 65. Serious game system framework in the main system based on the concept of SoS.

6.2.2. Game play workflow

Figure 66 shows the game play workflow of the serious game system. When starting the serious game system, the user was asked to create a new user or reuse the existing one. While creating a new user, the user was instructed to capture left, right, and front views for generating subject-specific biomechanical head model and face texture. The total generation time was $46.66 \pm 0.57s$. The model and texture generations were conducted in *Model Computation System*, as presented in Chapter 3 and Chapter 5. If reusing the existing subject-specific data, the user just fast scanned in front of the system to capture left, right, and front views for fast matching the generated data with his/her head states. The matching time was less than 1s. After the personalization procedure, the user selected a serious game through the general game GUI (Figure 66b). Game targets and difficulty levels were then set up in the GUIs of the appropriate game (Figure 66d) before starting the selected game (Figure 66c). While playing, the user could save his/her current results for later attempts. After playing, his/her results were saved and visualized on the game outcome interfaces (Figure 66d).

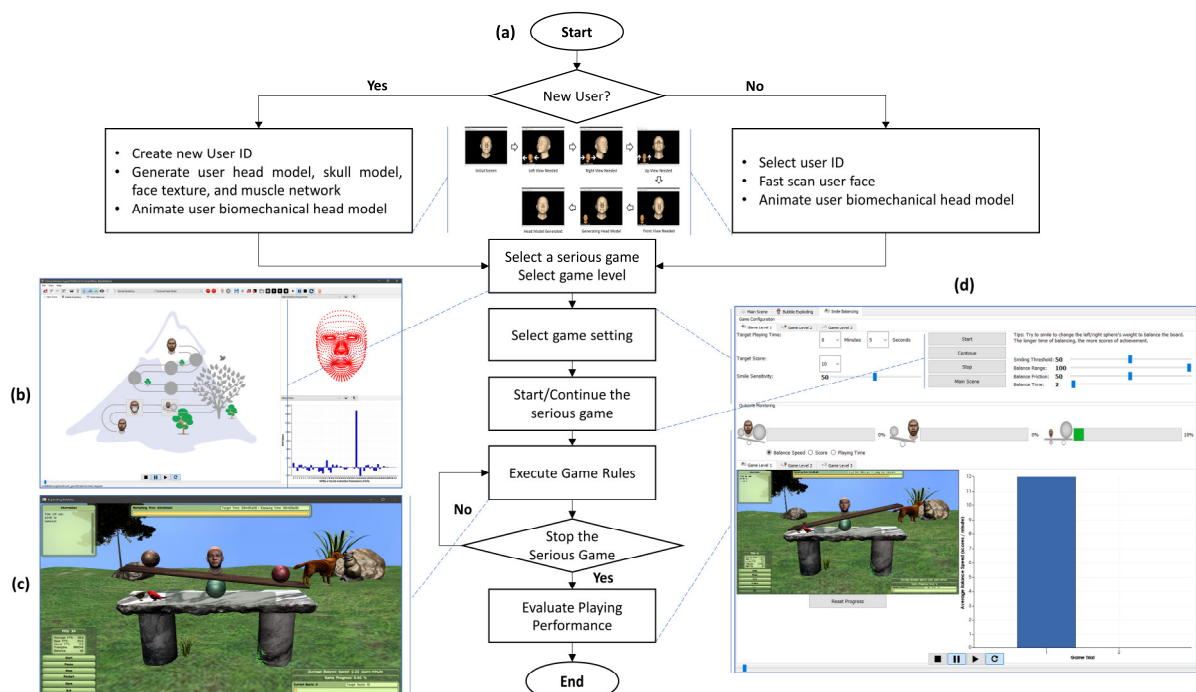


Figure 66. Game play workflow of the serious game system.

6.2.3. Game play GUIs

The serious game GUIs included the general and game GUIs. The general GUI (Figure 67) contained the *Game Browser Subsystem* interface for navigating among games, the *Mimic Visualizer* interface for rendering animated models in different visual options, and the *Graph Visualizer* interface for analyzing bio-feedbacks data in different graphing options. The game GUI (Figure 68) contains interfaces for setting game parameters, starting/stopping/continuing game rules, showing game tips, and reporting game progresses of the specific game level.

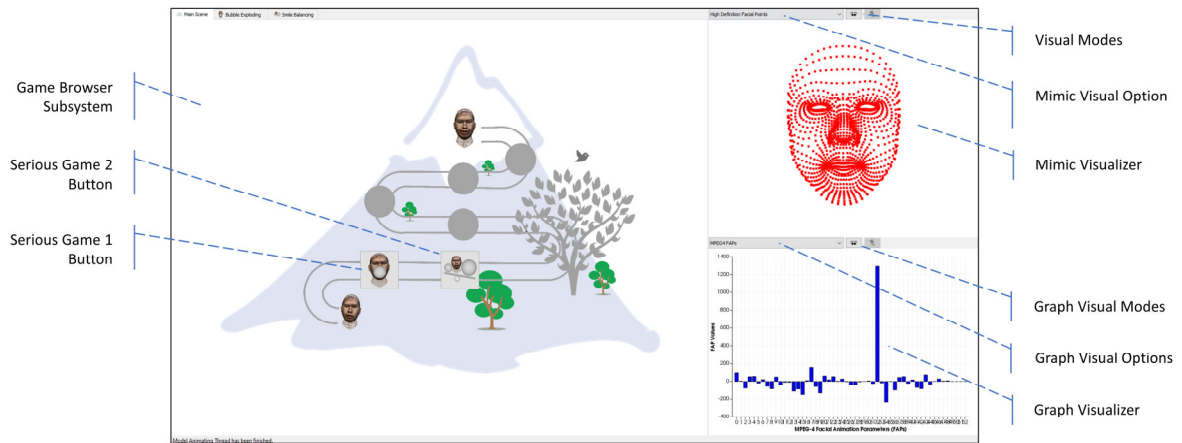


Figure 67. The general game GUI.

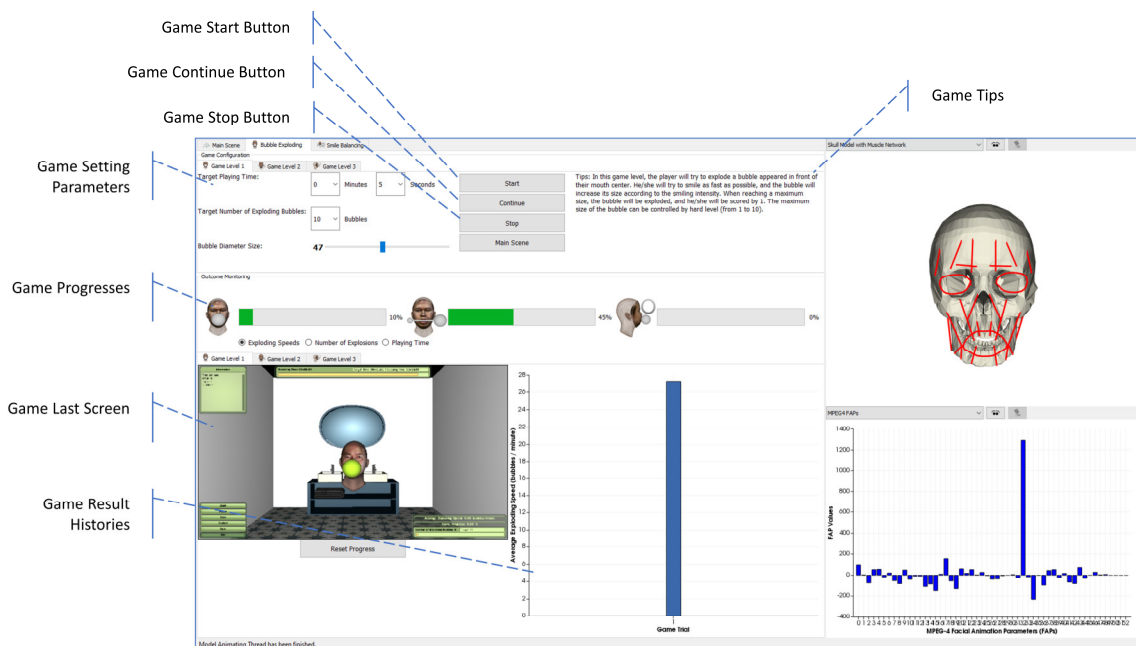


Figure 68. The GUI of each game.

6.2.4. Serious game 1: “Exploding bubbles”

The first game group called “*exploding bubbles*” games was designed for training stretching and relaxation of facial muscles responsible for smiling and kissing mimics. This game was designed into three difficulty levels called “*smile to explode bubbles*”, “*smile to explode left/right bubbles*”, and “*kiss to release bubbles*”.

6.2.4.1. Game level 1: “Smile to explode bubbles”

In the first level of the first game called “*smile to explode bubbles*”, the player tried to use his/her smiling mimics to explode the target bubble.

Figure 69b shows the rendering window of this game level. In the rendering window, the game objects included a user animated head model, an exploding bubble, a scoring panel, a timing panel, an information panel, a game control panel, and other decors. The game scene was designed in a luxury bath room for making positive cognitive effects to the player.

The game rule is illustrated in Figure 70a. In particular, after starting the game, smiling ranges that were computed as distances between left and right mouth corners were used to scale the exploding bubble (Figure 70b) while counting down the remaining time. When the bubble diameter reached the upper threshold value, it was exploded. The user was then scored as 1, and a new bubble with a different color was appeared at the mouth center when the smiling

range was in the neutral value (Figure 70c). If the total score reached the target score within the target playing time, the user was congratulated (Figure 70d). Otherwise, the user was failed and asked to play the game again (Figure 70e). While playing game, the remaining time was shown in the timing panel, the current score was shown in the scoring panel. The player could also start, pause, stop, restart, save, and exit the current game by using buttons in the game control panel. Moreover, game difficulties could be changed by tuning the target playing time, the target score, and the bubble diameter upper threshold could be tuned in the game parameter panel (Figure 69a).

As a result, by smiling to explode the bubble and neutralizing to create a new bubble repetitively to achieve the target score, muscles controlling the smiling mimics (left & right zygomatic minor and left & right zygomatic major [4,225,227]) would also be stretched and relaxed repetitively.

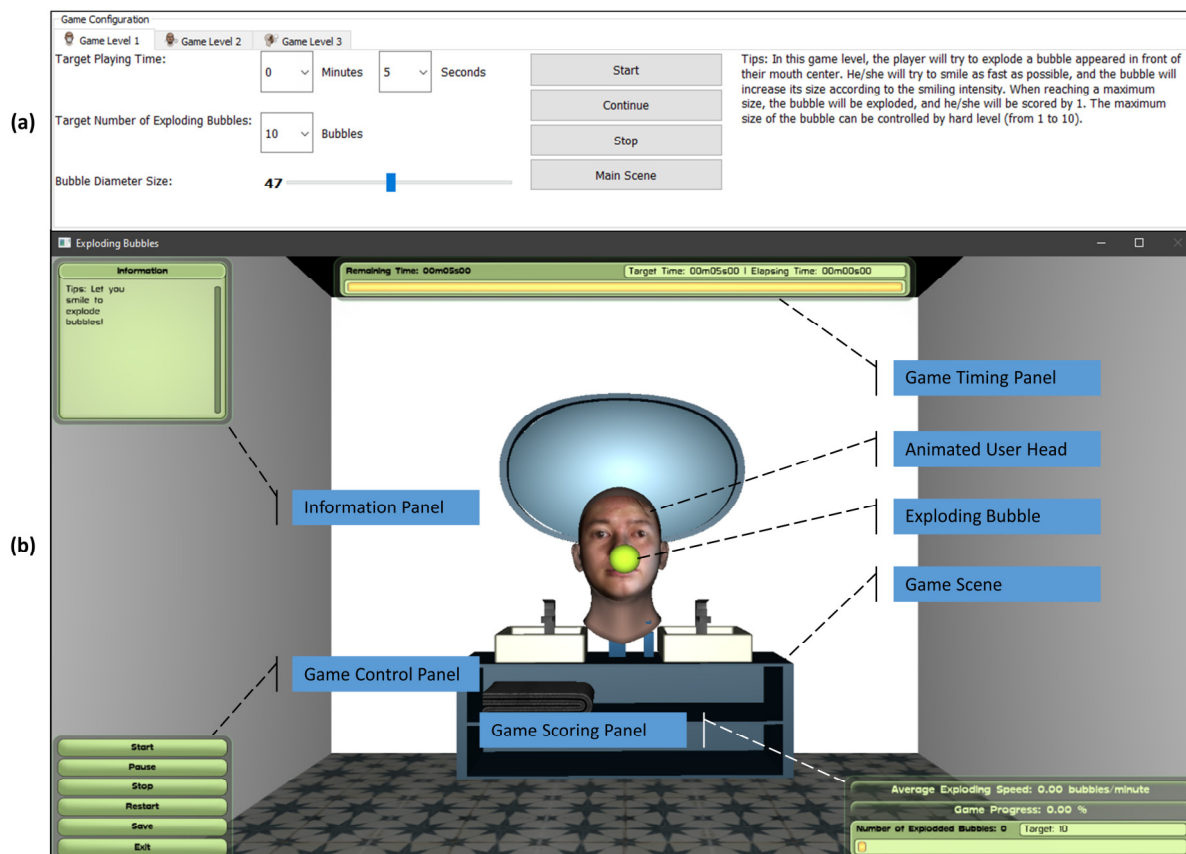


Figure 69. Interface of the “smile to explode bubbles” game: (a) game parameter interface and (b) game scene interface.

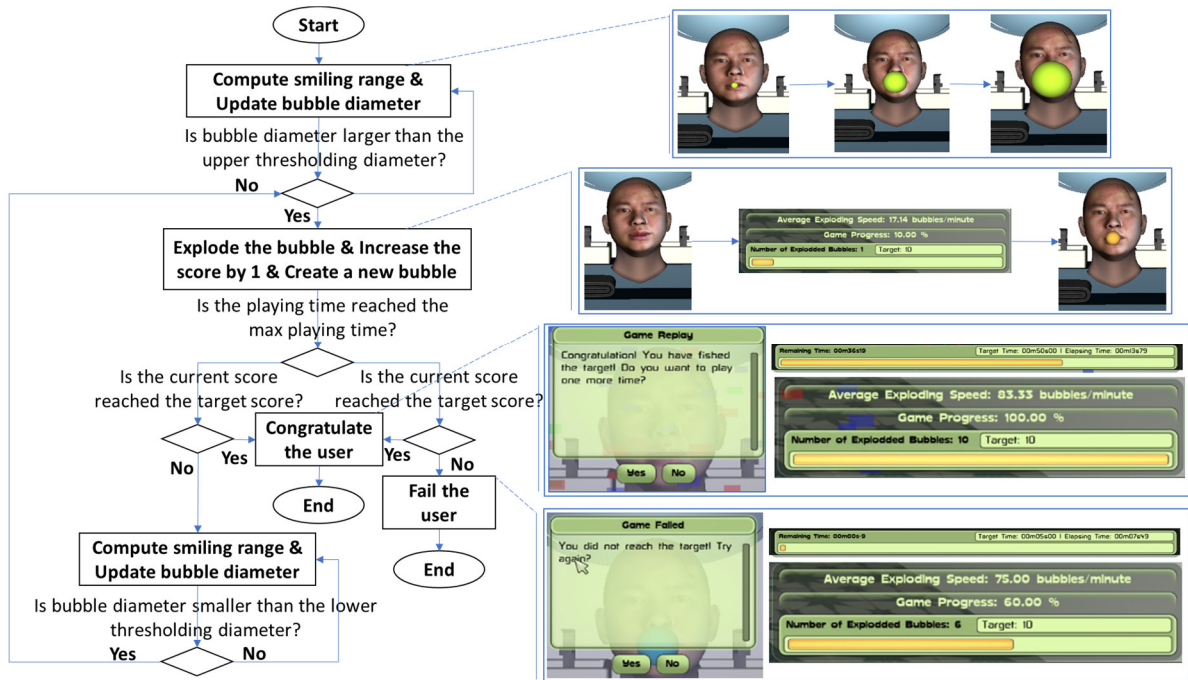


Figure 70. Game rule of the “smile to explode bubbles” game.

6.2.4.2. Game level 2: “Smile to exploded left/right bubbles”

In the second level of the first game, the “smile to explode left/right bubbles” was designed for training stretching and relaxation of left/right facial muscles responsible for controlling left/right smiling mimics. In this game, the player moved his/her mouth horizontally to the left/right side to explode the bubble on the left/right side of the mouth.

Figure 71b shows the rendering window of this game. The rendered game scene included an animated head model, left & right bubbles, left & right scoring panels, a timing panel, a game control panel, and other decors. This game was also designed in a scene of a luxury bath room for providing positive cognitive effects to the player.

The game rule is illustrated in Figure 72a. After starting, user head animations were tracked and animated on the animated head model. Left/right smiling ranges computed as distances between the left/right mouth corner and the mouth center were used to scale the left/right bubble size (Figure 72b). When the left/right bubble diameter reached its upper threshold value, it was exploded. The left/right score was then increased by 1. A new left/right bubble with a different color was appeared when left/right smiles become in the neutral mimic (Figure 72c). If both left and right total scores reached the target left & right scores within the target playing time, the player was congratulated (Figure 72d). Otherwise, he/she was failed and asked for next attempts (Figure 72e). The player could also set game parameters through the game parameter interface (Figure 71a). These game parameters are the target playing time, the left/right target score, and the left/right bubble diameter threshold.

As a result, by separately making left/right smiling mimics to explode the left/right bubble and returning into neutral mimics to create a new left/right bubble to achieve the left/right target score, the player would also train stretching and relaxation of his/her left/right zygomaticus minor & major muscles independently [4,225,227].

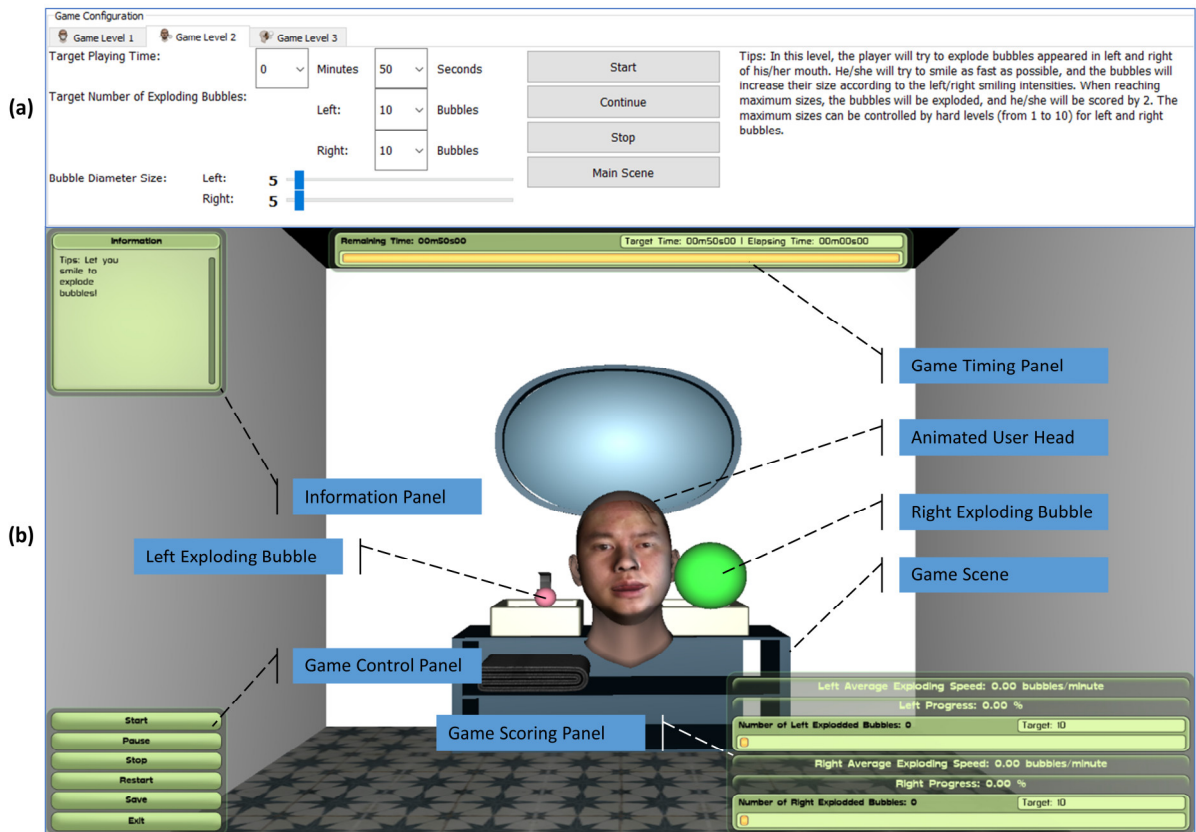


Figure 71. Interface of the “smile to exploded left/right bubbles” game: (a) game parameter interface and (b) game scene interface.

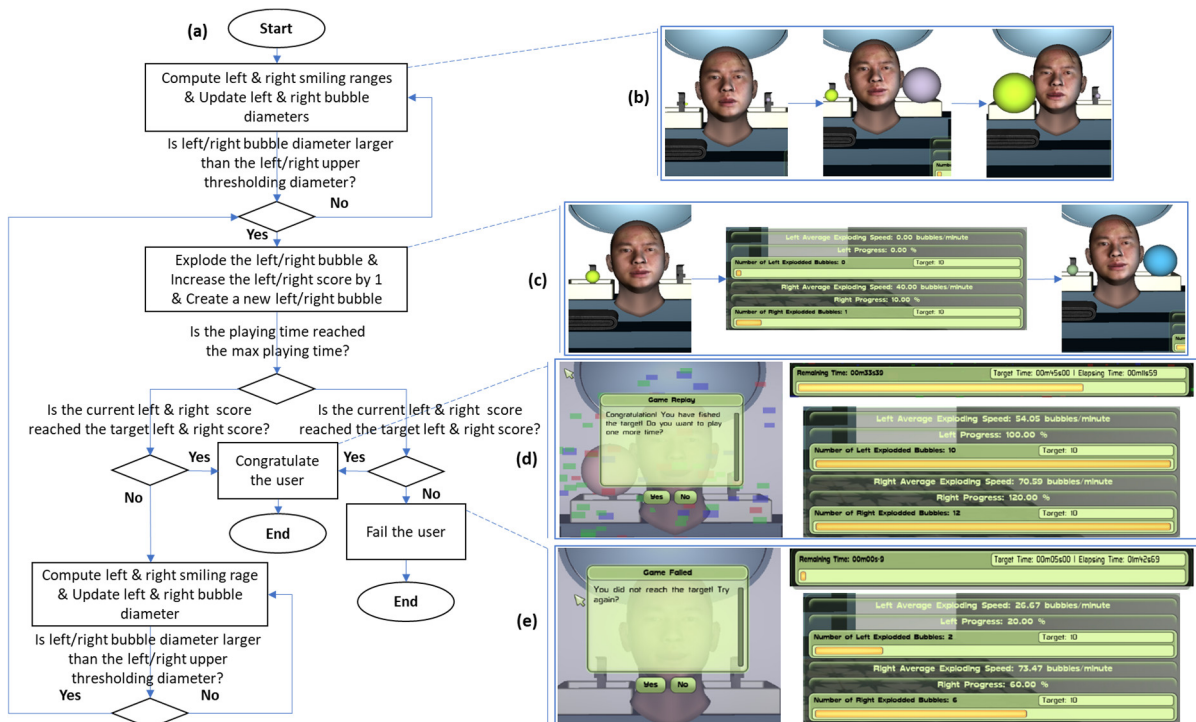


Figure 72. Game rule of the smile to exploded left/right bubbles” game.

6.2.4.3. Game level 3: “Kiss to release bubbles”

In the third level of the first game, the “kiss to release bubbles” game was designed for training kissing mimics. In this game, the player tried to horizontally shorten his/her mouth to downscale the target bubble so that it could pass through a ring.

Figure 73b shows the game render window. The game objects of this game included an animated head model, a releasing bubble, a passing ring, a scoring panel, an information panel, a game control panel, a game timing panel, and other decors. They were designed in a bath room for providing relaxed cognitive effects to the player.

The game rule is presented in Figure 74a. After starting the game, distances between the left and right mouth corners were computed in real-time for downscaling the target bubble (Figure 74b). When the bubble diameter was smaller than the ring diameter, it was released and passed across the ring. The total score was then increased by 1. If the facial mimic became to neutral, another bubble with a different color would appear at the mouth center (Figure 74c). If the total score reached the target score within the target playing time, the game target was achieved (Figure 74d). Otherwise, the play was failed and asked for next attempts (Figure 74e).

Overall, by conducting the game rules repetitively during the target playing time, the left & right zygomaticus minor & major muscles were also trained to elongate and relax more effectively.

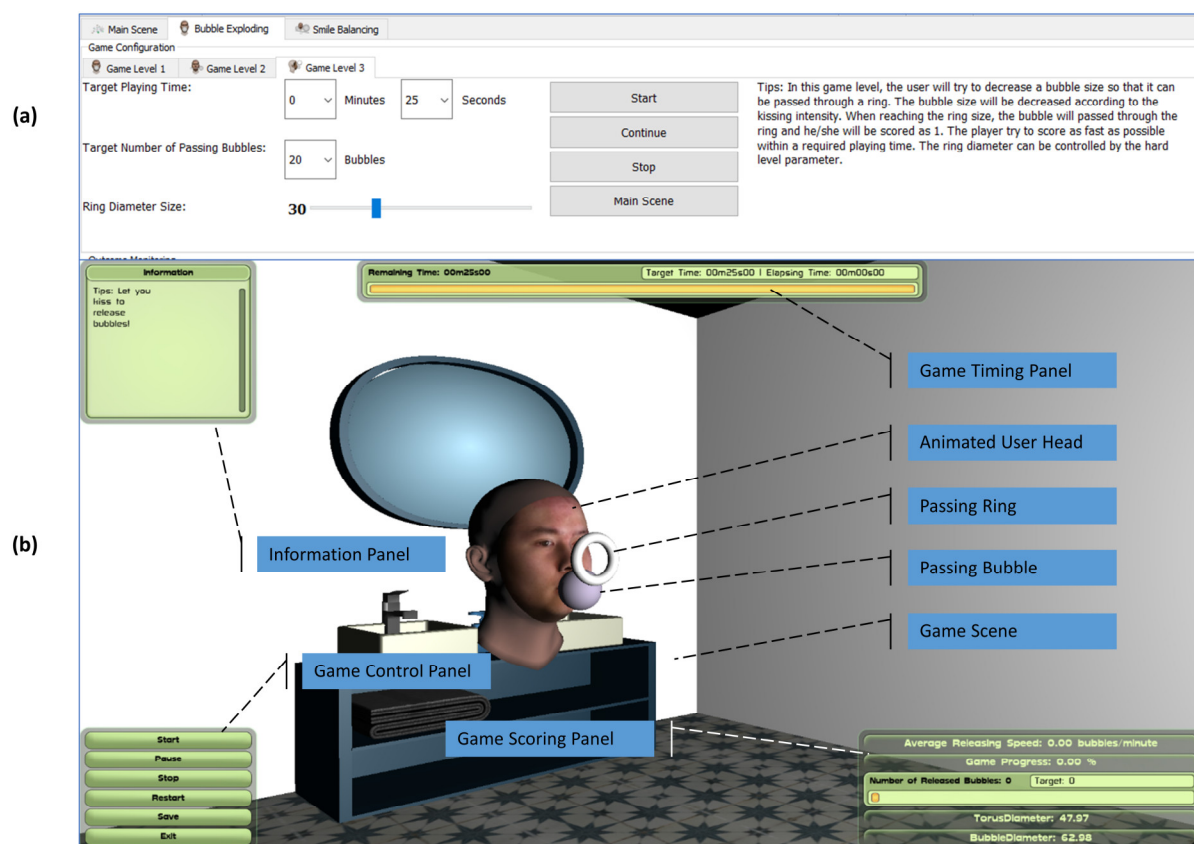


Figure 73. Interface of the “kiss to release bubbles” game: (a) game parameter interface and (b) game scene interface.

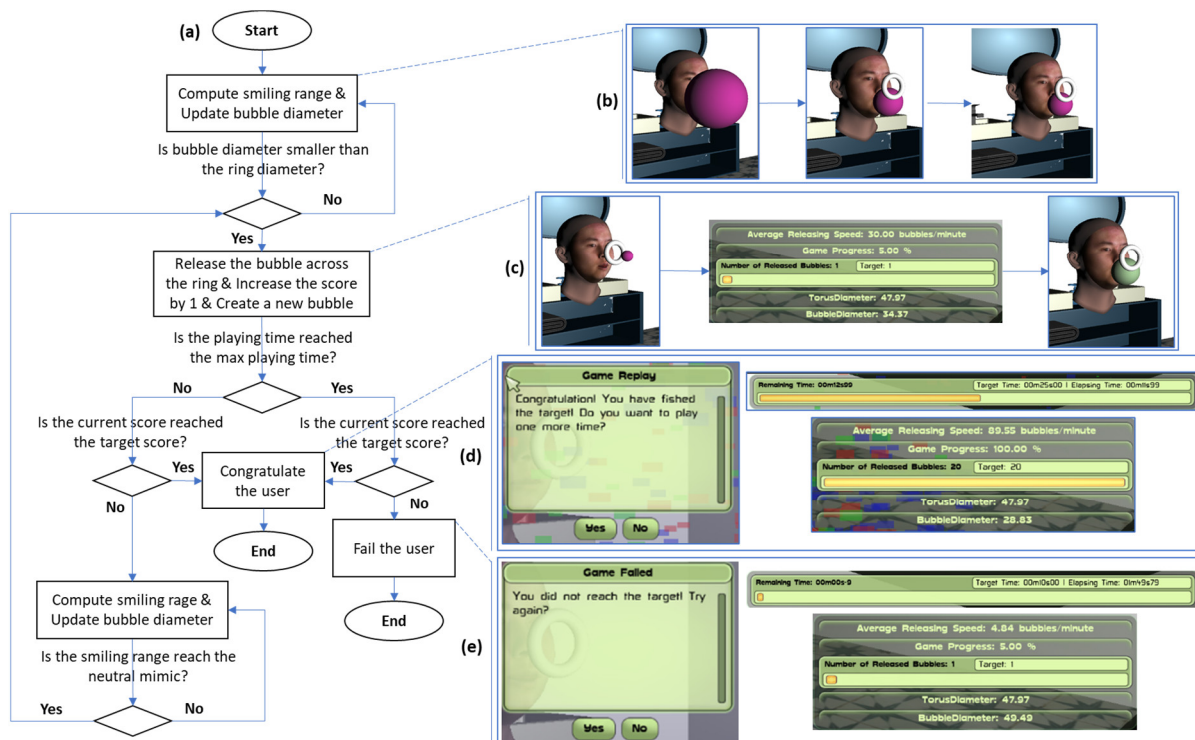


Figure 74. Game rule of the “kiss to release bubbles” game.

6.2.5. Serious game 2: “Smile balancing”

The second game called “*smile balancing*” games was designed for training symmetries of facial muscle movements and synkinesis management & movement control. In this game, the player tried to control his/her mouth for balancing a table in a target period. Three games were designed with different ways of balancing: “*smile to balance left & right balls*” game, “*smile to balance ball weight*” game, and “*smile to lift ball*” game.

6.2.5.1. Game level 1: “*Smile to balance left & right balls*”

In the “*smile to balance left & right balls*” game, the player tried to make smiling mimics as symmetrical as possible to balance a table in a target duration of time.

The game rendering window (Figure 75b) contains an animated head model, left & right balls, a balance board, a balance supporter, an information panel, a time-to-score timing panel, a game timing panel, a game control panel, a game scoring panel, and other decors. The game scene was designed in a green-grass garden with animals for providing positive and optimistic cognitive effects to the player. Physical effects were also integrated to the balance system. The left and right balls were fixed at left and right corners of the balance board. The balance board was connected to the balance supporter so that it could only rotate around the balance supporter. The balance board could not move if in contact with the stone table. The rotational friction could be set by changing the *balance friction* in the game parameter interface (Figure 75a).

The game rule is illustrated in Figure 76a. After starting, left/right smiling ranges that were computed as distances between the left/right mouth corner and the mouth center were used to scale the size and change the weight of the left/right ball. According to the physical simulations, the balance board was rotated around the balance supporter corresponding to the current weights of the left & right balls (Figure 76b). Angles between the balance board and the stone table were also computed. If the balancing angles were within the *balance range* and the left/right smiling ranges were larger than the *smiling threshold*, the time-to-score timer began to countdown from the *balance time*. After the *balance time*, the total score was added by 1 (Figure 76c). If the total score reached the target score within the target playing time, the player would successfully finish the game (Figure 76d). Otherwise, the player would be failed and

asked for next attempts (Figure 76e). *Smiling threshold*, *balance range*, *balance friction*, and *balance time* could be set in the game parameter interface (Figure 75a).

As a result, by trying to achieve the game target, the player would be trained to generate symmetrical smiling mimics. Moreover, while keeping balance states in a time period, the player would also train for controlling synkinesis during smiling mimics.

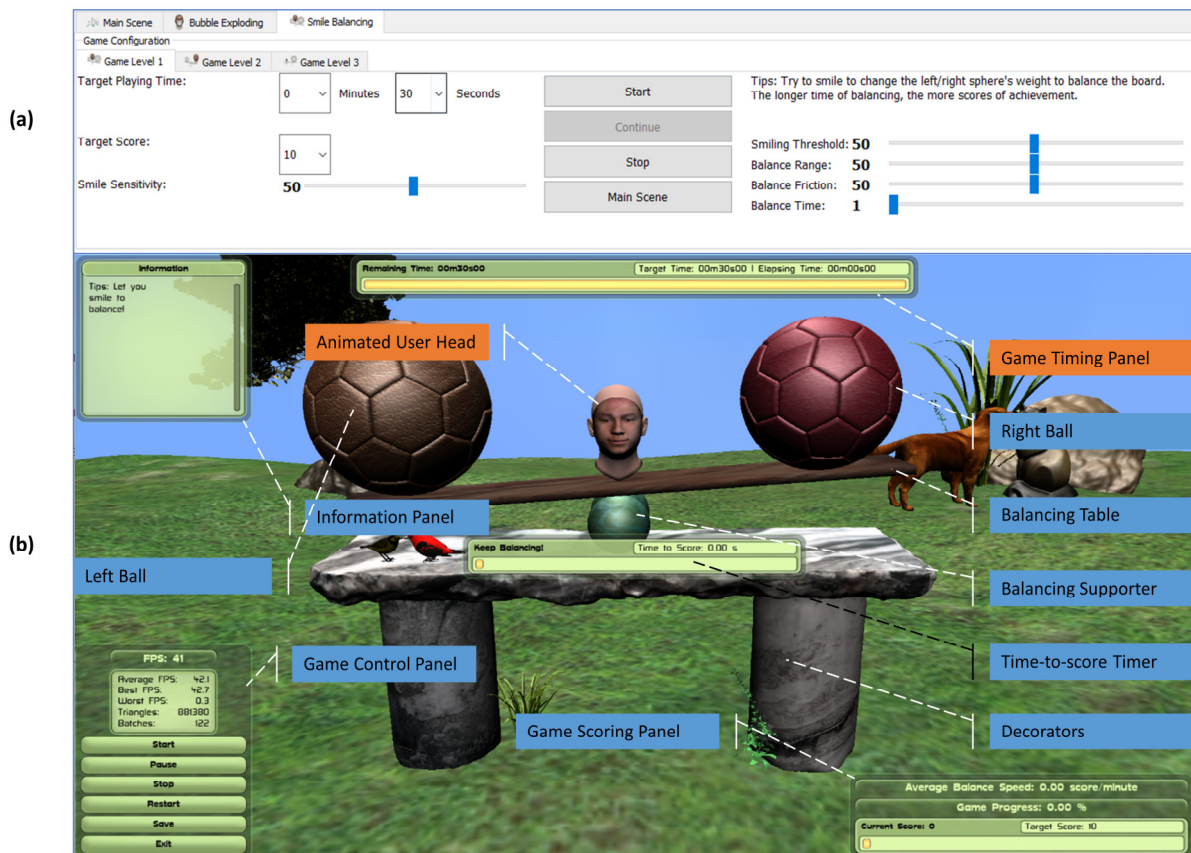


Figure 75. Interface of the “smile to balance left & right balls” game: (a) game parameter interface and (b) game scene interface.

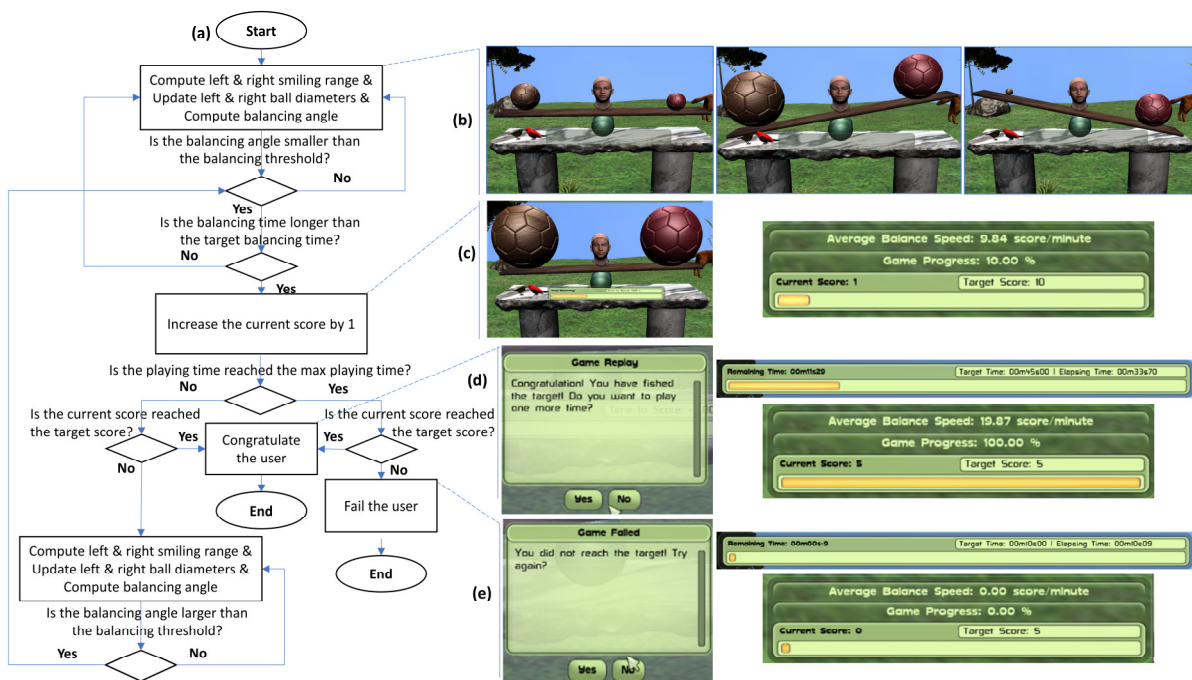


Figure 76. Game rule of the “smile to balance left & right balls” game.

6.2.5.2. Game level 2: “Smile to balance ball weights”

The “smile to balance ball weights” game was design for separately training the left/right muscles responsible for controlling smiling mimics.

In the game render window (Figure 77b), the game objects and the physical effects were the same as ones in the “smile to balance left & right balls” game; however, only one ball was fixed on the left/right corner of the balance board, and the animated head model was fixed on the other corner of the balance board.

The game rule is shown in Figure 78a. After starting, the left/right smiling ranges were computed to change the weight of the animated head. The balance states of the balance system were also updated according to the current weight of the animated head and the left/right ball (Figure 78b). If the balance angles between the balance board and the stone table were in the *balance range* as long as the *balance time*, the player was scored by 1 (Figure 78c). During the target playing time, if the player was able to achieve the target score, he/she would successfully complete the game target (Figure 78d). Otherwise, he/she would be failed and asked for next attempts (Figure 78e). Moreover, in this game, the player could choose the playing mode as preset, linear, or random. In the preset playing mode, the ball was optionally set at the left/right corner of the balance board with a static weight. In the linear playing mode, the weight of the left/right ball was linearly increased after each scoring. In the random playing mode, the ball was randomly set at left/right corner of the balance board with a random weight. All game parameters were selected in the game parameter interface (Figure 77a).

As a result, in this game, the ability of controlling synkinesis was mostly trained when the player tried to move his/her mouth to the left/right so that the animated head could be balanced with the left/right ball. His/her mimic stretching and speeds could also be improved when playing the game in the linear and random modes.

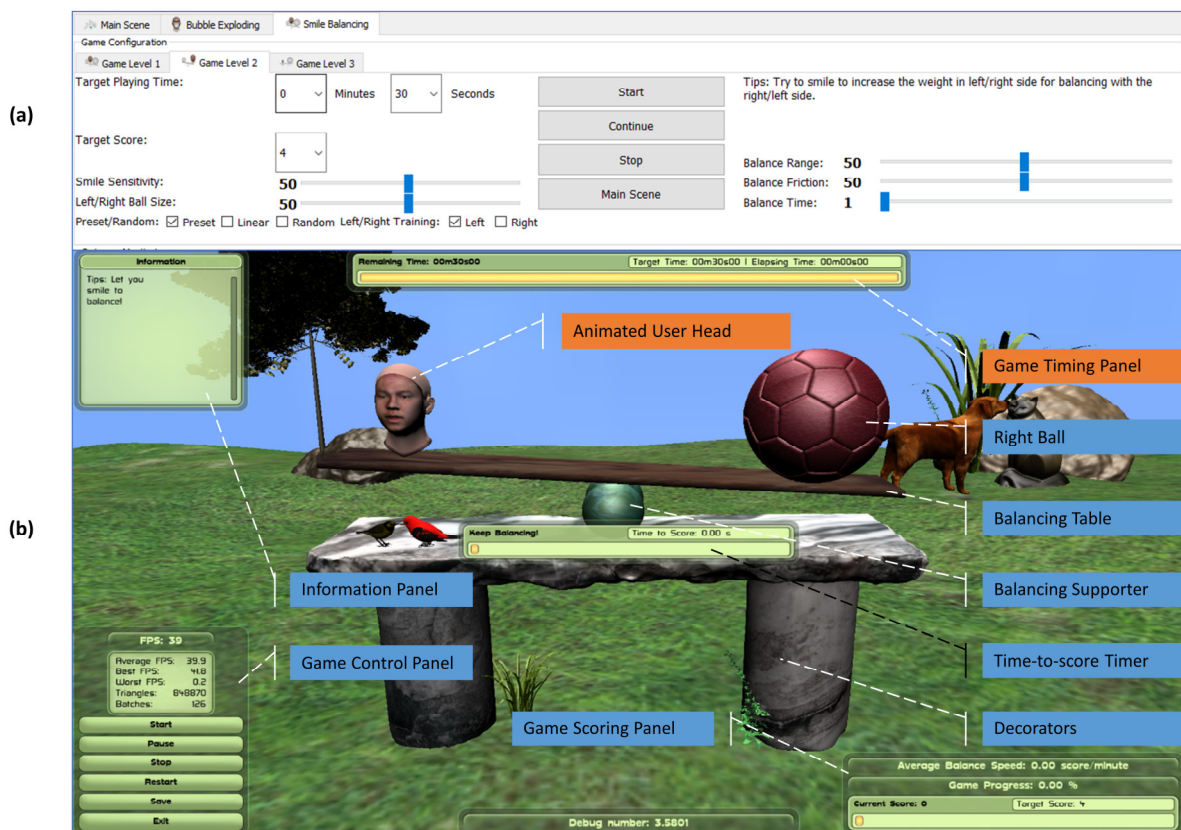


Figure 77. Interface of the “smile to balance ball weights” game: (a) game parameter interface and (b) game scene interface.

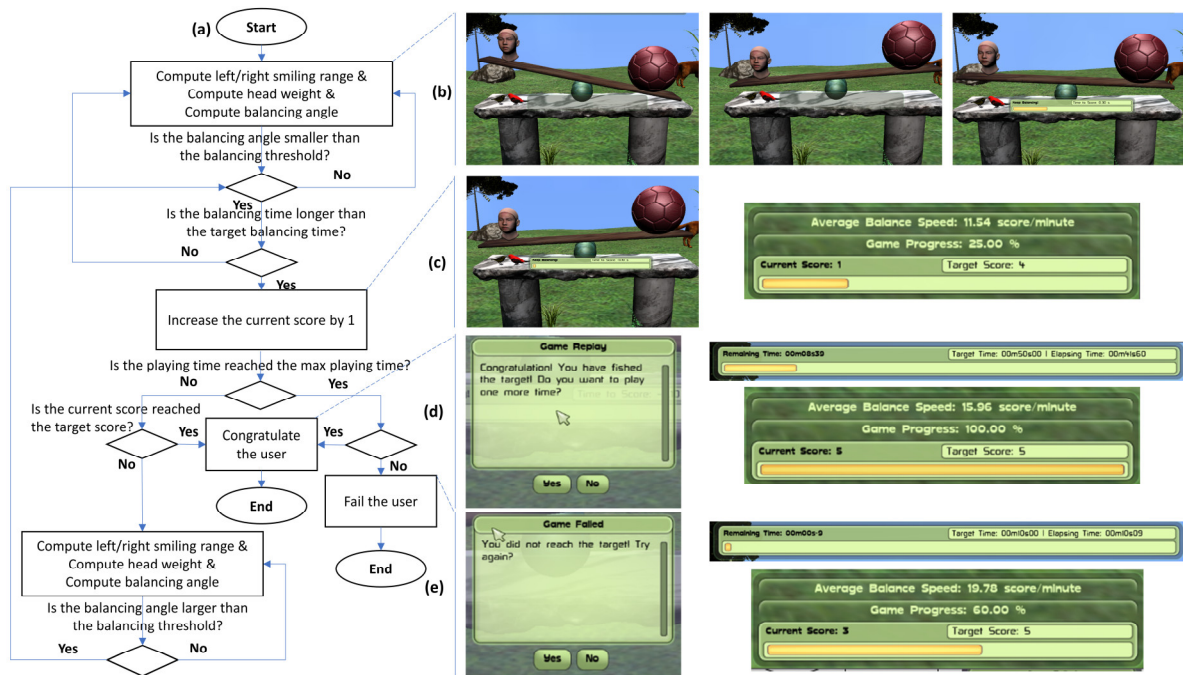


Figure 78. Game rule of the “smile to balance ball weights” game.

6.2.5.3. Game level 3: “Smile to lift balls”

In the “smile to lift balls” game, the game objects in the render window (Figure 79b) were the same as ones in the “smile to balance ball weights” game. However, in this game, the weight of the animated head was fixed with a weighted referencing ball, and it can be slid horizontally to the left/right along the balance board by making left and right smiling mimics. Based on the principle of leverage, the balance board would be balanced when distances between the animated head and the balance supporter were long enough. The game target was to move the animated head to a specific position for balancing the balance board.

The game rules are shown in Figure 80a. After starting, the left/right smiling ranges were computed to move the animated head along the balance board. While moving, the left/right ball was lifted by the animated head with the referencing ball. The player tried to move the head to a position so that the balance angles were in the *balance range* as long as *the target balance time* (Figure 80b), and he/she would be score by 1 (Figure 80c). If the total score reached the target score during the target playing time, the player would successfully finish the game (Figure 80d). Otherwise, he/she would not finish the game and be asked for later attempts (Figure 80e). Three playing modes were also designed like the “smile to balance ball weights” game. All game parameters could be tuned in the game parameter interface (Figure 79a).

As a result, by using left & right smiling mimics to move the head model for balancing the balance board during the target balance time, the player would train his/her effectiveness of controlling left & right zygomaticus minor & major muscles and managing synkinesis in these muscles.

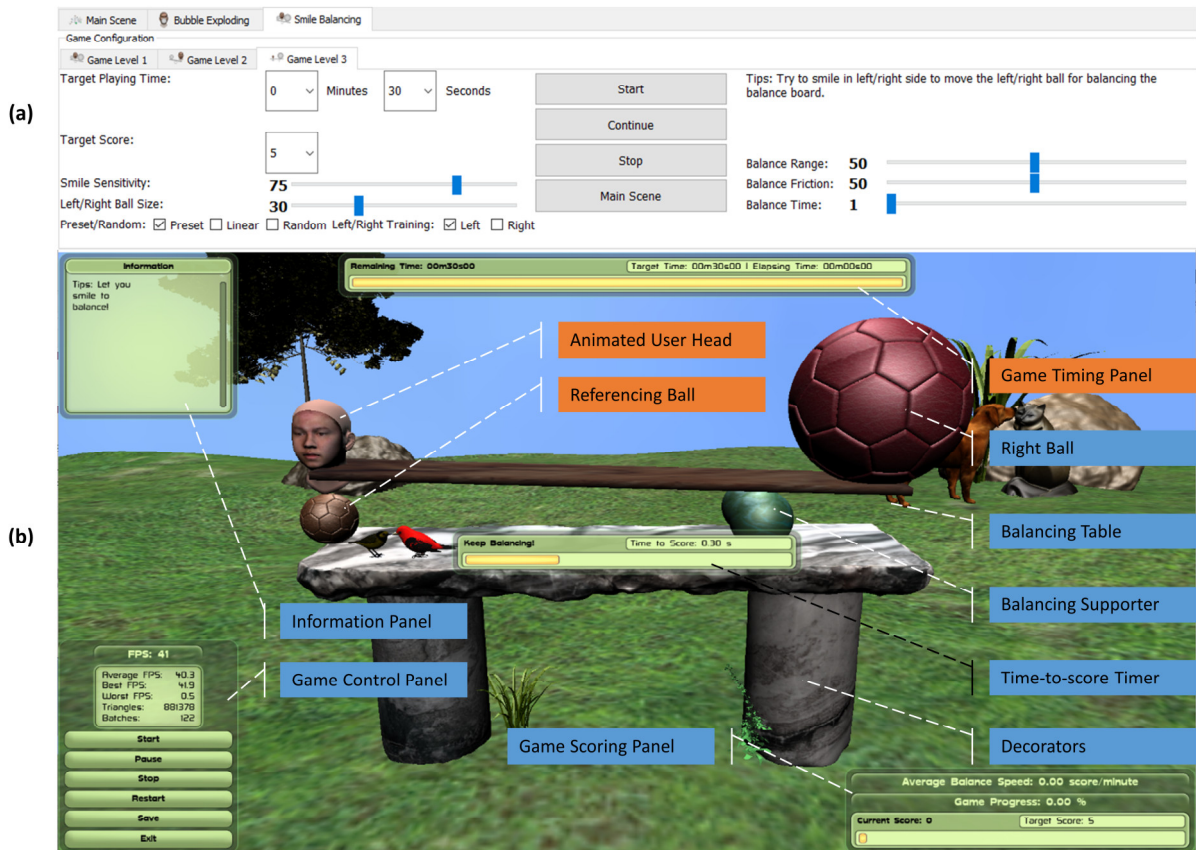


Figure 79. Interface of the “smile to left balls” game: (a) game parameter interface and (b) game scene interface.

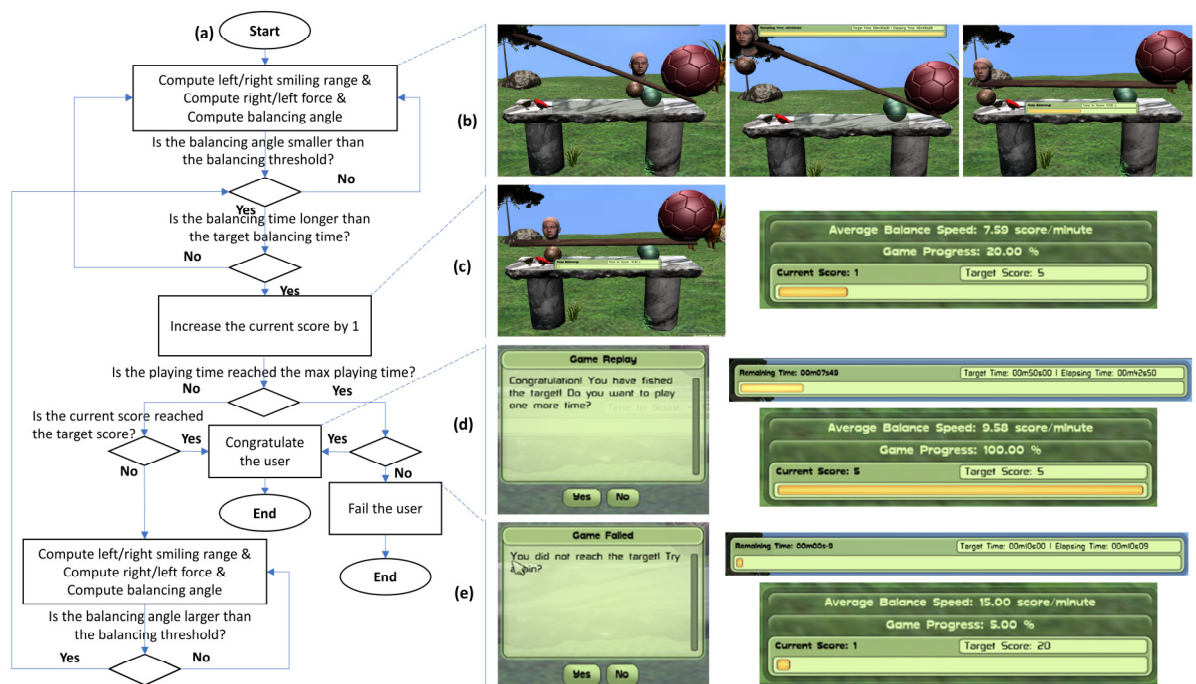


Figure 80. Game rule of the “smile to left balls” game.

6.3. Discussion and conclusions

Facial mimic rehabilitation needs a complex and long-term procedure of both physical and psychological treatments [21]. Physical rehabilitation exercises were specified for stretching and relaxation of facial muscles, symmetry of facial muscle movements, and synkinesis management & movement control [8]. These exercises needed repetitive and long-term movements for intensifying specific movements of appropriate facial muscles [8]. These

repetitive and extensive actions without motivations could reduce patients' ambition for continuing the recovering processes. Consequently, serious games specifically designed with challenged goals, scoring strategies, interesting scenes, and real-time interactions [129] could be one of the best solutions for improving motivations and effectiveness during rehabilitation exercises, especially for older adult [133] and pediatric patients [134]. Although numerous serious games have been developed for upper-limb stroke rehabilitation [93,94], trauma bones and soft-tissue injuries [95], traumatic brain injuries [96], home-based stroke rehabilitation [97], neuropsychological assessment, training and rehabilitation [131], and orthopedic trauma surgery [132], no serious game have not been developed for recovering facial mimics.

This chapter presented a novel serious game system for facial mimic rehabilitation. The system included two game groups specifically designed for training stretching & relaxation, symmetry, and synkinesis management & movement control of left & right zygomaticus minor & major muscles responsible for generating smiling and kissing facial mimics. These games also included adaptive challenge goals and difficulties that could be suitable for each patient abilities. Game results could also be personally stored and analyzed throughout different attempts of each player for progress managements (Figure 68). All serious games were designed in a luxury bath room (Figure 69, Figure 71, Figure 73) and a chilling green-grass garden (Figure 75, Figure 77, Figure 79) that could give relax and positive cognitive effects to the player. All games also contained realistic objects, such as real-time animated head models, dynamic colorful bubbles, and real-time physical balance system, integrated with sound effects that could improve motivations of the player. Especially, the game system could also support a fast patient-specific model generation procedure so that the player could use his/her model for interacting with other game objects to provide more user-friendly interfaces. Moreover, facial mimics and bio-feedbacks with multiple presentation modes could also be visualized to users in real-time for analyzing facial movements while playing games (Figure 68).

However, current game system developments also had some limitations. All serious games only designed for training smiling and kissing mimics. More serious games for other mimics should be developed. Because of hazardous situations from Covid-19 [228], system reliabilities and user acceptability of the game system have not been validated on facial palsy patients and clinical experts in short-term and long-term evaluation campaigns. In perspective, a clinical evaluation campaign for the serious game system will be conducted, and the serious game system will then be improved based on evaluation outcomes and user feedbacks.

In conclusion, this chapter, for the first time, presented a novel serious game system for facial mimic rehabilitation. Two serious game groups with three different levels for each game group were developed for training smiling and kissing mimics with realistic game scenes and user-friendly GUIs. Based on the concept of SoS, the game system could effectively cooperate with other systems to support multiple functions: subject-specific model generations, facial movement analysis, game playing data management, and serious game playing. In perspective, a clinical evaluation campaign for all games will be conducted with facial palsy patients in our clinical partner. More serious games for training other facial mimics will also be developed.

Chapter 7:

General Discussion

Facial palsy negatively affects personal and professional of human lives, so recovering of facial mimics into their normal and symmetrical movements helps improve life qualities of the involved patients. Current rehabilitation processes lacked of real-time bio-feedbacks and training motivation. This thesis was conducted to complement these drawbacks by developing a computer-aided clinical decision-support system coupled with external & internal real-time bio-feedbacks and serious games for facial mimic rehabilitation. However, developing the system is a scientific, technological, and clinical challenge. This challenge has been resolved by the completed system development tasks, as presented in Chapter 3, Chapter 4, Chapter 5, and Chapter 6. After completing all system development tasks, the thesis has six main contributions: (1) a novel real-time subject-specific head generation & animation system, (2) a novel head-to-skull prediction process, (3) a muscle-oriented patient-specific facial paralysis grading system, (4) novel serious games for facial mimic rehabilitation, (5) a novel clinical decision-support system for facial mimic rehabilitation, and (6) a reference for developing real-time soft-tissue simulation systems. Although all system development tasks have been successfully conducted, limitations were also risen from developing the main system and each of all sub-systems. This chapter will first present an overview of the thesis contents, and then discuss in details each of the six main contributions. Moreover, current system limitations will also be indicated.

7.1. Thesis overview

Facial expressions are important to personal identity, race, emotion, and health [1], so facial disorders due to stroke, accidental/sportive injures, facial post-transplantations, or sometimes without etiology, negatively affects the professional, social, and personal lives of involved patients [2]. Consequently, the recovery of facial expressions to normal and symmetrical mimics will significantly improve life and social qualities of these patients. Facial mimic rehabilitation is a vital clinical step that determines success of surgical interventions and drug therapies [14–18,21]. Rehabilitation of facial mimics includes both psychological and physical treatments that must be followed through numerous clinical steps [21]. In particular, facial paralysis degrees of the facial palsy patients have to be first evaluated using clinical and/or non-clinical facial paralysis grading systems before being taken into suitable rehabilitation programs. The rehabilitation programs include all or some of the treatment components: (1) patient education, (2) soft-tissue mobilization, (3) functional retraining, (4) facial expression retraining, and (5) synkinesis management [21]. Numerous challenges have been risen in all stages of conventional facial mimic rehabilitation. Facial paralysis grading system need to be more objective, easy-to-use, low-cost, real-time, accurate, and multifunctional [22]. Most conventional face exercises just focused on physical rehabilitation and included repetitive and nonmotivated movements in long-term periods. This can lead patients to be less motivated for continuing their rehabilitation programs. Moreover, quantitative and objective bio-feedbacks are rarely provided to patients/clinicians in real-time and patient-specific manners during face rehabilitation exercises. Additionally, current rehabilitation treatments require patients to meet their therapists directly for all above treatment components. This could be inconvenient for such a long-term treatment procedure in facial mimic rehabilitation.

To complement all above drawbacks and requirements, this thesis was conducted to answer two research questions: (1) how to help patients improve the psychological and physical effects of their facial mimic rehabilitation programs and (2) how to help clinicians supervise the rehabilitation progresses. A computer-aided clinical decision-support system for facial mimic rehabilitation was developed to answer these two research questions. The system had abilities of fast generating patient-specific biomechanical head model (comprising of a surface head

model, a skull model, and a muscle network) and real-time animating the generated models. The animated models could be used to compute facial muscle activation and other standard facial mimic parameters for analyzing facial paralysis degrees before and when conducting facial mimic rehabilitation programs. Facial rehabilitation serious games were also integrated into the system for motivating patients during their training processes. A user-friendly and multifunctional graphic user interface system was designed for instructing users so that the system can be used with/without supports of clinicians. The system also supported static & dynamic patient-specific data managements for clinicians to supervise and analyze training progresses of their patients. Based on the target system functions, a SoS-based system framework was designed (Figure 13). The main system included model computation system, serious game system, graphic user interface system, and central managing system. These systems should be both independent, connective, and optimized to accomplish all above target system functions in real-time with acceptable accuracies.

A systematic review was first conducted to investigate how real-time soft-tissue simulation systems have been developed in four system development aspects: (1) computational approaches, (2) interaction devices, (3) system frameworks, and (4) clinical validations [103]. By clearly analyzing advantages and drawbacks in each system development aspect of all related studies in literature, technological selections for developing the target system was proposed (

Table 2). In particular, the Microsoft Kinect sensor V2.0 controlled by the Kinect SDK 2.0 was used as the main input interaction devices. By using this type of sensor, no computational costs were spent for computing facial animations during tracking facial mimics. The system was developed for Microsoft Windows platform in C++ programming languages and taken advantages of multiple up-to-date programming frameworks (Qt 5.12.5 [161], VTK 7.1.1 [162], Ogre3D [151], irrKlang [163], Bullet real-time [150]). Multilevel validation procedures were also done on subsystems and the whole systems. System development tasks were accomplished using the standard IEC 62304:2006+A1:2015(E) [104].

For providing real-time external and internal bio-feedbacks, realistic patient-specific biomechanical head generations and animations were the first requirement in facial mimic analyzing systems. The models were generated and animated in the model computation system. In particular, a patient-specific head surface model coupled with face textures was generated from patient-specific data (HD facial points and face images) acquired from the Kinect V2.0 sensor. Geometrical validation results with MRI-based and point cloud-based models showed a good accuracy level (distance deviation of ~ 1 mm in neutral position and an error range of [2-3mm] for different facial mimic positions). The generated head model with face textures could be animated in real-time (up to 60 fps). Details were presented in Chapter 3. Internal structures (skull and muscle network) were also estimated from the generated head surface model based on a head-to-skull relationship and standard face anatomy structures (Chapter 5). The head-to-skull relationship was trained based on a statistical shape PLSR-based modelling method from 209 CT-reconstructed head & skull models (Chapter 4). Comparison between Kinect-generated and MRI-reconstructed skull models showed a good accuracy level (mean Hausdorff distances up to 3.12 mm). With the accuracy levels of both Kinect-generated head and skull models, lengths and strains of muscle action lines could be computed in real-time based on the Kinect-driven biomechanical head model (Chapter 5). The computed internal bio-feedbacks of facial muscles were in agreement with reported values in literature. Different facial mimic analysing metrics: mirror effects, MPEG-4 FAPs, FACS AUs, and MPEG-4 facial points could also be computed during real-time head animations.

For motivating facial palsy patients when training their facial mimics, a serious game system with two different games was specifically designed for training smiling and kissing mimics

(Chapter 6). The first game called “*exploding bubbles*” game was specified for training stretching and relaxation of facial muscles responsible for smiling and kissing mimics. The second game called “*smile balancing*” game was specified for training symmetries of facial muscle movements and synkinesis management & movement control. All games were designed with simple game rules, interesting game scenes, easy-to-use game interfaces, adaptive game difficulties, and player-specific game objects. These game components all provided more motivated and positive cognitive effects to the players. Moreover, because the serious game system was parallelly executed with other systems, users could analyze facial mimics while playing games. All games were designed with different parameters for adapting game difficulties with user abilities.

The target system must also be easy-to-use, so a user-friendly graphic user interface (GUI) system was integrated into the main system for effectively controlling system executions according to user commands (Appendix A). Using the GUI system, the user could be instructed to fast generate his/her subject-specific models (head, skull, and muscle network). The generated models could be rendered in the GUI system with different rendering modes. The user could also select his/her available models stored in the system to be animated. During real-time head animations, the GUI system could support multiple functions: serious game playing, mimic analyzing, bio-feedback graphing, system parameter editing, data capturing/recording, and user helping. Moreover, the GUI could save and manage user data statically and dynamically for each user so that patients/clinicians could reopen/replay the saved/recorded data to evaluate the previous training progresses.

In perspective, the serious game system, the GUI system, and the whole system will be validated with facial palsy patients and clinical experts in long-term & short-term user acceptability/safety and system reliability validations.

7.2. A novel real-time subject-specific head generation & animation system

Real-time simulations of subject-specific head and facial mimic have important applications in multiple disciplines: facial surgery simulation [34], internet communications [46], multimedia applications [47], human-computer interactions [50,53], and especially for facial mimic analyses. Numerous computer vision-based methods have been proposed for simulating facial animations (Table 4), but they hardly reached real-time computation speeds [31,37,41,43,51]. Although some studies was able to reach real-time framerates, no full head models were simulated [32,35–38,40–44,48,49,51–53], or the simulated head models were not subject-specific [31,45,46,50,54,55]. Especially, most studies could not integrate both subject-specific model generations and animations into a complete system, so their system hardly applied to new users. Last but not least, few studies conducted a multi-level validation process to evaluate their simulated models (Table 4). Consequently, a complete system with the abilities of subject-specific model generation and real-time model animations is required.

In Chapter 3, a real-time computer vision system for tracking simultaneously subject-specific rigid head and non-rigid facial mimic movements was presented. Using the Kinect V2.0 sensor coupled with Kinect SDK 2.0, high definition facial points could be acquired in real-time, no computation costs were needed for computing and tracking facial mimics. Consequently, the target system could be optimized to reach real-time framerates even being executed in slow hardware configurations. The developed system could also automatically generate the subject-specific head model with face texture of the target user based on his/her facial points and face images. The total model & texture generation time was 39.2 ± 0.5 s. After generated, his/her head model with face texture could be animated with framerates up to 60 fps. The system was evaluated through a multi-level validation procedure including geometrical validation in neutral and mimic positions, reproducibility & repeatability tests, and illumination effect analyses. The

validation results showed a good accuracy level for the generated head model (distance deviation of ~ 1 mm in neutral position and an error range of [2-3mm] for different facial mimic positions). The system was consistent with different light conditions and numerous times of model generations. In this thesis, the subject-specific fast head generation & real-time head animation system was predominately necessary for mirror effects, skull model generations, facial muscle computations, and serious game controlling. MPEG-4 FAPs and FACS AUs could be also computed from the real-time animated head model for multimedia applications. This real-time head animation system could also be used to generate dynamic facial mimics in time series for facial mimic analyses and facial expression classifications. Moreover, the subject-specific head generation procedure could be applied to generate full head models given only face models. This can be applicable to some datasets where only face regions were available, for example in the bosphorus database for 3D face analysis [229].

The real-time subject-specific head generation & animation system was contributed to literature in the Computer Method and Program in Biomedicine journal (Q1, IF@2018 = 3.424) [187] (<https://doi.org/10.1016/j.cmpb.2020.105410>).

7.3. A novel head-to-skull prediction process

Human skull is an important body structure for jaw movements and facial mimic simulations. In particular, the skull structure supports facial structures and protect internal brain [76]. This structure also combines with facial muscle contractions to generate facial mimics [73]. In physics-based face mimic simulations in which realistic facial animations could be generated based on physical deformations among biological structures inside head model, the skull layer was also primarily necessary for defining static constrains for the other soft-tissue models [74,77]. Moreover, the shape of mandible structure highly affects to jaw movements in facial mimics [78]. Skull models were mostly reconstructed from MRI/CT images, but the reconstruction procedures were relatively slow, even existing some powerful automatic reconstruction procedures [79,80]. Moreover, data acquisitions of MRI/CT techniques are slow, expensive, and harmful to human bodies. In the other than, external surface head model can be fast, safely, and easily reconstructed by portable sensors such as laser scanners [32,53], Kinect sensor [35], and color cameras [51], but they cannot acquire internal skull structures. Consequently, fast internal skull estimations from external surface heads based on a head-to-skull relationship are strongly recommended. Especially, most studies have only tried to estimate external face models based on available skull models in forensic facial reconstruction applications, yet the inversed issues have not been done.

In Chapter 4, a statistical shape modelling approach for predicting subject-specific human skull from head surface was presented. A complete procedure of data reconstruction, sampling, learning, hyper-parameter tuning, and cross-validation was conducted on 209 CT image datasets was conducted to estimate and validate the head-to-skull relationship. The partial least squared regression (PLSR) method showed outperformance over the other statistic shape modeling methods such as non-negative matrix factorization (NMF) [127,128] and principal component analysis (PCA) [124] in modeling this head-to-skull relationship. Based on the trained PLSR model, given a new head model, its internal skull model could be estimated. A 10-fold cross-validation with CT-reconstructed skull models showed a good accuracy level (mean Hausdorff distances of 2.09 ± 0.15 mm and 2.64 ± 0.26 mm for the best and worst predicted skull respectively). The head-to-skull prediction procedure have significant applications in physics-based facial mimic modeling where only head models are available or easy to be generated. Moreover, the PLSR head-to-skull model coefficient matrix can be inversed to the skull-to-head coefficient matrix for estimating head models from skull models in forensic applications.

This statistical shape modelling approach for predicting subject-specific human skull from head surface was published in the Medical & Biological Engineering & Computing Journal (Q2, IF@2018 = 2.022) [216] (<https://doi.org/10.1007/s11517-020-02219-4>).

7.4. A muscle-oriented patient-specific facial paralysis grading system

Patient-specific and objective facial paralysis grading is necessary for evaluating facial paralysis degrees before any rehabilitation procedures for proposing suitable training programs and during rehabilitation exercises for giving bio-feedbacks. The grading systems must also be executed in real-time with acceptable accuracy for tracking complex facial movements of facial palsy patients. However, because of lacking internal structures (skulls and muscle networks), most computer-aided grading systems just indirectly analyzed external 2-D/3-D motions from facial textures extracted from visual and/or depth data [57–68]. Although some standards were defined for evaluating facial mimics such as FACS [225] and MPEG-4 FAPs [167], they just measured external motions of facial movements. Moreover, most physics-based face modeling methods just computed facial animations from facial muscle contractions and were not real-time [4,5,72]. In fact, facial paralysis grading needs a novel evaluation method based directly on real-time facial muscle actions estimated from external facial mimics.

In chapter 5, patient-specific head, skull, and muscle network modeling using the Microsoft Kinect V2.0 sensor was presented. Taken advantages from the subject-specific head generation from Kinect HD facial points, as presented in Chapter 3, and the subject-specific skull generation from head models, as presented in Chapter 4, a muscle network was then defined as muscle action lines connecting between attachment points on the skull model and insertion points on the head model. During real-time head animations, these internal structures could be transformed according to current rigid head and non-rigid face movements, so facial muscle strains could be computed in real-time. The generated head and skull models were validated with the ground-truth models. In neutral positions, the best mean Hausdorff distance between Kinect-based and MRI-reconstructed head models is 1.91 mm, and one between Kinect-based and MRI-reconstructed skull models is 3.12 mm. In mimic positions, the best mean Hausdorff distance between Kinect-driven and point cloud-reconstructed head models is 2.02 mm for smiling mimics and 2.00 mm for [u]-pronouncing mimics. Moreover, based on analyses of muscle strains and comparisons with related studies, the Kinect-driven muscle network could acceptably describe muscle activities in smiling and [u]-pronouncing mimics for healthy and facial palsy subjects. The Kinect-driven biomechanical head model has multiple applications in muscle-oriented facial analyses. Facial paralysis degrees could be evaluated by analyzing muscle actions responsible for appropriate facial mimics. Moreover, multiple machine learning-based methods can be used to classify face emotions based on the computed muscle strains.

The complete patient-specific head, skull, and muscle network modeling was published in the Computer Method and Program in Biomedicine journal (Q1, IF@2018 = 3.632) [226]. (<https://doi.org/10.1016/j.cmpb.2020.105846>)

7.5. Novel serious games for facial mimic rehabilitation

Facial mimic rehabilitation needs both physical and psychological treatments. While basic oral competence referring to basic mouth functions needs short-term recovering procedures, facial expression retraining and synkinesis management were the most difficult and need long-term retraining processes [14,16]. Conventional facial mimic rehabilitation exercises specified for these difficulty and long-term retraining processes mostly included repetitive actions for intensifying specific movements of appropriate facial muscles [8]. This could make facial palsy patients less motivative after long-term retraining processes, especially in the case of retraining pediatric and elderly patients [21,30]. In the other hand, serious games with interesting

challenged goals, progressive scoring strategies, realistic scenes, and real-time interactions have been shown positive effects in most types of rehabilitation such as chronic, stroke, and traumatic diseases [90–92]. Although numerous serious games have been developed for upper-limb stroke rehabilitation [93,94], trauma bones and soft-tissue injuries [95], traumatic brain injuries [96], and home-based stroke rehabilitation [97], neuropsychological assessment, training and rehabilitation [131], and orthopedic trauma surgery [132], no serious games have been developed for facial mimic rehabilitation. This is because of lacking computational approaches and interaction devices that can realistically track and animate facial mimics. Nowadays, some well-calibrated 3-D sensors coupled with multifunctional SDKs such as Microsoft Kinect V2.0 and Kinect SDK 2.0 can acquire multiple types of object data in real-time. One of them is high definition facial points, which can be used to fast generate and real-time animate patient-specific biomechanical head models, as presented in Chapter 5. This real-time animated head model is especially suitable for designing serious games for facial mimic rehabilitation. Moreover, numerous game engines (Unity [144], Blender [145], Unreal [146], Ogre3D [151], etc.), sound engines (FMOD [230], OpenAL [231], BASS [232], irrKlang [233], etc.), and physics engines (Unity [144], XNA [147], Unreal [146], Blender [145], SOFA [148], PhysX [149], Bullet [150], etc.) have been recently developed for entertainment and serious game developments.

In Chapter 6, for the first time, a serious game system with six serious games classified into two game groups was presented. In the first game group, the player tried to use his/her horizontal mouth movements for up/down scaling the target bubbles to explode/release them. In the second game group, the player tried to smile as symmetrical as possible to balance a target balance system. When conducting the game rules, the player was also trained for stretching & relaxing (the first game) and synkinesis management & movement control (the second game) of facial muscles responsible for smiling and kissing mimics. These games were designed with relaxing and chilling scenes (Figure 69, Figure 71, Figure 73, Figure 75, Figure 77, and Figure 79) for providing positive cognitive effects to players. The games used subject-specific animated head model from the model computation system as one of their game objects for executing game rules, so the player could role-play their own facial mimics and appearances to complete the challenge goals. All games were designed with multiple game parameters for adapting game difficulties with player's abilities. Game play results were also stored and managed for each player or later analyzes. Moreover, the serious game system was designed to execute parallelly with the model computation system, so the player could analyze his/her facial mimics when playing games. Based on the developed serious game system framework, more serious games for training more facial mimics could be fast designed.

The serious game systems will contribute to literature as a journal article after conducting user acceptability/safety validations on facial palsy patients and clinical experts.

7.6. A novel clinical decision-support system for facial mimic rehabilitation

Clinical decision-support systems had intensive effectiveness for drug dosing, preventive care, and other aspects of medical care [234], so clinicians should implement them into their treatment routines whenever feasible and appropriate to improve treatment performance and reduce medical errors [235,236]. Computer-aided decision-support systems (CDSSs) for healthcare were highly required for delivering personalized healthcare services in both objective and automatic manners [237]. However, although facial mimic rehabilitation is essential for improving personal, social, and professional life qualities of facial palsy patients [1,2], no computer-aided decision-support systems have been developed for these conventional and long-term rehabilitation processes. Some of the most important requirements of these systems were real-time and patient-specific information of the current context [237], good data managements & presentations [237], and holistic overview of the patient [238].

Accomplishments of these requirements in facial mimic rehabilitation remain a major scientific, technological, and clinical challenge. In particular, no medical simulation system could provide realistic patient-specific bio-feedbacks of facial mimics supporting for clinical decision-making. Moreover, they also lacked of good data managements & presentations and patient motivations during facial mimic rehabilitation processes.

In this thesis, for the first time, a clinical decision-support system for facial mimic rehabilitation was presented to complement all above drawbacks. Based on the concept of SoS, the main system with multiple independent and complex systems (model computation system, serious game system, graphic user interface system, and central managing system) was optimized to have multiple functions with acceptable accuracy in real-time. In model computation system, taken advantages of the Microsoft Kinect V2.0 sensor with Kinect SDK 2.0, subject-specific head, skull, and muscle network could be fast generated and animated in real-time with acceptable accuracy, as presented in Chapter 3, Chapter 4, and Chapter 5. During real-time animations, multiple types of bio-feedbacks (facial muscle strains and standard facial mimic data, such as MPEG-4 FAPs, FACS AUs, mirror effects, etc.) could be provided to users in real-time. These bio-feedback data could be used to evaluate facial paralysis degrees before proposing facial mimic treatment programs and analyze facial movements during rehabilitation exercises. Serious games were also integrated in the serious game system for motivating facial palsy patients in their treatment processes, as presented in Chapter 6. Two serious game groups with three difficult levels of each group were designed with interesting & relaxing game scenes, adaptive & challenging game rules, patient-specific & interactive game objects, and personalized scoring managements for training smiling and kissing mimics. In the system, users could also both play serious games and analyze facial mimics simultaneously in real-time. Additionally, the main system was also integrated with a user-friendly, interactive, and multifunctional graphic user interface system. Particularly, the user could be interactively instructed by the GUI system to capture subject-specific data for generating his/her biomechanical head model. During real-time animations, the user could record/capture facial mimics, visualize facial mimics & bio-feedbacks, edit system parameters, and play serious games by navigating on appropriate user interfaces. Moreover, all subject-specific models, recorded/captured facial mimics, and game playing results were stored and managed for each user. The user could open/replay the stored static/dynamic mimic data for analyzing facial mimics throughout different times of usage. Last but not least, the GUI system also supported an interactive user help interface supporting for facial mimic & muscle educations during facial mimic rehabilitation treatments.

After conducting system reliability and user acceptability/safety in both short-term and long-term validation procedures on facial palsy patients and clinical experts, the whole system will be contributed to literature as a journal paper.

7.7. A reference for developing real-time soft-tissue simulation systems

Developing realistic simulation systems for soft-tissue deformations is a scientific, technological, and clinical challenge. To accomplish both real-time framerates and acceptable accuracy when simulating soft-tissues, the target systems must be optimal and compromised when selecting computational approaches, interaction devices, and system architectures. Multilevel validation procedures must also be conducted on the developed systems so that it can be employed in clinical applications. Currently, no studies have been analyzed how real-time framerates and acceptable accuracy could be achieved in complete simulation systems through all system development aspects. Some studies just described realistic soft-tissue modeling in medical simulations, proposed challenges risen from realistic soft-tissue simulations, or concerned about accuracies of soft-tissue simulation models, but solutions of these challenges to optimize both framerates and accuracies were not analyzed [121,239,240].

Up to now, with the abundant developments of software/hardware technologies and soft-tissue modeling methods, numerous studies have reasonably proposed as effective solutions for both achieving real-time computation speeds and acceptable system accuracy in soft-tissue simulation systems. They need to be summarized and analyzed to estimate general trends and weakness for future developments.

In Appendix C, a systematic review was conducted on all studies related to real-time medical simulations with soft-tissue deformation in four system development aspects: (1) computational approaches, (2) interaction devices, (3) system architectures, and (4) clinical validations. With careful selection of search terms, searching protocol, and quality assessments, high-quality articles most related to soft-tissue simulations were analyzed and summarized. Computational approaches were grouped into model development and model implementation approaches. In model development approaches, soft-tissue modeling methods were classified into mesh-based, meshfree-based, and combination-based modelling methods. All selected studies organized based on this classification were listed on Table 12, Table 13, and Table 14. All input/output interaction devices used in previous studies were also described and discussed about their abilities and contributions to the whole system performance. Moreover, system architectures including distributed and multithreaded also had their own advantages and disadvantages in optimizing system computation speeds and accuracies. In this review, most popular system programming frameworks for reducing time-to-market of the target systems were also presented. Multilevel validation procedures: geometrical validations, model behavior validations, and user acceptability/safety validations were also proposed to be one of system development tasks. Based on all above classifications and descriptions, system developers with their requirements of system framerates and accuracies will choose their suitable computational approaches, interaction devices, system architectures, and validation methods for their target systems. Last but not least, important trends, limitations, and future recommendations resultant from this systematic analysis were also proposed.

The content of this review was contributed to literature in the Applied Bionics and Biomechanics journal (Q3, IF@2018=1.525) [103] (<https://doi.org/10.1155/2020/5039329>).

7.8. Current limitations

In this thesis, limitations relating to drawbacks of the main system and other subsystems should also be indicated.

In the main system, the developed system functions just focused on generating and tracking subject-specific facial bio-feedbacks, visualizing & graphing bio-feedbacks on GUI interfaces, managing user data, and motivating user training processes. Clinicians must analyze these analyzed data using their own experts to diagnose facial paralysis, propose rehabilitation exercises, and evaluate training progresses. Actually, a fully active computer-aided decision-support system should also include pertinent knowledge and appropriate problem-solving skills for semi-automatically/automatically sending treatment recommendations to physicians [241]. Moreover, medical software systems should be validated in a multi-level validation procedure including geometrical validations, model behavior validations, and user acceptability/safety validations [103] to be able to be applied in clinical routines. However, because the current hazardous situations from the Covid-19, the developed system was just evaluated in geometrical and model behavior validations.

In the model computation system, *regarding to the subject-specific head generation procedure* (Chapter 3), subject-specific information such as hairs, ears, teeth, tongues, and irises of eyes lacked. Moreover, back-head regions were deformed from the generic head model based on the optimal affine transform representing relationship between the generic facial points and

the subject-specific facial points. Although the boundary vertices between back-head regions were optimally fitted with ones of the facial regions, the overall shapes of the back-head regions might not be subject-specific. This could be challenged because hairs always exist during data acquisition processes. *Regarding to the head-to-skull training and regressing processes*, one of the limitations of the head-to-skull relationship training process was the used of CT image database. In fact, medical imaging data was commonly acquired in supine position leading to shape artefacts due to gravity effect [215]. Moreover, the current head-skull datasets lacks of datasets from facial palsy patients. *Regarding to the muscle network definition process*, current muscle network was defined as muscle action lines connecting between attachment points on skull surfaces and insertion points on skin surfaces, so only muscle strains were computed during real-time head animations.

In the serious game system, currently, two serious game groups have been specifically designed for training smiling and kissing mimics. Due to the hazardous conditions from Covid-19, these serious games have not been played and evaluated by facial palsy patients in clinical environments. Moreover, developed serious games only specified for training smiling and kissing mimics. Because of the limitation of Kinect V2.0 and Kinect SDK 2.0, the developed serious games only support one player.

In the graphic user interface system, user acceptability validations have not been conducted for the developed user interfaces. Moreover, currently, all user subject-specific models and game playing results are stored offline in a local computer.

Overall, the thesis has been conducted to answer the two research questions: (1) how to help facial palsy patients improve the psychological and physical effects of their facial mimic rehabilitation programs and (2) how to help clinicians supervise the rehabilitation progresses. A computer-aided clinical decision-support system with real-time bio-feedbacks and serious games for facial mimic rehabilitation was successfully developed for solving these questions. In perspective, more system development tasks should be conducted for solving the current system limitations, and more functions should also be added into the current system to improve system qualities and multifunctionalities. Moreover, the thesis also opens new research areas relating to the automatic generation of patient specific head from visual sensor and internal structures using statistical shape modeling and real-time modeling and simulation for facial mimic rehabilitation.

8.1. Conclusions

In this thesis, I designed a clinical decision support system for facial mimic rehabilitation. This system was developed to answer two research questions: (1) how to help facial palsy patients improve the psychological and physical effects of their facial mimic rehabilitation programs and (2) how to help clinicians supervise the rehabilitation progresses. To answer the two research questions, a computer-aided clinical decision-support system coupled with real-time bio-feedbacks and serious games for facial mimic rehabilitation has been developed.

The system was designed with multiple functions to support both facial palsy patients in rehabilitation processes and clinicians in progress diagnosing. In particular, the system could automatically fast generate patient-specific models (head, skull, and muscle network) and animated the generated models in real-time using a Microsoft Kinect V2.0 sensor, and bio-feedbacks (facial muscle strains and standard mimic parameters) could be computed during the real-time animations. Moreover, the system also supported serious games with realistic game scenes, challenging game rules, and user-specific game objects specifically designed for motivating facial palsy patients during training smiling and kissing mimics. The system framework was designed based on the concept of system of systems for managing complexity and multifunctionality when executing all above system functions in real-time with acceptable accuracy. Additionally, user-friendly and multifunctional graphic user interfaces were designed and developed for effective interaction between user requirements and system functions.

The main contributions of the thesis relate to a systematic reference for developing real-time soft-tissue simulation systems, a novel real-time subject-specific head generation & animation system, a novel head-to-skull prediction process, a muscle-based patient-specific facial paralysis grading system, novel serious games for facial mimic rehabilitation, and a novel computer-aided clinical decision-support system for facial mimic rehabilitation.

The system also presents some limitations relating to developments of the main system and each of all sub-systems. The serious games and the whole system have not been validated with user acceptability/safety validations on facial palsy patients and clinical experts due to the difficulties of global health conditions. Moreover, some personal details such as hairs, eye lids, ears, teeth, and tongues have not been included in the model generation processes.

8.2. Perspectives

In perspectives, more system development tasks should be conducted for solving the current system limitations.

In the main system, a pertinent knowledge system should be developed and trained using the computed real-time & patient-specific bio-feedbacks and standard facial muscle behaviors based on machine learning approaches. A decision-making system should also be designed for automatically diagnosing facial paralysis before taking rehabilitation programs. This system should also support recommendations of suitable serious games with appropriate game parameters based on the current diagnosed results. The user acceptability/safety validations on facial palsy patients and clinical experts for the whole system will also be conducted in both short-term and long-term validation procedures.

In the model computation system, *regarding to the subject-specific head generation procedure*, relationship between facial regions and back-head regions should be more researched. Statistical shape modeling and geometrical deep learning approaches are two potential recommendations for resolving this issue. The more subject-specific head models in both facial and back-head regions can be generated, the more accurate skull models and muscle networks will be estimated. *Regarding to the head-to-skull training and regressing processes*, a postural transformation should be investigated in the future to correct the shape before using the reconstructed head/skull shapes for learning purpose. Moreover, a new database that focused on more various age & nationality groups and larger number of datasets will also improve the accuracy of statistical regression capacity for the head-to-skull prediction problem. *Regarding the muscle network definition process*, some novel real-time soft-tissue modeling methods such as Mass-Spring system with corrective modelling method (MSS-CS) will be applied to model a new muscle network. Based on the computed muscle strains, muscle forces can also be computed from the new muscle network. Standard muscle strain and force of each defined muscle according to different facial action units in FACS [225] will be computed in standard facial expression datasets [229]. From the real-time muscle strains and/or muscle forces, machine learning approaches will be employed to automatically evaluate facial paralysis degrees in comparison with these standard values. A standard FACS scoring sheet will also be automatically exported from the system based on these machine learning-based evaluations.

In the serious game system, user acceptability/safety validation procedures will be conducted on both facial palsy patients and clinical experts. Validation results will be used to improve all game components (game scenes, game rules, scoring strategies, game objects, and game interfaces). More serious games for training more types of facial mimics will be developed. Moreover, an Internet-based multiplayer framework will be developed so that multiple players can cooperate or contest together in multiplayer serious games. Moreover, game playing results and bio-feedbacks can be analyzed by cloud computing systems based on this Internet communication framework.

In the graphic user interface system, these interfaces will be evaluated by facial palsy patients and clinicians, and they will be edited based on their comments and recommendations. A user feedback interface will also be integrated into the GUI system so that users can fast give their feedbacks when using the system. These feedbacks will then be automatically sent to system developers for improving system user acceptability and reliability. Moreover, an online individualized healthcare management system will be developed so that each patient can create/login into his/her own account for managing all personal data.

Moreover, more system functions should also be developed to improve system qualities and functionalities.

First, a muscle-based facial mimic generation system should be developed. In this system, the user can use a PC mouse or his/her facial mimics to generate facial mimics on his/her own head model or the other head model. During controlling facial mimics, muscle strains and/or forces can also be feedbacked to the user for evaluating effects of current facial mimics. This function will be convenient for clinicians to design rehabilitation exercises and analyzed effects of the proposed exercises.

Second, to improve the accuracies of the subject-specific head, skull, and muscle network generation procedures, a novel skull-to-head generation process will be introduced. In this process, a skull model reconstructed from CT images will be provided from clinicians. Then, a subject-specific head model will be estimated from the provided skull based on the inversed skull-to-head relationship, as presented in Chapter 4. The estimated head model will be fused with the Kinect-based head model, whose generation process was presented in Chapter 3, to improve the subject-specific levels of the final generated head model, so as to the generated skull model and muscle network.

Third, the model computation system mainly relied on Kinect SDK 2.0 for tracking high-definition facial points, and this SDK only supports for the Microsoft Kinect V2.0 sensor. In perspectives, a novel SDK that has the input of depth and/or color images from other visual sensors such as stereo cameras, single cameras, or laser scanners & single cameras, will be developed for tracking high-definition facial points in real-time. This SDK will help the system be more flexible when being able to using different types of input interaction devices rather than the Microsoft Kinect V2.0 sensor.

8.3. Potential research recommendations

Based on trends and drawbacks from the systematical review about real-time soft-tissue simulation systems and challenges arisen when developing the target system, novel research recommendations are also discovered.

Hybrid modelling methods for real-time soft-tissue simulations. Realistic simulation soft-tissue is a complex engineering and scientific challenge. Numerous computational approaches have been developed to simulate soft-tissue in real-time with acceptable accuracies. The trends of the current computational approaches relate to: 1) mathematical formulation of physical laws applicable on image-based soft tissue geometries; 2) real-time simulation achievement of soft-tissue deformation with simple constitutive laws; 3) model implementations on specific hardware configuration to speed up the computational time. However, soft-tissue behavior is commonly anisotropic, viscoelastic, inhomogeneous, nearly incompressible with large deformation. In fact, the consideration of all physical aspects is practically difficult, particularly for a real-time simulation system. Thus, modeling assumptions related to constitutive laws, geometrical discretization, boundary and loading conditions were commonly performed for a specific application. Further studies need to be investigated to develop more accurate computational approaches for simulating complex soft-tissue behaviors in real-time conditions. The hybrid modelling approach in combining several methods is a potential solution leading to maximize the advantage of each method and overcome the limitations of the other. To support implementations of hybrid modelling approaches into one target soft-tissue, multifunctional system frameworks such as SOFA [158,242–244] and CHAI3D [245] frameworks are recommended. In these frameworks, multiple modelling methods could be easily cooperated to simultaneously simulate multiple behaviors of a soft-tissue. Moreover, they can support

input/output interaction device drivers, geometrical model libraries, modelling algorithms libraries, and graphic rendering libraries for designing complete real-time soft-tissue simulation systems.

Real-time internal bio-feedbacks estimation based on external evidences. Numerous visual sensors have been developed for fast acquiring multiple types of external subject-specific data, such as single cameras, stereo cameras, laser scanner, Microsoft Kinect sensors, and Asus XTION sensors. However, they cannot measure internal structures. Internal geometries can be accurately acquired by CT or MRI scanners, but the reconstruction procedures including labeling, smoothing, and voxelization cost so much processing time. Moreover, current sensors are difficult to acquire deep information on the soft tissues, which are crucial for *in vivo* modeling and simulation. In particular, there have been no sensors having ability of acquiring these data in real-time, so there is a need for a new type of sensor that can get the internal structures and/or textures in real-time. In fact, complex data processing schemes need to be investigated in the future to study the external-internal relationship of the soft tissues leading to a predictive solution of internal structures from external information. Statistical shape modeling (SSM) or artificial intelligence (AI)-based approaches are potential methods for such complex objective.

Real-time muscle-based face analyses based on machine learning methods. Currently, most facial analysis methods have been mainly relied on 2-D/3-D motions from facial features extracted from visual and/or depth data [57–68]. Moreover, some facial mimic analyzing standards such as MPEG-4 [167] and FACS [118] tried to parameterize facial mimics into basics facial animation parameters (FAPs) or facial action units (AUs), but they were still external motions. In fact, facial mimics are generated by deformation results of facial muscle contractions on facial skins [3–5]. However, due to lacks of subject-specific facial muscle actions when using common visual sensors, no studies have been tried to directly process facial muscle actions for analyzing facial mimics. In this thesis, internal muscle actions could be estimated in real-time using the Kinect V2.0 sensor, as presented in Chapter 5. This will be a framework for muscle-based facial mimic analyzing applications. Machine learning approaches are recommended for analyzing these muscle actions to detect facial paralysis degrees, classify facial emotions, recognize speeches, etc. Some standard 3-D facial mimics databases such as bosphorus database for 3D face analysis [229] can be used to train these machine learning-based models.

A serious game framework for facial mimic rehabilitation. Facial mimic rehabilitation needs both psychological and physical treatments in long-term rehabilitation procedures [14,16]. Serious games have been shown in most clinical rehabilitation procedures such as chronic, stroke, and traumatic diseases [90–92], but no serious games have been developed for facial mimic rehabilitation. Moreover, numerous game development frameworks have been published (Unity [144], Blender [145], Unreal [146], Ogre3D [151], etc.), but they just designed for general entertainment applications. In this thesis, only two novel serious games for facial mimic rehabilitation were designed for training smiling and kissing mimics. Development processes of these games were relatively difficult with various experts in graphical designs, programming, and clinical rehabilitation exercises. Actually, these serious games should be more adaptive with different types of facial mimic treatments and interests of players. To resolve this issue, a serious game framework for facial mimic rehabilitation should be developed. In this framework, different game scenes, game rules, game scoring strategies, and game objects will be designed so that clinicians can easily select suitable game components for their treatment programs. Moreover, based on the estimated facial paralysis of the current

patient, the serious game framework will be able to automatically propose some serious games suitable with this patient.

Publications

Journal articles (N = 4):

[ACL04] TN Nguyen, S. Dakpe, MC Ho Ba Tho, TT Dao. *Kinect-driven Patient-specific Head, Skull, and Muscle Network Modelling for Facial Palsy Patients*. *Comput. Methods Programs Biomed.* (Q1, IF@2019 = 3.424). DOI: 10.1016/j.cmpb.2020.105846.

[ACL03] TN Nguyen, VD Tran, HQ Nguyen, TT Dao. *A Statistical Approach for Predicting Subject-specific Human Skulls from Head Surfaces*. *Medical & Biological Engineering & Computing* (Q2, IF@2019 = 2.022). DOI: 10.1007/s11517-020-02219-4.

[ACL02] TN Nguyen, S Dakpé, MC Ho Ba Tho, TT Dao, *Real-time Computer Vision System for Tracking Simultaneously Subject-specific Rigid Head and Non-rigid Facial Mimic Movements using a Contactless Sensor and System of Systems approach*, *Comput. Methods Programs Biomed* (Q1, IF@2019 = 3.424). (2020) 105410. DOI: 10.1016/j.cmpb.2020.105410.

[ACL01] TN Nguyen, MC Ho Ba Tho, TT Dao, *A Systematic Review of Real-Time Medical Simulations with Soft-Tissue Deformation: Computational Approaches, Interaction Devices, System Architectures, and Clinical Validations*, *Appl. Bionics Biomech* (Q3, IF@2019=1.141). 2020 (2020) 1–30. DOI: 10.1155/2020/5039329.

Conference papers (N = 3):

[ACTI03] TN Nguyen, VD Tran, HQ Nguyen, TT Dao. *Virtual Human Skull Model: Prediction from Head Surface*. *Virtual Physiological Human (VPH2020)*, August 24–28, 2020, Paris, France (Poster)

[ACTI02] TN Nguyen, S Dakpé, MC Ho Ba Tho, TT Dao. *Real-time Subject-specific Head and Facial Mimic Animation System using a Contactless Kinect Sensor and System of Systems Approach*. *IEEE Engineering in Medicine and Biology Society (EMBC)*, July 23–27, 2019, Berlin, Germany (Oral presentation)

[ACTI01] TN Nguyen, S Dakpé, MC Ho Ba Tho, TT Dao. *Subject-specific head model development from a portable contactless sensor for facial mimic analysis and modeling*. *25th Congress of the European Society of Biomechanics (ESB)*, 7 – 10 July 2019, Vienna, Austria. (Oral presentation)

References

- [1] C. Frith, Role of facial expressions in social interactions, *Philos. Trans. R. Soc. B Biol. Sci.* 364 (2009) 3453–3458. doi:10.1098/rstb.2009.0142.
- [2] L.E. Ishii, J.C. Nellis, K.D. Boahene, P. Byrne, M. Ishii, The Importance and Psychology of Facial Expression, *Otolaryngol. Clin. North Am.* 51 (2018) 1011–1017. doi:10.1016/j.otc.2018.07.001.
- [3] T. Wu, A.P.L. Hung, P. Hunter, K. Mithraratne, Modelling facial expressions: A framework for simulating nonlinear soft tissue deformations using embedded 3D muscles, *Finite Elem. Anal. Des.* 76 (2013) 63–70. doi:10.1016/j.finel.2013.08.002.
- [4] A.X. Fan, S. Dakpé, T.T. Dao, P. Pouletaut, M. Rachik, M.C. Ho Ba Tho, MRI-based finite element modeling of facial mimics: a case study on the paired zygomaticus major muscles, *Comput. Methods Biomech. Biomed. Engin.* 20 (2017) 919–928. doi:10.1080/10255842.2017.1305363.
- [5] T.T. Dao, A.X. Fan, S. Dakpé, P. Pouletaut, M. Rachik, M.C. Ho Ba Tho, Image-based skeletal muscle coordination: case study on a subject specific facial mimic simulation, *J. Mech. Med. Biol.* 18 (2018) 1–15. doi:10.1142/S0219519418500203.
- [6] C. Rittey, The facial nerve, *Pediatr. ENT.* 83 (2007) 479–484. doi:10.1007/978-3-540-33039-4_47.
- [7] M. Constantinides, S.K.D. Galli, P.J. Miller, Complications of static facial suspensions with expanded polytetrafluoroethylene (ePTFE), *Laryngoscope.* 111 (2001) 2114–2121. doi:10.1097/00005537-200112000-00006.
- [8] M. Wernick Robinson, J. Baiungo, M. Hohman, T. Hadlock, Facial rehabilitation, *Oper. Tech. Otolaryngol. - Head Neck Surg.* 23 (2012) 288–296. doi:10.1016/j.otot.2012.10.002.
- [9] M. Lorch, S.J. Teach, Facial nerve palsy: etiology and approach to diagnosis and treatment, *Pediatr. Emerg. Care.* 26 (2010) 763–769.
- [10] M. Regina Garanhan, J. Cardoso Rosa, A. de Mello Guides Capelli, M. Claudia Ribeiro, Physical therapy in peripheral facial paralysis: retrospective study, *Braz. J. Otorhinolaryngol.* 73 (2007) 106–109. doi:10.1016/s1808-8694(15)31131-9.
- [11] E. Peitersen, Natural history of Bell's palsy, *Acta Otolaryngol.* 112 (1992) 122–124.
- [12] M. Morgan, D. Nathwani, Facial palsy and infection: the unfolding story, *Clin. Infect. Dis.* 14 (1992) 263–271.
- [13] D. Jayatilake, T. Isezaki, Y. Teramoto, K. Eguchi, K. Suzuki, Robot assisted physiotherapy to support rehabilitation of facial paralysis, *IEEE Trans. Neural Syst. Rehabil. Eng.* 22 (2014) 644–653. doi:10.1109/TNSRE.2013.2279169.
- [14] J.M. Dubernard, B. Lengelé, E. Morelon, S. Testelin, L. Badet, C. Moure, J.L. Beziat, S. Dakpé, J. Kanitakis, C. D'Hauthuille, A. El Jaafari, P. Petruzzo, N. Lefrancois, F. Taha, A. Sirigu, G. Di Marco, E. Carmi, D. Bachmann, S. Cremades, P. Giraux, G. Burloux, O. Hequet, N. Parquet, C. Francès, M. Michallet, X. Martin, B. Devauchelle, Outcomes 18 months after the first human partial face transplantation, *N. Engl. J. Med.* 357 (2007) 2451–2460. doi:10.1056/NEJMoa072828.
- [15] H. Anderl, Reconstruction of the face through cross-face-nerve transplantation in facial paralysis, *Chir. Plast.* 2 (1973) 17–45. doi:10.1007/BF00280913.
- [16] S. Khalifian, P.S. Brazio, R. Mohan, C. Shaffer, G. Brandacher, R.N. Barth, E.D. Rodriguez, Facial transplantation: The first 9 years, *Lancet.* 384 (2014) 2153–2163. doi:10.1016/S0140-6736(13)62632-X.
- [17] J. Lopez, E.D. Rodriguez, A.H. Dorafshar, Facial Transplantation, Elsevier Inc., 2019. doi:10.1016/B978-0-323-49755-8.00051-7.
- [18] M.Z. Siemionow, F. Papay, R. Djohan, S. Bernard, C.R. Gordon, D. Alam, M. Hendrickson, R. Lohman, B. Eghtesad, J. Fung, First U.S. near-total human face transplantation: A paradigm shift for massive complex injuries, *Plast. Reconstr. Surg.* 125 (2010) 111–122. doi:10.1097/PRS.0b013e3181c15c4c.
- [19] Laurent Lantieri, Man with 3 faces: Frenchman gets 2nd face transplant, AP NEWS. (2018). <https://apnews.com/97155af29cc243b08df41f891a7cefbd/Man-with-3-faces:-Frenchman-gets-2nd-face-transplant>.
- [20] M. Sosin, E.D. Rodriguez, The Face Transplantation Update: 2016, *Plast. Reconstr. Surg.* 137 (2016) 1841–1850. doi:10.1097/PRS.0000000000002149.
- [21] M.W. Robinson, J. Baiungo, Facial Rehabilitation: Evaluation and Treatment Strategies for the

- Patient with Facial Palsy, *Otolaryngol. Clin. North Am.* 51 (2018) 1151–1167. doi:10.1016/j.otc.2018.07.011.
- [22] W.S.W. Samsudin, K. Sundaraj, Clinical and non-clinical initial assessment of facial nerve paralysis: A qualitative review, *Biocybern. Biomed. Eng.* 34 (2014) 71–78. doi:10.1016/j.bbe.2014.02.005.
- [23] J.G. Neely, N.G. Cherian, C.B. Dickerson, J.M. Nedzelski, Sunnybrook facial grading system: Reliability and criteria for grading, *Laryngoscope.* 120 (2010) 1038–1045. doi:10.1002/lary.20868.
- [24] C.A. Banks, P.K. Bhama, J. Park, C.R. Hadlock, T.A. Hadlock, Clinician-graded electronic facial paralysis assessment: The eFACE, *Plast. Reconstr. Surg.* 136 (2015) 223e–230e. doi:10.1097/PRS.0000000000001447.
- [25] E. Marsk, L. Hammarstedt-Nordenvall, M. Engström, L. Jonsson, M. Hulcrantz, Validation of a Swedish version of the Facial Disability Index (FDI) and the Facial Clinimetric Evaluation (FaCE) scale, *Acta Otolaryngol.* 133 (2013) 662–669. doi:10.3109/00016489.2013.766924.
- [26] J.M. VanSwearingen, J.S. Brach, The Facial Disability Index: Reliability and Validity of a Disability Assessment Instrument for Disorders of the Facial Neuromuscular System, *Phys. Ther.* 76 (1996) 1288–1298. doi:10.1093/ptj/76.12.1288.
- [27] G. Hunter, Specific soft tissue mobilization in the management of soft tissue dysfunction, *Man. Ther.* 3 (1998) 2–11. doi:10.1054/math.1998.0310.
- [28] E. Charters, S. Coulson, Oral competence following facial nerve paralysis: Functional and quality of life measures, *Int. J. Speech. Lang. Pathol.* 0 (2020) 1–11. doi:10.1080/17549507.2020.1728380.
- [29] P.J. Byrne, Importance of facial expression in facial nerve rehabilitation, *Curr. Opin. Otolaryngol. Head Neck Surg.* 12 (2004) 332–335. doi:10.1097/01.moo.0000134829.61048.64.
- [30] C.A. Banks, T.A. Hadlock, Pediatric facial nerve rehabilitation, *Facial Plast. Surg. Clin. North Am.* 22 (2014) 487–502. doi:10.1016/j.fsc.2014.07.006.
- [31] J. Zhang, J. Yu, J. You, D. Tao, N. Li, J. Cheng, Data-driven facial animation via semi-supervised local patch alignment, *Pattern Recognit.* 57 (2016) 1–20. doi:10.1016/j.patcog.2016.02.021.
- [32] A. Matsuoka, F. Yoshioka, S. Ozawa, J. Takebe, Development of three-dimensional facial expression models using morphing methods for fabricating facial prostheses, *J. Prosthodont. Res.* 63 (2019) 66–72. doi:10.1016/j.jpor.2018.08.003.
- [33] D. Terzopoulos, K. Waters, Physically-based facial modelling, analysis, and animation, *J. Vis. Comput. Animat.* 1 (1990) 73–80. doi:10.1002/vis.4340010208.
- [34] W.S. Lee, N. Magnenat-Thalmann, Fast head modeling for animation, *Image Vis. Comput.* 18 (2000) 355–364. doi:10.1016/S0262-8856(99)00057-8.
- [35] L. Turban, D. Girard, N. Kose, J.L. Dugelay, From Kinect video to realistic and animatable MPEG-4 face model: A complete framework, 2015 IEEE Int. Conf. Multimed. Expo Work. ICMEW 2015. (2015) 1–6. doi:10.1109/ICMEW.2015.7169783.
- [36] M. Hernandez, J. Choi, G. Medioni, Near laser-scan quality 3-D face reconstruction from a low-quality depth stream, *Image Vis. Comput.* 36 (2015) 61–69. doi:10.1016/j.imavis.2014.12.004.
- [37] L. Yin, A. Basu, S. Bernögger, A. Pinz, Synthesizing realistic facial animations using energy minimization for model-based coding, *Pattern Recognit.* 34 (2001) 2201–2213. doi:10.1016/S0031-3203(00)00139-4.
- [38] C.-W. LUO, J. YU, Z.-F. WANG, Synthesizing Performance-driven Facial Animation, *Acta Autom. Sin.* 40 (2014) 2245–2252. doi:10.1016/S1874-1029(14)60361-X.
- [39] J. Yu, Z. Wang, A Video , Text , and Speech-Driven Realistic 3-D Virtual Head for Human – Machine Interface, *IEEE Trans. Cybern.* 45 (2015) 991–1002. doi:10.1109/TCYB.2014.2341737.
- [40] H. Liang, R. Liang, M. Song, Coupled Dictionary Learning for the Detail-Enhanced Synthesis of 3-D Facial Expressions, *IEEE Trans. Cybern.* 46 (2016) 890–901. doi:10.1109/TCYB.2015.2417211.
- [41] S. Zhan, L. Chang, J. Zhao, T. Kurihara, H. Du, Y. Tang, J. Cheng, Real-time 3D face modeling based on 3D face imaging, *Neurocomputing.* 252 (2017) 42–48. doi:10.1016/j.neucom.2016.10.091.
- [42] H. Jin, X. Wang, Z. Zhong, J. Hua, Robust 3D face modeling and reconstruction from frontal and side images, *Comput. Aided Geom. Des.* 50 (2017) 1–13. doi:10.1016/j.cagd.2016.11.001.
- [43] L. Jiang, J. Zhang, B. Deng, H. Li, L. Liu, 3D Face Reconstruction with Geometry Details from a Single Image, *IEEE Trans. Image Process.* 27 (2018) 4756–4770.

- doi:10.1109/TIP.2018.2845697.
- [44] P. Dou, I.A. Kakadiaris, Multi-view 3D face reconstruction with deep recurrent neural networks, *Image Vis. Comput.* 80 (2018) 80–91. doi:10.1016/j.imavis.2018.09.004.
- [45] T. Goto, S. Kshirsagar, N. Magnenat-thalmann, Using Real-Time Facial Feature Tracking and Speech Acquisition, *IEEE Signal Process. Mag.* (2001) 17–25.
- [46] N.P. Chandrasiri, T. Naemura, M. Ishizuka, H. Harashima, I. Barakonyi, Internet communication using real-time facial expression analysis and synthesis, *IEEE Multimed.* 11 (2004) 20–29. doi:10.1109/MMUL.2004.10.
- [47] K.H. Choi, J.N. Hwang, Automatic creation of a talking head from a video sequence, *IEEE Trans. Multimed.* 7 (2005) 628–637. doi:10.1109/TMM.2005.850964.
- [48] Y. Liu, G. Xu, Personalized Multi-View Face Animation with Lifelike Textures, *Tsinghua Sci. Technol.* 12 (2007) 51–57. doi:10.1016/S1007-0214(07)70008-1.
- [49] H. Zha, P. Yuru, Transferring of speech movements from video to 3D face space, *IEEE Trans. Vis. Comput. Graph.* 13 (2007) 58–69. doi:10.1109/TVCG.2007.22.
- [50] Y. Fu, R. Li, T.S. Huang, M. Danielsen, Real-time multimodal humanavatar interaction, *IEEE Trans. Circuits Syst. Video Technol.* 18 (2008) 467–477. doi:10.1109/TCSVT.2008.918441.
- [51] S.F. Wang, S.H. Lai, Reconstructing 3D face model with associated expression deformation from a single face image via constructing a low-dimensional expression deformation manifold, *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (2011) 2115–2121. doi:10.1109/TPAMI.2011.88.
- [52] M. Song, D. Tao, X. Huang, C. Chen, J. Bu, Three-dimensional face reconstruction from a single image by a coupled RBF network, *IEEE Trans. Image Process.* 21 (2012) 2887–2897. doi:10.1109/TIP.2012.2183882.
- [53] S. Marcos, J. Gómez-García-Bermejo, E. Zalama, A realistic, virtual head for human-computer interaction, *Interact. Comput.* 22 (2010) 176–192. doi:10.1016/j.intcom.2009.12.002.
- [54] X.M. Wan, S.J. Liu, J.X. Chen, X.G. Jin, Geodesic Distance Based Realistic Facial Animation Using RBF Interpolation, *Comput. Sci. Eng.* 14 (2012) 49–55. doi:10.1145/1275808.1276419.6.
- [55] D. Li, C. Sun, F. Hu, D. Zang, L. Wang, M. Zhang, Real-time performance-driven facial animation with 3ds Max and Kinect, 2013 3rd Int. Conf. Consum. Electron. Commun. Networks, CECNet 2013 - Proc. (2013) 473–476. doi:10.1109/CECNet.2013.6703372.
- [56] S. Ouni, G. Gris, Dynamic lip animation from a limited number of control points: Towards an effective audiovisual spoken communication, *Speech Commun.* 96 (2018) 49–57. doi:10.1016/j.specom.2017.11.006.
- [57] S. Wang, H. Li, F. Qi, Y. Zhao, Objective facial paralysis grading based on P face and eigenflow, *Med. Biol. Eng. Comput.* 42 (2004) 598–603. doi:10.1007/BF02347540.
- [58] G.C. Littlewort, M.S. Bartlett, K. Lee, Automatic coding of facial expressions displayed during posed and genuine pain, *Image Vis. Comput.* 27 (2009) 1797–1803. doi:10.1016/j.imavis.2008.12.010.
- [59] P.A. Desrosiers, Y. Bennis, M. Daoudi, B. Ben Amor, P. Guerreschi, Analyzing of facial paralysis by shape analysis of 3D face sequences, *Image Vis. Comput.* 67 (2017) 67–88. doi:10.1016/j.imavis.2017.08.006.
- [60] Z. Zhang, Microsoft kinect sensor and its effect, *IEEE Multimed.* 19 (2012) 4–10. doi:10.1109/MMUL.2012.24.
- [61] J. Hamm, C.G. Kohler, R.C. Gur, R. Verma, Automated Facial Action Coding System for dynamic analysis of facial expressions in neuropsychiatric disorders, *J. Neurosci. Methods.* 200 (2011) 237–256. doi:10.1016/j.jneumeth.2011.06.023.
- [62] T. Wang, S. Zhang, J. Dong, L. Liu, H. Yu, Automatic evaluation of the degree of facial nerve paralysis, *Multimed. Tools Appl.* 75 (2016) 11893–11908. doi:10.1007/s11042-015-2696-0.
- [63] G. Storey, R. Jiang, A. Bouridane, Role for 2D image generated 3D face models in the rehabilitation of facial palsy, *Healthc. Technol. Lett.* 4 (2017) 145–148. doi:10.1049/htl.2017.0023.
- [64] N. Dagnes, F. Marcolin, E. Vezzetti, F.R. Sarhan, S. Dakpé, F. Marin, F. Nonis, K. Ben Mansour, Optimal marker set assessment for motion capture of 3D mimic facial movements, *J. Biomech.* 93 (2019) 86–93. doi:10.1016/j.jbiomech.2019.06.012.
- [65] M. Frey, C.H. John Tzou, M. Michaelidou, I. Pona, A. Hold, E. Placheta, H.B. Kitzinger, 3D Video Analysis of Facial Movements, *Facial Plast. Surg. Clin. North Am.* 19 (2011) 639–646. doi:10.1016/j.fsc.2011.07.007.
- [66] M.D. Salgado, S. Curtiss, T.T. Tollefson, Evaluating symmetry and facial motion using 3D videography, *Facial Plast. Surg. Clin. North Am.* 18 (2010) 351–356.

- doi:10.1016/j.fsc.2010.01.011.
- [67] C.A. Trotman, J. Faraway, T. Hadlock, C. Banks, N. Jowett, H.J. Jung, Facial soft-tissue mobility: Baseline dynamics of patients with unilateral facial paralysis, *Plast. Reconstr. Surg. - Glob. Open.* 6 (2018) 1–12. doi:10.1097/GOX.0000000000001955.
- [68] T. Al-Anezi, B. Khambay, M.J. Peng, E. O’Leary, X. Ju, A. Ayoub, A new method for automatic tracking of facial landmarks in 3D motion captured images (4D), *Int. J. Oral Maxillofac. Surg.* 42 (2013) 9–18. doi:10.1016/j.ijom.2012.10.035.
- [69] S.A. King, R.E. Parent, Creating Speech-Synchronized Animation, *IEEE Trans. Vis. Comput. Graph.* 11 (2005) 341–352. doi:10.1109/TVCG.2005.43.
- [70] M.D. Cordea, E.M. Petriu, A 3-D Anthropometric-Muscle-Based Active Appearance Model, *IEEE Trans. Instrum. Meas.* 55 (2006) 91–98. doi:10.1109/TIM.2005.860861.
- [71] M.D. Cordea, E.M. Petriu, D.C. Petriu, Three-dimensional head tracking and facial expression recovery using an anthropometric muscle-based active appearance model, *IEEE Trans. Instrum. Meas.* 57 (2008) 1578–1588. doi:10.1109/TIM.2008.923784.
- [72] T. Erkoç, D. Ağdoğan, M.T. Eskil, An observation based muscle model for simulation of facial expressions, *Signal Process. Image Commun.* 64 (2018) 11–20. doi:10.1016/j.image.2018.02.009.
- [73] Y. Lee, D. Terzopoulos, K. Waters, Constructing physics-based facial models of individuals, *Proc. Gr Aphics Interface’93 Conf.* (1993) 1–8. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.2662>.
- [74] K. Kähler, J. Haber, H.-P. Seidel, Geometry-based muscle modeling for facial animation, in: *Graph. Interface, 2001*: pp. 37–46.
- [75] P. Claes, D. Vandermeulen, S. De Greef, G. Willems, J.G. Clement, P. Suetens, Computerized craniofacial reconstruction: Conceptual framework and review, *Forensic Sci. Int.* 201 (2010) 138–145. doi:10.1016/j.forsciint.2010.03.008.
- [76] M. Wei, Y. Liu, H. Dong, A. El Saddik, Human Head Stiffness Rendering, *IEEE Trans. Instrum. Meas.* 66 (2017) 2083–2096. doi:10.1109/TIM.2017.2676258.
- [77] H.Y. Ping, L.N. Abdullah, P.S. Sulaiman, A.A. Halin, Computer Facial Animation: A Review, *Int. J. Comput. Theory Eng.* 5 (2013) 658–662. doi:10.7763/ijcte.2013.v5.770.
- [78] K. Waters, D. Terzopoulos, A physical model of facial tissue and muscle articulation, in: [1990] *Proc. First Conf. Vis. Biomed. Comput.*, 1990: pp. 77–82. doi:10.1109/VBC.1990.109305.
- [79] J.D. Nielsen, K.H. Madsen, O. Puonti, H.R. Siebner, C. Bauer, C.G. Madsen, G.B. Saturnino, A. Thielscher, Automatic skull segmentation from MR images for realistic volume conductor models of the head: Assessment of the state-of-the-art, *Neuroimage.* 174 (2018) 587–598. doi:10.1016/j.neuroimage.2018.03.001.
- [80] J. Minnema, M. van Eijnatten, W. Kouw, F. Diblen, A. Mendrik, J. Wolff, CT image segmentation of bone for medical additive manufacturing using a convolutional neural network, *Comput. Biol. Med.* 103 (2018) 130–139. doi:10.1016/j.compbimed.2018.10.012.
- [81] P. Claes, D. Vandermeulen, R. Suetens, G. Willems, S. De Greef, Volumetric deformable face models for cranio-facial reconstruction, *ISPA 2005. Proc. 4th Int. Symp. Image Signal Process. Anal.* 2005. (2008) 353–358. doi:10.1109/ispa.2005.195437.
- [82] R. Liang, Y. Lin, L. Jiang, J. Bao, X. Huang, Craniofacial model reconstruction from skull data based on feature points, *Proc. - 2009 11th IEEE Int. Conf. Comput. Des. Comput. Graph. CAD/Graphics 2009.* (2009) 602–605. doi:10.1109/CADCG.2009.5246828.
- [83] Yuru Pei, Hongbin Zha, Zhongbiao Yuan, Facial feature estimation from the local structural diversity of skulls, in: *2008 19th Int. Conf. Pattern Recognit., IEEE, 2008*: pp. 1–4. doi:10.1109/ICPR.2008.4761858.
- [84] Y.F. Zhang, M.Q. Zhou, G.H. Geng, J. Feng, Face appearance reconstruction based on a regional statistical craniofacial model (RCSM), *Proc. - Int. Conf. Pattern Recognit.* (2010) 1670–1673. doi:10.1109/ICPR.2010.413.
- [85] W. Shui, M. Zhou, Q. Deng, Z. Wu, F. Duan, 3D craniofacial reconstruction using reference skull-face database, *Int. Conf. Image Vis. Comput. New Zeal.* (2010) 1–7. doi:10.1109/IVCNZ.2010.6148864.
- [86] Y. Yang, M. Zhou, K. Lu, F. Duan, Y. Li, Y. Tian, Z. Wu, Skull Identification via Correlation Measure Between Skull and Face Shape, *IEEE Trans. Inf. Forensics Secur.* 9 (2014) 1322–1332. doi:10.1109/tifs.2014.2332981.
- [87] F. Duan, D. Huang, Y. Tian, K. Lu, Z. Wu, M. Zhou, 3D face reconstruction from skull by regression modeling in shape parameter spaces, *Neurocomputing.* 151 (2015) 674–682.

- doi:10.1016/j.neucom.2014.04.089.
- [88] F. Bai, C. Zhengxin, X. Qiao, D. Qingqiong, F. Duan, Y. Tian, Face reconstruction from skull based on Least Squares Canonical Dependency Analysis, 2016 IEEE Int. Conf. Syst. Man, Cybern. SMC 2016 - Conf. Proc. (2017) 3612–3617. doi:10.1109/SMC.2016.7844794.
- [89] D. Madsen, M. Lüthi, A. Schneider, T. Vetter, Probabilistic Joint Face-Skull Modelling for Facial Reconstruction, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018: pp. 5295–5303.
- [90] K. Laver, S. George, S. Thomas, J.E. Deutsch, M. Crotty, Virtual reality for stroke rehabilitation, *Stroke*. 43 (2012) 20–21. doi:10.1161/STROKEAHA.111.642439.
- [91] G. Saposnik, M. Levin, Virtual reality in stroke rehabilitation: A meta-analysis and implications for clinicians, *Stroke*. 42 (2011) 1380–1386. doi:10.1161/STROKEAHA.110.605451.
- [92] C.M. Bleakley, D. Charles, A. Porter-Armstrong, M.D.J. McNeill, S.M. McDonough, B. McCormack, Gaming for health: A systematic review of the physical and cognitive effects of interactive computer games in older adults, *J. Appl. Gerontol.* 34 (2015) NP166–NP189. doi:10.1177/0733464812470747.
- [93] B. Sheng, Y. Zhang, W. Meng, C. Deng, S. Xie, Bilateral robots for upper-limb stroke rehabilitation: State of the art and future prospects, *Med. Eng. Phys.* 38 (2016) 587–606. doi:10.1016/j.medengphy.2016.04.004.
- [94] I. Jakob, A. Kollreider, M. Germanotta, F. Benetti, A. Cruciani, L. Padua, I. Aprile, Robotic and Sensor Technology for Upper Limb Rehabilitation, *PM R.* 10 (2018) S189–S197. doi:10.1016/j.pmrj.2018.07.011.
- [95] H.A. Meijer, M. Graafland, J.C. Goslings, M.P. Schijven, Systematic Review on the Effects of Serious Games and Wearable Technology Used in Rehabilitation of Patients With Traumatic Bone and Soft Tissue Injuries, *Arch. Phys. Med. Rehabil.* 99 (2018) 1890–1899. doi:10.1016/j.apmr.2017.10.018.
- [96] M.G. Maggio, R. De Luca, F. Molonia, B. Porcari, M. Destro, C. Casella, R. Salvati, P. Bramanti, R.S. Calabro, Cognitive rehabilitation in patients with traumatic brain injury: A narrative review on the emerging use of virtual reality, *J. Clin. Neurosci.* 61 (2019) 1–4. doi:10.1016/j.jocn.2018.12.020.
- [97] Y. Chen, K.T. Abel, J.T. Janecek, Y. Chen, K. Zheng, S.C. Cramer, Home-based technologies for stroke rehabilitation: A systematic review, *Int. J. Med. Inform.* 123 (2019) 11–22. doi:10.1016/j.ijmedinf.2018.12.001.
- [98] V. Wattanasoontorn, I. Boada, R. García, M. Sbert, Serious games for health, *Entertain. Comput.* 4 (2013) 231–247. doi:10.1016/j.entcom.2013.09.002.
- [99] Wikipedia, List of game engines, (2020). https://en.wikipedia.org/wiki/List_of_game_engines.
- [100] O. Barzilay, A. Wolf, Adaptive rehabilitation games, *J. Electromyogr. Kinesiol.* 23 (2013) 182–189. doi:10.1016/j.jelekin.2012.09.004.
- [101] S.W. Popper, S.C. Bankes, R. Callaway, D. DeLaurentis, System of systems symposium: Report on a summer conversation, *Potomac Inst. Policy Stud. Arlington, VA.* 320 (2004).
- [102] Y. Hata, S. Kobashi, H. Nakajima, Human Health Care System of Systems, *IEEE Syst. J.* 3 (2009) 231–238. doi:10.1109/JSYST.2009.2017389.
- [103] T.-N. Nguyen, M.-C. Ho Ba Tho, T.-T. Dao, A Systematic Review of Real-Time Medical Simulations with Soft-Tissue Deformation: Computational Approaches, Interaction Devices, System Architectures, and Clinical Validations, *Appl. Bionics Biomech.* 2020 (2020) 1–30. doi:10.1155/2020/5039329.
- [104] INTERNATIONAL IEC STANDARD 62304 ISBN: 978 0 580 83868 2, Medical device software - Software life-cycle processes, (2015).
- [105] Kinect, Wikipedia, Free Encycl. (2010). <https://en.wikipedia.org/wiki/Kinect>.
- [106] M. Kocón, Z. Emirsajłow, Facial expression animation overview, in: *IFAC Proc. Vol.*, 2009: pp. 312–317. doi:10.3182/20090819-3-PL-3002.00055.
- [107] M.S. Joel Brown, Stephen Sorkin, Jean-Claude Latombe, Kevin Montgomery, Algorithmic tools for real-time microsurgery simulation, 6 (2002) 289–300.
- [108] R. Min, N. Kose, J.L. Dugelay, KinectfaceDB: A kinect database for face recognition, *IEEE Trans. Syst. Man, Cybern. Syst.* 44 (2014) 1534–1548. doi:10.1109/TSMC.2014.2331215.
- [109] G. Goswami, M. Vatsa, R. Singh, RGB-D face recognition with texture and attribute features, *IEEE Trans. Inf. Forensics Secur.* 9 (2014) 1629–1640. doi:10.1109/TIFS.2014.2343913.
- [110] P. Krishnan, S. Naveen, RGB-D face recognition system verification using kinect and FRAV3D databases, in: *Procedia Comput. Sci.*, Elsevier Masson SAS, 2015: pp. 1653–1660.

- doi:10.1016/j.procs.2015.02.102.
- [111] S.R. Bodhi, S. Naveen, Face detection, registration and feature localization experiments with RGB-D face database, *Procedia Comput. Sci.* 46 (2015) 1778–1785. doi:10.1016/j.procs.2015.02.132.
- [112] Sujono, A.A.S. Gunawan, Face Expression Detection on Kinect Using Active Appearance Model and Fuzzy Logic, *Procedia Comput. Sci.* 59 (2015) 268–274. doi:10.1016/j.procs.2015.07.558.
- [113] M. Hayat, M. Bennamoun, A.A. El-Sallam, An RGB-D based image set classification for robust face recognition from Kinect data, *Neurocomputing.* 171 (2016) 889–900. doi:10.1016/j.neucom.2015.07.027.
- [114] D. Kim, B. Comandur, H. Medeiros, N.M. Elfiky, A.C. Kak, Multi-view face recognition from single RGBD models of the faces, *Comput. Vis. Image Underst.* 160 (2017) 114–132. doi:10.1016/j.cviu.2017.04.008.
- [115] N. Nourbakhsh Kaashki, R. Safabakhsh, RGB-D face recognition under various conditions via 3D constrained local model, *J. Vis. Commun. Image Represent.* 52 (2018) 66–85. doi:10.1016/j.jvcir.2018.02.003.
- [116] K. Mishima, T. Sugahara, Analysis methods for facial motion, *Jpn. Dent. Sci. Rev.* 45 (2009) 4–13. doi:10.1016/j.jdsr.2009.03.003.
- [117] B. Hontanilla, C. Aubá, Automatic three-dimensional quantitative analysis for evaluation of facial movement, *J. Plast. Reconstr. Aesthetic Surg.* 61 (2008) 18–30. doi:10.1016/j.bjps.2007.03.037.
- [118] R. Ekman, *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*, Oxford University Press, USA, 1997.
- [119] Artec3D, *Portable 3D Scanners*, (n.d.) <https://www.artec3d.com/>.
- [120] C. Jiang, J. Wu, W. Zhong, M. Wei, J. Tong, H. Yu, L. Wang, Automatic Facial Paralysis Assessment via Computational Image Analysis, *J. Healthc. Eng.* 2020 (2020). doi:10.1155/2020/2398542.
- [121] H. Delingette, Toward realistic soft-tissue modeling in medical simulation, *Proc. IEEE.* 86 (1998) 512–523. doi:10.1109/5.662876.
- [122] K. Pearson, LIII. On lines and planes of closest fit to systems of points in space, *London, Edinburgh, Dublin Philos. Mag. J. Sci.* 2 (1901) 559–572.
- [123] H. Hotelling, Analysis of a complex of statistical variables into principal components., *J. Educ. Psychol.* 24 (1933) 417.
- [124] R. Soummer, L. Pueyo, J. Larkin, Detection and characterization of exoplanets and disks using projections on Karhunen-Loève eigenimages, *Astrophys. J. Lett.* 755 (2012) L28.
- [125] S. Wold, P. Geladi, K. Esbensen, J. Öhman, Multi-way principal components-and PLS-analysis, *J. Chemom.* 1 (1987) 41–56.
- [126] P.H. Garthwaite, An interpretation of partial least squares, *J. Am. Stat. Assoc.* 89 (1994) 122–127.
- [127] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, *Nature.* 401 (1999) 788–791.
- [128] P. Paatero, U. Tapper, Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values, *Environmetrics.* 5 (1994) 111–126.
- [129] C.C. Abt, *Serious Games*, Viking Press. New York City, New York, USA, 1st Ed. (1970).
- [130] V. Wattanasoontorn, R.J.G. Hernández, M. Sbert, Serious Games for e-Health Care, in: 2014: pp. 127–146. doi:10.1007/978-981-4560-32-0_9.
- [131] F. Ferreira-Brito, M. Fialho, A. Virgolino, I. Neves, A.C. Miranda, N. Sousa-Santos, C. Caneiras, L. Carriço, A. Verdelho, O. Santos, Game-based interventions for neuropsychological assessment, training and rehabilitation: Which game-elements to use? A systematic review, *J. Biomed. Inform.* 98 (2019) 103287. doi:10.1016/j.jbi.2019.103287.
- [132] J. Negrillo-Cárdenas, J.R. Jiménez-Pérez, F.R. Feito, The role of virtual and augmented reality in orthopedic trauma surgery: From diagnosis to rehabilitation, *Comput. Methods Programs Biomed.* 191 (2020). doi:10.1016/j.cmpb.2020.105407.
- [133] N. Skjæret, A. Nawaz, T. Morat, D. Schoene, J.L. Helbostad, B. Vereijken, Exercise and rehabilitation delivered through exergames in older adults: An integrative review of technologies, safety and efficacy, *Int. J. Med. Inform.* 85 (2016) 1–16. doi:10.1016/j.ijmedinf.2015.10.008.
- [134] S. Jurdi, J. Montaner, F. Garcia-Sanjuan, J. Jaen, V. Nacher, A systematic review of game technologies for pediatric patients, *Comput. Biol. Med.* 97 (2018) 89–112.

- doi:10.1016/j.compbimed.2018.04.019.
- [135] S. Moya, D. Tost, S. Grau, Interactive graphical design of 3D serious neurorehabilitation games, *Presence Teleoperators Virtual Environ.* 21 (2012) 58–68. doi:10.1162/PRES_a_00079.
- [136] N.A. Borghese, M. Pirovano, R. Mainetti, P.L. Lanzi, IGER: An intelligent game engine for rehabilitation, *Biosyst. Biorobotics.* 1 (2013) 947–950. doi:10.1007/978-3-642-34546-3_154.
- [137] D. González-Ortega, F.J. Díaz-Pernas, M. Martínez-Zarzuela, M. Antón-Rodríguez, A Kinect-based system for cognitive rehabilitation exercises monitoring, *Comput. Methods Programs Biomed.* 113 (2014) 620–631. doi:10.1016/j.cmpb.2013.10.014.
- [138] M. Pedraza-Hueso, S. Martín-Calzón, F.J. Díaz-Pernas, M. Martínez-Zarzuela, Rehabilitation Using Kinect-based Games and Virtual Reality, *Procedia Comput. Sci.* 75 (2015) 161–168. doi:10.1016/j.procs.2015.12.233.
- [139] H. Tannous, D. Istrate, M.C. Ho Ba Tho, T.T. Dao, Feasibility study of a serious game based on Kinect system for functional rehabilitation of the lower limbs, *Eur. Res. Telemed. / La Rech. Eur. En Télémédecine.* 5 (2016) 97–104. doi:10.1016/j.eurtel.2016.05.004.
- [140] S.S. Esfahlani, B. Muresan, A. Sanaei, G. Wilson, Validity of the Kinect and Myo armband in a serious game for assessing upper limb movement, *Entertain. Comput.* 27 (2018) 150–156. doi:10.1016/j.entcom.2018.05.003.
- [141] R. Cabrera, A. Molina, I. Gómez, J. García-Heras, Kinect as an access device for people with cerebral palsy: A preliminary study, *Int. J. Hum. Comput. Stud.* 108 (2017) 62–69. doi:10.1016/j.ijhcs.2017.07.004.
- [142] V. Fernandez-Cervantes, N. Neubauer, B. Hunter, E. Stroulia, L. Liu, VirtualGym: A kinect-based system for seniors exercising at home, *Entertain. Comput.* 27 (2018) 60–72. doi:10.1016/j.entcom.2018.04.001.
- [143] M. Xu, Y. Zhai, Y. Guo, P. Lv, Y. Li, M. Wang, B. Zhou, Personalized training through Kinect-based games for physical education, *J. Vis. Commun. Image Represent.* 62 (2019) 394–401. doi:10.1016/j.jvcir.2019.05.007.
- [144] Unity Technologies, Unity3D, (2005). www.unity.com.
- [145] The Blender Foundation, Blender, (1994). www.blender.org.
- [146] Epic Games, Unreal Engine, (1998). www.unrealengine.com.
- [147] Microsoft, Microsoft XNA Game Studio 4.0, (2004). <https://www.microsoft.com/en-us/download/details.aspx?id=23714>.
- [148] Inria, SOFA Framework, (2005). www.sofa-framework.org.
- [149] NVIDIA, PhysX SDK, (2005). <https://developer.nvidia.com/physx-sdk>.
- [150] Erwin Coumans, Bullet Real-Time Physics Simulation, (2014). www.bulletphysics.org.
- [151] The OGRE Team, OGRE (Object-Oriented Graphics Rendering Engine), (2005). www.ogre3d.org.
- [152] ISTI - CNR, MeshLab, (2002). www.meshlab.net.
- [153] J. Boardman, B. Sauser, System of Systems - The meaning of of, *Proc. 2006 IEEE/SMC Int. Conf. Syst. Syst. Eng. 2006* (2006) 118–123. doi:10.1109/sysose.2006.1652284.
- [154] J.A. Lane, What is a System of Systems and Why Should I Care?, *Univ. South. Calif.* (2013) 1–14.
- [155] J. Dahmann, System of Systems Pain Points, *INCOSE Int. Symp.* 24 (2014) 108–121. doi:10.1002/j.2334-5837.2014.tb03138.x.
- [156] S. Cotin, H. Delingette, N. Ayache, A hybrid elastic model for real-time cutting, deformations, and force feedback for surgery training and simulation, *Vis. Comput.* 16 (2000) 437–452. doi:10.1007/PL00007215.
- [157] B. Zhu, L. Gu, A hybrid deformable model for real-time surgical simulation, *Comput. Med. Imaging Graph.* 36 (2012) 356–365. doi:10.1016/j.compmedimag.2012.03.001.
- [158] P. Bensoussan, C. Duriez, L. Grisoni, P. Bensoussan, SOFA - an Open Source Framework for Medical Simulation SOFA – an Open Source Framework for Medical Simulation, *MMVR 15 - Med. Meets Virtual Real.* (2007). <https://hal.inria.fr/inria-00319416>.
- [159] Microsoft, Kinect, (2010). <https://en.wikipedia.org/wiki/Kinect>.
- [160] Microsoft, Kinect for Windows SDK 2.0, (2010). www.microsoft.com.
- [161] N. Haavard, C.-E. Eirik, Qt, (1995). www.qt.io.
- [162] Kitware Inc, VTK Visualization Tool Kit, (2006). www.vtk.org.
- [163] G. Nikolaus, irrKlang high level audio engine, (2006). www.ambiera.com/irrklang/.
- [164] Bjarne Stroustrup, C++, (1985). <https://en.wikipedia.org/wiki/C%2B%2B>.
- [165] M.W. Maier, Architecting principles for systems-of-systems, *Syst. Eng.* 1 (1998) 267–284.

- doi:10.1002/(SICI)1520-6858(1998)1:4%3C267::AID-SYS3%3E3.0.CO;2-D.
- [166] N. Aspert, D. Santa-Cruz, T. Ebrahimi, Mesh: Measuring errors between surfaces using the hausdorff distance, in: Proceedings. IEEE Int. Conf. Multimed. Expo, 2002: pp. 705–708.
- [167] Generic Coding of Audio-Visual Objects: (MPEG-4 video), (1999).
- [168] A. Myronenko, X. Song, Point set registration: Coherent point drifts, *IEEE Trans. Pattern Anal. Mach. Intell.* 32 (2010) 2262–2275. doi:10.1109/TPAMI.2010.46.
- [169] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, W. Stuetzle, Multiresolution Analysis of Arbitrary Meshes, *World Dredg. Mar. Constr.* 19 (2005).
- [170] S. Schaefer, T. McPhail, J. Warren, Image deformation using moving least squares, *ACM SIGGRAPH 2006 Pap. - SIGGRAPH '06.* (2006) 533. doi:10.1145/1179352.1141920.
- [171] P.J. Burt, E.H. Adelson, A multiresolution spline with application to image mosaics, *ACM Trans. Graph.* 2 (1983) 217–236. doi:10.1145/245.247.
- [172] A.S. Householder, Unitary Triangularization of a Nonsymmetric Matrix, *J. ACM.* 5 (1958) 339–342. doi:10.1145/320941.320947.
- [173] N. Haavard, E. ChambeEng, Qt, (1995). www.qt.io.
- [174] The Slicer Community, 3D Slicer, (2000). www.slicer.org.
- [175] P.J. Besl, N.D. McKay, A Method for registration of 3-D shapes, in: P.S. Schenker (Ed.), *Sens. Fusion IV Control Paradig. Data Struct.*, 1992: pp. 586–606. doi:10.1117/12.57955.
- [176] R.B. Rusu, Z.C. Marton, N. Blodow, M. Dolha, M. Beetz, Towards 3D Point cloud based object maps for household environments, *Rob. Auton. Syst.* 56 (2008) 927–941. doi:10.1016/j.robot.2008.08.005.
- [177] M. Gopi, S. Krishnan, C.T. Silva, Surface Reconstruction based on Lower Dimensional Localized Delaunay Triangulation, *Comput. Graph. Forum.* 19 (2000) 467–478. doi:10.1111/1467-8659.00439.
- [178] J. Vollmer, R. Mencl, H. Muller, Improved Laplacian Smoothing of Noisy Surface Meshes, *Comput. Graph. Forum.* 18 (2003) 131–138. doi:10.1111/1467-8659.00334.
- [179] J. Stam, Evaluation of loop subdivision surfaces, in: *SIGGRAPH'98 CDROM Proc.*, 1998.
- [180] PrimeSense, Wikipedia. (n.d.). <https://en.wikipedia.org/wiki/PrimeSense>.
- [181] M.G. Diaz, F. Tombari, P. Rodriguez-Gonzalvez, D. Gonzalez-Aguilera, Analysis and Evaluation between the First and the Second Generation of RGB-D Sensors, *IEEE Sens. J.* 15 (2015) 6507–6516. doi:10.1109/JSEN.2015.2459139.
- [182] A.X. Fan, S. Dakpé, T.T. Dao, P. Pouletaut, M. Rachik, M.C. Ho Ba Tho, MRI-based finite element modeling of facial mimics: a case study on the paired zygomaticus major muscles, *Comput. Methods Biomech. Biomed. Engin.* 20 (2017) 919–928. doi:10.1080/10255842.2017.1305363.
- [183] L. Tran, F. Liu, X. Liu, Towards high-fidelity nonlinear 3D face morphable model, in: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019: pp. 1126–1135.
- [184] F. Wu, L. Bao, Y. Chen, Y. Ling, Y. Song, S. Li, K.N. Ngan, W. Liu, Mvf-net: Multi-view 3d face morphable model regression, in: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019: pp. 959–968.
- [185] Y. Zhou, J. Deng, I. Kotsia, S. Zafeiriou, Dense 3D Face Decoding over 2500FPS: Joint Texture & Shape Convolutional Mesh Decoders, in: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019: pp. 1097–1106.
- [186] B. Gecer, S. Ploumpis, I. Kotsia, S. Zafeiriou, GANFIT: Generative adversarial network fitting for high fidelity 3D face reconstruction, in: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019: pp. 1155–1164.
- [187] T.-N. Nguyen, S. Dakpé, M.-C. Ho Ba Tho, T.-T. Dao, Real-time computer vision system for tracking simultaneously subject-specific rigid head and non-rigid facial mimic movements using a contactless sensor and system of systems approach, *Comput. Methods Programs Biomed.* 191 (2020) 105410. doi:10.1016/j.cmpb.2020.105410.
- [188] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, L. Tarbox, F. Prior, The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository., *J. Digit. Imaging.* 26 (2013) 1045–57. doi:10.1007/s10278-013-9622-7.
- [189] T. Ju, S. Schaefer, J. Warren, Mean value coordinates for closed triangular meshes, in: *ACM SIGGRAPH 2005 Pap. - SIGGRAPH '05*, ACM Press, New York, New York, USA, 2005: p. 561. doi:10.1145/1186822.1073229.
- [190] M. Vallières, E. Kay-Rivest, L.J. Perrin, X. Liem, C. Furstoss, H.J.W.L. Aerts, N. Khaouam,

- P.F. Nguyen-Tan, C.-S. Wang, K. Sultanem, J. Seuntjens, I. El Naqa, Radiomics strategies for risk assessment of tumour failure in head-and-neck cancer, *Sci. Rep.* 7 (2017) 10117. doi:10.1038/s41598-017-10371-5.
- [191] S. Pieper, M. Halle, R. Kikinis, 3D Slicer, (2005) 632–635. doi:10.1109/isbi.2004.1398617.
- [192] T.T. Dao, P. Pouletaut, F. Charleux, Á. Lazáry, P. Eltes, P.P. Varga, M.C. Ho Ba Tho, Multimodal medical imaging (CT and dynamic MRI) data and computer-graphics multi-physical model for the estimation of patient specific lumbar spine muscle forces, *Data Knowl. Eng.* 96–97 (2015) 3–18. doi:10.1016/j.datak.2015.04.001.
- [193] W.E. Lorensen, H.E. Cline, Marching cubes: A high resolution 3D surface construction algorithm, *ACM SIGGRAPH Comput. Graph.* 21 (1987) 163–169. doi:10.1145/37402.37422.
- [194] D.A. Field, Laplacian smoothing and Delaunay triangulations, *Commun. Appl. Numer. Methods.* 4 (1988) 709–712. doi:10.1002/cnm.1630040603.
- [195] K.E. Smith, G. Bhatia, M.W. Vannier, Assessment of mass properties of human head using various three-dimensional imaging modalities, *Med. Biol. Eng. Comput.* 33 (1995) 278–284. doi:10.1007/BF02510500.
- [196] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, G. Taubin, The ball-pivoting algorithm for surface reconstruction, *IEEE Trans. Vis. Comput. Graph.* 5 (1999) 349–359. doi:10.1109/2945.817351.
- [197] A. Fabri, G.-J. Giezeman, L. Kettner, S. Schirra, S. Schnherr, On the design of CGAL a computational geometry algorithms library, *Softw. Pract. Exp.* 30 (2000) 1167–1202. doi:10.1002/1097-024X(200009)30:11<1167::AID-SPE337>3.0.CO;2-B.
- [198] S. Marden, J. Guivant, Improving the performance of ICP for real-time applications using an approximate nearest neighbour search, *Australas. Conf. Robot. Autom. ACRA.* (2012) 3–5.
- [199] P. Alliez, É.C. De Verdière, O. Devillers, M. Isenburg, Isotropic surface remeshing, *Proc. - SMI 2003 Shape Model. Int. 2003.* 2003 (2003) 49–58. doi:10.1109/SMI.2003.1199601.
- [200] R.B. Rusu, S. Cousins, 3D is here: Point Cloud Library (PCL), in: 2011 IEEE Int. Conf. Robot. Autom., IEEE, 2011: pp. 1–4. doi:10.1109/ICRA.2011.5980567.
- [201] P. Geladi, Bruce R. Kowalski, Partial Least-Squares Regression: A tutorial, *J. Optoelectron. Adv. Mater.* 10 (1986) 1–17. doi:10.1016/0003-2670(86)80028-9.
- [202] F. Lindgren, P. Geladi, S. Wold, The kernel algorithm for PLS, *J. Chemom.* 7 (1993) 45–59. doi:10.1002/cem.1180070104.
- [203] B.S. Dayal, J.F. Macgregor, Improved PLS algorithms, *J. Chemom.* 11 (1997) 73–85. doi:10.1002/(SICI)1099-128X(199701)11:1%3C73::AID-CEM435%3E3.0.CO;2-%23.
- [204] Free 3D Models, (n.d.). www.free3d.com.
- [205] M.S. Floater, Mean value coordinates, *Comput. Aided Geom. Des.* 20 (2003) 19–27. doi:10.1016/S0167-8396(03)00002-5.
- [206] N. Aspert, D. Santa-Cruz, T. Ebrahimi, MESH: measuring errors between surfaces using the Hausdorff distance, in: *Proceedings. IEEE Int. Conf. Multimed. Expo, IEEE, 1978:* pp. 705–708. doi:10.1109/ICME.2002.1035879.
- [207] I.S. Pandzic, R. Forchheimer, MPEG-4 Facial Animation, 2003. doi:10.1002/ima.10072.
- [208] P.M. Prendergast, Facial anatomy, *Adv. Surg. Facial Rejuvenation Art Clin. Pract.* 9783642178 (2012) 3–14. doi:10.1007/978-3-642-17838-2_1.
- [209] T.W. Sederberg, S.R. Parry, Free-form deformation of solid geometric models, *ACM SIGGRAPH Comput. Graph.* 20 (1986) 151–160.
- [210] J.L. Godoy, J.R. Vega, J.L. Marchetti, Relationships between PCA and PLS-regression, *Chemom. Intell. Lab. Syst.* 130 (2014) 182–191. doi:10.1016/j.chemolab.2013.11.008.
- [211] F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, *IEEE Trans. Neural Networks.* 20 (2008) 61–80.
- [212] R. Bhalodia, S.Y. Elhabian, L. Kavan, R.T. Whitaker, Deepssm: A deep learning framework for statistical shape modeling from raw images, in: *Int. Work. Shape Med. Imaging, 2018:* pp. 244–257.
- [213] M. Reuter, C. Wachinger, H. Lombaert, B. Paniagua, M. Lüthi, B. Egger, Shape in Medical Imaging: International Workshop, ShapeMI 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, *Proceedings, Springer, 2018.*
- [214] M.M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, P. Vandergheynst, Geometric deep learning: going beyond euclidean data, *IEEE Signal Process. Mag.* 34 (2017) 18–42.
- [215] T.T. Dao, P. Pouletaut, Á. Lazáry, M.C.H.B. Tho, Multimodal Medical Imaging Fusion for Patient Specific Musculoskeletal Modeling of the Lumbar Spine System in Functional Posture,

- J. Med. Biol. Eng. 37 (2017) 739–749. doi:10.1007/s40846-017-0243-3.
- [216] T.-N. Nguyen, V.-D. Tran, H.-Q. Nguyen, T.-T. Dao, A statistical shape modeling approach for predicting subject-specific human skull from head surface, *Med. Biol. Eng. Comput.* In Press (2020). doi:10.1007/s11517-020-02219-4.
- [217] M. Fratarcangeli, M. Schaerf, Realistic modeling of animatable faces in MPEG-4, *Comput. Animat. Soc.* (2004) 1–11. <http://www.fratarcangeli.net/pubs/casa04.mf.pdf>.
- [218] W. Happak, J. Liu, G. Burggasser, A. Flowers, H. Gruber, G. Freilinger, Human facial muscles: Dimensions, motor endplate distribution, and presence of muscle fibers with multiple motor endplates, *Anat. Rec.* 249 (1997) 276–284. doi:10.1002/(SICI)1097-0185(199710)249:2<276::AID-AR15>3.0.CO;2-L.
- [219] S. Pieper, M. Halle, R. Kikinis, 3D Slicer, in: 2004 2nd IEEE Int. Symp. Biomed. Imaging Nano to Macro (IEEE Cat No. 04EX821), 2004: pp. 632–635 Vol. 1. doi:10.1109/ISBI.2004.1398617.
- [220] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, G. Ranzuglia, Meshlab: an open-source mesh processing tool., in: *Eurographics Ital. Chapter Conf.*, 2008: pp. 129–136. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.649.4449>.
- [221] G. Freilinger, H. Gruber, W. Happak, U. Pechmann, Surgical anatomy of the mimic muscle system and the facial nerve: Importance for reconstructive and aesthetic surgery, *Plast. Reconstr. Surg.* 80 (1987) 686–690. doi:10.1097/00006534-198711000-00005.
- [222] P.C.M. Benington, J.E. Gardener, N.P. Hunt, Masseter muscle volume measured using ultrasonography and its relationship with facial morphology, *Eur. J. Orthod.* 21 (1999) 659–670. doi:10.1093/ejo/21.6.659.
- [223] T.T. Dao, Rigid musculoskeletal models of the human body systems: a review, *J. Musculoskelet. Res.* 19 (2016) 1630001. doi:10.1142/S0218957716300015.
- [224] A. Ballit, I. Mougharbel, H. Ghaziri, T.T. Dao, Fast Soft Tissue Deformation and Stump-Socket Interaction Toward a Computer-Aided Design System for Lower Limb Prostheses, *Irbm.* 1 (2020) 1–10. doi:10.1016/j.irbm.2020.02.003.
- [225] M. Allen, Facial Action Coding System, 2017. doi:10.4135/9781483381411.n178.
- [226] T.-N. NGUYEN, S. DAKPE, M.-C. HO BA THO, T.-T. DAO, Kinect-driven Patient-specific Head, Skull, and Muscle Network Modelling for Facial Palsy Patients, *Comput. Methods Programs Biomed.* (2020) 31. doi:10.1016/j.cmpb.2020.105846.
- [227] T. Wu, A.P.L. Hung, P. Hunter, K. Mithraratne, Modelling facial expressions: A framework for simulating nonlinear soft tissue deformations using embedded 3D muscles, *Finite Elem. Anal. Des.* 76 (2013) 63–70. doi:10.1016/j.finel.2013.08.002.
- [228] D.S. Hui, E.I. Azhar, Z.A. Memish, A. Zumla, Human Coronavirus Infections—Severe Acute Respiratory Syndrome (SARS), Middle East Respiratory Syndrome (MERS), and SARS-CoV-2, in: *Ref. Modul. Biomed. Sci.*, Elsevier, 2020. doi:10.1016/B978-0-12-801238-3.11634-4.
- [229] A. Savran, N. Alyüz, H. Dibekliouglu, O. Çeliktutan, B. Gökberk, B. Sankur, L. Akarun, Bosphorus database for 3D face analysis, in: *Eur. Work. Biometrics Identity Manag.*, 2008: pp. 47–56.
- [230] Firelight Technologies, FMOD - Sound. Logic., (1995). <https://www.fmod.com/>.
- [231] Loki Software, OpenAL, (2009). <http://www.openal.org/>.
- [232] un4seen developments, BASS, (n.d.). <http://www.un4seen.com/>.
- [233] N. Gebhardt, irrKlang high level audio engine, (2006). <http://ambiera.com/irrklang/>.
- [234] R. Randell, N. Mitchell, D. Dowding, N. Cullum, C. Thompson, Effects of computerized decision support systems on nursing performance and patient outcomes: A systematic review, *J. Heal. Serv. Res. Policy.* 12 (2007) 242–249. doi:10.1258/135581907782101543.
- [235] K. Kawamoto, C.A. Houlihan, E.A. Balas, D.F. Lobach, Improving clinical practice using clinical decision support systems: A systematic review of trials to identify features critical to success, *Br. Med. J.* 330 (2005) 765–768. doi:10.1136/bmj.38398.500764.8f.
- [236] A. Berlin, M. Sorani, I. Sim, A taxonomic description of computer-based clinical decision support systems, *J. Biomed. Inform.* 39 (2006) 656–667. doi:10.1016/j.jbi.2005.12.003.
- [237] A. Nikfarjam, E. Emadzadeh, G. Gonzalez, *Biomedical Informatics Insights*, 2014. doi:10.4137/BII.S8981.
- [238] E. Kilsdonk, L.W. Peute, M.W.M. Jaspers, Factors influencing implementation success of guideline-based clinical decision support systems: A systematic review and gaps analysis, *Int. J. Med. Inform.* 98 (2017) 56–64. doi:10.1016/j.ijmedinf.2016.12.001.
- [239] W. Sun, P. Lal, Recent development on computer aided tissue engineering — a review, *Comput. Methods Programs Biomed.* 67 (2002) 85–103. doi:10.1016/S0169-2607(01)00116-X.

- [240] A. Nealen, M. Müller, R. Keiser, E. Boxerman, M. Carlson, Physically Based Deformable Models in Computer Graphics, (2005).
- [241] P. Fraccaro, D. O’Sullivan, P. Plastiras, H. O’Sullivan, C. Dentone, A. Di Biagio, P. Weller, Behind the screens: Clinical decision support methodologies - A review, *Heal. Policy Technol.* 4 (2015) 29–38. doi:10.1016/j.hlpt.2014.10.001.
- [242] S. Marchesseau, T. Heimann, S. Chatelin, R. Willinger, H. Delingette, Fast porous visco-hyperelastic soft tissue model for surgery simulation: Application to liver surgery, *Prog. Biophys. Mol. Biol.* 103 (2010) 185–196. doi:10.1016/j.pbiomolbio.2010.09.005.
- [243] S. Niroomandi, I. Alfaro, E. Cueto, F. Chinesta, Accounting for large deformations in real-time simulations of soft tissues based on reduced-order models, *Comput. Methods Programs Biomed.* 105 (2012) 1–12. doi:10.1016/j.cmpb.2010.06.012.
- [244] H. Courtecuisse, J. Allard, P. Kerfriden, S.P.A. Bordas, S. Cotin, C. Duriez, Real-time simulation of contact and cutting of heterogeneous soft-tissues, *Med. Image Anal.* 18 (2014) 394–410. doi:10.1016/j.media.2013.11.001.
- [245] A.K. Ho, H. Alsaffar, P.C. Doyle, H.M. Ladak, S.K. Agrawal, Virtual reality myringotomy simulation with real-time deformation: development and validity testing., *Laryngoscope.* 122 (2012) 1844–51. doi:10.1002/lary.23361.
- [246] D. Moher, A. Liberati, J. Tetzlaff, D.G. Altman, Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement, *Int. J. Surg.* 8 (2010) 336–341. doi:https://doi.org/10.1016/j.ijso.2010.02.007.
- [247] J. Berkley, G. Turkiyyah, D. Berg, M. Ganter, S. Weghorst, Real-time finite element modeling for surgery simulation: An application to virtual suturing, *Ieee Trans. Vis. Comput. Graph.* 10 (2004) 314–325. doi:10.1109/tvcg.2004.1272730.
- [248] G.R. Joldes, A. Wittek, K. Miller, Suite of finite element algorithms for accurate computation of soft tissue deformation for surgical simulation, *Med. Image Anal.* 13 (2009) 912–919. doi:10.1016/j.media.2008.12.001.
- [249] K. Miller, G. Joldes, D. Lance, A. Wittek, Total Lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation, *Commun. Numer. Methods Eng.* 23 (2007) 121–134. doi:10.1002/cnm.887.
- [250] Z.A. Taylor, M. Cheng, S. Ourselin, High-Speed Nonlinear Finite Element Analysis for Surgical Simulation Using Graphics Processing Units, *IEEE Trans. Med. Imaging.* 27 (2008) 650–663. doi:10.1109/TMI.2007.913112.
- [251] S. Cotin, H. Delingette, N. Ayache, Real-time elastic deformations of soft tissues for surgery simulation, *IEEE Trans. Vis. Comput. Graph.* 5 (1999) 62–73. doi:10.1109/2945.764872.
- [252] M.A. Audette, V. Hayward, O. Astley, M. Doyon, G.A. McCallister, K. Chinzei, A PC-based system architecture for real-time finite element-based tool-specific surgical simulation, *Int. Congr. Ser.* 1268 (2004) 378–383. doi:10.1016/j.ics.2004.03.294.
- [253] M. Sedef, E. Samur, C. Basdogan, Real-Time Finite-Element Simulation of Linear Viscoelastic Tissue Behavior Based on Experimental Data, *IEEE Comput. Graph. Appl.* 26 (2006) 58–68. doi:10.1109/MCG.2006.135.
- [254] G. Sela, J. Subag, A. Lindblad, D. Albocher, S. Schein, G. Elber, Real-time haptic incision simulation using FEM-based discontinuous free-form deformation, *CAD Comput. Aided Des.* 39 (2007) 685–693. doi:10.1016/j.cad.2007.05.011.
- [255] M. García, O.D. Robles, L. Pastor, A. Rodríguez, MSRS: A fast linear solver for the real-time simulation of deformable objects, *Comput. Graph.* 32 (2008) 293–306. doi:10.1016/j.cag.2008.01.008.
- [256] G.R. Joldes, A. Wittek, K. Miller, An efficient hourglass control implementation for the uniform strain hexahedron using the total Lagrangian formulation, *Int. j. Numer. Method. Biomed. Eng.* 24 (2008) 1315–1323.
- [257] G.R. Joldes, A. Wittek, K. Miller, Real-time nonlinear finite element computations on GPU – Application to neurosurgical simulation, *Comput. Methods Appl. Mech. Eng.* 199 (2010) 3305–3314. doi:10.1016/j.cma.2010.06.037.
- [258] A. Wittek, G. Joldes, M. Couton, S.K. Warfield, K. Miller, Patient-specific non-linear finite element modelling for predicting soft organ deformation in real-time; Application to non-rigid neuroimage registration, *Prog. Biophys. Mol. Biol.* 103 (2010) 292–303. doi:10.1016/j.pbiomolbio.2010.09.001.
- [259] I. Peterlík, M. Sedef, C. Basdogan, L. Matyska, Real-time visio-haptic interaction with static soft tissue models having geometric and material nonlinearity, *Comput. Graph.* 34 (2010) 43–54.

- doi:10.1016/j.cag.2009.10.005.
- [260] R.J. Lapeer, P.D. Gasson, V. Karri, Simulating plastic surgery: From human skin tensile tests, through hyperelastic finite element models to real-time haptics, *Prog. Biophys. Mol. Biol.* 103 (2010) 208–216. doi:10.1016/j.pbiomolbio.2010.09.013.
- [261] H. Courtecuisse, H. Jung, J. Allard, C. Duriez, D.Y. Lee, S. Cotin, GPU-based real-time soft tissue deformation with cutting and haptic feedback, *Prog. Biophys. Mol. Biol.* 103 (2010) 159–168. doi:10.1016/j.pbiomolbio.2010.09.016.
- [262] G.M. Turkiyyah, W.B. Karam, Z. Ajami, A. Nasri, Mesh cutting during real-time physical simulation, *CAD Comput. Aided Des.* 43 (2011) 809–819. doi:10.1016/j.cad.2010.10.005.
- [263] K. Morooka, Y. Nakasuka, R. Kurazume, Xian Chen, T. Hasegawa, M. Hashizume, Navigation system with real-time finite element analysis for minimally invasive surgery, in: 2013 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., IEEE, 2013: pp. 2996–2999. doi:10.1109/EMBC.2013.6610170.
- [264] R. Mafi, S. Sirouspour, GPU-based acceleration of computations in nonlinear finite element deformation analysis, *Int. j. Numer. Method. Biomed. Eng.* 26 (2014) 807–827. doi:10.1002/cnm.
- [265] V. Strbac, J. Vander Sloten, N. Famaey, Analyzing the potential of GPGPUs for real-time explicit finite element analysis of soft tissue deformation using CUDA, *Finite Elem. Anal. Des.* 105 (2015) 79–89. doi:10.1016/j.finel.2015.07.005.
- [266] A. Karami, M. Eghtesad, S.A. Haghpanah, Prediction of muscle activation for an eye movement with finite element modeling, *Comput. Biol. Med.* 89 (2017) 368–378. doi:10.1016/j.compbio.2017.08.018.
- [267] F. Martínez-Martínez, M.J. Rupérez-Moreno, M. Martínez-Sober, J.A. Solves-Llorens, D. Lorente, A.J. Serrano-López, S. Martínez-Sanchis, C. Monserrat, J.D. Martín-Guerrero, A finite element-based machine learning approach for modeling the mechanical behavior of the breast tissues under compression in real-time, *Comput. Biol. Med.* 90 (2017) 116–124. doi:10.1016/j.compbio.2017.09.019.
- [268] D. Lorente, F. Martínez-Martínez, M.J. Rupérez, M.A. Lago, M. Martínez-Sober, P. Escandell-Montero, J.M. Martínez-Martínez, S. Martínez-Sanchis, A.J. Serrano-López, C. Monserrat, J.D. Martín-Guerrero, A framework for modelling the biomechanical behaviour of the human liver during breathing in real time using machine learning, *Expert Syst. Appl.* 71 (2017) 342–357. doi:10.1016/j.eswa.2016.11.037.
- [269] M. Tonutti, G. Gras, G.Z. Yang, A machine learning approach for real-time modelling of tissue deformation in image-guided neurosurgery, *Artif. Intell. Med.* 80 (2017) 39–47. doi:10.1016/j.artmed.2017.07.004.
- [270] V. Luboz, M. Bailet, C. Boichon Grivot, M. Rochette, B. Diot, M. Bucki, Y. Payan, Personalized modeling for real-time pressure ulcer prevention in sitting posture, *J. Tissue Viability.* (2017). doi:10.1016/j.jtv.2017.06.002.
- [271] L.P. Nedel, D. Thalmann, Real time muscle deformations using mass-spring systems, *Proceedings. Comput. Graph. Int. (Cat. No.98EX149).* (1998) 156–165. doi:10.1109/CGI.1998.694263.
- [272] C. Monserrat, U. Meier, M. Alcaniz, F. Chinesta, M.C. Juan, A new approach for the real-time simulation of tissue deformations in surgery simulation., *Comput. Methods Programs Biomed.* 64 (2001) 77–85. doi:10.1016/S0169-2607(00)00093-6.
- [273] T. Goto, W.S. Lee, N. Magnenat-thalmann, Facial feature extraction for quick 3D face modeling, *Signal Process. Image Commun.* 17 (2002) 243–259. doi:10.1016/S0923-5965(01)00021-2.
- [274] C. Bonamico, M. Costa, F. Lavagetto, R. Pockaj, Real-time MPEG-4 facial animation with 3D scalable meshes, *Signal Process. Image Commun.* 17 (2002) 743–757. doi:10.1016/S0923-5965(02)00075-9.
- [275] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, H.-P. Seidel, Laplacian surface editing, in: *Proc. 2004 Eurographics/ACM SIGGRAPH Symp. Geom. Process. - SGP '04*, ACM Press, New York, New York, USA, 2004: p. 175. doi:10.1145/1057432.1057456.
- [276] W. Mollemans, F. Schutyser, N. Nadjmi, P. Suetens, Very fast soft tissue predictions with mass tensor model for maxillofacial surgery planning systems, *Int. Congr. Ser.* 1281 (2005) 491–496. doi:10.1016/j.ics.2005.03.048.
- [277] P. Chen, K.E. Barner, K. V Steiner, A Displacement Driven Real-Time Deformable Model For Haptic Surgery Simulation, *Haptic Interfaces Virtual Environ. Teleoperator Syst.* 2006 14th Symp. (2006) 499–505. doi:10.1109/HAPTIC.2006.1627099.

- [278] M. López-Cano, J. Rodríguez-Navarro, A. Rodríguez-Baeza, M. Armengol-Carrasco, A. Susín, A real-time dynamic 3D model of the human inguinal region for surgical education, *Comput. Biol. Med.* 37 (2007) 1321–1326. doi:10.1016/j.compbiomed.2006.12.006.
- [279] Y.J. Lim, S. De, Real time simulation of nonlinear tissue response in virtual surgery using the point collocation-based method of finite spheres, *Comput. Methods Appl. Mech. Eng.* 196 (2007) 3011–3024. doi:10.1016/j.cma.2006.05.015.
- [280] A. Murai, K. Kurosaki, K. Yamane, Y. Nakamura, Musculoskeletal-see-through mirror: Computational modeling and algorithm for whole-body muscle activity visualization in real time, *Prog. Biophys. Mol. Biol.* 103 (2010) 310–317. doi:10.1016/j.pbiomolbio.2010.09.006.
- [281] E. Basafa, F. Farahmand, Real-time simulation of the nonlinear visco-elastic deformations of soft tissues, *Int. J. Comput. Assist. Radiol. Surg.* 6 (2011) 297–307. doi:10.1007/s11548-010-0508-6.
- [282] J. Wang, S. Liao, X. Zhu, Y. Wang, C. Ling, X. Ding, Y. Fang, X. Zhang, Real time 3D simulation for nose surgery and automatic individual prosthesis design, *Comput. Methods Programs Biomed.* 104 (2011) 472–479. doi:10.1016/j.cmpb.2010.09.001.
- [283] B.H. Le, M. Zhu, Z. Deng, Marker optimization for facial motion acquisition and deformation, *IEEE Trans. Vis. Comput. Graph.* 19 (2013) 1859–1871. doi:10.1109/TVCG.2013.84.
- [284] Y. Zhang, W. Lin, B. Zhou, Z. Chen, B. Sheng, J. Wu, Facial expression cloning with elastic and muscle models, *J. Vis. Commun. Image Represent.* 25 (2014) 916–927. doi:10.1016/j.jvcir.2014.02.010.
- [285] Y. Weng, C. Cao, Q. Hou, K. Zhou, Real-time facial animation on mobile devices, *Graph. Models.* 76 (2014) 172–179. doi:10.1016/j.gmod.2013.10.002.
- [286] F. Goulette, Z.-W. Chen, Fast computation of soft tissue deformations in real-time simulation with Hyper-Elastic Mass Links, *Comput. Methods Appl. Mech. Eng.* 295 (2015) 18–38. doi:10.1016/j.cma.2015.06.015.
- [287] J. Zhang, Y. Zhong, J. Smith, C. Gu, A new ChainMail approach for real-time soft tissue simulation, *Bioengineered.* 7 (2016) 246–252. doi:10.1080/21655979.2016.1197634.
- [288] A. Woodward, Y.H. Chan, R. Gong, M. Nguyen, T. Gee, P. Delmas, G. Gimel'farb, J.A. Marquez Flores, A low cost framework for real-time marker based 3-D human expression modeling, *J. Appl. Res. Technol.* 15 (2017) 61–77. doi:10.1016/j.jart.2017.01.002.
- [289] J. Zhou, Z. Luo, C. Li, M. Deng, Real-time deformation of human soft tissues: A radial basis meshless 3D model based on Marquardt's algorithm, *Comput. Methods Programs Biomed.* 153 (2018) 237–252. doi:10.1016/j.cmpb.2017.09.008.
- [290] G. Yarnitzky, Z. Yizhar, A. Gefen, Real-time subject-specific monitoring of internal deformations and stresses in the soft tissues of the foot: A new approach in gait analysis, *J. Biomech.* 39 (2006) 2673–2689. doi:10.1016/j.jbiomech.2005.08.021.
- [291] I.S. Pandzic, R. Forchheimer, *MPEG-4 facial animation: the standard, implementation and applications*, John Wiley & Sons, 2003.
- [292] M. Doblaré, E. Cueto, B. Calvo, M.A. Martínez, J.M. Garcia, J. Cegoñino, On the employ of meshless methods in biomechanics, *Comput. Methods Appl. Mech. Eng.* 194 (2005) 801–821. doi:10.1016/j.cma.2004.06.031.
- [293] G.Y. Zhang, A. Wittek, G.R. Joldes, X. Jin, K. Miller, A three-dimensional nonlinear meshfree algorithm for simulating mechanical responses of soft tissue, *Eng. Anal. Bound. Elem.* 42 (2014) 60–66. doi:10.1016/j.enganabound.2013.08.014.
- [294] M.H. Doweidar, B. Calvo, I. Alfaro, P. Groenenboom, M. Doblaré, A comparison of implicit and explicit natural element methods in large strains problems: Application to soft biological tissues modeling, *Comput. Methods Appl. Mech. Eng.* 199 (2010) 1691–1700. doi:10.1016/j.cma.2010.01.022.
- [295] F. Roohbakhshan, R.A. Sauer, Efficient isogeometric thin shell formulations for soft biological materials, *Biomech. Model. Mechanobiol.* 16 (2017) 1569–1597. doi:10.1007/s10237-017-0906-6.
- [296] D. Kamensky, F. Xu, C.H. Lee, J. Yan, Y. Bazilevs, M.C. Hsu, A contact formulation based on a volumetric potential: Application to isogeometric simulations of atrioventricular valves, *Comput. Methods Appl. Mech. Eng.* 330 (2018) 522–546. doi:10.1016/j.cma.2017.11.007.
- [297] T.S. Sørensen, J. Mosegaard, An Introduction to GPU Accelerated Surgical Simulation, in: M. Harders, G. Székely (Eds.), *Biomed. Simul.*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006: pp. 93–104.
- [298] T.T. Dao, A.-X. Fan, S. Dakpé, P. Pouletaut, M. Rachik, M.C. Ho Ba Tho, Image-based Skeletal

- Muscle Coordination: Case Study on a Subject Specific Facial Mimic Simulation, *J. Mech. Med. Biol.* 18 (2018) 1850020. doi:10.1142/S0219519418500203.
- [299] T.T. Dao, M.-C.H.B. Tho, A Systematic Review of Continuum Modeling of Skeletal Muscles: Current Trends, Limitations, and Recommendations, *Appl. Bionics Biomech.* 2018 (2018) 1–17. doi:10.1155/2018/7631818.
- [300] Y.-J. Lim, S. De, Real time simulation of nonlinear tissue response in virtual surgery using the point collocation-based method of finite spheres, *Comput. Methods Appl. Mech. Eng.* 196 (2007) 3011–3024. doi:10.1016/j.cma.2006.05.015.
- [301] B. Wan, M. Shahmoradi, Z. Zhang, Y. Shibata, B. Sarrafpour, M. Swain, Q. Li, Modelling of stress distribution and fracture in dental occlusal fissures, *Sci. Rep.* 9 (2019) 4682.
- [302] Y. Zhang, Y. Liu, Y. She, Y. Liang, F. Xu, C. Fang, The Effect of Endodontic Access Cavities on Fracture Resistance of First Maxillary Molar Using the Extended Finite Element Method, *J. Endod.* 45 (2019) 316–321. doi:10.1016/j.joen.2018.12.006.
- [303] S. Niroomandi, I. Alfaro, D. González, E. Cueto, F. Chinesta, Real-time simulation of surgery by reduced-order modeling and X-FEM techniques, *Int. j. Numer. Method. Biomed. Eng.* 28 (2012) 574–588. doi:10.1002/cnm.1491.
- [304] R. Avazmohammadi, D.S. Li, T. Leahy, E. Shih, J.S. Soares, J.H. Gorman, R.C. Gorman, M.S. Sacks, An integrated inverse model-experimental approach to determine soft tissue three-dimensional constitutive parameters: application to post-infarcted myocardium, *Biomech. Model. Mechanobiol.* 17 (2018) 31–53. doi:10.1007/s10237-017-0943-1.
- [305] R. Avazmohammadi, J.S. Soares, D.S. Li, S.S. Raut, R.C. Gorman, M.S. Sacks, A Contemporary Look at Biomechanical Models of Myocardium, *Annu. Rev. Biomed. Eng.* 21 (2019) 417–442. doi:10.1146/annurev-bioeng-062117-121129.
- [306] R. Hazemi, L. Macaulay, Requirements for graphical user interface development environments for groupware, *Interact. Comput.* 8 (1996) 69–88. doi:10.1016/0953-5438(95)01018-1.

Appendices

Appendix A

A Systematic Review of Real-time Medical Simulations with Soft-tissue Deformation: Computational Approaches, Interaction Devices, System Architectures, and Clinical Validations

Real-time facial mimic simulation is primarily necessary for developing a clinical decision-support system for facial mimic rehabilitation. In fact, facial mimics are also formed by complex soft-tissue deformations on facial skins, so realistic simulation of facial mimics need both real-time framerates and acceptable accuracies when simulating soft-tissues. This appendix summarizes and analyzes achievements in literature for developing real-time soft-tissue simulation systems. A systematic review process was conducted using the PRISMA protocol with three reliable scientific search engines (ScienceDirect, PubMed, and IEEE). Fifty-five relevant papers were finally selected and included into the review process, and a quality assessment procedure was also performed on them. Four system development aspects: (1) computational approaches, (2) interaction devices, (3) system architectures, and (4) clinical validations were analyzed. As a result, the computational approaches were categorized into mesh-, meshfree- and hybrid approaches. The interaction devices concerned about combination between virtual surgical instruments and force-feedback devices, 3-D scanners, biomechanical sensors, human interface devices, 3-D viewers and 2-D/3-D optical cameras. System architectures were analyzed based on the concepts of system execution schemes and system frameworks. In particular, system execution schemes included distribution-based, multithread-based, and multi-model-based executions. System frameworks are grouped into the input and output interaction frameworks, the graphic interaction frameworks, the modelling frameworks, and the hybrid frameworks. Clinical validation procedures are ordered as three levels: geometrical validation, model behavior validation, and user acceptability/safety validation. The review provides useful information to characterize how real-time medical simulation systems with soft tissue deformations have been developed. By clearly analyzing advantages and drawbacks in each system development aspect for real-time simulation of soft-tissues, this review can be used to provide technology selections for developing the target system.

A1. Materials and methods

A systematic review method was conducted using the PRISMA protocol [246] (Figure 81). Three scientific databases were chosen as ScienceDirect, PubMed, and IEEE. In more details, a focus on human soft tissues like upper/lower limb muscles, facial muscles, livers, and skins was done. A special attention was also given on the contributions related to the improvement of computational methods and/or employing effective hardware/software system architectures for real-time medical simulation systems. Finally, other articles focused on analyzing applications of real-time soft-tissue models for system validation, user acceptability and safety requirements were included. Note that in this present review, method refers to the development strategy of mathematical constitutive formulations of soft-tissue deformations based on a specific computational approach. Reviewed studies relate to mesh-based and mesh-free methods. Algorithm concerns the procedure to compute soft-tissue deformations using specific modeling methods. Model refers to the mathematical representation of soft-tissue deformations using mesh-based and mesh-free based methods. A set of search terminologies was defined for the literature investigation, and then each terminology was presented in a search term by using AND/OR operators. The used search terminologies and their appropriate search terms are listed

in Table 8. For the systematic information retrieval process, journal articles published up to December 2017 were assessed.

Table 8. *The search terms used for the systematic review process.*

#	Search terminologies (Terms)	Search terms (STs)
1	Term#1: Computer-aided medical simulations/systems	ST#1: Real-time AND computer-aided AND medical AND (simulations OR systems)
2	Term#2: Real-time biomedical simulations/systems	ST#2: Real-time AND biomedical AND (simulations OR systems)
3	Term#3: Real-time facial simulations	ST#3: Real-time AND facial AND simulations
4	Term#4: Real-time liver deformation models	ST#4: Real-time AND liver AND deformation AND models
5	Term#5: Real-time medical simulations/systems	ST#5: Real-time AND medical AND (simulations OR systems)
6	Term#6: Real-time muscle deformation models	ST#6: Real-time AND muscle AND deformation AND models
7	Term#7: Real-time surgery	ST#7: Real-time AND surgery
8	Term#8: Real-time finite element methods	ST#8: Real-time AND finite AND element AND methods
9	Term#9: Real-time soft tissue deformations	ST#9: Real-time AND soft AND tissue AND deformations

A1.1. Selection methodology

Selection was the most significant procedure for choosing both qualitatively and quantitatively appropriate articles for the systematic review. After identified from the search engines, retrieved articles were automatically saved to their suitable folders using the Mendeley paper management system. Two independent reviewers (TNN and TTD) screened and selected relevant papers for this review study. They also participated into the quality assessment. Consensus discussion was done when necessary for solving disagreements. The number of included/excluded articles is summarized in the Table 8. Firstly, the duplicates were checked with the duplication tool in the Mendeley software. The number of duplicated papers at this stage was 1,610 for all search terms. Then, the general and specific eligibility criteria were applied to the all unduplicated articles. The title inclusion criteria were first used for filtering out the irrelevant articles. The included articles at this phase were 973, which were then enrolled to the abstract filtering criteria for selecting the most pertinent articles. After reading all the abstracts, 92 included articles were then read in full-text to select the best qualitative and quantitative articles for systematically reviewing. Finally, the number of included articles was 55. Specifically, the flow chart of the selection procedure illustrating the number of included/excluded articles after each selection stage is shown in Figure 81. To answer the identified research questions, the selected 55 papers were categorized into four classes. The first category concerns the computational approaches for modeling deformations of human soft-tissues in real-time. The second category relates to the disadvantages and advantages of interaction devices for getting the external data from soft-tissues and visualizing the processed data. The third category deals with the characteristics of medical hardware/software systems consisting of graphic user interfaces (GUIs), programming languages, programming frameworks, and other techniques for developing soft-tissue simulation systems. The final category composes of system validations in clinical contexts and the analyses of user acceptability and safety requirements of developed systems. Additionally, each selected paper could also be grouped on multiple categories if their contents related to more than one category.

Table 9. *The number of included/excluded articles according to the selection procedure.*

Search Terms	Science Direct	PubMed	IEEE	All	Duplicates	Duplication included	Title Excluded	Title Included	Abstract Excluded	Abstract Included	Content excluded	Content Included
ST#1	5,537	67	13	5,617	21	5,596	5,465	131	128	3	2	1
ST#2	10,873	2,873	284	14,030	447	13,583	13,542	41	37	4	3	1
ST#3	3,638	87	14	533	19	514	335	179	169	10	0	10
ST#4	1,689	39	10	1,738	19	1,719	1,543	176	171	5	4	1
ST#5	32,857	9,762	290	42,909	605	42,304	42,209	95	83	12	7	5
ST#6	4,560	21	3	4,583	19	4,564	4,431	133	125	8	5	3
ST#7	104,034	31,339	367	135,727	371	135,356	135,265	91	76	15	3	12
ST#8	146,837	261	153	147,251	60	147,191	147,136	55	45	10	7	3
ST#9	9,185	160	23	9,368	49	9,319	9,247	72	47	25	6	19
Total	319,210	44,609	1,157	361,756	1,610	360,146	359,173	973	881	92	37	55

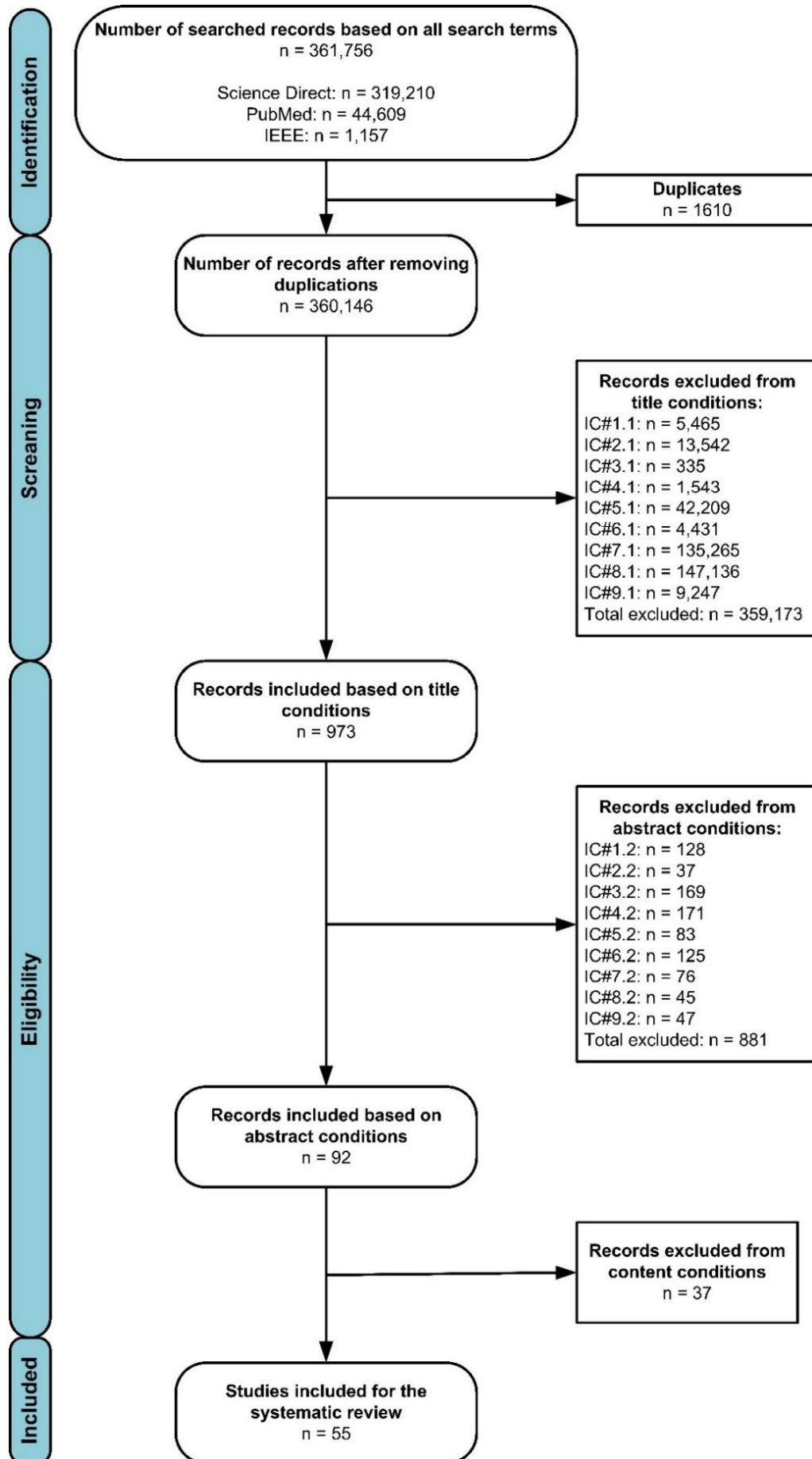


Figure 81. Workflow of the selection process using PRISMA protocol for the performed systematic review.

A1.2. Eligibility criteria

The inclusion/exclusion criteria were clearly defined based on the meaning of each search terminology. The list of inclusion criteria for each search terminology is shown in Table 10. In addition, to keep the literature at a high academic level, only journal articles were considered for the present review. Moreover, the articles in conferences with a couple of pages are initially eliminated. Other kinds of low-quality written forms such as letters, judgements, and book chapters were also not selected. Other than that, the articles were not written in English were excluded from the literature review.

Table 10. *The inclusion criteria for each search terminology*

#	Search Terms (STs)	Inclusion conditions (ICs)
1	ST#1	<p>IC#1.1: The title must satisfy all following conditions: 1) The title contains "real-time", "medical", "simulations" and "computer-aided" keywords; and 2) The title concerns the supports of computers in soft-tissue simulations executing in real-time.</p> <p>IC#1.2: The abstract must satisfy all following conditions: 1) The abstract concerns the supports of computer in medical systems, medical simulations, and medical applications so that they can be executed in real-time; 2) The abstract describes the medical system architectures, the interactions of computer's input/output devices in clinical environments; and 3) The system developed in the paper focuses on simulating human soft-tissues.</p>
2	ST#2	<p>IC#2.1: The title must satisfy all following conditions: 1) The title contains "real-time", "biomedical", and "simulations" keywords; and 2) The title concerns the issues of real-time simulation in biomedical applications.</p> <p>IC#2.2: The abstract must satisfy all following conditions: 1) The abstract concerns the analyses of real-time in biomedical applications/systems; and 2) The abstract focuses on analyzing the computational approaches, the system architectures, or the characteristics of real-time in biomedical applications.</p>
3	ST#3	<p>IC#3.1: The title must satisfy all following conditions: 1) The title contains "real-time" and "facial" keywords; and 2) The title concerns the computational approaches to simulate the human faces.</p> <p>IC#3.2: The abstract must satisfy all following conditions: 1) The abstract concerns the development of computational techniques or system designs for modelling the facial mimics/expressions/muscles; and 2) The developed techniques must be able to execute in real-time.</p>
4	ST#4	<p>IC#4.1: The title must satisfy all following conditions: 1) The title contains "real-time", "liver", and "models" keywords; and 2) The title concerns the modelling methods of human liver in real-time.</p> <p>IC#4.2: The abstract must satisfy all following conditions: 1) The abstract concerns the issues of computational approaches for modelling the human liver; and 2) The computational approaches must be executed in real-time.</p>
5	ST#5	<p>IC#5.1: The title must satisfy all following conditions: 1) The title contains "real-time", "medical", and "simulations"/"systems" keywords; and 2) The title aims at developing the computational methods for modelling the soft-tissue in medical environments.</p> <p>IC#5.2: The abstract must satisfy all following conditions: 1) The abstract concerns computational approaches or system architectures for modelling soft-tissues in medical environments; and 2) The system must be run in real-time.</p>
6	ST#6	<p>IC#6.1: The title must satisfy all following conditions: 1) The title contains "real-time", "muscle", and "models" keywords; and 2) The title considers the computational methods for modelling the human muscles in real-time.</p> <p>IC#6.2: The abstract must satisfy all following conditions: 1) The abstract concerns the developments of computational techniques for modelling and simulating human muscles so that</p>

#	Search Terms (STs)	Inclusion conditions (ICs)
		they can run in real-time; and 2) The abstract shows the implementations of muscle deformable models in clinical environments.
7	ST#7	<p>IC#7.1: The title must satisfy all following conditions: 1) The title contains "real-time" and "surgery" keywords; and 2) The title illustrates the surgical simulations/systems applied in human soft-tissues executed in real-time.</p> <p>IC#7.2: The abstract must satisfy all following conditions: 1) The abstract describes the surgical simulations/systems for human soft-tissues; and 2) The abstract concerns system architectures of surgical simulations or systems so that they can execute in real-time.</p>
8	ST#8	<p>IC#8.1: The title must satisfy all following conditions: 1) The title contains "real-time" and "finite element" keywords; and 2) The title concerns the finite element modelling methods for human soft-tissues in real-time.</p> <p>IC#8.2: The abstract must satisfy all following conditions: 1) The abstract concerns the human soft-tissue modelling method in real-time based on the finite element modelling methods; and 2) The abstract aims at development, generation, and analysis the variations of finite element modelling methods to get the real-time requirements.</p>
9	ST#9	<p>IC#9.1: The title must satisfy all following conditions: 1) The title contains "real-time", "soft tissue", and "deformations"/"models" keywords; and 2) The title considers the modelling methods of the human soft-tissue deformations executing in real-time.</p> <p>IC#9.2: The abstract must satisfy all following conditions: 1) The abstract illustrates the computational approaches for development the models of human soft-tissue deformations; and 2) The abstract aims at developing, analyzing, and generating the modelling methods.</p>

A1.3. Quality assessment

The quality assessment procedure was established to rate the quality of each analyzed paper. Eighteen Yes-No assessment items were defined and used. Papers related to computational approaches bias were evaluated using the following four items: 1) Was the method adequately used/developed and described for the involved tissue behavior?, 2) Was the verification well performed for the used/developed method?, 3) Was the validation systematically performed for the used/developed method?, and 4) Did the method really satisfy the real-time constraint?. Papers related to interaction devices bias were evaluated using the following four items: 5) Was the devices well selected for the system?, 6) Was the device accuracy adequate for the real-time constraints?, 7) Was the device use easy enough for a clinical routine practice?, and 8) Is the device price suitable for a clinical setting?. Papers related to system architectures bias were evaluated using the following four items: 9) Was the system adequately described?, 10) Was the system developed with the participation of the end users?, 11) Was the system scalable?, and 12) Were the system frameworks adequately selected for implementing the system of interest?. Papers related to clinical validation bias were evaluated using the following six items: 13) Was the study adequately validated with in vitro data?, 14) Was the study adequately validated with in vivo data?, 15) Was the study adequately validated with patient data?, 16) Was the level of validation suitable for translating the outcomes into clinical routine practices?, 17) Was the user acceptability performed for patients?, and 18) Was the user acceptability performed for clinical experts?

Note that the user acceptability validation is commonly conducted after developing a full simulation system. This validation targets at validating the acceptability level related to graphic system's user interfaces, system's ease-of-use, system's functions, system's robustness, etc., during short-term and/or long-term evaluation campaigns for clinicians. Regarding the verification of the developed method, an error check list related to the input data, algorithm

execution, output visualization is defined. The “well performed” category is assigned to a paper if all these three elements are satisfied.

A2. Results

A2.1. Overall quality assessment analysis

Statistical results of the quality assessment procedure are presented in Table 11. Overall, most selected articles well described, verified, and validated the computational approaches. Tissue behaviors were well described in selected studies. Over 80% of articles modelled the tissue physical characteristics in the methods while the others just focused on soft-tissue deformations. Most authors all well conducted verifications (76%) and validations steps (89%). For examples, in the study of S. Cotin et al., 1999¹², after clearly described the developed methods, the authors designed an example system using the method and analyzed the computed results. Their outputs were compared with other methods and showed faster computation time and higher accuracy level. Visualizations were also clearly presented to show computed deformations and collisions with virtual surgical tool. System performance and accuracy were also measured and verified. Thus, the verification was well-performed in this study. The verification procedure was not well-performed in Bensoussan et al., 2008⁵⁸ because they mainly introduced the SOFA framework, and the authors just verified their results by visual assessments. Although the real-time constraint was strongly required in the study objectives, only 65% of the developed computational approaches really satisfied this constraint. The others just nearly reached the real-time conditions. For example, the computation framerate was nearly 30 fps. In addition, all interaction devices were all accurate enough for using in clinical routines with acceptable prices, and they were also well selected for appropriate computational approaches and system architectures. Moreover, the data transmission bandwidths of these selected devices were relatively much faster than the computational and graphical rendering speeds, so they were all suitable for real-time applications. Over 50% of articles were implemented their developed computational approaches into a simulation system. They also well-described the architectures and frameworks of the implemented systems for future developments. However, these systems were rarely developed with the participation of end-users. They were mainly tested with the developers and did not have many feedbacks from users. Most of implemented simulation systems could not be directly transferred into the clinical routine practices due to lack of validations with *in vitro*, *in vivo*, and real patient data. The computed results of simulation systems were often validated with *in vitro* data acquired from phantom tissues with physical testing machines. Due to difficulties of acquiring data from living organs, only 13% of studies conducted clinical validations using *in vivo* data. Moreover, only external data such as deformations were available. Finally, the user and expert acceptability aspects were occasionally (i.e. only 4% and 7% of studies) investigated. Note that most developed systems were initially designed for testing and verifying the computational approaches rather than for developing real clinical applications.

Table 11. *Summary of the statistical results of the quality assessment procedure.*

Quality assessment criteria	% of 'yes' scores (%)
Computational approaches bias	
1. Was the method adequately used/developed and described for the involved tissue behavior?	82
2. Was the verification well performed for the used/developed method?	76
3. Was the validation systematically performed for the used/developed method?	89
4. Did the method really satisfy the real-time constraints?	65
Interaction devices bias	
5. Was the devices well selected for the system?	49
6. Was the device accuracy adequate for the real-time constraint?	47
7. Was the device use easy enough for a clinical routine practice?	47
8. Is the device price suitable for a clinical setting?	47
System architectures bias	
9. Was the system adequately described?	65
10. Was the system developed with the participation of the end users?	15
11. Was the system scalable?	53
12. Were the system frameworks adequately selected for implementing the system of interest?	45
Clinical applications bias	
13. Was the study adequately validated with in vitro data?	33
14. Was the study adequately validated with in vivo data?	13
15. Was the study adequately validated with patient data?	18
16. Was the level of validation suitable for translating the outcomes into clinical routine practices?	29
17. Was the user acceptability performed for patients?	4
18. Was the user acceptability performed for clinical experts?	7

A2.2. Computational approaches

To achieve real-time computation speed when rendering and computing soft-tissue deformations, two modeling approaches have been commonly adopted. The first approach that we called model development (MD) mainly focuses on geometry discretization strategy and mathematical constitutive formulations of soft-tissue stress-strain relationships. Soft-tissue models developed using this approach are commonly executed with a single-thread platform in a faster and/or more accurate manner. The second approach that we named as constitutive model implementations (MI) relates to the algorithmic implementations of the existing constitutive models using developed methods for soft-tissue deformations onto a more powerful hardware configuration such as Graphic Processing Unit (GPU) system. Thus, systems can compute soft-tissue models faster and more robustly than the traditional ones. Particularly, this concept refers to a family of more suitable programming algorithms to parallelize the execution tasks of a developed modeling method, which was traditionally running on a single-thread platform. For examples, Berkley et al., 2004, addressed a MD study related to the development of a Linearized FEM (L-FEM) method built from the reduced object kinematics [247]. The L-FEM method is suitable for modeling linear elasticity of soft-tissues. This method is faster than the FEM. Moreover, in the study of Joldes et al., 2008, the Total Lagrangian (TL) formulation was applied to improve the computation speed of the traditional FEM [248]. Additionally, the Total Lagrangian Explicit Dynamic FEM (TLED-FEM) formulation was developed by Miller et al., 2007, and it could run faster than the FEM when executing on the same CPU-based platform [249]. Regarding model implementation (MI) approach, only the implicit time integration of FEM method has been proven to be the most suitable for parallel implementation. This method was implemented in a GPU platform by Taylor et al., 2008 [250].

It is interesting to note that most studies focused at developing new mathematical methods for modeling the soft-tissue deformations rather than implementing the developed modeling

methods into a specific hardware configuration to accelerate computation speed. The distribution of the two approaches throughout the selected literature is illustrated in Figure 82. Obviously, among the total of 55 studies, over 80% of the studies proposed the model development of soft tissue deformations while only 18% of studies took advantages of specific hardware to accelerate available modeling methods. Regarding the MD approach, we grouped all developed computational methods into three categories: mesh-, meshfree-, and hybrid modeling methods (Table 12, Table 13, and Table 14). In more details, the mesh-based modeling methods refer to the development of the finite element method (FEM) and its variations to simulate the soft-tissue deformations in real-time (Figure 83). The meshfree-based modeling techniques refer to the decomposition of soft-tissue model into simpler physical sub-models or representations without meshing the domains of interest (Figure 84). The hybrid modeling methods take advantages of cooperating multiple modeling methods to increase both computation speeds and model accuracy. The distribution of selected studies according to each modeling method is shown in Figure 8. The result shows that up to 51% of the studies related to the mesh-based methods. The use of the meshfree-based methods reaches over 42%. Finally, the percentage of hybrid methods is around 7%.

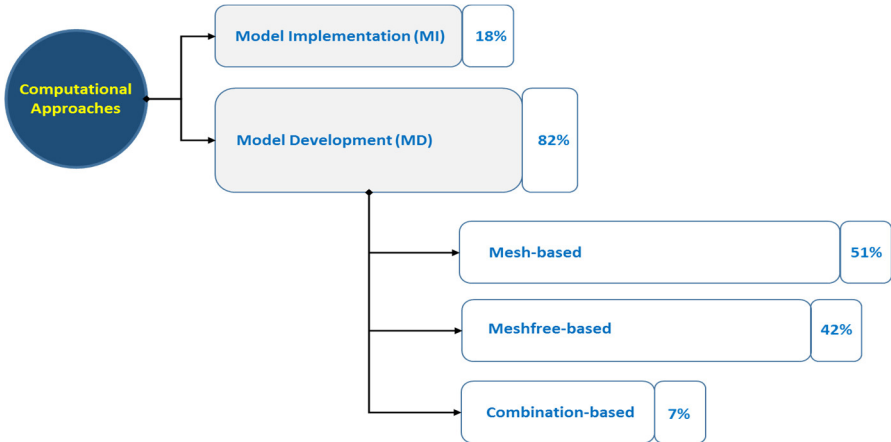


Figure 82. Distribution of computational approaches (MD and MI) and associated techniques for MD approach in the literature.

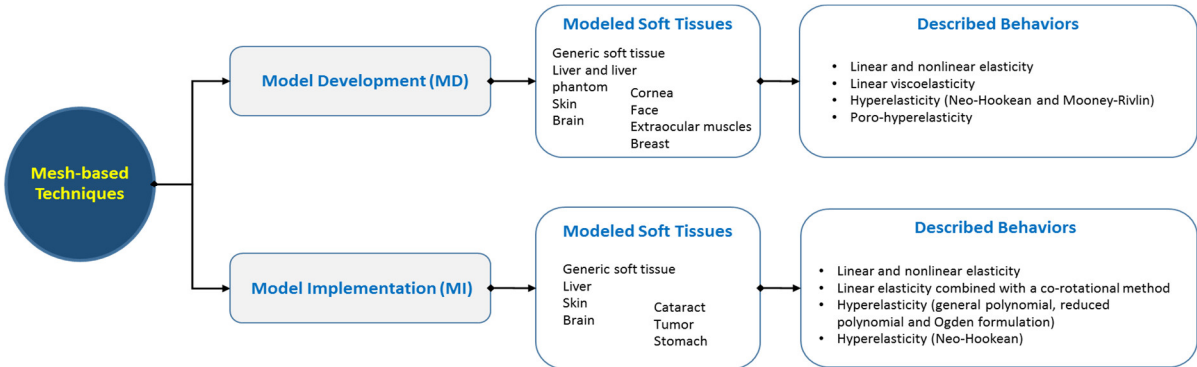


Figure 83. Overview of all modeled soft tissues and different described behaviors for mesh-based studies.

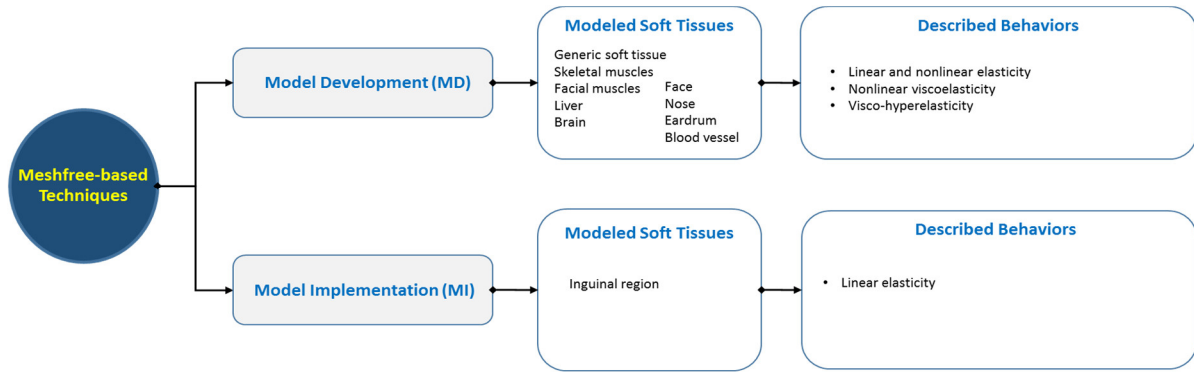


Figure 84. Overview of all modeled soft tissues and different described behaviors for meshfree-based studies.

Table 12. Classification of developed modelling methods for soft tissue deformations in real time: mesh-based techniques.

Reference	Approach	Modelling methods	Soft-tissue Types	Tissue Behaviours	Computation Time/Speed	Geometry Discretization	Hardware Configurations
S Cotin et al., 1999 [251]	MD	Precomputation-based FEM (Pre-comp FEM) approximated by linear functions	The human liver	Linear elasticity Nonlinear elasticity	7ms (force feedback) 8ms (force feedback)	1400 N* 6,500 tetrahedral elements	Dec AlphaStation 400MHz
Berkley et al., 2004 [247]	MD	Linearized FEM (L-FEM)	The human skin	Linear elasticity	1 kHz (force feedback) 30 Hz (model rendering)	863 N Surface triangle elements	1 GHz Athelon CPU
Audette et al., 2004 [252]	MI	Multi rate FEM (MR-FEM)	The human brain	Linear elasticity	10 kHz (force feedback)	NI**	Dual Pentium PC
Sedef et al., 2006 [253]	MD	Precomputation-based FEM (Pre-Comp FEM) using linear viscoelastic formulations	The soft-tissue cube	Linear viscoelasticity	1 kHz (force feedback) 100 Hz (model rendering)	51 N 153 DOF*** 136 tetrahedral elements	Pentium IV 2.4GHz dual CPU
Sela et al., 2007 [254]	MD	Precomputation-based FEM (Pre-Comp FEM) using discontinuous free-form deformations	The human skin	Linear elasticity	1 kHz (force feedback and model cutting)	12,108 polygons	P4-2.8 GHz CPU, 1GB RAM
Karol Miller et al., 2007 [249]	MD	Total Lagrangian explicit dynamics (TLED) FEM	NI	Nonlinear elasticity	16ms (model deformation)	6000 E****, 6741 N Hexahedral elements	3.2GHz Pentium IV
Garcia et al., 2008 [255]	MD	Matrix system reduction FEM (MSR-FEM)	NI	Linear elasticity	3.8ms – 35.7ms (solving the system)	From 266 N – 1579 E To 110 N – 587 E	2.4 GHz Pentium IV CPU, 1 GB
Joldes et al., 2008 [256]	MD	Total Lagrangian (TL) FEM	NI	Nonlinear elasticity	2.1ms (one system time-step)	2200 E - 2535 N Hexahedral elements	CPU
Taylor et al., 2008 [250]	MI	Total Lagrangian explicit dynamic (TLED) FEM	The human brain	Nonlinear elasticity	From 14.0 to 10.7 times faster than CPU	From 11,168 E to 46,655 E Tetrahedral elements	3.2 GHz P4 CPU, 2GB RAM NVIDIA GeForce 7900 GT GPU
Joldes et al., 2009 [248]	MD	Total Lagrangian explicit dynamics FEM (TLED-FEM)	The human brain	Hyperelasticity (Neo-Hookean)	12ms (model deformation) 1 kHz (haptic feedback)	15,050 E, 16,710 N 7000 DOF	3 GHz Intel Core Duo CPU
Joldes et al., 2010 [257]	MI	FEM (NL-FEM) implemented on GPU	The human brain	Nonlinear elasticity	3.54s (3000 system time-step running) 19.95s (3000 system time-step running)	16,825 E – 2,693 N 125,292 E – 5,669 N	GPU NVIDIA CUDA Tesla C1060 (240 1.296GHz cores, 4GB High-Speed Memory)

Reference	Approach	Modelling methods	Soft-tissue Types	Tissue Behaviours	Computation Time/Speed	Geometry Discretization	Hardware Configurations
Wittek et al., 2010 [258]	MI	Total Lagrangian explicit dynamics FEM (TLED-FEM) implemented on GPU	The human brain	Nonlinear elasticity	<4s (deformation prediction)	18,000 N – 30,000 E – ~50,000 DOF	GPU NVIDIA CUDA Tesla C870 (128 600 MHz Cores, 1.5 GB Memory)
Peterlík et al., 2010 [259]	MD	Precomputation-based FEM (Pre-comp FEM) using radial basic functions (RBF)	The human liver	Nonlinear elasticity	0.54s 9.89s (stiffness and tangent stiffness matrix computing) 1 kHz (haptic feedback) 30 Hz (model rendering)	1,777 E – 501 N 10,270 E – 2011 N Surface triangle elements	AMD Opteron 2 GHz CPU, 8 GB RAM
Lapeer et al., 2010 [260]	MI	Total Lagrangian FEM (TL-FEM)	The human skin	Hyperelasticity (general polynomial, reduced polynomial and Ogden formulation)	>1 kHz (Haptic feedback)	100 E – 50,000 E	GPU
Marchesseau et al., 2010 [242]	MD	Multiplicative Jacobian Energy Decomposition FEM (MJED-FEM)	The human liver	Poro-hyperelasticity, Visco-hyperelasticity	13 FPS (model deformation)	20,700 E – 4300 N Tetrahedral elements	CPU
Courteuisse et al., 2010 [261]	MI	Linearized FEM (L-FEM)	The human cataract The human liver The brain tumor	Linear elasticity combined with a co-rotational method	1.4 FPS (model computing on CPU) 46.15 FPS (model computing on GPU) 64ms (model computing on GPU)	41,000 N Tetrahedral elements 3,874 N Tetrahedral elements	GPU
Turkiyyah et al., 2011 [262]	MD	Discontinuous basic function FEM (DBF-FEM)	The human skin	Linear elasticity	13.9ms (model computing and mesh updating)	31,008 N Surface triangle elements	CPU
Niroomandi et al., 2012 [243]	MD	Order reduction method (ORM) FEM	The human cornea The human liver	Nonlinear elasticity	20 Hz (model and graphic updating)	7,182 E – 8,514 N Hexahedral elements 10,519 E - 2853 N Tetrahedral elements	2 GHz CPU, 2 GB RAM
Wu et al., 2013 [3]	MD	Finite-element method (FEM)	The superficial fascia in a face	Nonlinear elasticity	NI	560 E – 1180 N 28,320 DOF	CPU
Morooka et al., 2013 [263]	MD	Precomputation-based FEM (Pre-comp FEM) using neuro networks	The phantom liver	NI	NI	15,616 E - 4,804 N	CPU
Mafi et al., 2014 [264]	MI	Element-by-element precondition conjugate gradient FEM (EbE PCG-FEM)	The human stomach	Linear elasticity	10 times faster than CPU for model computing	6361 E – 13,3784 E 1295 N – 25462 E	NVIDIA GTX 470
Courteuisse et al., 2014 [244]	MI	Precondition FEM (Pre-Cond FEM)	The heterogeneous soft-tissues	Linear elasticity combined with a co-rotational method	70 FPS (system iteration) 1 kHz (haptic feedback) 22ms (node adding or removing)	1,300 tetrahedral elements 150 contact points 3,874 N	256 core GPU

Reference	Approach	Modelling methods	Soft-tissue Types	Tissue Behaviours	Computation Time/Speed	Geometry Discretization	Hardware Configurations
Strbac et al., 2015 [265]	MI	Total Lagrangian explicit dynamic (TLED) FEM	A general cube mesh	Hyperelasticity (Neo-Hookean)	0.309s – 163.402s (one solution time-step)	125 E – 91,125 E	NVIDIA GTX460 GPU
Karami et al., 2017 [266]	MD	Finite-element modelling method (FEM)	The extraocular muscles (EOMs) in an eye	Linear elasticity	20ms (model deformation)	Eyeball: 8638 E – 1970 N Muscle: 2673E - 864N Tetrahedral elements	CPU
Martínez et al., 2017 [267]	MD	Precomputation-based FEM (Pre-Comp FEM) using artificial neuro networks	The human breast	Hyperelastic (Mooney-Rivlin)	<0.2 s (model compression)	313,000 E - 62,000 N Tetrahedral elements	2.6 GHz Intel (R) Xeon (R) CPU
Lorente et al., 2017 [268]	MD	Precomputation-based FEM (Pre-Comp FEM) using artificial neuro networks	The human liver	Nonlinear elasticity	2.89 s (model computing using machine learning) 51.63 s (model computing using FEM)	From 379,800 N to 420,690 N	3.4 GHz Intel Core i7, 8 GB RAM, OS X El Capitan
Tonutti et al., 2017 [269]	MD	Precomputation-based FEM (Pre-comp FEM) using artificial neuro networks and support vector regression	The brain tumour	Nonlinear elasticity	<10ms (model prediction using neural network)	6,442 N - 1,087 E Tetrahedral elements	Core i7 2.9 GHz CPU
Luboz et al., 2017 [270]	MD	Precomputation-based FEM (Pre-Comp FEM) using the reduced order modelling method	The butt area	Nonlinear elasticity	<1s (strain field computing)	27,649 E Hex-dominant elements	CPU

*N: Nodes

**NI: No Information

***DOF: Degree-of-Freedom

****E: Elements

Table 13. Classification of developed modelling methods for soft tissue deformations in real time: *meshfree-based techniques.*

Reference	Approach	Modelling methods	Soft-tissue Types	Tissue Behaviours	Computation Time/Speed	Geometry Discretization	Hardware Configurations
Nedel and Thalmann 1998 [271]	MD	Mass-spring system method (MSM)	The muscle	Linear elasticity	16 FPS (model deformation) 84 FPS (model deformation)	82 mass points 17 mass points	SGI Impact workstation, MIPS R10000 CPU
Monserrat et al., 2001 [272]	MD	Boundary element method (BEM)	The general cube mesh	Linear elasticity	15 Hz (model deformation)	< 150 N Surface triangle elements	R-4400 CPU, 64 MB RAM
Goto et al., 2002 [273]	MD	Statistical analysis method (SAM) - Muscle	The human face	NI	1 minute (facial feature detection)	NI	Pentium II, 333 MHz CPU
Bonamico et al., 2002 [274]	MD	Mesh Geometry VRML-like representation (VRML) & Radial Basis Function (RBF) - Muscle	The human face	Linear elasticity	475ms (facial deformation) 1,430ms (facial deformation)	1253 V - 2444 F 4152 V - 8126 F	Pentium II 450 MHz CPU, 128 MB RAM
Brown et al., 2002 [107]	MD	Mass-spring system method (MSM)	The blood vessel	Nonlinear viscoelasticity	24 FPS (system iteration) 6 FPS (system iteration)	216 N - 1440 E Surface triangle elements 8000 N - 66,120 E Surface triangle elements	Sun Ultra 60 Workstation 450 MHz CPU, 1GB RAM
Sorkine et al., 2004 [275]	MD	Laplacian surface deformation (LSD)	The face model	Linear elasticity	0.07 s (model solving)	~10,000 V Surface triangle elements	2.0 GHz Pentium IV CPU

Reference	Approach	Modelling methods	Soft-tissue Types	Tissue Behaviours	Computation Time/Speed	Geometry Discretization	Hardware Configurations
Chandrasiri et al., 2004 [46]	MD	Personal Facial Expression Space method (PEES) - Muscle	The human face	Linear elasticity	12 FPS (facial animation)	NI	1 GHz Athlon CPU
Mollemans et al., 2005 [276]	MD	Mass tensor method (MTM)	The cube The human face	Linear elasticity	From 24.57 s to 2.3 s	From 53,3380 N to 10,368 N Tetrahedral mesh	CPU
Chen et al., 2006 [277]	MD	Mass-spring system method (MSM) combined with quasi-static algorithm	The human brain	Linear elasticity	48 Hz – 3000 Hz (haptic feedback)	8000 N Surface triangle elements	SGI Prism Server 4 GPU, 8 CPU, 32 GB RAM
López-Cano et al., 2007 [278]	MI	Mass-spring system method (MSM)	The human inguinal region	Linear elasticity	73 FPS (system iteration)	4,891 V Surface triangle elements	GPU NVIDIA 6800, Pentium IV 3.0 GHz CPU, 1 GB RAM
Lim and De 2007 [279]	MD	Point collocation-based finite spheres (PCMFS)	The human liver	Nonlinear elasticity	1ms (model deformation)	1186 polygons Polygon elements	Intel Xeon 3.33 GHz CPU, NVIDIA Quadro4 XGL
Murai et al., 2010 [280]	MD	Inverse dynamic computation (IDC)	The human muscles	Linear elasticity	16ms (muscle tension estimation) 15 FPS (model rendering)	274 muscles	Intel Xeon 3.33 GHz CPU, 3.25 GB RAM, NVIDIA Quadro FX3700 GPU
Basafa and Farahmand 2011 [281]	MD	Mass-spring-damper method (MSD)	The cube model	Nonlinear viscoelasticity	5ms (model deformation) 150 Hz (haptic feedback) 30 Hz (model rendering)	96 N - 270 E Tetrahedral mesh 500 N Tetrahedral mesh	3.2 GHz Core Duo CPU, 1GB RAM
Wang et al., 2011 [282]	MD	Laplacian surface deformation (LSD)	The human nose	Linear elasticity	NI	NI	Windows 2000 or Windows XP, 512 MB RAM or 250 MB
Ho et al., 2012 [245]	MD	Mass-spring system method (MSM)	The human eardrum	Linear elasticity	1 kHz (haptic feedback) 30 Hz (model rendering)	917 E Surface triangle elements	Intel Core2 Q6600 CPU, NVIDIA GeForce 9600
Wan et al., 2012 [54]	MD	Radial basic function (RBF) & Geodesic distance - Muscle	The human face	Linear elasticity	0.0316 s (one system frame computing)	5272 V - 10,330 F Surface triangle elements	Intel Core 2 Duo E7200 2.53 GHz CPU, 2 GB RAM
Le, et al., 2013 [283]	MD	Thin-shell deformation method (TSD) - Muscle	The human face	Linear elasticity	73.8 FPM (facial animation) 164.3 FPM (facial animation)	40 markers 100 markers	Intel Xeon 2.4 GHz 16-Core CPU, NVIDIA Tesla C1060 240-Core GPU
Y. Zhang et al., 2014 [284]	MD	Elastic-plus-muscle-distribution-based (E+MD)	The facial muscles	Linear elasticity	NI	NI	NI
Weng et al., 2014 [285]	MD	Facial motion regression algorithm (FMR) - Muscle	The human face	NI	>200 FPS (graphic rendering on PC) 30 FPS (graphic rendering on mobile devices)	75 facial markers	Core i7 3.5 CPU Intel Atom 2.0 GHz CPU
Goulette and Chen 2015 [286]	MD	Hyper-elastic mass link method for FEM (HEML-FEM)	The cube model	Visco-hyperelasticity	4.02ms (one model computation iteration) 21.24ms (one model computation iteration)	4,430 E – 1,128 N Tetrahedral mesh 21,436 E – 5,591 N Tetrahedral mesh	Core 2 Duo 2.40 GHz CPU, 3.45 GB RAM
J. Zhang et al., 2016 [287]	MD	The time-saving volume-energy conserved ChainMail method (TSVE-Chainmail) method	The cube model	Nonlinear elasticity	30 Hz (model rendering)	NI	Core i7-4700 3.4 GHz CPU

Reference	Approach	Modelling methods	Soft-tissue Types	Tissue Behaviours	Computation Time/Speed	Geometry Discretization	Hardware Configurations
Woodward et al., 2017 [288]	MD	Radial basis function mapping approach (RBF) - Muscle	The human face	Linear elasticity	2 minutes (system initializing) Up to 30 Hz (Facial feature detection)	NI	NI
Zhou et al., 2018 [289]	MD	Marquardt radial basis meshless method (MRM)	The general cube model	Nonlinear elasticity	0.1509 s (model deformation)	121 nodes Tetrahedral mesh	Core i7-4790 3.60 GHz CPU, 8 GB RAM, Intel HD Graphics 4600 (64 MB)

Table 14. Classification of developed modelling methods for soft tissue deformations in real time: combination-based techniques.

Reference	Approach	Modelling methods	Soft-tissue Types	Tissue Behaviours	Computation Time/Speed	Geometry Discretization	Hardware Configurations
Cotin et al., 2000 [156]	MD	Precomputation-based FEM (Pre-comp FEM) & Mass Tensor Method (MTM) & Hybrid Modelling Method (HMM)	The blood vessel	Linear elasticity	40 Hz (model deformation) 500 Hz (haptic feedback)	760 Vertices – 4,000 Edges 8,000 tetrahedral elements	233 MHz DEC Alpha Workstation
Yarnitzky et al., 2006 [290]	MD	Dynamics-based & FEM	The foot soft-tissue	Linear elasticity	<25ms (model deformation)	100 nodes	1.6 GHz Dothan Pentium IV CPU, 1 GB RAM NI
Bensoussan et al., 2008 [158]	MD	Multi-cooperative methods (Multi-Corp)	NI	NI	NI	NI	NI
Zhu and Gu 2012 [157]	MD	Boundary element method (BEM)& Mass-spring system (MSM) & Particle surface interpolation (PSI)	The human liver	Linear elasticity with an extra mass-spring model	From 0.99ms to 4.17ms (model deformation)	From 200 to 1,200 nodes	2.26 GHz Pentium M CPU, GeForce 9650M GPU, 2 GB RAM

A2.3. Model development approaches

A2.3.1. Mesh-based modelling methods

Mesh-based modeling methods are grouped into four common computation strategies: the finite element modeling method (FEM), the pre-computation-based FEM, the formulation-adapted FEM, and the boundary element methods (Figure 85). Note that in this present review, “deformation models” terminology relate to soft-tissue models developed using a specific modeling method while “simulation models” terminology refers to numerical models in general meaning.

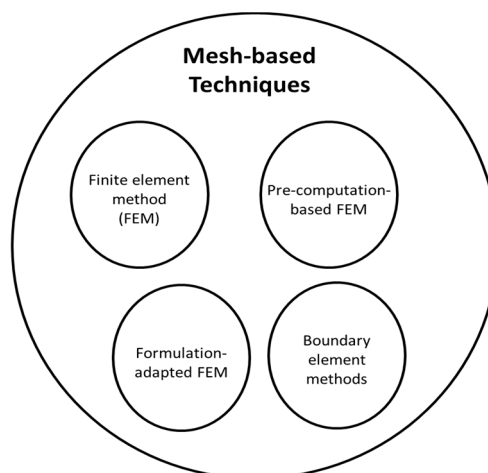


Figure 85. Overview of common computation strategies for mesh-based studies.

The finite element method (FEM) has been popularly employed in the literature despite of its very high computational cost. Deformable objects are geometrically meshed by a set of elementary components called finite elements. These elements are connected by nodes whose quantity defines the size of the FE model. Material properties are commonly assigned into each finite element. Then, the physical behavior of solid object deformations is described by a set of constitutive equations. Finally, the resolution of these equations on the nodes with prescribed boundary and loading conditions leads to the stress-strain relationships of the deformable objects. FEM provides a very high level of accuracy and realistic deformations in both linear and nonlinear cases. For example, Wu *et al.*, 2013 [3], modeled the facial muscles by FEM to animate the facial expressions. Each single muscle was considered as an incompressible and hyper-elastic material. Each muscle model includes 1,180 nodes and 28,320 DOF. Note that the computing time could not be achieved in real-time. Karami *et al.*, 2017, employed also the FEM for modeling the extraocular muscles (EOMs) in an eye to estimate the muscular activations and directions [266]. The eye-ball model includes 1,970 nodes and 8,638 elements. Each muscle model includes 1,100 nodes and 2,673 elements. The computation time needed to solve the model was 20ms.

The pre-computation-based FEM is the most popular variation of FEM. This method uses the relationship between the mechanical forces and the deformations pre-computed from the accurate FEM with full physical and biomechanical characteristics to train an approximate model. To achieve this goal, a database of the accurate FE simulation outcomes needs to be constructed a priori. The computational accuracy and speed of the simulated model depend on the types of employed approximate techniques such as linear/nonlinear regression functions and machine learning (ML). By using this strategy, Cotin *et al.*, 1999, developed a liver surgical simulation system [251]. Sedef *et al.*, 2006, provided a solution for real-time and realistic FEM for simulating viscoelastic tissue behavior in medical training based on the experimental data collected from a robotic tester [253]. Sela *et al.*, 2007, proposed an effective solution for dealing with the topological changes in cutting simulations [254]. Peterlik *et al.*, 2010, simulated the human liver with realistic haptic feedback and deformations embedded with both nonlinear geometric and material parameters [259]. Morooka *et al.*, 2013, designed a navigation system for the minimally invasive surgeries using a neural network model [263]. Martinez *et al.*, 2017, used the decision tree and two tree-based ensemble methods for simulating the breast compression [267]. Lorente *et al.*, 2017, applied decision trees, random forests, and extremely randomized trees models to simulate biomechanical behaviors of a human liver during the breathing action [268]. Tonutti *et al.*, 2017, also applied artificial neural networks (ANNs) and support vector regression (SVR) algorithms for learning the precomputed data from the FEM model of a human tumor [269]. Luboz *et al.*, 2017, used a set of pressure frames compressed into a small number of modes by proper orthogonal decomposition [270]. This method allows the summarized modes to be described by a linear set of scalar coefficients, and this reduced set of pressure map modes was then inputted to the FE to compute the strain field modes.

The formulation-adapted FEM has been developed by mathematically alternating the FEM formulations with the other modeling methods. One of them is called linearized FEM (L-FEM) in which the kinematic behavior of the simulated object is linearized to the first order of approximations during a specific timing period. Thus, the FEM model built from the reduced object kinematic is also simplified and executed much faster than the original one. Due to the simplification, the L-FEM is only suitable for modeling the soft-tissues with linear elastic materials. For instance, Berkley *et al.*, 2004, applied the L-FEM to the virtual suturing application [247]. Moreover, Audette *et al.*, 2004, divided a FEM model into multiple sub-meshes [252]. All sub-meshes were computed independently in parallel threads of a real-time

operating system to output the local deformations. Garcia *et al.*, 2008, presented another reduction method called matrix system reduction FEM (MSR-FEM) [255]. The method focused rather on computing the regions of interest than the whole model. The Order Reduction Method (ORM) was developed by Niroomandi *et al.*, 2012, to reduce the complex computation of nonlinear FEM for real-time simulations [243]. The total Lagrangian (TL) formulation was also applied in a FE model to improve the computation speed. Joldes *et al.*, 2008, used this approach to develop a FE model for an efficient hourglass control application [248]. A variation of this method, called total Lagrangian explicit dynamic FEM (TLED-FEM) was also developed by Miller *et al.*, 2007, for an image-guided surgery applications [249]. This method was also employed by Joldes *et al.*, 2009, to simulate the deformations of a human brain [248]. They all successfully improved both the sizes and computation speeds of the developed models. Another version of TL-FEM proposed by Marchesseau *et al.*, 2010, was called multiplicative Jacobian Energy Decomposition (MJED) FEM [242]. This approach optimizes the generation of stiffness matrix in TL-FEM to solve the linear system of equations during each iteration. Turkiyyah *et al.*, 2011, aimed at physically simulating the mesh cutting in real-time thank to the controlled discontinuities in the basic functions and the fast incremental methods for updating the global deformations [262]. Finally, element-by-element precondition conjugate gradient FEM (Ebe PCG-FEM) was developed by Mafi *et al.*, 2013, [264]. This method combined the FEM with a conjugate gradient method by alternating the mesh topological computation at run time by iterations. Thus, the developed model would be faster than the original one using FEM and required less system memories during execution. A new preconditioning technique (Pre-Cond FEM) was also proposed by Courtecuisse *et al.*, 2014, for improving the computational time of soft-tissue deformations [244]. This technique could simulate topologically changes and haptic feedbacks of homogeneous and heterogeneous materials in acceptable accuracy.

The boundary element methods are based on surface deformations to deduce the internal deformations in real-time. Monserrat *et al.*, 2001, developed a surgery simulation system using this method [272]. Compared with the FEM, the BEM only required the discretization of the object's surface so that it could provide an optimized, fast, and easy implementation. Another surface-based method for developing the soft-tissue models was called Laplacian surface deformation (LSD) was first proposed in Sorkine *et al.*, 2004 [275]. The method represented the object surface based on the Laplacian of the mesh. Wang *et al.*, 2011, also employed the LSD method for nose surgery in a complete surgical system for automatic individual prosthesis design [282]. Goto *et al.*, 2002, used statistical analysis method (SAM) for detecting features on the facial surface through 2D images, and then the detected features were mapped to a generic 3-D facial model for generating the expressions using the surface deformation method [273]. Moreover, the computation speed of the facial expression estimators was enhanced by using a scaling polygon mesh method based on iterative edge contractions by Bonamico *et al.*, 2002 [274]. Chandrasiri *et al.*, 2004, proposed a strategy for converting the acquired facial expressions to the MPEG-4 FAP [291], stream to deform the 3D surface facial models robustly and in real-time [46]. Wan *et al.*, 2012 [54], and Woodward *et al.*, 2017 [288], used the landmark-based and muscle-based facial expression estimation to animate the 3-D surface facial model. The used methods were radial basic function (RBF) and geodesic distance. Le *et al.*, 2013, took advantages of the thin-shell linear deformation model to reconstruct the facial pose via the facial marker displacements [283].

A2.3.2. Meshfree-based modelling methods

Compared to the mesh-based modeling methods, meshfree-based modeling methods uses discrete points for representing continuum, and it takes advantages of interpolation methods to solve the partial differential equations (PDEs) [157]. Thus, a simulated soft-tissue object is

commonly modeled as a distribution of discrete nodes inside to form a complete volumetric model. These nodes are embedded with a shape function to form the model's stiffness matrix and to describe biomechanical characteristics of the soft-tissue object [289]. In particular, this approach does not need to pre-process all cell elements to estimate the global deformations like mesh-based modeling methods do. Consequently, the meshfree-based modeling techniques are much faster than the mesh-based modeling strategies, and they can simulate large deformations in real-time. Because of these advantages, the meshfree-based modeling methods have been received much attentions from research community in the recent years. One of the most popular method using the meshfree-based strategy is the mass-spring system modeling (MSM) method. Nedel *et al.*, 1998, applied the MSM method to model the muscle deformations in real-time [271]. Brown *et al.*, 2002, applied MSM method for a surgical training system [107]. Chen *et al.*, 2006, also used the MSM for developing a deformable model for haptic surgery simulation [277]. The MSM was also applied to simulate the 3-D model of the human inguinal region by Lopez-Cano *et al.*, 2007 [278]. Ho *et al.*, 2012, developed a deformable tympanic membrane using the MSM method for simulating the real-time deformation and cutting in a virtual reality myringotomy simulator [245]. Another well-known method of meshfree-based method called Mass Tensor method (MTM) in which the modeled object is approximated into a tetrahedron mesh. Inside each tetrahedron in the MTM, the displacement vectors of four vertices are linearly interpolated into the displacement field of this tetrahedron [156]. The MTM was used by Mollemans *et al.*, 2005, to simulate the soft tissue deformations after bone displacement [276]. An improvement of MSM called Mass-spring-damper (MSD) modeling methods was proposed by Basafa *et al.*, 2010 [281]. The results illustrated that with a simple cube model including 96 nodes and 270 tetrahedrons the computation time was just about 0.005s for each step. Another improvement of MSM was developed by Goulette *et al.*, 2015, called Hyper-Elastic Mass Links (HEML) in which the forces at a specific node are considered as a sum of force functions from the neighboring nodes connected with it [286]. Experiments showed that with the 21,436-tetrahedron HEML model, the computation time was at 21.24ms corresponding with 47 fps. A different aspect for meshfree-based methods proposed by Lim and De, 2007, known as the point collocation-based method of finite sphere (PCMFS) [279]. The technique was based on the combination between the multi-resolution approaches and the fast analysis strategies for nonlinear deformations for the active regions where being contacted by the surgical tool-tip. A distinctive modeling method for meshfree-based method was inverse dynamic computation (IDC) proposed by Murai *et al.*, 2010, for the musculoskeletal system [280]. Zhang *et al.*, 2014, developed an elastic-plus-muscle-distribution-based (E+MD) to model the facial muscle distribution for generating the facial expressions in real-time [284]. Another method called the time-saving volume-energy conserved ChainMail (TSVE-ChainMail) was proposed by Zhang *et al.*, 2016 [287]. The method was developed from traditional ChainMail method in which the model is represented as a spring system. Zhou *et al.*, 2018, has also proposed a Marquardt radial basis meshless method (MRM) for the soft-tissue cutting [289]. In addition to these studies, it is important to note that a large range of soft-tissue models (brain, ligament, atrioventricular valves) were also developed using element-free Galerkin method and isogeometric method [292–296]. Due to the used keywords, this present review does not include these works. Thus, interested readers could use more specific keywords to get information about these methods.

A2.3.3. Hybrid modelling methods

Hybrid methods have been intensively investigated in the literature due to its cooperative functions which take advantages from multiple methods. For instance, although the mass-tensor method (MTM) is fast and suitable for simulation the soft-tissue deformation in real-time, it is still lacked the realistic biomechanical characteristics, especially when simulating the nonlinear materials. On the other hand, the FEM has realistic simulation of biomechanical behaviors of soft-tissues, but it has high computation cost. Additionally, the pre-computation-based methods (Pre-Comp FEM) have very high performances for simulating the soft-tissue deformations in

real-time based on the pre-computed data from the FEM, but they cannot handle the topological changes. Consequently, the combination between MTM, FEM, and Pre-Comp FEM can not only simulate the deformation in real-time but also handle cutting and tearing realistically with nonlinear materials. This approach was first developed by Cotin *et al.*, 2000 [156]. The result of this study showed that the update frequency was able to reach at 40 Hz with an MTM having 760 vertices and 4000 edges. Yarnitzky *et al.*, 2006, combined the physically kinematic model with the local FE model to estimate the stresses and deformations inside a plantar foot's soft-tissues during gait [290]. Bensoussan *et al.*, 2008 introduced a well-known SOFA framework supporting modularly and flexibly for biomedical researchers to develop new soft-tissue deformation models [158]. The framework was comprised of multiple modeling methods combined effectively to simulate the soft-tissues according to their requirement level of real-time constraints. Zhu and Gu, 2012, also applied multiple modeling methods to develop a hybrid deformable model for real-time surgical simulation [157]. Different cooperative components exist in the system such as boundary the element method (BEM), the mass-spring method (MSM), and a particle surface interpolation algorithm.

A2.4. Model implementation approaches

The model implementation (MI) approach mainly focusses on algorithmic implementation of soft-tissue models based on developed modeling methods onto a more powerful hardware configuration. This approach can improve the computational performance of the developed soft-tissue deformation models, even faster and more robust than the MD approach. In particular, MI approach mostly aims at finding more suitable programming algorithms to parallelize the execution functions of the soft-tissue deformation models onto a graphic processing unit (GPU) platform rather than onto a central processing unit (CPU) platform. Basically, GPUs are comprised of highly parallel architectures. Each separate GPU unit contains numerous processors and memory segmentations, and each of processor works independently on its own data distribution. Consequently, although the clock frequencies of GPUs are often smaller than CPUs, the overall computation speed of GPUs are much faster than CPUs, even when CPUs can be composed of multiple processing cores up to now. Furthermore, various programming frameworks supported for model implementations have been improved in an easier and flexible ways. Two classical interfaces have been employed for programming on GPUs have been OpenGL, application programming interfaces (APIs), DirectX, CUDA from NVIDIA, and CTM from ATI. These frameworks have been written in high-level C-programming language which bring many benefits for modelers to implement their developed methods executing on GPU effectively [250]. An analysis of GPU implementations for surgical simulations was reviewed by Sørensen *et al.*, 2006 [297]. They concluded that GPUs would become much powerful and cost-effective platforms for implementing the soft-tissue deformation models in real-time medical environments. However, to be able to achieve benefits from this implementation approach, the developed modeling methods must be compatible and be able to reconfigured with parallel computations [297]. The first model implementation strategy was proposed by Taylor *et al.*, 2008 [250]. The authors implemented a model using the total Lagrangian explicit dynamic (TLED) FEM onto a NVIDIA GeForce 7900 GT GPU platform, and the results showed that the computation speed of the implemented model was much faster than the CPU-implemented model. A human brain model using the TLED-FEM was also implemented on the NVIDIA Tesla C870 GPU platform by Wittek *et al.*, 2010, and the computational performance was also accelerated significantly [258]. For instance, with the brain model of 18,000 nodes and 30,000 elements (approximately 50,000 degrees of freedom) the average time for estimating the brain deformations was less than 4s when implemented on GPU, and the time implemented on the CPU platform was up to 40s. A model using the explicit FEM in a real-time skin simulator was also implemented on a GPU platform by Lapper *et al.*, 2010, and the simulation results could be accelerated to reach real-time goal [260]. Joldes *et al.*, 2010, employed a GPU platform using a programming guide

NVIDIA Compute Unified Device Architecture (CUDA) to speed up the TLED-FEM human brain model [257]. Strbac *et al.*, 2015, also employed the model using TLED-FEM onto multiple General-Purpose Graphics Processing Units (GPGPUs) to evaluate the efficiency of this implementation with the current commercial solutions [265]. The experimental evaluations showed that when the size of model increased from 125 to 91,125 elements, the computational time was from 1 s to 1h39min37s running on Abaqus commercial software, from 0.149s to 34.143s running on the most powerful GPU of GTX980. Courtecuisse *et al.*, 2010, proposed an implementation, called linearized FEM (L-FEM), which was the combination between the linear elastic material with a FEM [261]. Mafi *et al.*, 2013, deployed a model using element-by-element preconditioned conjugate gradients (EbE PCG) FEM method in the GTX470 GPU platform for speeding up the deformation computation in real-time [264]. The implicit time integration of a non-linear FEM on the GPU platform was also performed by Courtecuisse *et al.*, 2014 [244].

A2.5. Interaction devices

In addition to the model development methods and the implementation approaches, the interaction devices contribute significantly to the whole system accuracy and computational time. After user commands are transferred to the computer system through input devices, the computer system must execute the simulated model according to the commanded strategies. Once each simulation iteration is completed, the estimated feedbacks from the simulated model are transmitted to the user through the output devices. Consequently, the total accuracy of both input/output interaction devices and soft-tissue models must be at least equal to the desired accuracy tolerances in each medical application. Different interaction devices have been used in the reviewed studies. However, due to the focused objectives on developing the modeling methods, up to 42% of reviewed studies did not use interaction devices in their simulation systems. The interaction devices in the remained studies could be divided into different types: the virtual surgical instruments and force-feedback devices, the 3-D scanners, the biomechanical sensors, the PC's human interface devices, the 3-D viewers and the 2-D and 3-D optical cameras. The statistic distribution of used interaction devices is shown in Figure 86. It is clearly showed that the virtual surgical instruments have been popularly used with 19 studies, and the least used device was the 3-D viewers and the 2-D optical cameras with only 3 studies. The second most popular interaction devices are the force feedback devices which were found on 15 studies. Other remaining interaction devices have been utilized by only from 4 to 6 studies. In fact, most of studies taken advantages of the force-feedback devices always combined with the virtual surgical instruments for interacting with the simulated model. Moreover, other interaction devices have certainly used in computer systems, such as computer screens, computer keyboards, etc. are not deeply analyzed in this review due to their obviously contributions to the simulation system.

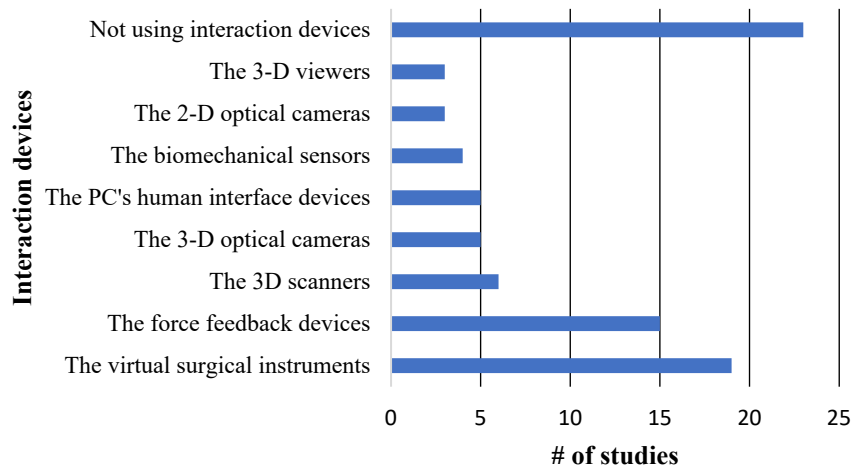


Figure 86. The distribution of using interaction devices in the chosen literature.

A2.5.1. Virtual surgical instruments and force-feedbacks devices

The virtual surgical tools have been widely combined with force-feedback devices to communicate between a user and a simulated model so that the simulation system could become more flexible and realistic. The functions of virtual surgical instruments are to transfer the controlled signals from external real devices to the simulated model and to feedback the calculated biomechanical parameters from the simulated model to the external haptic devices [107,247,277,281]. The speed of transmitting data from/to simulated models must be relatively high so that the visualization and haptic feedback can be simulated realistically [107,247]. Force-feedback devices are the input/output devices having a function of interfacing between a user and a virtual surgical tool. When received the interactions from the virtual surgical tool, the simulated model will react and calculate haptic forces during each simulation iteration. These computed haptic forces are finally feed-backed to the device through the virtual surgical tool [251–254,261,277,286]. As a result, the user will feel like they are interacting with a real soft-tissue when received the reaction forces from the force-feedback device [251,259]. For examples, Cotin *et al.*, 1999, used this combination in surgical simulation to provide haptic sensations for the surgeon [251]. Audette *et al.*, 2004, used a 7-degree-of-freedom (DOF) haptic device combined with a surgical tool whose tip fixed at the end of the haptic device to make the simulation system more realistically [252]. In the study of Chen *et al.*, 2006, a commercial PHANTOM haptic device with 3 DOFs force feedback and 6 DOF position and orientation was used for haptic surgery simulation [277]. Sela *et al.*, 2007, developed a new force-feedback system called SensAble™ PHANTOM Desktop™ haptic device [254]. The device was combined with a pen-sized handle, and they are all connected to a robot arm with flexible engine-forced joints to simulate a virtual surgical scalpel. Courtecuisse *et al.*, 2010, used a virtual laparoscopic grasper, which was managed by a Xitact IHP haptic device [261]. Peterlik *et al.*, 2010, used a virtual haptic interface point (HIP) controlled by a PHANTOM haptic device to calculate haptic forces reacted from a liver model based on displacements between the HIP's position and the 3D surface model of the liver [259].

A2.5.2. 3-D scanners

The 3-D scanners can be divided into two categories: structural and surface scanners. The most widely-used structural scanners in the literature have been CT and MRI. Cotin *et al.*, 1999, created anatomical model of a human liver from MRI images [251]. Marchesseu *et al.*, 2010, used CT images for creating the geometrical model of a liver [242]. Besides, 3-D surface

scanners were also used in the studies relating to surface-based soft-tissue modeling methods. The most popular surface-based scanners employed were laser scanners and ultrasonic scanners. They took advantages of measuring the time-of-flight of laser/ultrasound beams for estimating the distance between the laser/ultrasound sources and the object's surface. These scanners are fast and able to acquire the object surfaces in real-time. The laser scanners are much more accurate than ultrasound scanners, but lasers beams can be very harmful to the living soft-tissues during long acquisition period. For example, Monserrat *et al.*, 2001, employed the 3-D ultrasonic scanner (SAC GP10, Smart EDDY System, USA) for capturing the 3-D outside surface of the simulated object based on the boundary element modeling method [272]. The combination between surface scanners and structural scanners was also proven to be effective for accurate reconstructing both surface and structural details. Wang *et al.*, 2011, combined a 3-D laser scanner with the lateral X-ray scanners in their methods [282]. In fact, the 3-D laser images containing both 3-D geometrical point cloud and colors of the human face were transformed to the lateral X-ray image for comparing and cutting the nose part on the face. This combination provided high quality and patient-specific model of the human face appearance.

A2.5.3. Biomechanical sensors

Most biomechanical sensors used in soft-tissue modeling systems are electromagnetic sensors, force sensors, and electromyography sensors. Brown *et al.*, 2002, used an electromagnetic tracker (miniBRID of Ascension Technology Cooperation) to track behaviours of a real surgical forceps [107]. Sedef *et al.*, 2006, attached a force sensor (ATI Industrial Automation's Nano 17) at the end of a real surgical probe for measuring forces inside a surgical trocar so that the user could feel like being in a real minimally invasive surgery [253]. Yarnitzky *et al.*, 2006, employed ultra-thin force sensors arranged under the bony prominences of each foot for measuring the forces under the calcaneus, metatarsal heads, and phalanges in real-time in an application of monitoring foot's internal deformations under outside interactions [290]. Electromyography (EMG) was also used in the study of Murai *et al.*, 2010, for acquiring muscle tensions [280].

A2.5.4. PC's human interface devices

Some PC's human interface devices have also been used widely in the real-time soft-tissue simulation systems. They are all flexible and easy for user manipulation. For example, Lopez-Cano *et al.*, 2007, used a PC mouse as a surgical tool interacted with virtual pointer to deform the 3-D model of human inguinal region [278]. This configuration could simulate vertical and horizontal stretching deformations of the simulated model. In some simulated systems, the PC mouse could only be controlled to rotate the view angle of the simulated model [255]. Moreover, it could be used for drawing the cut shapes onto the surface model [282].

A2.5.5. 3-D viewers

The 3D viewers are the output interaction devices. They are composed of two separate high-resolution screens or two glasses attached together horizontally for displaying two different image frames to human eyes. Based on stereo geometrical model from human vision, the 3-D viewers can create depth perceptions for human brains, so when using the 3-D viewer for visualizing the simulated model, the graphic rendering can become more realistic. A classic 3-D viewer was presented by Brown *et al.* using the stereo glasses for enhancing the illusion of depth in the video frames [107]. In the visualization system, there were two image frames displayed: one image frame was colored in red, and the others was colored in cyan. The stereo

glasses included two different color filter glasses for the left and right lens, so at the same time each human eye would see a different image frame. Each pair of image frame was shifted horizontally for creating the depth information. A different 3-D viewing device called a 3-D stereo visor was used to visualize a simulated model in the virtual reality myringotomy simulation in the study of Ho *et al.*, 2012 [245]. This device included two different high-resolution screens for displaying two different image frames at the same time.

A2.5.6. 2-D and 3-D optical cameras

The 2-D optical cameras are the input interaction devices having the functions of acquiring 2-D image frames of object surfaces. In the application of facial expression recognition, Chandrasiri *et al.*, 2004, mounted a complementary metal oxide semiconductor (CMOS) camera to a headphone to capture 2-D color video frames of a user face [46]. An ordinary web camera available on a mobile device was also used by Weng *et al.*, 2014, in the application of real-time facial animations [285]. In particular, the offline 2-D images acquired from a camera were also analyzed for detecting the facial expressions and cloning them to other 2-D facial images in the study of Zhang *et al.*, 2014 [284]. One of the most drawback of 2-D optical cameras is the inability of reconstructing depth information from a single view of vision, so multiple optical cameras have been cooperated to form a 3-D optical camera system for detecting the 3-D data. A motion capture device was utilized to capture 3-D motions of a human during dynamic movements in the study of Murai *et al.*, 2010 [280]. A stereo optical motion capture was also combined with facial markers in the study of Wan *et al.*, 2012, for detecting facial animations [54]. Over 36 facial markers were detected and followed by mocap, and their motions were then converted to MPEG-4 standard's definition of facial animations. Woodward *et al.*, 2017, used an off-the-shelf stereo webcam for marker-based facial animation application [288]. Applied to minimally invasive surgeries in the study of Moroka *et al.*, 2013, the 3-D optical camera was integrated into a stereo endoscopy whose size was small enough to be used in restricted navigation spaces [263].

A2.6. System architectures

Computational approaches and interaction devices have been developed throughout the literature to improve computational accuracy and speed of soft-tissue deformation models, but they will not operate effectively and robustly in real-time if soft-tissue models and interaction devices are not well-cooperated in a system architecture. This section will synthesize system execution schemes and programming frameworks of the system architectures developed in the literature.

A2.6.1. System execution schemes

Cotin *et al.*, 1999, designed the first execution scheme called the distributed execution scheme in which a computer system and a Dec Alpha station were closely cooperated [251]. The computer system aimed to compute the haptic forces and exchange data with the haptic device while the Dec Alpha station visualized the deformations of this soft-tissue model in real-time. The communication environment between two computer systems was the Ethernet connection. Chen *et al.*, 2006, distributed a developed haptic surgery simulation onto two computing systems [277]. While an SGI Prism Visualization Server with 4 ATI FireGL GPUs covered graphical simulations, a windows computer system controlled the haptic devices and simulated the haptic feedbacks. The system could manage more than one simulated model by using a new peripheral protocol called Virtual Reality Peripheral Network (VRPN) developed by the University of North Carolina. Two workstation systems were also cooperated on a simulation system for the minimally invasive surgery proposed by Morooka *et al.*, 2013 [263]. All model

computations were performed by the first workstation while the second workstation only performs visualization of deformations and virtual tools in form of 3D stereo vision for improving depth sensations. Note that the limitation of transmission bandwidth led to the latency between the visualization force-feedback. To solve this issue, the multithread execution scheme was proposed, Brown *et al.*, which distributed two tasks of deformation visualization and collision detection on two different execution threads on a single dual-processor machine (Sun Ultra 60 with two 450 MHz processors) [107]. The system included three inter-cooperative simulators: a deformable object simulator, a tool simulator, and a collision detection module. The idea of multiple-thread executing on a single computer system was also applied by Sedef *et al.*, 2006, in a soft-tissue simulation system including a phantom haptic device, a computer screen, and a simulated model [253]. Peterlik *et al.*, 2010, also implemented two asynchronous computation threads executed on an AMD Opteron Processor 250 (2GHz) PC to operate a liver simulation system [259]. The main thread called haptic thread acquiring positions of a haptic device, detecting collisions, calculating haptic forces, and computing model's deformations. The simulation system designed in the study of Goulette *et al.*, 2015, also included multiple computation modules for accelerating the system execution [286]. Two modules were threaded to execute in parallel on an Intel Core 2 Duo at 2.40 GHz, 3.45 GB of RAM. As a result, the visual rate could be reached up to 47 fps. Audette *et al.* designed a surgical simulation system which included their own developed haptic device with 7 DOFs [252]. To control this haptic device, an intelligent I/O board, called the DAP5216a/626, operating individually with the computer system was proposed. Another scheme for system execution called a multi-model representation, was proposed by Bensoussan *et al.*, 2008, within the SOFA framework [158]. In this scheme, each soft-tissue simulation components could be represented by multiple modeling methods related to real-time deformation simulation, accurate collision detection, or realistic interaction computation. Finally, the task of programmers was to design a switcher to effectively alternate modeling methods according to each appropriate simulation issue.

A2.6.2. System framework

System development frameworks must be selected carefully so that the system could be developed both in high productivity and short time-to-market. Generally speaking, a software framework is a generalization software structure in which programmers can contribute their written codes to modify this structure to a specific application. Taking advantages of available configurations and pre-built libraries, simulation systems could be developed much more flexibly and faster than in traditional development procedures. The system frameworks can be divided into four groups: the input/output interaction frameworks, the graphic interaction frameworks, the modelling frameworks, and the hybrid frameworks. Regarding the input/output interaction frameworks, haptic devices have been commonly used in the literature, and they are often interfaced with computer systems by GHOST [253,259,277] and PHANTOM [277] input interaction framework. These input interaction frameworks are all free and open source. Moreover, other standard input interaction devices, such as keyboards, PC mouse, web cameras, microphones, etc., can also interface with computer systems through Application Programming Interfaces (APIs) supported by Microsoft Windows systems [46,278,282]. Regarding the graphic interaction frameworks, the most employed graphic framework was OpenGL in which 2-D and 3-D vector graphics can be rapidly rendered by GPU-platform boards. The rendering tasks can be executed on a separate computer system or on a local thread [157,245,252,253,259,271,277,278]. In particular, the OpenGL framework can be embedded in multiple types of operating systems such as Android, iOS, Linux, Windows, and various embedded operating systems. Moreover, it can also support for writing in multiple programming languages (e.g. C++, Python, C#, Cg, etc.). In addition, the CUDA™ graphic framework was developed by Nvidia Cooperation. There have been numerous studies using CUDA framework for implementing their simulation system on of-the-shelf graphic GPU

boards and achieving great benefits from parallel execution structure in real-time computations [250,257,258,261,264,265]. Another general graphic framework called OpenCL™ was also developed for flexible parallel implementation. An image processing framework called Virtual Place (AZE Co.) was also used for converting 3D deformations to stereo video frames for creating depth feeling on human visions [263]. Regarding the modelling frameworks, GHS3D [242], TetGen [281], and CDAJ-Modeler [263] were employed for generating mesh models from CT/MRI images. Additionally, Maxilim software could also support for boundary condition simulations [276]. Moreover, the CHOLMOD open source library could also be used for solving the linear systems in real-time [283]. Finally, the combinational frameworks have been developed to provide a much more flexible and multi-functional environment for developing a whole system. MATLAB is a powerful combinational framework including a facial analysis toolbox was used for facial expression analysis [46]; a toolbox called *iso2mesh* was used to generate a tetrahedral mesh of a brain and its tumor [269]; an artificial neuro network toolbox was employed to train the force-deformation data [269]; an optimization toolbox was used to obtain optimal parameters of simulation models that represent for simulated physical quantities; the OpenGL graphic library could be supported in the MATLAB environment for simulating interaction between soft-tissue model and surgical tools [281]. An Android programming platform was also used [285]. Additionally, supporting for FEM physical modelling, the Fast FE Modelling Software Platform [247] and GetFEM++ [259] were also employed. For parallel-threading, the RTAI-patched Linux was used for satisfying the hard-real-time requirements [252]. Other powerful and more multi-functional system frameworks are the SOFA [158,242–244] and CHAI3D [245] frameworks. In fact, they support various libraries and modules for implementing a complete simulation system including input/output interaction device drivers, geometrical model libraries, modelling algorithms libraries, and graphic rendering modules.

A2.7. Clinical validations

To translate the outcomes of the developed simulation systems into clinical routine practices, a systematic validation must be required. All validation efforts done in the literature were ordered as three validation levels: geometrical validations, model behavior validations, and user acceptability/safety validations.

The most important components in a simulation system are the geometrical and physical models. To accurately simulate the target soft-tissues, the geometrical appearances of both the simulated models and real objects must be well-fitted. Geometries at a specific state are commonly compared with the real *in vivo/in vitro* data acquired from a relatively accurate measurement method. CT/MRI were usually used to reconstruct the real 3-D geometrical models of the tissues of interest. Due to expensive processing time and resources, this scheme is just suitable for offline geometrical validation. For example, real data from patients under maxillofacial surgeries were stored and compared with the predicting appearances for improving the reproducibility capacity of the simulation system [276]. Moreover, the soft-tissue phantom could be used to give the validating data for the geometrical validation [269]. Regarding the model behaviour validations, the physical characteristics of the simulated models must be assessed with the real physical data at different deforming states. One of the most popular schemes is to use the calculated data from a standard commercial simulation software as baseline data. For example, a model using linear viscoelastic FEM was validated through a compression test solved by both the proposed computation approach and the ANSYS finite-element software package. Obtained results showed that the maximum error of displacement was less than 1% [253]. The ANASYS software package was also used for validating a model based on machine learning based FEM method [267]. Recently, the Marquardt-based model has also been validated by ANSYS in a liver simulation system [289]. The Abaqus software was also used for validation purpose [250,258,264]. Other used FE packages relate to MSC

NASTRAN 2003 which was used by Yarnitzky *et al.*, 2006 [290], and LS-DYNA™ which was used by Joldes *et al.*, 2009 [248]. Note that open source packages were also employed. The SOFA framework was the execution environment for performance evaluation between the developed method and the previous modelling methods [242]. In addition to geometrical and model validations, user acceptability/safety validation needs to be performed to evaluate the quality of interfaces between the system and its users in real clinical applications. One of the most popular schemes of this validation level is to collect feedbacks from experts and patients who have been experienced with the developed system. Ho *et al.*, 2012, validated their virtual reality myringotomy simulation system by a face-validity study in which a validated questionnaire was delivered to eight otolaryngologists and four senior otolaryngology residents for evaluating the system after a long period interaction with the simulator [245]. Tonutti *et al.*, 2017, conducted their validation procedure on surgeons with and without implementing the developed system, and the differences results were evaluated for proving the effectiveness of the simulation system [269].

A3. Discussion

A3.1. Computational approaches

The FE modelling methods and its variations have been commonly used for developing soft-tissue deformation models. However, the trade-off between system accuracy and computation speed remains a challenging issue. The FEM can simulate deformations for complex soft tissues [298,299]. In particular, commercial FE solver was commonly used to evaluate the accuracy of new FE algorithm [255,256,258,264]. However, using FE method, real-time requirements are only satisfied if being modelled with a smaller number of elements [3,266] or being accelerated by GPU-implementation [257,258]. In general, the computational cost of the FEM increases exponentially with the expansion of the number of nodes especially in case of simulation of the nonlinear materials. Consequently, various development techniques have been developed to improve their computation speeds. The most used approach was the pre-computation-based technique. This approach has been proved as a fast and robust technique for simulating deformations in real-time, but large deformations with complex material properties and constitutive laws and topological changes on the fly could not be handled during the system iterations because the trained model cannot be updated online [251,253,254,259,263,267–270]. Other FEM variations could significantly improve the performance of the FEM. One of them was the case of linearizing the kinematic of the simulated object [247] in which the model could be cut faster than the original model using FEM, but the speed was not fast enough for realistic visualization in medical applications. The idea of dividing a FEM mesh into multiple sub-meshes to be executed in parallel [252] could initially increase the computation speeds, but this method led to the limited number of threads being able to handle on a real-time operation system. Other development methods such as matrix system reduction (MSR-FEM) [255] and the order reduction method (ORM-FEM) [243] based on the reduction of the FEM's stiffness matrix could improve the processing time, but they just simulated small deformations. The total Lagrangian algorithm in conjunction with FEM [256] and its modification known as total Lagrangian explicit dynamic (TLED-FEM) [249] allowed element pre-computations, so less computation cost would be required for each time step. It is important to note that hyperelastic and viscoelastic constitutive models were implemented with explicit time integration schemes in a straightforward manner using home-made or commercial FE solvers. Moreover, Jacobian Energy Decomposition (MJED-FEM) [242] with implicit time integration schemes could be used to model hyperelastic, viscoelastic, and poro-elastic behaviours of the soft tissues. However, these methods could not handle interactions with other simulated objects and topological changes. The topological changes could be handle in the method proposed by Turkiyyard *et al.*, 2011 [262], but they could not solve effectively the cured cuts, partial cuts, and multiple cuts inside elements. More effectively for simulating the topological changes was Element-by-element precondition conjugate gradient FEM (EbE PCG-FEM) [264] method, but

it was not suitable for simulating the heterogeneous materials. Another potential method called pre-conditioning FEM (Pre-Cond FEM) [244] could solve this problem dramatically. It could both simulate the topological changes and the haptic feedback of homogeneous and heterogeneous materials with acceptable accuracy and real-time frame rates.

On the other hand, the meshfree-based techniques have been achieved great attention in the recent years. All the meshfree-based methods have been very fast and highly adaptive to topological changes, but they are less realistic than the mesh-based methods from biomechanical point of view. The most popular meshfree-based method was mass-spring system (MSM) [107,245,271,277,278]. This method could handle the deformations and topological changes, but it could not simulate accurately with nonlinear material characteristics. The improvements of MSM method were the mass-tensor method (MTM) [276] and mass-spring-damper (MSD) method [281]. They could handle the nonlinear material more effectively than the MSM due to the use of nonlinear mass springs in the method. Another improvement of MSM was Hyper-Elastic Mass Links (HEML) method [286]. This method could be considered for the compromising solution between the biomechanical accuracy and computation efficiency in real-time. A different technique was point collocation-based method of finite sphere (PCMFS) [300]. By just focused on the local region of interests, the simulation time could be decreased significantly, but the method for detecting the region of interest was still not defined effectively. More globally, the method called inverse dynamic computation (IDC) [280] could compute the muscle tensions based on the external data from sensors. Despite of the acceptable accuracy, the method could not analyze a single muscle. This idea could be found in the method called elastic-plus-muscle-distribution-based (E+MD) [284] in which the facial expression could be used for investigate the internal muscle tensions. A recent method called time-saving volume-energy conserved ChainMail (TSVE-ChainMail) [287] was powerful in handling both topological changes and simulating the isotropic, anisotropic, and heterogeneous materials at the real-time rate. At this stage, the real-time deformations could be achieved but the interactions among the simulated objects remains a difficult task. To overcome this drawback surface-based methods (e.g. boundary element method (BEM) [272], Laplacian surface deformation (LSD) [275], and Marquardt radial basis meshless method (MRM) [289]) have been proposed. These approaches could estimate the internal deformations based on the surface changes, and it could handle the interaction between modelled soft-tissues through surface interactions, but they were not able to simulate inhomogeneous materials, nonlinear elastics, and topological changes on the fly. In addition, model cutting and needle penetration issues were also studied using extended finite element method (XFEM) and meshfree-based approaches for soft tissues. In particular, extended finite element method (XFEM) has been used to study complex hard tissue (tooth [301], maxillary molar and endodontic cavities [302]) models with fracture and crack propagation behaviors and soft tissue (cornea [303]) models with cutting simulation. This open new avenue to model biological tissues with more complex interaction behaviors.

In addition, many studies have been conducted for the implemented model based on developed soft-tissue deformation method on to the GPU-parallel computing platform, and they can all achieve much better accelerations when compared with the conventional developing approach. Not all developed modeling methods are suitable for this approach, so the implicit time integration of nonlinear FEM method have been proven to be the most suitable for parallel implementation. Furthermore, the additional reconfigurations must be approved to the current methods to adapt with the implemented hardware platforms. When the model developing approach reaches its limitation, new implementation strategies will be necessary for accelerating their current computation performances.

It is important to note that the computational speed and resources depend on each particular application (e.g. surgical planning or surgical simulation) of soft-tissue deformation systems. For example, real-time soft tissue deformation behavior, high-speed device interaction and skill-based training ability could be more important criteria to be achieved for a computer-aided surgical simulation system. Besides, surgical planning system focuses on the whole workflow from data acquisition, pre-visualization of a specific surgical intervention and then predefine the optimal surgical steps.

Generally speaking, a large range of methods were developed to simulate the soft tissue deformations. Each method showed its robustness and accuracy for a specific case study. There are no universal methods and the selection of the methods depends directly on the application. It is important to note that real-time deformations with topological changes on the fly, and accurate object interactions remain challenging issues. One of potential solutions relates to the use of multiple modeling methods in a whole simulation system. However, an effective cooperation strategy should be established, and the requirement of advanced computational resources needs to be satisfied.

Finally, computation speed of a soft-tissue simulation system depends strongly on the use of constitutive behavior laws for modeling soft tissue physiology. Elastic, hyperelastic and viscoelastic laws were commonly used in the developed real-time simulation systems for upper/lower limb muscle, facial muscle, liver, and skin tissues. It is important to note that more complex constitutive laws such as electromechanical models could be used in general for modeling the skeletal muscle [70] or myocardium [304,305]. However, these complex models deal with additional computational need and requirements to reach a real-time ability for medical simulation systems. Linear and non-linear stress-strain relationships were described in the elastic material. Hyperelastic material was described using Noe-Hookean and Mooney-Rivlin formulations. It is important to note that some additional components were integrated into linear elastic law to improve the computation speed and model accuracy. For example, the combination of a linear elastic law with a co-rotational method was performed (Courtecuisse et al., 2014 [244]) or an extra mass-spring model was integrated into a linear elastic law (Zhu and Gu 2012 [157]). Regarding all analyzed simulation systems for soft-tissues, the most used law is the linear elastic one. The use of more complex laws (hyperelastic and viscoelastic) leads to a larger number of model parameters and of course computation speed.

A3.2. Interaction devices in real-time simulation systems

There have been various kinds of interaction devices contributing differently to the system's reality and computation performance. While the output interaction devices mainly provide realistic visualizations and reactions to human senses, the input interactions have the fundamental involvement to the computation performance, especially in both model accuracy and computation speed. Regarding the output interaction devices, the computer screens display appearances of simulated models and their deformations when interacted with virtual surgical instruments and/or other surrounding structures [245]. However, their lacking of depth information makes visualizations possible only in 2-D space. The 3-D viewers can complement this drawback. Like human visions, this interaction devices can create 3-D virtual sensation for human vision based on the differences between left and right scenes [107,245]. Even more realistically, the haptic feedback devices receive calculated haptic forces from simulated models to create collision feeling for human tactile [156,251,252,254,259,261,277,286]. Consequently, the cooperation between the 3-D viewers and the haptic feedback devices will become much more powerful in generating realistic sensations for human [107,245]. In particular, force-

feedback devices have been commonly used for many medical applications (e.g. surgical simulation, surgical trainings, or minimally invasive surgeries [252,277,286]). Force-feedback devices have been flexibly cooperated with various types of virtual surgical tools (e.g. virtual haptic interface point (HIP), the virtual blade, or the virtual scalpel). The most widely haptic device used in the literature is the SensAble™ PHANTOM Desktop™ haptic device, and its flexibilities are depended on the number of DOFs. It is important to note that to simulate force-feedbacks realistically, the haptic forces must be estimated and transferred to the force-feedback device at speeds from 500 Hz to 1,000 Hz, so this means that the computation speeds of simulated models must be faster than the those speeds [156,252]. Furthermore, a separate controller must be installed and executed one or multiple computer system to keep the real-time computation speed [253,259,261,277,286].

On the other hand, several biomechanical quantities have not been measured directly from the biomechanical sensors, so simulated models are often used to infer internal physical characteristics based on external knowledges. For example, in the case of musculoskeletal tracking, EMG sensors are often fused with musculoskeletal models for inferring individual muscle tensions according to the markers' motions, which are tracked by the 3-D optical camera system [280,290]. The soft-tissue physical parameters can also be inferred from soft-tissue deformation models by the movements of surface markers instead of direct measurements from the sensors. The surface makers are proven to be very robust and flexible for estimating outside deformations, but the limited number of markers being able to put on a soft-tissue surface leads to decreasing the estimated deformation resolutions, and so are the resultant calculations [54,263,288]. The 3-D scanners such as MRI/CT scanners [242,251,268] and 3-D ultrasonic scanners [272] have been employed for detail surface reconstruction in 3-D spaces, but their slow acquisition times (in case of MRI/CT scanners), lacking of surface characteristics (in case of 3-D laser scanners and ultrasonic scanners), and harmful infections to human health (in case of CT and laser scanners) make them not suitable for tracking external deformations of soft-tissue in real-time and in long-term use. This issue was initially resolved by the combination between the 2-D optical cameras and the X-ray images for adding more surface characteristics [282], but the appearances were static and could not estimate deformations on the fly. Consequently, other devices having the ability of acquiring both detail surface deformations in 3-D spaces and surface characteristics online are substantially required for improving computation speeds and model accuracy of soft-tissue simulation systems.

A3.3. Suitable execution schemes in simulation systems

To manage the data transmission from/to I/O interaction devices and to compute the simulated model in an optimal way, a suitable system execution scheme must be developed. There are two main system execution schemes. In the distribution-based scheme [251,263,277], system tasks are highly parallelized in multiple computing machines, which are interfaced through a limited-bandwidth and slow transmission environments. This scheme allows system tasks execute independently and take advantages of multiple computing hardware, but the problems appear when having delays in communication between multiple machines. Thus, data transmissions are still not fast enough for effectively communicating among multiple computer systems. This issue has been initially solved by numerous attempts such as high-speed Ethernets, high-speed data transmission protocols, etc., but they are not efficient enough for transmitting large information in real-time. In the multi-thread scheme [107,253,259,286], system tasks are executed on multiple computing threads. Because all threads are connected through a very high-speed internal bus, there is nearly delay in data transmission among threads. However, because the limitation of computation strength and memory capacity of a single thread inside a computer system, the simulation task(s) must be simplified and optimized to be able to execute on a single thread. This can be a challenging task for model developments and

implementations. Fortunately, this challenge can be easily resolved by the development of hardware technology with more threads integrated on a single CPU or even more CPU facilitated on a single computer system. In addition, cooperation of the two execution schemes was also found in the literature [158,252]. In this case, various types of data acquisition boards have been developed to fast manage the input/output data streams. These boards are designed in a mobile hardware and can be easily plugged in to a computing machine through a specific high-speed transmission channel and a software driver. Consequently, this system configuration can take advantages of both distribution-based and multithread-based execution schemes.

A3.4. Clinical validations

Generally speaking, the clinical validation is the final system development stage to determine whether the simulation system is acceptably suitable for clinical routine practices. Current clinical validation procedures were grouped into three levels: geometrical validations, model behavior validations, and user acceptability/safety validations. Regarding geometrical and model behavior validations, the validation data have been commonly acquired from standard simulation software, phantom soft-tissue organs, or post-operation data. There is a lack of *in vivo* data for accurately validation the simulation system in real medical environments. The use of accurate CT/MRI data is promising but this approach is not suitable for online validation. The use of standard simulation software to validate the physical behaviors of simulated models also faces some problems. It is important to note that most of soft tissue materials (e.g. muscle, fat, skin) are unavailable in these types of software. Thus, only simplified behaviors were validated with classical mechanical laws (e.g. linear elastic or hyperelastic laws). Consequently, more experimental protocols should be investigated to characterize the soft tissue behaviors and use them for enhancing model behavior validations. Finally, the user acceptability/safety validation was performed with the end users including patients, trainees, and experts through questionnaires. Note that this approach is relatively subjective and qualitative. In addition to these validation levels, system validation should be performed in which the whole system was evaluated rather than each system's components. This stage targets at analyzing system functions, system robustness, and system computation performances during short-term and/or long-term working durations. While the system functions are relatively easy to verify by comparing with the proposed development functions at the designing stage, the system robustness and computation performances must be tested after short-term and long-term working durations. Although this validation process is necessary for a stable and robust system, rarely studies in the literature conduct this validation.

A4. Conclusions

This appendix summarized the literature works about real-time soft-tissue deformation systems. Throughout this review, studies relating to real-time soft-tissues simulations have been analyzed according to four system engineering aspects: computational approaches, interaction devices, system architectures, and clinical validations. This review provides useful information to describe how each aspect has been developed and how they have been cooperated for both executing in real-time and keeping realistic behaviors of soft-tissues. By clearly analyzing advantages and drawbacks in each system development aspect, this review can be used as a reference for developing real-time soft-tissue simulation systems. The contents of this review was published in the Applied Bionics and Biomechanics journal (Q3, IF@2019=1.141) [103].

Appendix B

Cage-based Mesh Deformation Method

This thesis applied cage-based deformation method for parameterizing a generic skull model. The generic skull model (Figure 87a) supported by Free3D [204] had 6,122 vertices and 9,537 with full detail structures. The cage of the skull model (Figure A.1b) was generated from the skull's convex hull after isometrically remeshing and up-scaling. The number of the cage vertices was I , and the cage facets was K . The generic skull model must be totally surrounded by its cage (Figure A.1b c). The mean value coordinates [189] were employed to parameterize the generic skull model. After parameterized the vertices of the skull model ($V_{SM}[6,122 \times 3]$) could be controlled by the vertices of the skull's cage ($V_{SC}[I \times 3]$) and the mean coefficient matrix ($M[I \times 6,122]$) using Equation 36.

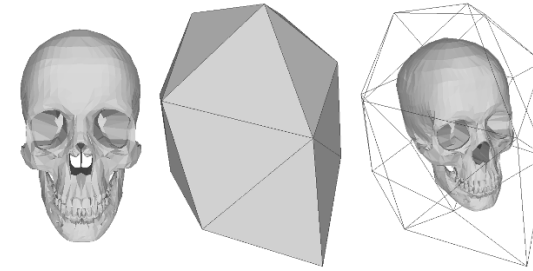
$$V_{SM} = M^T V_{SC} \quad (36)$$


Figure 87 consists of three 3D models arranged horizontally. Model (a) is a detailed, shaded skull mesh. Model (b) is a polyhedral cage structure. Model (c) shows the skull mesh from (a) positioned inside the cage structure from (b). Below each model is a label: (a) Generic Skull Model (6,122 V; 9,537 F), (b) Generic Skull Cage (I V, K F), and (c) Generic Skull Model (6,122 V; 9,537 F).

Figure 87. The generic skull model and its cage: (a) generic skull model, (b) Generic skull cage, and (c) Generic skull model inside its cage

Appendix C

Graphic User Interface (GUI) System

Design & development of GUI system are one of the most demanding, difficult, and expensive tasks for interactive system developments. The system GUI must be multifunctional and user-friendly to communicate between the user commands and system functions [306]. This chapter presents the GUI system of the main system. First, overall system execution workflow and GUI will be introduced. All system functions and their appropriate interfaces will then be presented. The GUI system was programmed in Visual Studio C++ 2015 based on the Qt 4.7.0 C++ framework [173]. Models and textures were rendered in the graphic card using VTK 7.1.1 embedded with Qt 4.7.0 [162]. Serious game scenes were rendered using Ogre3D game graphic engine [151].

C1. Overall system functions

The computer-aided clinical decision-support system for facial mimic rehabilitation was developed with the target functions of

- fast generating user-specific head, facial texture, skull, and muscle network of a target user based on user-specific data acquired from a Microsoft Kinect V2.0 sensor [105],
- real-time animating the generated models according to rigid head and non-rigid face movements of the tracked user,
- easily modifying different modes of model visualizations including
 - head model with/without face texture,
 - head model with skull and muscle network,
 - skull model with muscle network,
 - high-definition (HD) facial points,
 - and MPEG-4 [167] facial points,
- real-time measuring facial symmetries by the left/right mirror effects and displaying asymmetries in Hausdorff distance [166] color maps,
- real-time displaying facial mimics in different metrics:
 - MPEG-4 facial animation parameters (FAPs) [167],
 - facial action coding system (FACS) action units (AUs) [225],
 - action line lengths of linear facial muscles,
 - and horizontal/vertical lengths, and areas of orbicularis facial muscles,
- capturing static facial postures and showing the captured data with different visual modes,
- recording dynamic facial mimics and replaying the recorded data in time series with different visual modes,
- allowing the user to play facial rehabilitation serious games using his/her biomechanical head model,
- and organizing mimic analyzing and game playing data for each user.

C2. Overall system execution workflow and GUI

Figure 88 shows the overall system execution workflow of the GUI system. From the main interface, the user will be asked for *selecting a working directory* before going to the other system functions. In the main interface, the user can choose one of the four starting modes: *normal starting animation*, *quickly starting animation*, *playing dynamic data*, or *opening static data*. After selecting a starting mode, his/her biomechanical head model can be real-time animated on the main interface in a static or dynamic manner. While dynamically animating the user head model, the system can support *playing serious games*, *rendering*

mimic data, graphing bio-feedbacks, capturing static data, recording dynamic data, editing system parameters, and showing helps simultaneously. The system will continue its requested commands until being stopped by the user.

The main interface of the GUI system is shown in Figure 89. This interface includes a *biomechanical head model rendering window*, a *mimic rendering window*, a *biofeedback graphing window*, *system menu bars*, and *system toolbars*. From the menu file (Figure 90a) the user can access all basic system functions (start/stop rendering, open/play saved data, and edit system parameters). The view menu (Figure 90b) contains commands for controlling visual modes of all rendering windows and system interfaces. The help menu (Figure 90c) contains helps of facial mimics and facial muscle definitions. Moreover, the main interface also contains combo boxes for selecting *mirror effect rendering modes* (Figure 90c), *biomechanical head rendering modes* (Figure 90d), *mimic rendering modes* (Figure 90f), and *biofeedback graphing modes* (Figure 90g).

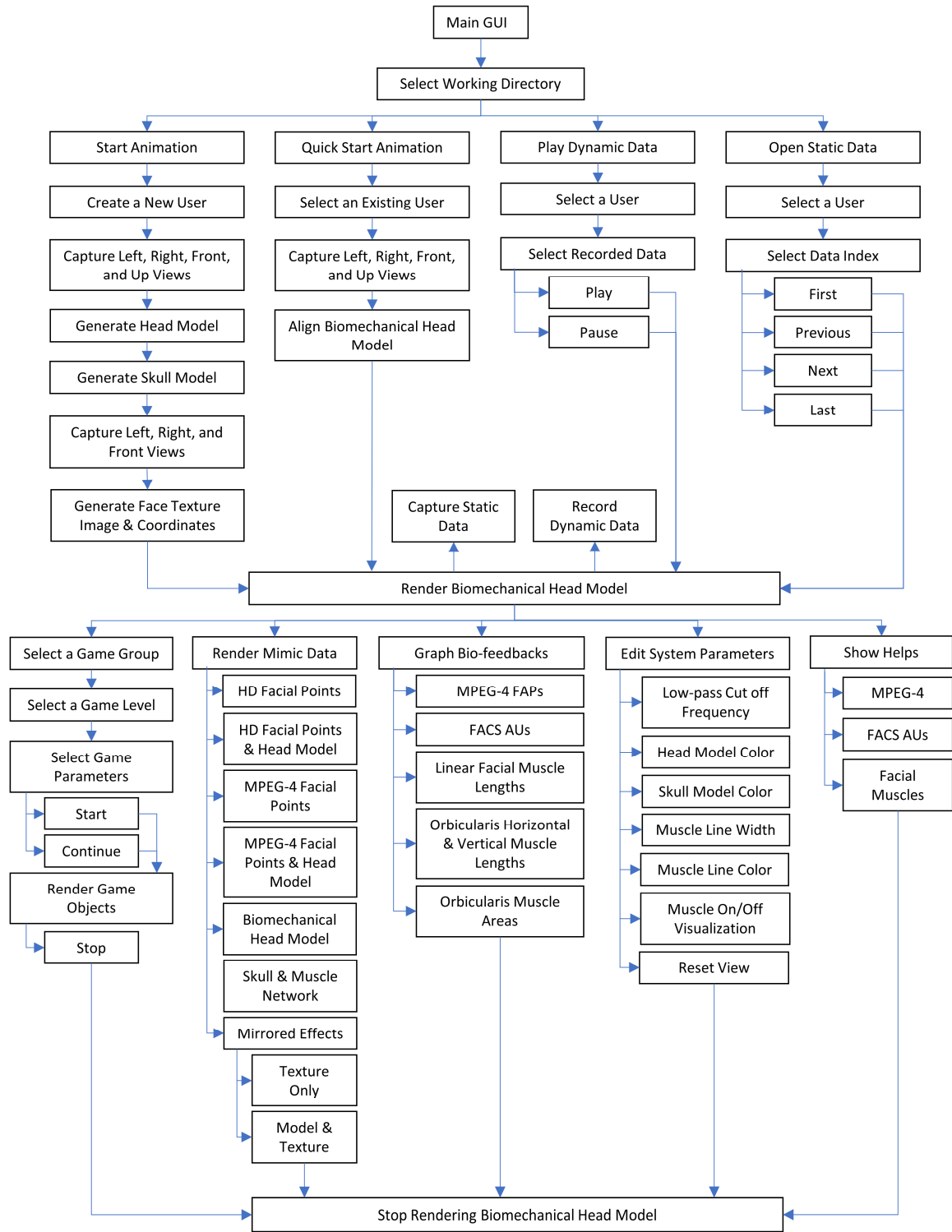


Figure 88. General execution workflow of the main system

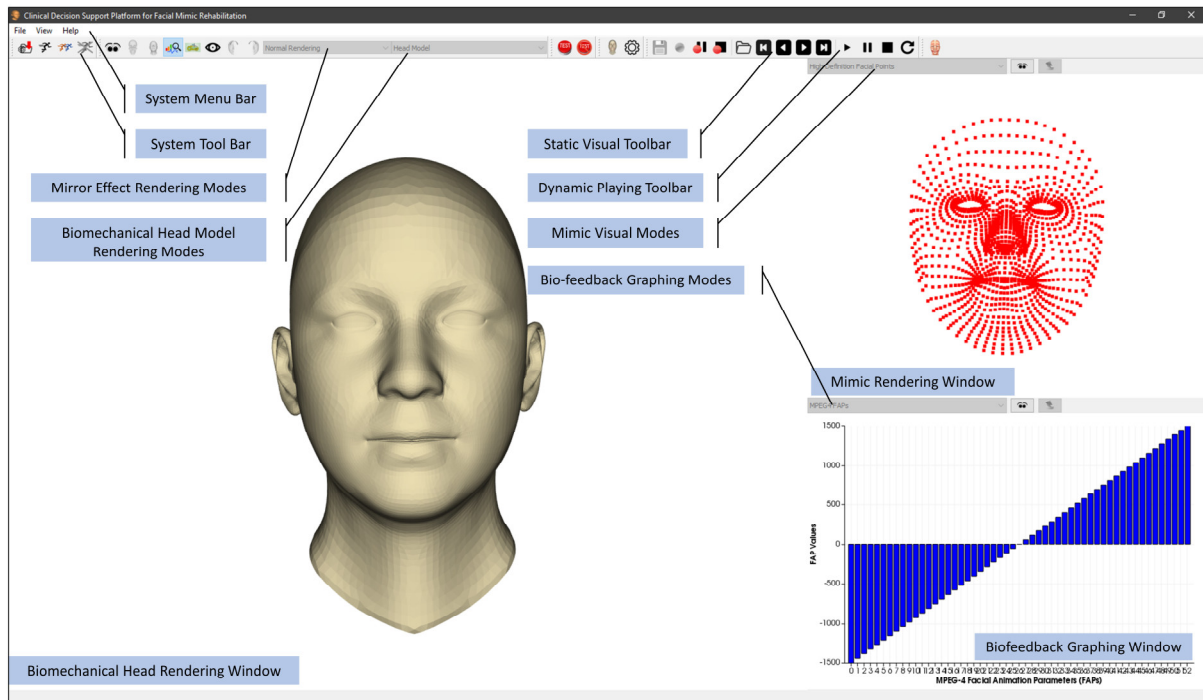


Figure 89. Main interface of the GUI system

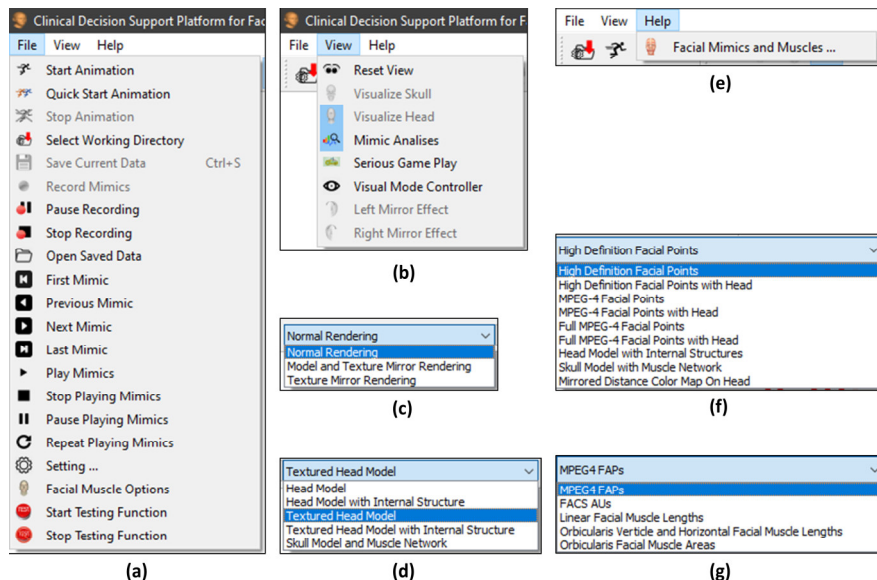


Figure 90. Submenus of the main interface: (a) file menu, (b) view menu, (c) biomechanical head rendering modes, (d) mirror effect rendering modes, (e) help menu, (f) mimic data rendering modes, and (g) biofeedback rendering modes.

C3. Biomechanical head rendering

Before rendering the biomechanical head model, the system will ask the user to select a working directory by using the “*select working directory*” command button (Figure 91a). The user can create a new directory or select an existing directory in the directory interface (Figure 91b). The structure of a selected working directory is shown in Figure 91c. After selecting a working directory, the user can also select one of four starting modes: *normal starting*, *quick starting*, *opening*, and *playing modes*.

In the normal starting mode, all subject-specific models will be generated for the current user before being rendered. First, the user presses the “*start animation*” command button (Figure 92a) to begin the model generation procedures. He/she will be asked for inputting a user identification (ID) in the “*New User Name*” sub-window (Figure 92b). Each user has his/her

personal data directories: dynamic data for saving dynamic recorded mimics, game play data for saving game results & parameters, model generation data for saving subject-specific models & texture, and static data for saving static captured mimics (Figure 92c). After input a new user ID, the Kinect sensor will then be opened, and an instruction screen will appeared for asking the user to raise his/her hand so that the system can recognize the current user (Figure 92d). The system will then instruct the detected user to move his/her head to acquire data in the left, right, up, and front views. These data will be used to generate subject-specific head, skull, and muscle network (Figure 92e). Moreover, the system will then generate the user’s face texture from his/her left, right, and front facial images (Figure 92f). After generating head, skull, muscle network, and face texture, they are rendered in the *biomechanical head render window* in different rendering modes: non-textured head model, non-textured head model with internal structures, textured head model, textured head model with internal structure, and skull model & muscle network (Figure 93).

In the quick starting mode, the subject-specific models and texture will be reused from the previous model and texture generation procedures. First, the use presses the *quick start animation* command button for starting the fast alignment procedure (Figure 94a). He/she will then select his/her user ID in the “*select one user*” sub-window (Figure 94b). The Kinect sensor will be opened, and an instruction scene will also be appeared to ask the user raise his/her hand for being detected (Figure 94c). The detected user will be instructed for capturing left, right, up, and front views to fast model alignments (Figure 94d). The aligned models can be animated according to head and facial mimics of the current user in different rendering modes (Figure 93).

In the starting modes of opening static data and playing dynamic data, the user can re-visualize his/her previous facial mimics in both static and dynamic manners. By pressing the “*open saved data*” command button (Figure 95a), a duo-textbox sub-window will be shown for selecting a user ID and his/her appropriate captured mimics (Figure 95b). By pressing the *play button* (Figure 95c), the user can choose a dynamic recorded data of each user for replaying (Figure 95d). The selected static/dynamic data can be rendered in the main interface in different rendering modes (Figure 93). In the static opening mode, the user can go to the next, previous, first, and last saved mimics by using the next, previous, first, and last command buttons (Figure 95a). In the dynamic playing mode, the user can pause, stop, or replay the recorded data by using the pause, stop, and replay command buttons (Figure 95c). Moreover, he/she can navigate the playing data to a specific frame by using the dynamic timing slider (Figure 95e).

As a result, while rendering the biomechanical head model, the system can support game playing, facial mimic analyzing, biofeedback graphing, data capturing/recording functions simultaneously in real-time.

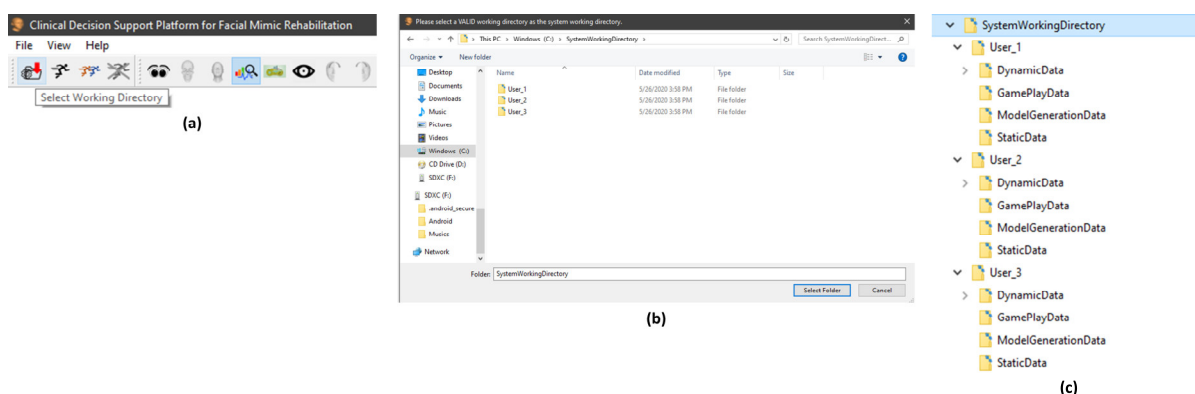


Figure 91. System working directory selections: (a) the “select working directory” command button, (b) the selecting/creating folder interface, (c) the structure of a working directory.

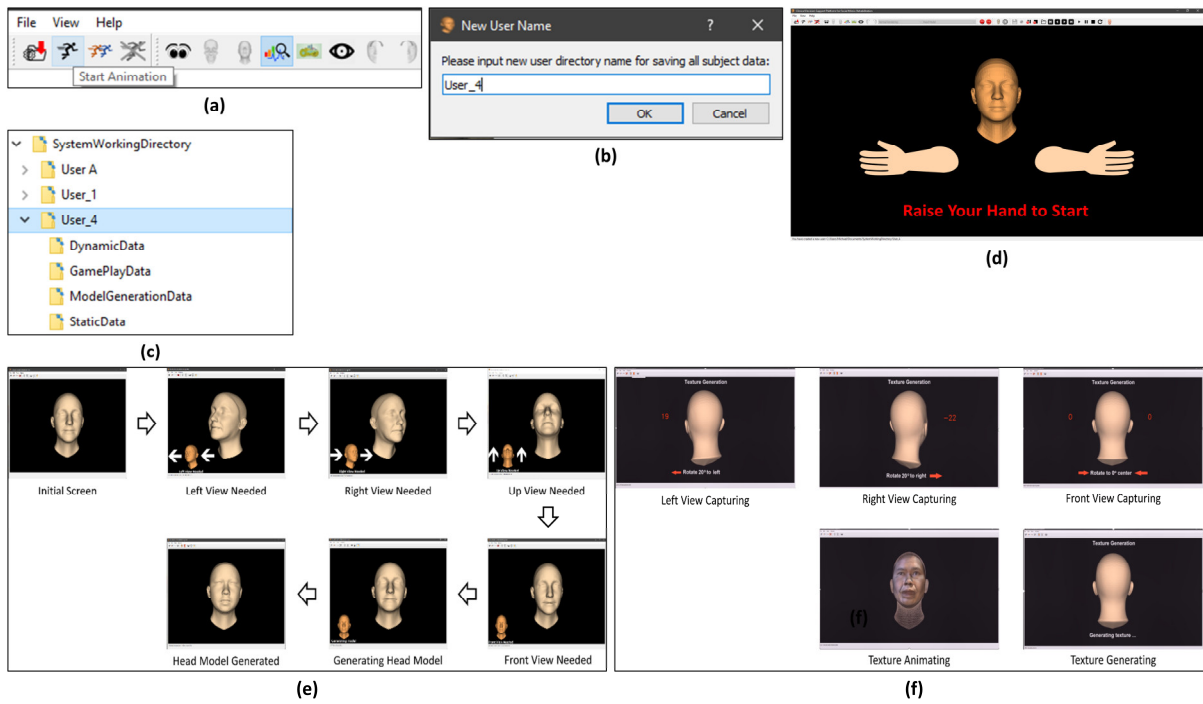


Figure 92. User supporting interfaces for generating new subject-specific models: (a) normal starting animation command, (b) user identification input interface, (c) folder structure of a new user directory, (d) user detection interface, (e) data capturing interface for model generations, and (f) data capturing interface for face texture generation

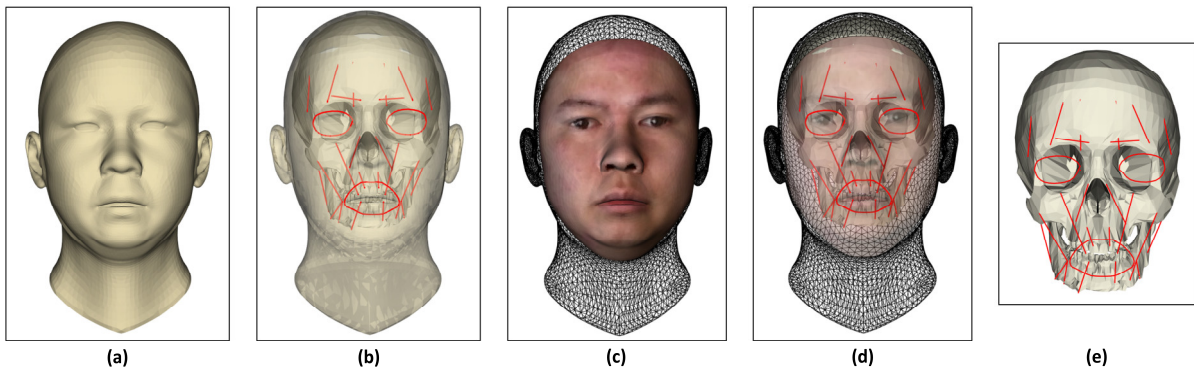


Figure 93. Rendering modes of biomechanical head models: (a) non-textured head model, (b) non-texture head model with internal structures, (c) textured head model, (d) textured head model with internal structures, and (e) skull model & muscle network.

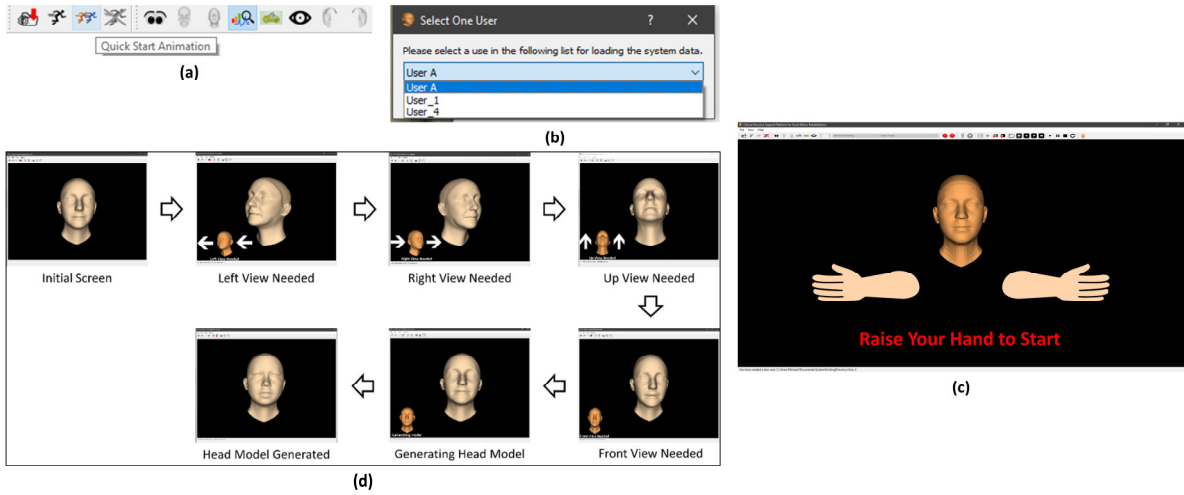


Figure 94. User supporting interfaces for fast aligning subject-specific models: (a) quick starting animation command, (b) user selection interface, (c) user detection interface and (d) data capturing interface for fast model alignments.

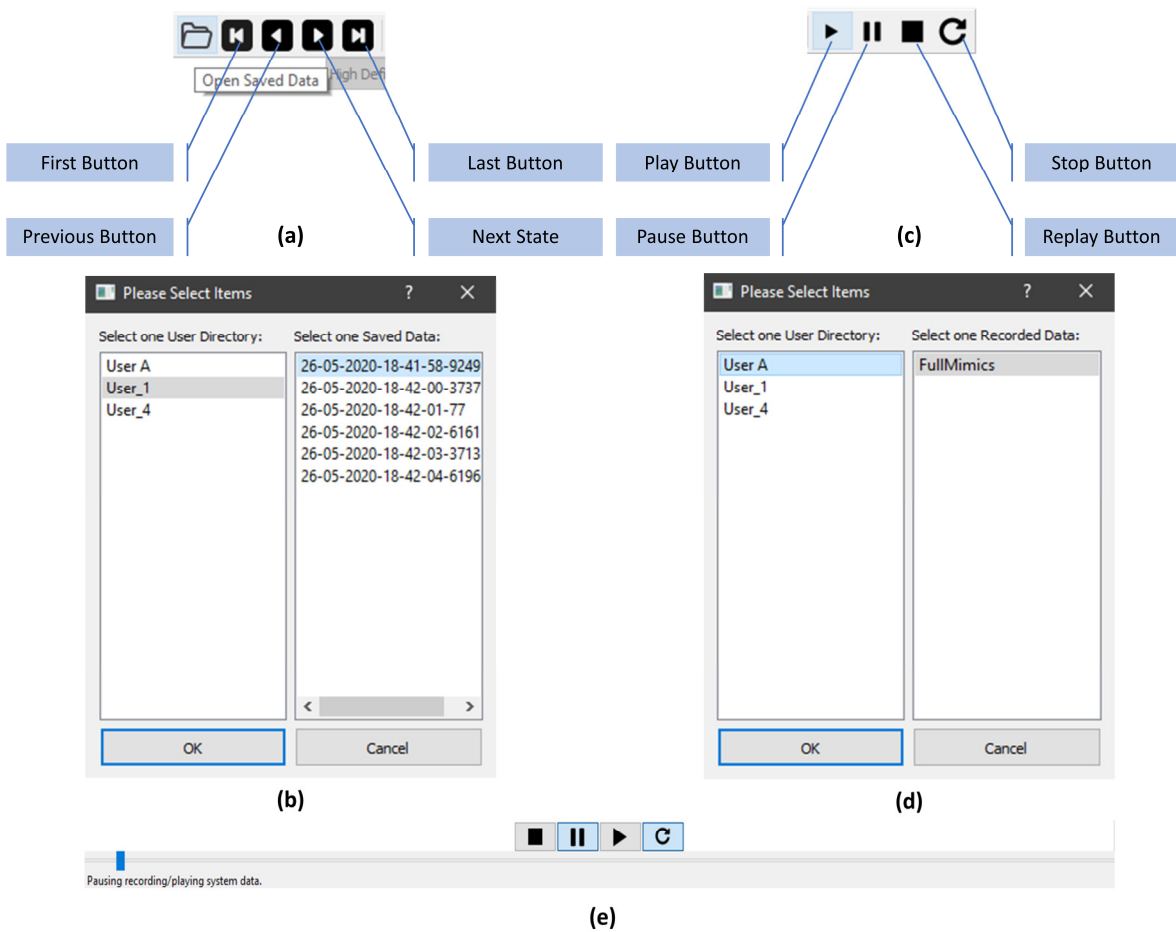


Figure 95. User supporting interfaces for opening static data: (a) open, first, previous, next, and last navigating buttons and (b) user & static data selection interface and for playing dynamic data: (c) play, pause, stop, and replay controlling buttons, (d) user & dynamic data selection interface, and (e) dynamic timing slider.

C4. Serious game playing

The rendered biomechanical head model can be used for playing serious games. The use will change the main interface to the game playing mode by pressing the *serious game play* command button (Figure 96a). The serious game play interfaces include game browser (Figure 96b) and game group

interface (Figure 96c). Game browser contains game browser scene and buttons for navigating to all game groups. Each game group interface supports game configuration and outcome interfaces for setting game parameters and analyzing game results. The detailed explanations of GUIs of all game levels of these game groups were presented in Chapter 6.

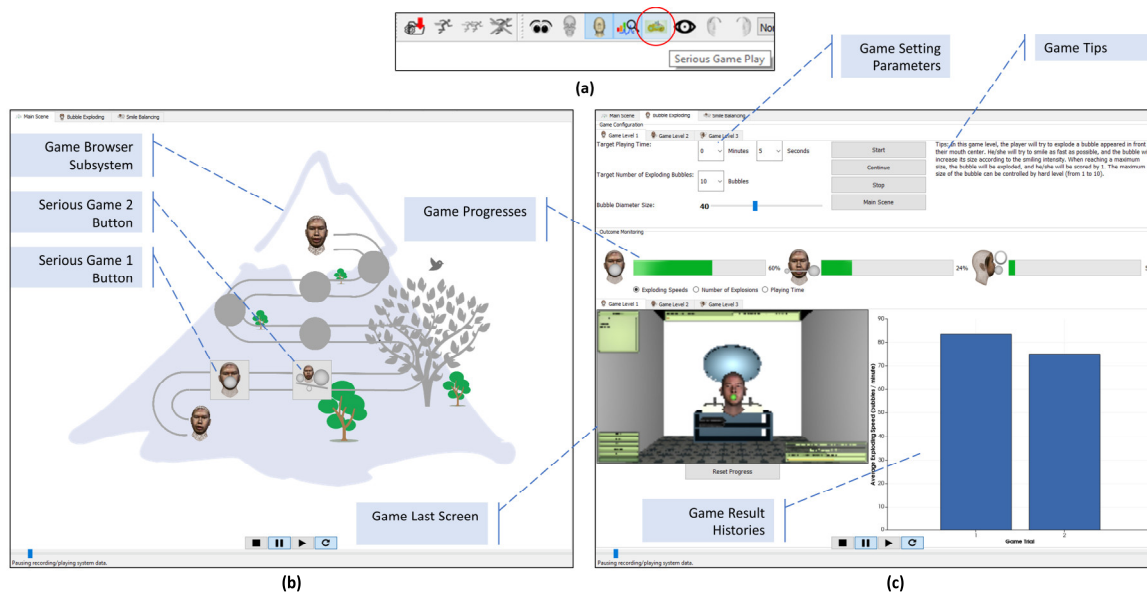


Figure 96. Serious game play interfaces: (a) serious game play command button, (b) game browser, and (c) game group interfaces.

C5. Mimic data & bio-feedback rendering

While computing the biomechanical head model, the user can select to analyze facial mimic in different modes: high definition facial points, high definition facial points with head model, MPEG-4 facial points, MPEG-4 facial points with head model, head model with internal structures, skull model with muscle network, and distance color maps for left/right mirror effects (Figure 97). Moreover, bio-feedbacks computed from the biomechanical can also be graphed and visualized in real-time. The bio-feedbacks information includes MPEG-4 FAPs, FACS AUs, linear facial muscle lengths, orbicularis horizontal & vertical muscle lengths, and orbicularis muscle areas (Figure 98).

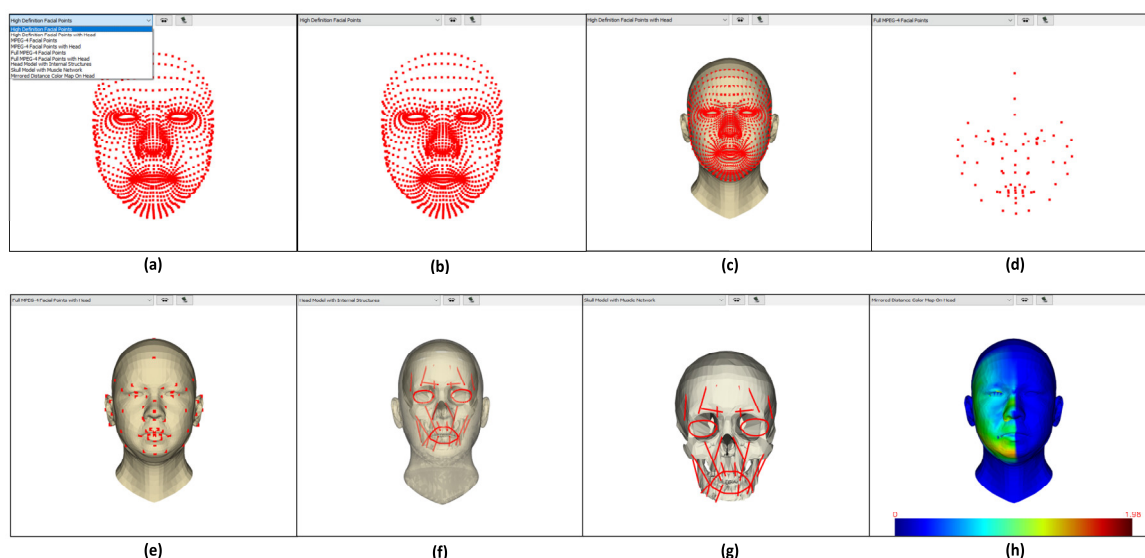


Figure 97. Rendering modes of facial mimics: (a) facial mimic rendering modes, (b) high definition facial points, (c) high definition facial points with head model, (d) MPEG-4 facial points, (e) MPEG-4 facial points with head models, (f) head model with internal structures, (g) skull model with muscle network, and (h) distance color maps for left/right mirror effects.

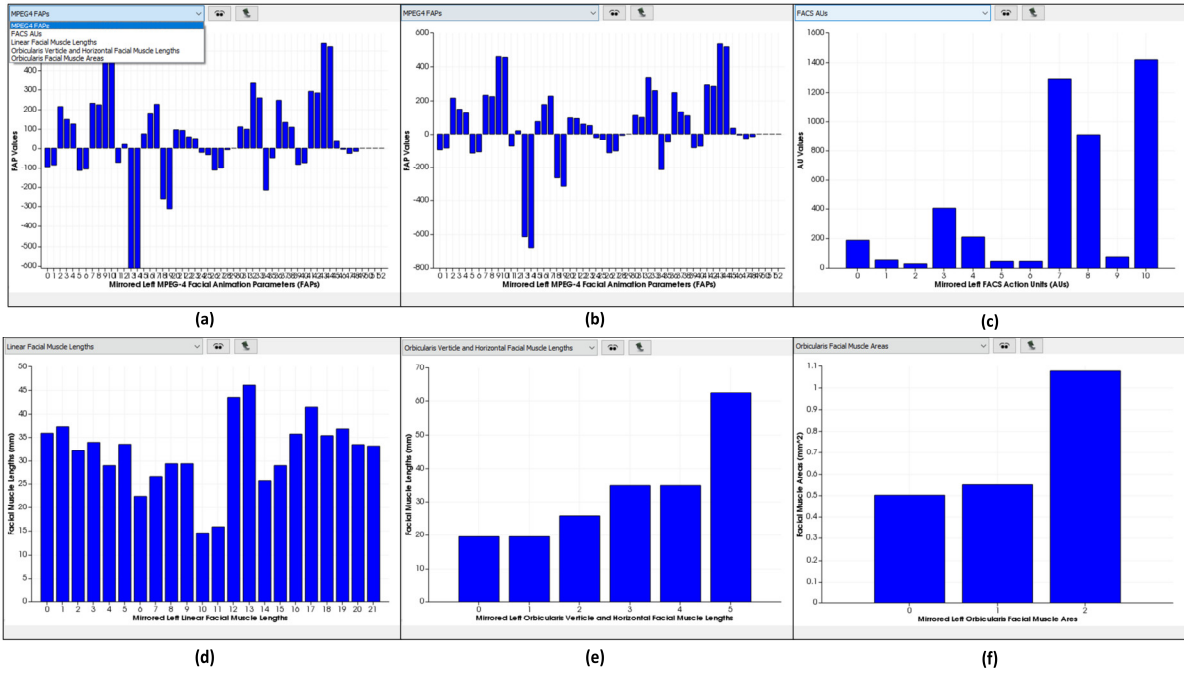


Figure 98. Graphing modes of bio-feedbacks: (a) bio-feedback graphing modes, (b) MPEG-4 FAPs, (c) FACS AUs, (d) linear facial muscle lengths, (e) orbicularis horizontal and vertical muscle lengths, and (f) orbicularis muscle areas.

C6. Data capturing & recording

The animated biomechanical head model can be captured as static data or recorded as dynamic data by using data capturing & recording command button (Figure 99a). For each time of pressing the data capturing button, the current mimic will be automatically saved on the static data folder of the current user directory and grouped as the current system time (Figure 99b). The captured data include subject-specific models, muscle attachment & insertion points, muscle lengths & areas, MPEG-4 FAPs, and FACS AUs. When pressing the data recording button, all above mimic information will be recorded as time serial data and saved to the dynamic data folder of the current user directory (Figure 99b).

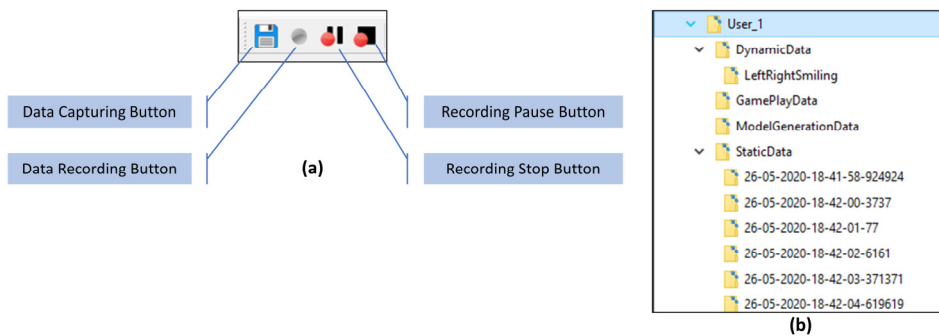


Figure 99. Data capturing & recording interfaces: (a) data capturing & recording command buttons and (b) data organization structure of the captured & recorded data.

C7. System parameter setting

During system executions, the user can edit system parameters and facial muscles through setting interfaces as shown in Figure 100. System parameters can be set up by pressing the *system parameter setting button* (Figure 100f). These parameters are grouped into main system, animation screen, mimic screen, and plotting screen parameters, and they can be accessed through appropriate window tabs. The user can set the *low-pass cut off frequency* for the model computation system (Figure 100a). *Background color*, *head model color*, *skull model color*, *muscle line color*, *muscle point color*, *muscle line width*, *muscle point size*, *head rendering subdivision value*, and *head opacity percentage* of the biomechanical

head rendering window can be set in the animation screen parameters (Figure 100b). The background color, facial point color, and facial point size of the facial mimic rendering window can be set up in the mimic screen parameter tab (Figure 100c). Moreover, background and chart color of the graphing rendering window can be set up in the plotting screen parameter tab (Figure 100d). Facial muscles can also be edited for being visible/invisible by checking/unchecking appropriate muscle names in the facial muscle selection interface through the facial muscle setting button (Figure 100f).

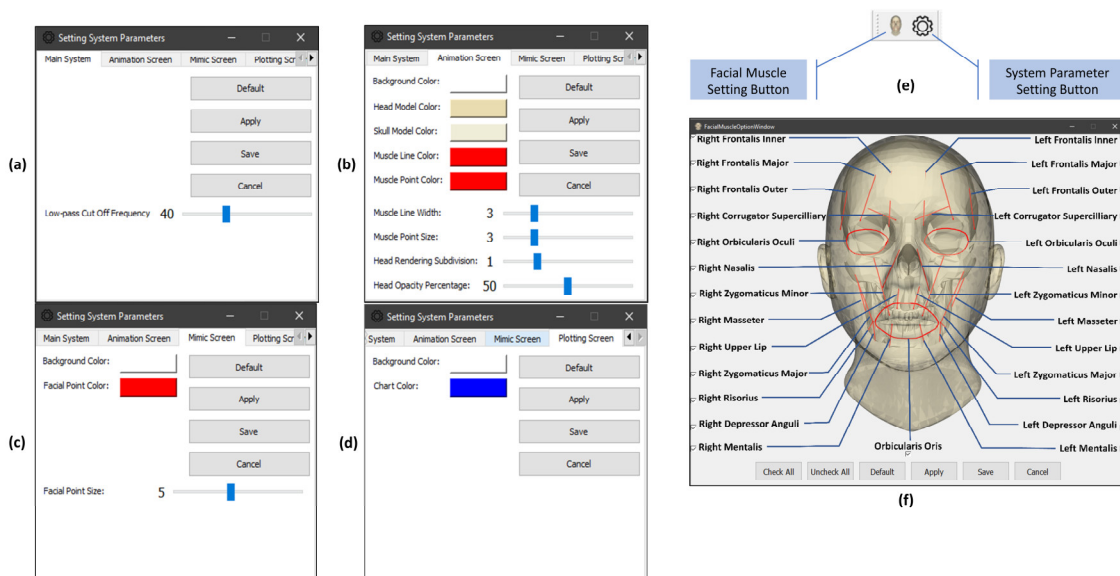


Figure 100. System and muscle setting interfaces: (a) setting parameters for main system, (b) setting parameters for the biomechanical head rendering window, (c) setting parameters for the mimic analyzing window, (d) setting parameters for the biofeedback graphing window, (e) system & facial muscle setting command buttons, and (f) facial muscle setting interface.

C8. User help visualizing

In the user help interfaces, the system can support to the user definitions of MPEG-4 facial animation parameters (Figure 101a), definitions of FACS action units (Figure 101b), and definitions of major facial muscles (Figure 101d) by clicking the *facial mimic and muscle* command button (Figure 101c).

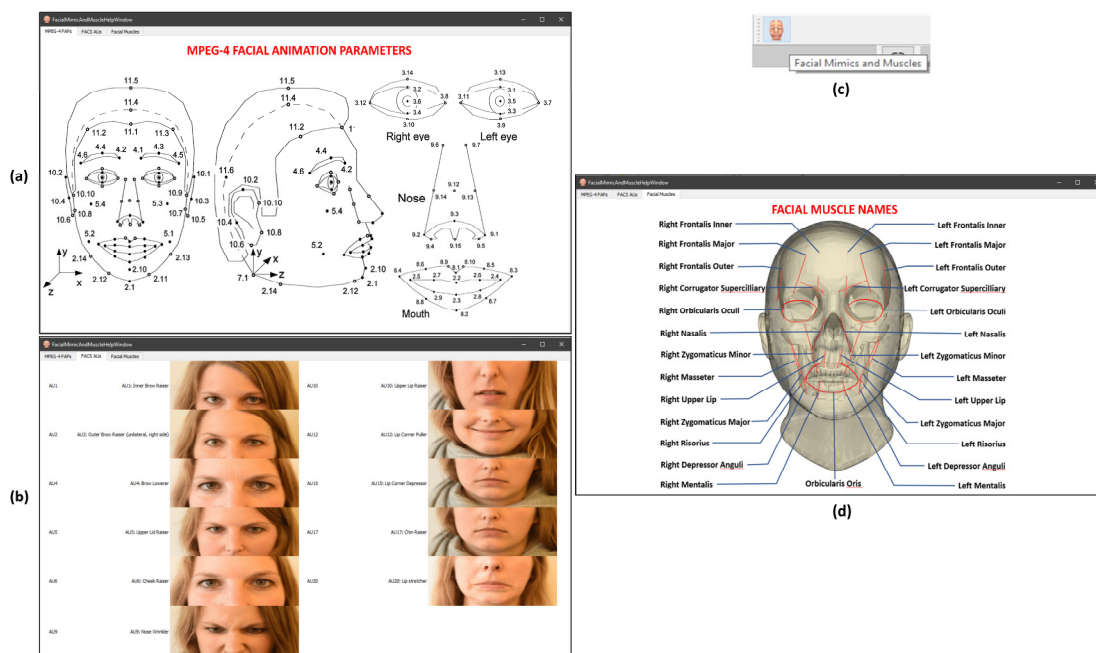


Figure 101. User help interfaces: (a) MPEG-4 facial parameter definitions, (b) FACS Action Unit definitions, and (c) facial muscle definitions.

C9. Discussion and conclusions

GUIs are important to design and develop interactive computer-aided systems. They provide interaction interfaces between the human commands and the system functions. This chapter presented designs and interfaces of the GUI system in the context of the whole SoS-based system. Based on the concept of SoS, this GUI system was designed to be executed both independently and cooperatively with other systems to optimize interfacing between the main system and users.

In particular, by executed in parallel with the data acquisition subsystem, the user instruction interfaces can provide interactive feedbacks to the user for self-controlling his/her head positions & orientations during data capturing processes. Moreover, the GUI system can provide a multifunctional model animation. After selecting a working directory, the user can either create his/her subject-specific models in the normal starting mode or reuse the models of an existing user in the quick starting mode. Consequently, a clinician can use his/her facial mimics to control subject-specific models of his/her patient for designing rehabilitation exercises. In the replaying modes, the user can easily select and replay his/her own or another one's recorded mimics in the static/dynamic manner. All data types of the replayed mimics can easily be visualized and navigated to different time point using the navigation buttons. During real-time biomechanical head rendering, the system can simultaneously support capturing/recording facial mimic data, playing serious games, visualizing facial mimics, graphing bio-feedbacks, editing system parameters, and showing helps. Each system function also provides multiple execution modes for adapted with user's requirements. For example, the facial mimic rendering and bio-feedback graphing interfaces can support multiple types of mimic representations, so multi-quantitative facial analyses could be conducted on the current facial mimics. All types of data relating to current facial mimics can also be stored in both static and dynamic manners for the current user using the capturing/recording interfaces. Last but not least, the GUI system can also provide the user help interface supporting for facial muscle and facial mimic educations during rehabilitation treatments.

In perspective, the user acceptability validation needs to be conducted by both facial palsy patients and clinical experts for improving the GUI designs and the system reliabilities. User evaluation interfaces will also be integrated as questionnaires into the GUI system for the user acceptability and system reliability validations. A user management interfaces with Internet connections will be developed for storing and analyzing patient-specific data in cloud computing platforms.