



HAL
open science

Automatic determination of optimal viewing angle for the coronary lesion observation in rotationnal X-ray angiography

Hélène Feuillâtre

► **To cite this version:**

Hélène Feuillâtre. Automatic determination of optimal viewing angle for the coronary lesion observation in rotationnal X-ray angiography. Signal and Image processing. Université de Rennes, 2016. English. NNT: 2016REN1S039 . tel-01427735

HAL Id: tel-01427735

<https://theses.hal.science/tel-01427735>

Submitted on 6 Jan 2017

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.

THÈSE / UNIVERSITÉ DE RENNES 1

sous le sceau de l'Université Bretagne Loire

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Traitement du Signal et Télécommunications

Ecole doctorale MATISSE

présentée par

Hélène Feuillâtre

Préparée à l'unité de recherche LTSI – INSERM U1099

Laboratoire Traitement du Signal et de l'Image

ISTIC UFR Electronique et Informatique

**Détermination
automatique de
l'incidence optimale
pour l'observation
des lésions
coronaires en
imagerie rotationnelle
R-X.**

**Thèse à soutenir à Rennes
le 10 juin 2016**

devant le jury composé de :

Su RUAN

PU, LITIS, Université de Rouen / *rapporteur*

David SARRUT

DR, CREATIS, INSA Lyon / *rapporteur*

Valérie BURDIN

P, LaTIM, TELECOM Bretagne / *examineur*

Christophe LOHOU

PU, ISIT, Université d'Auvergne / *examineur*

Christine TOUMOULIN

MCU-HDR, LTSI, Université de Rennes 1 / *co-directeur de thèse*

Jean-Claude NUNES

MCU, LTSI, Université de Rennes 1 / *co-directeur de thèse*

TABLE DES MATIERES

Chapitre 1 : Problématique	1
1.1. Contexte médical.....	2
1.1.1. Système cardiovasculaire	2
1.1.2. Pathologies	10
1.1.3. Diagnostics.....	12
1.1.4. Traitements	14
1.1.5. Modalité d'acquisition.....	16
1.2. Motivation de la thèse	19
1.2.1. Limitations et difficultés liées à l'angiographie RX.....	19
1.2.2. Projet	21
1.2.3. Reconstruction 3D des artères coronaires	22
1.2.4. Données : séquence 3D+T d'arbres coronaires.....	25
1.2.4.1. Définitions : Notions de graphe et d'arbre.....	25
1.2.4.2. Obtention de la séquence d'arbres coronaires 3D.....	27
1.2.4.3. Représentation des données sous forme d'arbres	30
1.2.5. Contribution	36
Chapitre 2 : Etat de l'Art	39
2.1. Appariement d'arbres vasculaires.....	40
2.1.1. Algorithme exact	41
2.1.2. Algorithme inexact	41
2.1.3. Applications aux artères coronaires.....	46
2.2. Problème de la clique maximum (MCP).....	47
2.2.1. Algorithmes exacts	48
2.2.2. Approche heuristique.....	50
2.2.3. Maximum weight clique problem (MWCP).....	50
2.3. Recherche d'incidences optimales.....	52

Chapitre 3 : Mise en correspondance d'artères coronaires.....	57
3.1. Matching of two vascular trees.....	60
3.1.1. Hierarchical tree approach.....	61
3.1.2. Association graph approach.....	63
3.1.2.1. Association graph building.....	63
3.1.2.2. Maximum clique extraction.....	66
3.1.3. Adaptation to our coronary matching problem.....	67
3.1.3.1. Similarity measure choice.....	67
3.1.3.2. Artificial node insertion in coronary trees.....	71
3.2. Matching of the coronary tree sequence.....	75
3.2.1. Mosaic-based matching of coronary tree sequence.....	76
3.2.2. Characteristics of merging and union nodes.....	78
3.2.3. Building of the union tree $UT_{i,i+1}$ from the matching result $M_{i,i+1}$ between T_i and T_{i+1} ..	80
3.2.4. Building of the mosaic tree $US_{0,i+1}$ of the tree sequence $S_{0,i+1}$	83
3.2.4.1. Node merging from $US_{0,i}$ and $UT_{i,i+1}$ related by their common phase i in $US_{0,i+1}$	85
3.2.4.2. Insertion in $US_{0,i+1}$ of not merged node from $US_{0,i}$ and $UT_{i,i+1}$	85
3.2.4.3. Merging of newly inserted union nodes with an existing node of $US_{0,i+1}$	86
3.2.4.4. Correction of the newly inserted node from $UT_{i,i+1}$ with artificial node insertion in $US_{0,i-1}$	87
3.2.4.5. Last tree T_{i+1} correction thanks to supplementary information from $US_{0,i}$	89
3.3. Conclusion	89
Chapitre 4 : Recherche d'incidences optimales	91
4.1. Presentation of the global algorithm	94
4.2. Object of interest.....	96
4.2.1. Vascular edges (segment).....	96
4.2.2. Bifurcations.....	97
4.3. Selection criteria.....	97
4.3.1. Foreshortening	99
4.3.2. External overlap.....	100
4.3.3. Internal overlap	101
4.3.4. Bifurcation opening angle	101
4.4. Optimal angiographic view determination	102

4.4.1.	Segment optimal view	103
4.4.2.	Bifurcation optimal view	107
4.4.3.	Time complexity analysis.....	110
4.5.	Conclusion	111
Chapitre 5 : Résultats		113
5.1.	Pair-wise matching of successive trees	115
5.1.1.	Evaluation of the hierarchical tree method	115
5.1.2.	Evaluation of the association graph method	119
5.1.2.1.	Unary constraints	119
5.1.2.2.	Binary constraints.....	121
5.1.2.3.	Pair-wise matching algorithm.....	125
5.1.3.	Comparison of the two methods.....	126
5.1.4.	Matching evaluation with artificial nodes.....	127
5.1.5.	Result synthesis	131
5.2.	Mosaic-based matching of the coronary sequence	132
5.2.1.	Evaluation of real sequence	134
5.2.2.	Evaluation from simulated sequences	136
5.2.3.	Result synthesis	137
5.3.	Optimal viewing angle results	140
5.3.1.	Result from segment of interest	140
5.3.2.	Results for bifurcation of interest	142
5.3.3.	Results for multiple segments of interest	145
5.3.4.	Result synthesis	150
5.4.	Conclusion	151
Conclusions et perspectives		153
6.1.	Bilans et conclusions	154
6.2.	Perspectives.....	157
6.2.1.	Amélioration des méthodes	157
6.2.2.	Validation des méthodes.....	158
6.2.3.	Autres applications.....	159

Annexe A Interface graphique	161
Bibliographie.....	167
Liste des publications	181

TABLE DES FIGURES

Figure 1 : Schémas du cœur (Inserm, 2016).....	2
Figure 2 : Circulation sanguine dans le corps humain représentée par un réseau d'artères, de capillaires et de veines.	3
Figure 3 : Vascularisation du cœur : artères et veines coronaires (source : http://www.sanofi-cardiologie.fr/cardiologie/Contenus/Menu/SERVICES/Coeur-et-Anatomie)	4
Figure 4 : Cycle cardiaque et mouvement du cœur (Belin SVT 2de, 2010 p. 209).....	6
Figure 5 : Schéma du tissu nodal du cœur (source : eurekasante.vidal.fr).....	7
Figure 6 : Activité électrique du cœur couplé à l'ECG en anglais. Onde P : début de la contraction des ventricules (impulsion électrique produite au niveau du nœud sinusal). Ondes QRS : début de la contraction des ventricules (l'impulsion électrique se propage du nœud auriculo-ventriculaire au faisceau de His et au réseau de Purkinje). Onde T : relaxation des ventricules et repolarisation. Intervalle PQ : temps de contraction auriculaire. Intervalle QT : temps de contraction ventriculaires.	8
Figure 7 : Représentation des trois mouvements principaux du cœur durant la systole. CR : contraction radiale, CL : contraction longitudinale, TL : torsion longitudinale. [Cast12]	9
Figure 8 : Exemple de champs de déplacement systolique (A et C) et diastolique (B et D) pour un volontaire sain d'après [Peti03] (Figure 5.15 et 5.16). A et B représentent les champs de déplacement sur une séquence petit axe et C et D sur une séquence grand axe.	10
Figure 9 : Coupe schématique d'une artère saine et d'une artère athérosclérose (Inserm, 2016).	11
Figure 10 : Schéma de l'évolution d'un athérome [Libb02].	12
Figure 11 : Exemples de projection 2D des artères coronaires sous deux incidences différentes réalisées pendant une coronarographie rotationnelle pour un même patient avec une seule injection de produit de contraste.	14
Figure 12 : Schéma des étapes d'une angioplastie avec pose de stent.	15
Figure 13 : Paramètres géométriques du C-arm. A) Définition des angles OAG/OAD et CRAN/CAU. B) Schéma d'un C-arm avec définition des distances [SSRB05].	17
Figure 14 : Exemples de C-arm. A) C-arm mobile. B) C-arm motorisé. (http://www3.gehealthcare.com et http://www.healthcare.siemens.com)	18
Figure 15 : Etapes diagnostiques de l'atteinte coronarienne et procédure de revascularisation.	21
Figure 16 : Séquence de projections acquise sur une durée de 4 cycles cardiaques. Les projections (orange) représentant la même phase sont utilisées pour la reconstruction de l'arbre coronaire à cet instant t.	23
Figure 17 : Exemples de séquence d'arbres coronaires 3D droit et gauche représentant différents instants du cycle cardiaque obtenus après modélisation 3D.....	28

Figure 18 : Lignes centrales de la séquence d'arbre coronaire 3D représentée dans la Figure 17.....	28
Figure 19 : A) Extraction des lignes centrales sous forme de spline de couleur différentes. B) Détection des bifurcations et construction de la hiérarchie du graphe.	29
Figure 20 : Arbres coronaires non isomorphes : difficulté pour la mise en correspondance.	29
Figure 21 : Détails d'une séquence $S_{0,9}$ de 10 arbres coronaires gauches T_i (lignes centrales). L'image de droite nous montre la superposition de cette séquence permettant d'observer le mouvement des artères selon les phases du cycle cardiaque.	31
Figure 22 : Schéma expliquant le calcul de la hiérarchie des splines : calcul de la distance associée à la matrice distMap.	32
Figure 23 : Artère coronaire gauche représenté sous forme d'arbre. L'image de droite représente la structure d'arbre et l'image de gauche les différentes informations stockées (les points noirs correspondent aux nœuds du graphe et les segments de couleurs aux arcs).....	36
Figure 24 : Classification des méthodes d'appariement de graphe selon Oyarzun Laura [Oyar15].	42
Figure 25 : (a) un graphe. (b) une clique en vert d'ordre 3. (c) une clique maximale d'ordre 3. (d) une clique maximum d'ordre 4.	48
Figure 26 : Exemple de cliques maximums multiples dans un graphe.....	51
Figure 27: Three main steps to obtain the matching result $M_{i,i+1}$ with the association graph approach based on Metzen et al 's algorithm [MKSZ09].	63
Figure 28: Representation of the similarity measures used to match two nearby coronary trees. A) First selection: the current node $v_i^{\alpha_i}$ can be assigned with the nodes $v_{i+1}^{\alpha_{i+1}}$ and $v_{i+1}^{\beta_{i+1}}$. S represents the diameter selected to determine the region of interest (threshold value associated to the constraint). B) Application of the DTW's algorithm between two paths to evaluate the similarity between the two nodes $v_i^{\alpha_i}$ and $v_{i+1}^{\alpha_{i+1}}$ and for C) respectively for the two nodes $v_i^{\alpha_i}$ and $v_{i+1}^{\beta_{i+1}}$. The green path is longer than the current purple path: the result of their matching will be penalized.	68
Figure 29: Example of wrong branch assignment by the computation of the Euclidean distance of only node coordinate by DTW. The dark branch between the blue node and the purple node is matched with the incorrect red branch. Normally, this is the dark branch between the blue node and the yellow node which must be matched with the correct red branch.	68
Figure 30: Illustration of the Dynamic Time Warping's algorithm from [KeRa05] (Fig. 3). A) The two sequences C and Q are similar but out of phase. B) Distance matrix between the two sequences. The optimal warping path (point by point optimal matching) is shown in solid squares. C) Alignment result from the distance matrix.....	69
Figure 31: Step pattern used by the Dynamic Time Warping's algorithm to compute the distance matrix.	70

Figure 32: Illustration of artificial node insertion a_{i+1}^ϕ associated to an unmatched bifurcation node v_i^ϕ	72
Figure 33: Global mosaic-based matching algorithm for a 3D+T coronary tree sequence.....	76
Figure 34: Example of a built mosaic tree from the tree sequence $S_{0,2} = \{T_0, T_1, T_2\}$ with artificial nodes (pink and blue) inserted during the matching process. B) Building of the two union trees $UT_{0,1}$ and $UT_{1,2}$ from respectively the two trees T_0, T_1 and T_1, T_2 . C) Built mosaic tree $US_{0,2}$ from the sequence $S_{0,2}$ derived from $UT_{0,1}$ and $UT_{1,2}$	79
Figure 35: First correction step: the single union node with 6_2 as main node is newly inserted, we search if it can be matched with another union node in previous phases. After the Dynamic Time Warping (DTW) computation based on Euclidean distance of node coordinates, tangent and direction, the two paths in blue (from main nodes 4_0 and 6_2) and in purple (from main nodes 4_0 to 8_0) are considered as similar, so nodes 8_0 and 6_2 must be merged. The algorithm searches if there are intermediary nodes between the two similar paths: in the example the node 7_2 in the tree T_2 . As previously, we search if this node can be matched with another node. In this example, an artificial node 10_0 is inserted in T_0 to match with 7_2	86
Figure 36: Second correction step: we search if an artificial node can be added in previous tree to complete the matching result. The algorithm takes into account the intermediary nodes between the two similar paths determined by DTW before the insertion of artificial nodes.	87
Figure 37: summary steps of the optimal viewing angle determination.	95
Figure 38: Schema and notation for a segment (A) and a bifurcation (B) of interest.	97
Figure 39: schematic representation of two foreshortening rates for two different viewing angles (θ_j, φ_k) . (a) The gantry orientation corresponds to a high foreshortening rate and it better than (b) which represent a bad foreshortening rate.	98
Figure 40: Comparison of binary constraints cb^δ for the nine matching tests with different values of selectivity parameters $s(cb^\delta)$. The better selectivity parameter for each constraint are in bold.	122
Figure 41: Different combinations of binary constraints cb^δ for the nine matchings.....	124
Figure 42: Comparison of constraints for the association graph algorithm. A1 corresponds to the algorithm with $s(cu^1) = 10$ and $s(cb^4) = 5$ for the nine matchings. A2 is based on the combination of the three binary constraints with $s(cu^1) = 10$ and $s(cb^4) = 5$, $s(cb^5) = 15$ and $s(cb^6) = 15$ for the nine matchings.	125
Figure 43: Comparison between Lohe et al.'s algorithm (A3) with $s(cc^2) = 10$ and $s(sm^2) = 5$, $s(sm^4) = 15$ and $s(sm^5) = 15$ and the algorithm based on Metzen et al. and Tschirren et al. (A2) with $s(cu^1) = 10$ and $s(cb^4) = 5$, $s(cb^5) = 15$ and $s(cb^6) = 15$ for the nine matchings.	126
Figure 44: Comparison between our association graph algorithm without artificial nodes (A2) and with artificial nodes (A4 and A5). Constraints and associated selectivity parameters used to determine if two branches are similar are fixed for the nine tree matchings. A4 uses only the fourth constraint with	

$s(cb^4) = 5$. A5 uses a combination of node-to-node constraints: $s(cb^4) = 5$, $s(cb^5) = 15$ and $s(cb^6) = 15$ with $w(cb^4) = 0.5$, $w(cb^5) = 0.25$ and $w(cb^6) = 0.25$ 128

Figure 45: Comparison of node distribution with the matching between two successive trees of the sequence $S_{0,9}$. A) corresponds to the matching result with only one criteria, node coordinates, based on the DTW's algorithm as the binary constraint and the artificial node insertion constraint. B) corresponds to our new algorithm with the combination of the three criteria based on the DTW's algorithm and the hierarchy correction step (A5). We have the same node number in the two cases. CM corresponds to the correctly matched nodes. CAN represents the correct matched node with an artificial nodes (CBAN+CLAN). IM corresponds to the incorrectly matched nodes and UM the unmatched nodes..... 129

Figure 46: A) Real coronary tree sequence $S_{0,9}$ of 10 extracted left coronary trees T_i at different ti cardiac phase and B) Simulated tree sequence $S'_{0,9}$ from modification of $S_{0,9}$ 129

Figure 47: Node distribution in percentage for the 36 matching tests (mean of the total number for each classification). A) represents the real distribution (ground truth) and B) the node distribution after the matching of two successive trees (means of the 36 tests). CM corresponds to the correctly matched nodes, CBAN and CLAN respectively to correct matched bifurcation and leaf nodes with an artificial node. IM represents the incorrect matched nodes and UM the unmatched nodes. 130

Figure 48: Node classification details for the 36 matching tests. 131

Figure 49: Node distribution in percent for the 42 tree sequences. A) represents the real distribution and B) the node distribution after our matching algorithm. 134

Figure 50: Matching result of the tree sequence $S_{0,9}$. A) Superimposition of the tree sequence with the matched edges represented in the same color. B) The global mosaic tree $US_{0,9}$ of the sequence $S_{0,9}$ based on the tree T_0 completed with information from $S_{1,9}$. The same colors are used for the edge matching..... 135

Figure 51: Result of the matched sequence $S_{0,9}$ with node classification for each tree. 135

Figure 52: A) Node distribution in percent for the 9 matchings of two successive trees of the sequence $S_{0,9}$ computed in section 5.1. B) Node distribution in percent for the global matching algorithm of the sequence $S_{0,9}$ with mosaic tree building. In this figure, CAN represents all correct artificial nodes (CBAN + CLAN), IM and UM respectively the total incorrect matched node number and the total unmatched node number..... 136

Figure 53: Matching result of the simulated tree sequence $S'_{0,9}$. A) Superimposition of the simulated trees with edge matching represented with same color. B) Global mosaic tree $US'_{0,9}$ of the sequence $S'_{0,9}$ built from the tree T_0 completed with information from $S'_{1,9}$. The same color as A) are used. 137

Figure 54: Node classification after the matching of the simulated tree sequence $S'_{0,9}$ 137

Figure 55: Box plots for the matching result in percentage for the 42 matched sequences. In the left, we have global percent of correctly matched nodes (Total CM) and after the details for real nodes (CM), artificial nodes corresponding to a bifurcation in another tree (CBAN) and artificial nodes corresponding to a leaf in another tree (CLAN). In the right, we have the global percent of incorrectly matched nodes (Total IM) and its associated details (IM, IBAN and ILAN), and after the global percent of unmatched nodes (Total UM) and its detail (UM, UBAN, ULAN)..... 138

Figure 56: Examples of matching errors. The three zooms show the localization errors in different viewing angles. A) The leaf of the tree in bold is matched with the leaf in other trees. This incorrect matched leaf should be matched with an artificial node in some others trees which have longer terminal branch. B) The error is the same of the previous case A). The leaf of the tree in bold is incorrectly matched and the same artificial node (which corresponds to the current leaf) is missing in all the other trees of the sequence. C) Contrary to the two previous cases, errors are only located on the tree in bold (errors similar to the Figure 51). The leaf node is incorrectly matched and consequently an artificial node corresponding to a bifurcation is missed. 139

Figure 57: Process of the optimal view map determination for a segment in the whole cardiac cycle. A) The foreshortening and external overlap maps are computed successively without threshold for each matched phase. The global foreshortening and external overlap maps are calculated with the minimum value of the previous temporary maps. The global optimal view map corresponds to the mean of the two last maps. B) Same process as A) but with two successive thresholds. First the foreshortening maps are computed with a threshold of 90% for each phase to perform a high selection of angles until the global foreshortening maps. Second, the external overlap maps are calculated with a threshold of 30% to only remove worse values. The global optimal view map is computed with the two previous global map. $nb_{(\theta_j, \varphi_k)}$ represents the selected gantry orientation number after each threshold. 141

Figure 58: Optimal viewing angles for a segment of interest matched in five cardiac phases. The optimal view map is obtained in the Figure 57. Two optimal gantry orientations are proposed and the left coronary artery with the segment in pink is shown under these two views. A) Optimal view map with the 5% better gantry orientation (551 retained angles). B) Optimal view map with the 2% better gantry orientation (220 retained angles). 142

Figure 59: Process to optimal view map determination for a bifurcation in the whole cardiac cycle. The bifurcation is matched in all the ten phases. A) The four criteria are computed without threshold for each matched phase. B) The four criteria are calculated successively with a threshold to remove worse values under 30%. First the foreshortening map and bifurcation opening map are computed successively for each map and the threshold is performed. Next, when the global foreshortening and bifurcation opening maps are obtained with their minimum value on the sequence, the two overlap are computed in the same way. The global optimal view map is obtained with the mean of the four global maps. $nb_{(\theta_j, \varphi_k)}$ represents the selected gantry orientation number after each threshold.... 144

Figure 60: Optimal viewing angles for a bifurcation in the whole cardiac cycle. The optimal view map corresponds to the Figure 59. In top, the left coronary tree with the bifurcation in pink is display under the best angiographic view for the different q phases. 145

Figure 61: Optimal view map for different segments. In the right, the five last columns represent the optimal view maps for five successive segments in the tree hierarchy. In the left, the first column corresponds to the optimal view maps when the five previous segments are merged into one (multiple segments case). In the middle, the two last columns represent the optimal view map from the linear combination of the optimal view map of the five segments (same weights for the first and weights calculated from the two criteria proposed in section 4.4). The first line represents the optimal view map without threshold and the others with different thresholds..... 147

Figure 62: Global External Overlap map for different segments. In right, the last five columns represent the global external overlap maps for the five successive segments in the tree hierarchy. In the left, the

first column corresponds to the global external overlap map when the five segments are merged into one single (multiple segment case). In the middle, the two last columns represent the global external overlap map from the linear combination of the global external overlap map of the five segments (same weights for the first and weights calculated from the two criteria proposed in section 4.4).. 147

Figure 63: Optimal viewing angles for a multiple segment. The optimal view maps in the left correspond to the 5% better angiographic projection angles and the following maps represent the 2% better values. A) Optimal viewing angle for the multiple segment as object of interest. B) Optimal viewing angle for the combination between the six segments (same weight). C) Optimal viewing angle for the combination between the six segments with different weights computed in relation to the hierarchical level of the segment according to the selected segment (in pink) and the length segments. The different tree sequences in the right show the left coronary tree under the found optimal gantry orientation. The pink segments represent the selected segment (segment 1), the green segment is the father of the selected segment (segment 2), the segment 3 is in purple, the segment 4 in orange, the segment 5 in blue and the root segment (segment 6) in yellow. 148

Figure 64: Interface graphique de l'application développée permettant de charger les données, de visualiser l'arbre en 3D sous différentes incidences et d'exécuter les différents algorithmes programmés au cours de cette thèse..... 162

Figure 65: Capture d'écran de trois onglets de l'interface permettant d'exécuter les algorithmes développés pour la mise en correspondance (A) et la recherche d'incidence optimale (C). L'onglet (B) permet de modifier interactivement les données réelles pour générer des données simulées. 163

Figure 66: Exemple de fichier contenant les données réelles. A) les dossiers correspondent aux différentes phases du cycle cardiaque. B) les fichiers contiennent le scanner coronarien CT ainsi que le volume (surfaces des contours vasculaires) et les lignes centrales des arbres droits et gauches..... 164

TABLE DES TABLEAUX

<p>Table 1: Comparison of different consistency constraints cc^λ and similarity measures sm^μ for the nine matchings with different threshold values $s(cc^\lambda)$ and $s(sm^\mu)$. CM is the number of correctly matched nodes, CMWA is the number of correctly matched nodes without assignment, IM corresponds to the incorrectly matched nodes and UM to the unmatched nodes. The distribution percentage of each classification is given in brackets. The last column is the execution time in ms for the nine matchings.</p>	117
<p>Table 2: Matching results for different similarity measures sm^μ with the combination of the two consistency constraints cc^1 and cc^2 and their threshold $s(cc^1) = 7.5$ and $s(cc^2) = 10$ obtain in relation to the Table 1. CM is the number of correctly matched nodes, CMWA is the number of correctly matched nodes without assignment, IM corresponds to the incorrectly matched nodes and UM to the unmatched nodes.</p>	117
<p>Table 3: Matching results for different combinations of similarity measures sm^μ and their associated thresholds $s(sm^\mu)$ obtained with the consistency constraint cc^2 and its associated threshold $s(cc^2) = 10$. This constraint gives better result according to the Table 1 and Table 2. CM is the number of correctly matched nodes, CMWA is the number of correctly matched nodes without assignment, IM corresponds to the incorrectly matched nodes and UM to the unmatched nodes.</p>	118
<p>Table 4: Comparison of different unary constraint results cu^γ with different associated selectivity parameter values $s(cu^\gamma)$ for the nine matchings.</p>	120
<p>Table 5: Example of the better combination of unary constraints cu^γ for the nine matchings according to a weight factor.</p>	121
<p>Table 6: Runtime in ms of binary constraints cb^δ with different selectivity parameter values $s(cb^\delta)$ for the nine matchings.</p>	123
<p>Table 7: Comparison of the execution time (in ms) for four algorithms.</p>	127
<p>Table 8: The different weights used to the optimal view map combination.</p>	147
<p>Table 9: Foreshortening and external overlap rates obtained for the optimal viewing angle (-43°, 16°) determined from the optimal view map of the multiple segment (Figure 63-A).</p>	149
<p>Table 10: Foreshortening and external overlap rates obtained for the optimal viewing angle (-74°, 17°) determined from the mean of the global optimal view maps of the six segments (same weight factor) (Figure 63-B).</p>	149
<p>Table 11: Foreshortening and external overlap rates obtained for the optimal viewing angle (-60°, 20°) determined from the merging of the global optimal view maps of the six segments (different weight factors) (Figure 63-C).</p>	149

TABLE DES ALGORITHMES

Algorithm 1: Détermination de la matrice de distance	33
Algorithm 2: Détermination de la distance euclidienne entre deux splines	33
Algorithm 3: Calcul du vecteur de connectivité et de la racine de l'arbre	34
Algorithm 4: Artificial node insertion.	74
Algorithm 5: Mosaic-based matching algorithm of the whole tree sequence (equation (4) and (5)) .	78
Algorithm 6: function BuildUnionOfTwoTrees()	81
Algorithm 7: function ComputeMosaicTree().....	83
Algorithm 8: Optimal view algorithm for a matched segment of interest, function OptimalViewMapSegment().....	103
Algorithm 9: Optimal view algorithm for a matched bifurcation of interest, function OptimalViewMapBifurcation().	107

CHAPITRE 1

PROBLEMATIQUE

SOMMAIRE

1.1.	Contexte médical.....	2
1.1.1.	Système cardiovasculaire	2
1.1.2.	Pathologies	10
1.1.3.	Diagnostics.....	12
1.1.4.	Traitements	14
1.1.5.	Modalité d’acquisition.....	16
1.2.	Motivation de la thèse	19
1.2.1.	Limitations et difficultés liées à l’angiographie RX.....	19
1.2.2.	Projet	21
1.2.3.	Reconstruction 3D des artères coronaires	22
1.2.4.	Données : séquence 3D+T d’arbres coronaires.....	25
1.2.4.1.	Définitions : Notions de graphe et d’arbre.....	25
1.2.4.2.	Obtention de la séquence d’arbres coronaires 3D.....	27
1.2.4.3.	Représentation des données sous forme d’arbres	30
1.2.5.	Contribution	36

Ce premier chapitre a pour objectif de présenter le contexte clinique de cette thèse. Dans un premier temps, le système cardio-vasculaire et plus particulièrement la physiologie des artères coronaires sont abordés. Les pathologies cardiaques atteignant les artères coronaires, comme l'athérosclérose, sont ensuite décrites avec les différents diagnostics et traitements possibles. Dans un second temps, ce chapitre est consacré à notre problématique. Le projet est ainsi décrit pour comprendre au mieux notre contribution.

1.1. CONTEXTE MEDICAL

1.1.1. SYSTEME CARDIOVASCULAIRE

Le cœur est un des organes vitaux du corps humain. Situé au sein du thorax, entre les deux poumons, ce muscle a pour objectif de faire circuler le sang dans tout l'organisme par le biais de ses contractions. Le cœur est connecté aux autres organes via un réseau d'artères, de capillaires et de veines, distribuant grâce au sang l'oxygène et les nutriments nécessaires au fonctionnement des cellules du corps et permettant l'élimination de certains déchets (CO₂ entres autres).

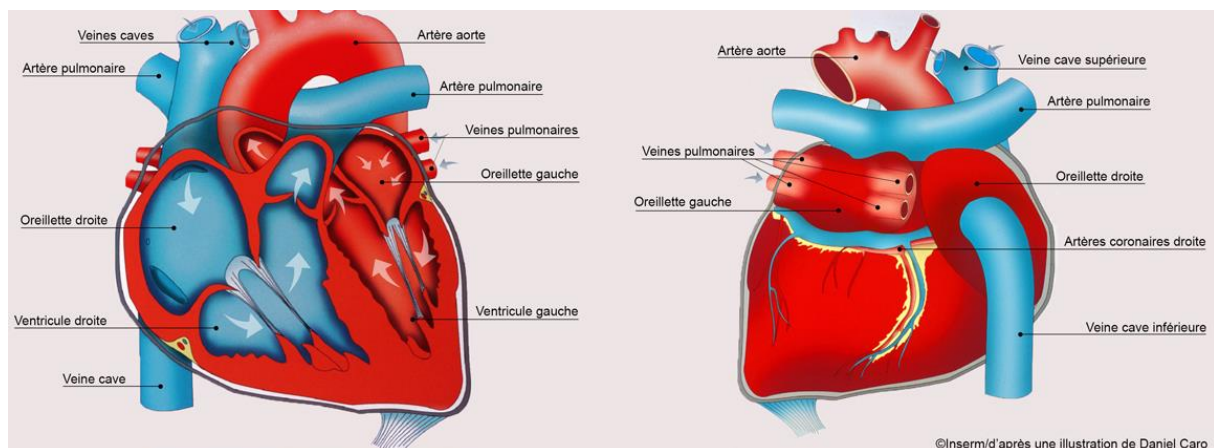


Figure 1: Schémas du cœur (Inserm, 2016).

Le cœur est composé de deux moitiés indépendantes nommées cœur gauche et cœur droit séparés par une cloison étanche. Chacune de ces parties est composée de deux cavités : une oreillette

et un ventricule (Figure 1 décrivant la physiologie cardiaque et la circulation sanguine intracardiaque). Ces dernières sont séparées par des valves (valve tricuspide, valve pulmonaire, valve mitrale et valve aortique) qui assurent le transfert du sang entre les cavités et donc le bon fonctionnement de la pompe cardiaque. Le sang, appauvri en oxygène et enrichi en CO₂ par son passage dans les organes, est acheminé via la veine cave dans le cœur, et plus particulièrement dans l'oreillette droite. La veine cave supérieure récupère le sang désaturé en oxygène provenant des membres supérieurs, tandis que la veine cave inférieure reconduit le sang provenant des membres inférieurs. Ce sang est ensuite guidé jusqu'aux poumons grâce aux artères pulmonaires. De nouveau riche en oxygène grâce aux poumons, il retourne dans le cœur via les veines pulmonaires qui sont reliées à l'oreillette gauche. Par la suite, le sang est amené au niveau du ventricule gauche où il est propulsé en direction des autres organes par l'aorte (Figure 2 décrivant la circulation cardiovasculaire).

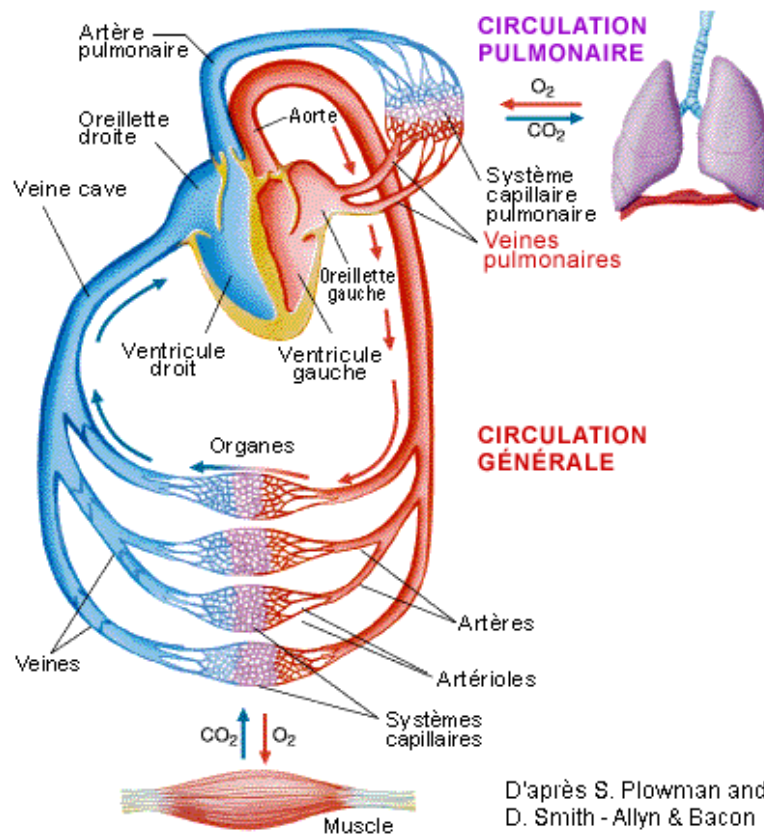


Figure 2: Circulation sanguine dans le corps humain représentée par un réseau d'artères, de capillaires et de veines.

Le cœur est composé principalement de trois tuniques :

- **l'endocarde** : tunique interne en contact direct avec le sang. Il recouvre donc les cavités cardiaques ainsi que les valves du cœur.
- **le myocarde** : tunique située au milieu. Ce muscle est responsable de l'action de pompage du cœur. Son épaisseur varie en fonction de sa localisation, il est très épais au niveau des ventricules, mais mince au niveau des oreillettes.
- **le péricarde**: tunique externe composée de plusieurs feuillets protégeant le cœur et permettant les mouvements du cœur liés au cycle cardiaque sans entrave. Le péricarde viscéral (Épicarde) adhère au myocarde. Il est séparé du péricarde pariétal et du péricarde fibreux par un espace péricardique (espace de glissement).

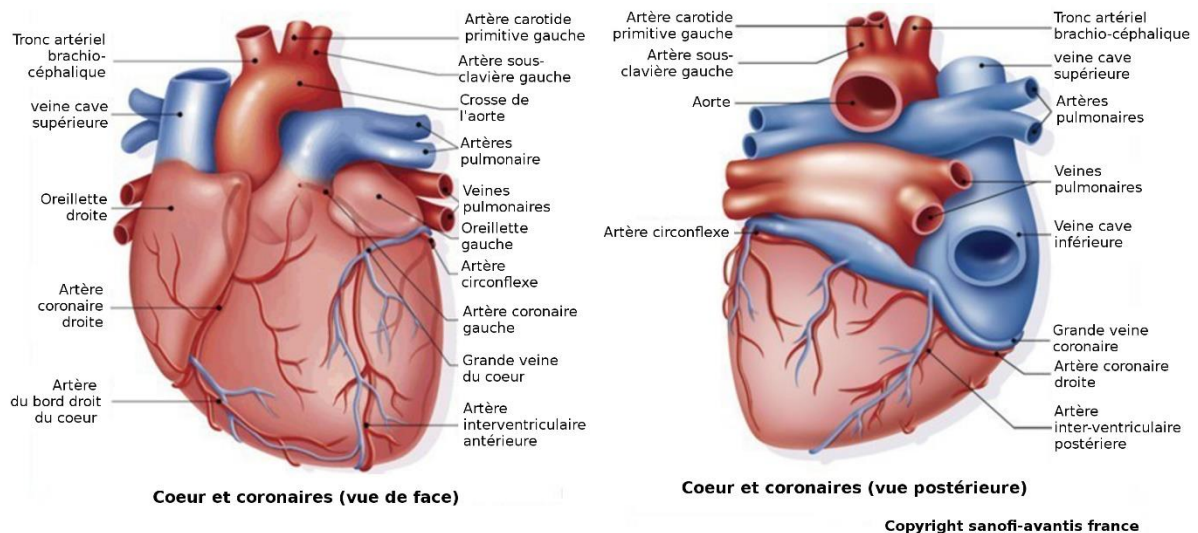


Figure 3: Vascolarisation du cœur : artères et veines coronaires (source : <http://www.sanofi-cardiologie.fr/cardiologie/Contenus/Menu/SERVICES/Coeur-et-Anatomie>)

Le cœur, étant un muscle, est aussi vascularisé (Figure 3). Effectivement, pour pouvoir fonctionner correctement, ce dernier a besoin d'oxygène et de nutriments. Les artères coronaires, naissant à la base de l'aorte (sinus de Valsalva), parcourent la surface externe du cœur (Épicarde) et alimentent directement le myocarde grâce à ses nombreuses ramifications et les capillaires. Le sang, de nouveau riche en CO₂, est ramené vers l'oreillette droite via les veines coronaires (Figure 3). Dans notre sujet de recherche, nous nous intéressons plus particulièrement aux artères coronaires. Ces dernières, décrivant une couronne autour du cœur, sont dites terminales. La survenue d'une obstruction aura immédiatement une répercussion sur l'organe et sur son fonctionnement. A leurs

origines, ces artères possèdent un diamètre allant de 3 à 5 mm. Ces structures de petites tailles sont liées à la surface du myocarde. Leurs formes tortueuses leur permettent d'englober et de couvrir la totalité de cette surface. Les artères coronaires sont séparées en deux branches principales (Figure 3) : l'artère coronaire gauche (LCA pour Left Coronary Artery) et l'artère coronaire droite (RCA pour Right Coronary Artery).

L'artère coronaire gauche est composée de trois segments principaux. Elle débute dans le sinus de Valsalva antérogauche par un tronc commun (TC) et se divise en une artère interventriculaire antérieure (IVA) et une artère circonflexe (CX).

- L'IVA chemine dans le sillon du même nom, l'interventriculaire antérieure, jusqu'à la pointe du ventricule gauche (l'apex) pour rejoindre l'artère interventriculaire postérieure (branche de l'artère coronaire droite). Ses branches perforantes, dites septales, irriguent les deux tiers antérieurs du septum interventriculaire et ses branches de surfaces, dites diagonales, irriguent quant à elles la face antérieure du ventricule gauche.
- L'artère circonflexe chemine le sillon atrioventriculaire gauche jusqu'à généralement la face inférieure du ventricule gauche. Elle se divise en branches marginales (branche de surface) irriguant la face latérale du ventricule gauche et en branches auriculaires irriguant l'oreillette gauche.

L'artère coronaire droite, née dans le sinus de Valsalva droit, chemine le sillon atrioventriculaire droit. Dans cette portion, l'artère donne naissance à une branche appelée marginale de bord droit irriguant la face antérieure du ventricule droit. L'artère coronaire droite se divise ensuite en deux branches à la croix des sillons:

- l'artère interventriculaire postérieure (IVP) vascularisant la face inférieure du ventricule droit,
- l'artère rétroventriculaire gauche (RVG) irriguant la partie médiane et basale de la face inférieure du ventricule gauche.

L'anatomie des artères coronaires est très variable inter-individu. Outre les déformations des artères coronaires liées à l'âge (artères plus tortueuses, perte d'élasticité, etc.) ou aux anomalies et variantes anatomiques, l'importance des deux branches gauches et droites varie selon les individus. Le terme de dominance pour le réseau coronaire apparaît alors [SMKM05]. La circulation coronarienne est ainsi à dominance droite pour environ 70% de la population, à dominance gauche pour 20% et enfin équilibrée pour seulement 10% de la population.

Ces artères coronaires solidaires du myocarde subissent les différents mouvements non rigides du cœur (contraction, relaxation). Chaque battement du cœur entraîne une séquence d'évènements

appelée révolution cardiaque (Figure 4). Le fonctionnement du cœur est continu et périodique. Deux périodes peuvent être mises en évidence, les phénomènes de :

- contraction, appelés **systoles**,
- relâchement, appelés **diastoles**.

Le repère initial de la révolution cardiaque commence par une phase de propulsion du sang vers les ventricules due à la contraction des oreillettes. Cette étape est appelée systole auriculaire. La phase suivante correspond à la contraction des ventricules et à l'éjection du sang dans les artères, c'est la systole ventriculaire. Elle se déroule en deux étapes :

- la systole ventriculaire iso-volumique (phase de contraction). Les valves tricuspides et mitrales se ferment empêchant le reflux du sang dans les oreillettes. Les valves pulmonaires et aortiques, quant à elles s'ouvrent. Les valves provoquent alors le premier bruit du cœur.
- la systole ventriculaire isotonique (phase d'éjection systolique). Les ventricules se contractent permettant alors au sang, sous forte pression, d'être éjecté du cœur via l'aorte ou l'artère pulmonaire. Cette phase se termine par la fermeture des valves provoquant le deuxième bruit du cœur.

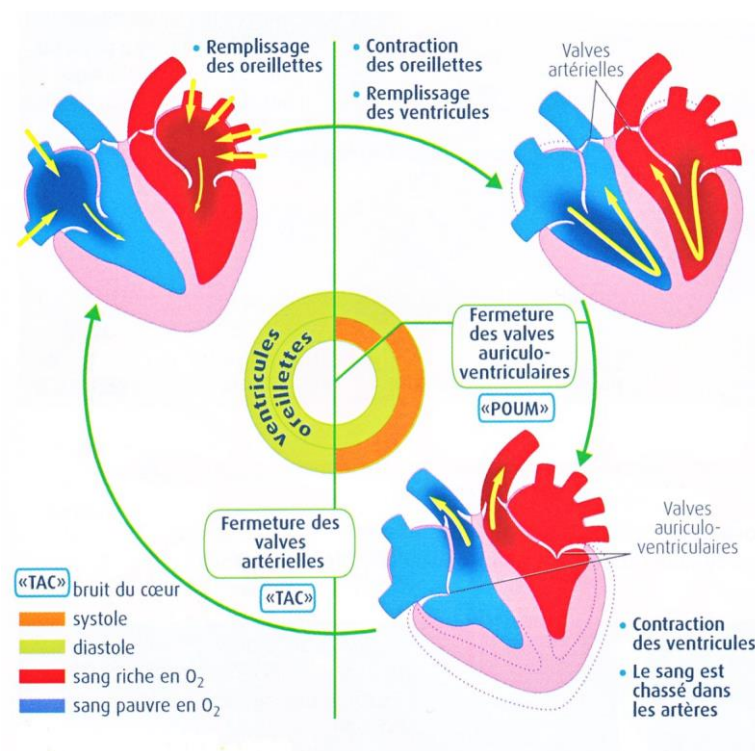


Figure 4: Cycle cardiaque et mouvement du cœur (Belin SVT 2de, 2010 p. 209).

La diastole est la dernière étape de la révolution cardiaque. Elle consiste en la relaxation de toutes les parties du cœur, permettant un remplissage passif des oreillettes. Elle peut être également séparée en deux phases :

- la diastole iso-volumique (phase de relaxation) où toutes les valves sont fermées. Le sang s'écoule donc uniquement dans les oreillettes,
- la diastole isotonique (phase de remplissage) où les valves tricuspides et mitrales s'ouvrent permettant au sang de s'écouler lentement dans les ventricules.

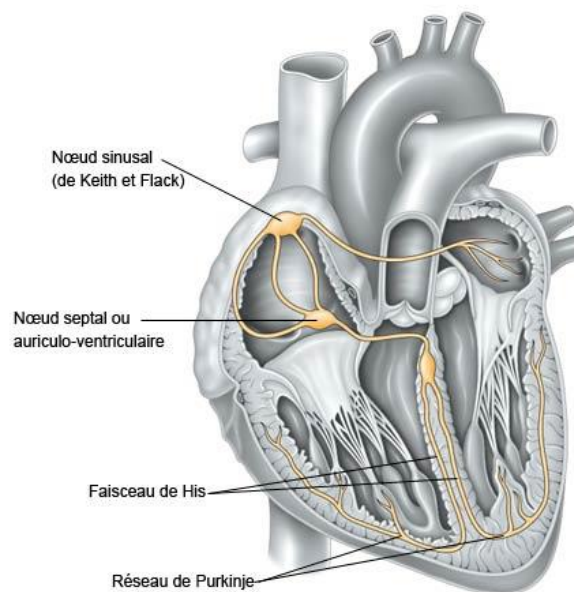


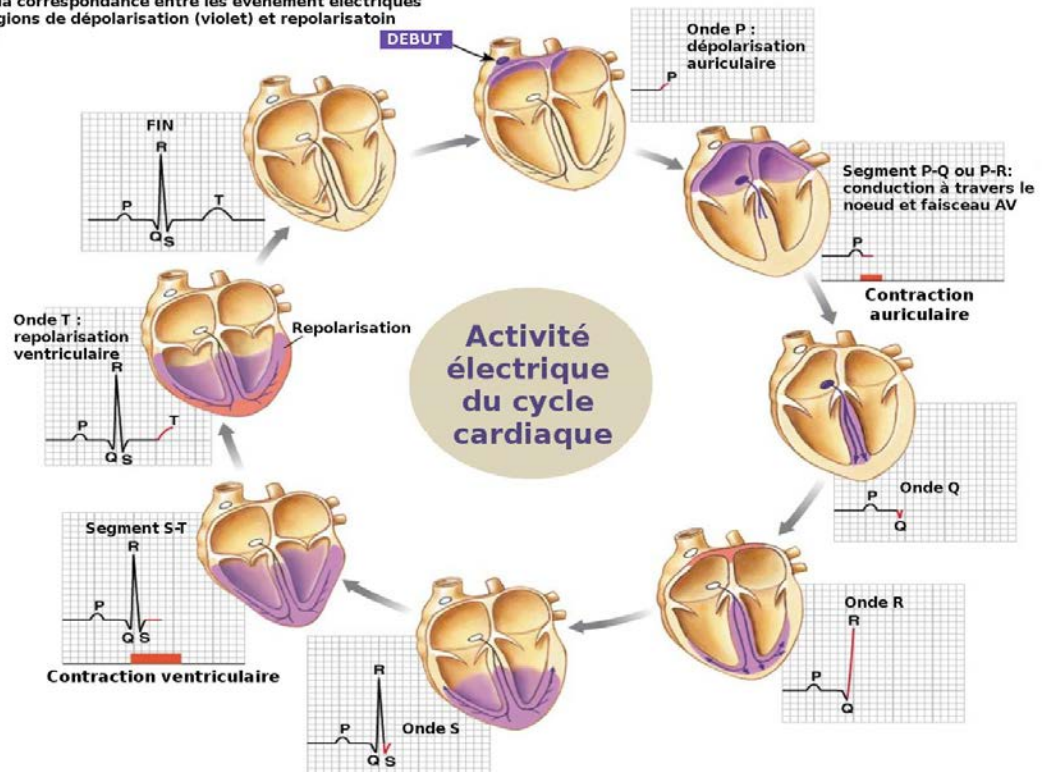
Figure 5 : Schéma du tissu nodal du cœur (source : eureka.sante.vidal.fr).

Pour réaliser ces différentes contractions, le muscle cardiaque est doué d'automatisme. Le tissu nodal en est l'origine (Figure 5). Il donne naissance à des impulsions électriques puis les conduit à grande vitesse vers les cellules myocardiques, les dépolarisant et engendrant ainsi les contractions cardiaques. Le courant électrique prend naissance dans le nœud sinusal situé dans le haut de la paroi de l'oreillette droite, près de la veine cave supérieure. Une fois l'impulsion produite, elle se répand dans les oreillettes du cœur qui se contractent (systole auriculaire) éjectant ainsi le sang vers les cavités inférieures (onde P, voir Figure 6). La dépolarisation arrive ensuite au niveau du nœud auriculo-ventriculaire, seul point de passage possible pour le courant électrique entre les oreillettes et les ventricules. Après une courte pause, l'impulsion électrique se propage dans le septum par le Faisceau de His et ensuite le réseau de Purkinje. L'impulsion électrique est ainsi propagée dans plusieurs points

du ventricule permettant ainsi une dépolarisation quasi instantanée du muscle ventriculaire. Cette contraction provoque la systole ventriculaire (intervalle QT, Figure 6). Pour finir l'activité électrique du cœur liée au cycle cardiaque, les fibres musculaires se repolarisent revenant à leur état initial.

Correlation entre un ECG et l'activité électrique du cœur

La figure montre la correspondance entre les événements électriques de l'ECG et les régions de dépolarisation (violet) et repolarisation (rouge) du cœur.



2013, Pearson Education, Inc.

Figure 6 : Activité électrique du cœur couplé à l'ECG en anglais. Onde P : début de la contraction des ventricules (impulsion électrique produite au niveau du nœud sinusal). Ondes QRS : début de la contraction des ventricules (l'impulsion électrique se propage du nœud auriculo-ventriculaire au faisceau de His et au réseau de Purkinje). Onde T : relaxation des ventricules et repolarisation. Intervalle PQ : temps de contraction auriculaire. Intervalle QT : temps de contraction ventriculaire.

L'activité électrique peut varier selon les individus. Effectivement, certaines personnes peuvent souffrir d'arythmie (tachycardie, bradycardies) ou autres anomalies (extrasystole, fibrillation ventriculaire ou auriculaire, etc.) provoquant un battement irrégulier du cœur. Les impulsions électriques se produisent de façon désordonnées ou ne passent pas par les circuits électriques habituels.

Les mouvements cardiaques, liés à l'activité électrique, ne sont pas homogènes. Pour avoir une idée plus précise de ces mouvements complexes, de nombreuses études ont été menées portées sur

le mouvement du ventricule gauche [Cast12, HADA99, Peti03, Tava12]. Ce mouvement peut être séparé en trois composantes (Figure 7).

- Le mouvement de contraction longitudinale (CL) dirigé le long du grand axe du ventricule. La base du cœur descend vers l'apex en systole avec une amplitude de mouvement de 12 à 15 mm. Au contraire, l'apex est peu mobile, ne bougeant que de quelques millimètres.
- Le mouvement de contraction radiale (CR) qui correspond au déplacement de l'endocarde vers le grand axe du ventricule avec des amplitudes comprises entre 4 et 8 mm.
- Le mouvement de torsion correspondant à une rotation autour du grand axe du ventricule. Durant la diastole, la rotation de la base s'effectue dans le sens horaire et la rotation de l'apex diffère. Cette dernière est réalisée dans le sens inverse avec une angulation supérieure.

A ses différents mouvements non rigides, il faut ajouter la translation du cœur liée aux phénomènes respiratoires. La Figure 8 issue du rapport de thèse de Petitjean [Peti03] nous montre ces mouvements complexes en systole et en diastole selon différentes coupes du ventricules.

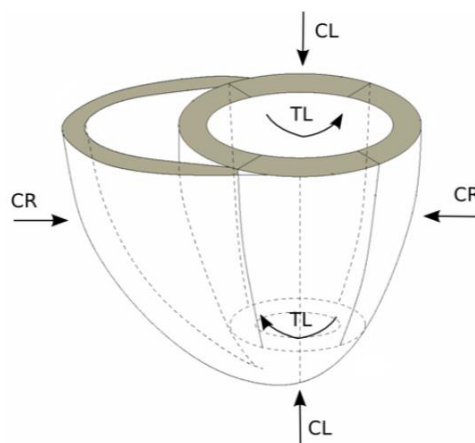


Figure 7: Représentation des trois mouvements principaux du cœur durant la systole. CR : contraction radiale, CL : contraction longitudinale, TL : torsion longitudinale. [Cast12]

Les artères coronaires suivent les différentes contractions du cœur en lien direct avec les phases de l'ECG. Elles sont donc également animées d'un mouvement irrégulier complexe (non rigide et non linéaire). Leurs déplacements peuvent être variables d'un individu à un autre (malformation, arythmie, stress, infarctus, etc.). De plus, les différentes étapes du cycle cardiaque font varier le diamètre des artères coronaires. Ce dernier évolue dans l'intervalle [0,8 mm - 5mm]. Toutes ces difficultés doivent être prises en considération dans notre projet.

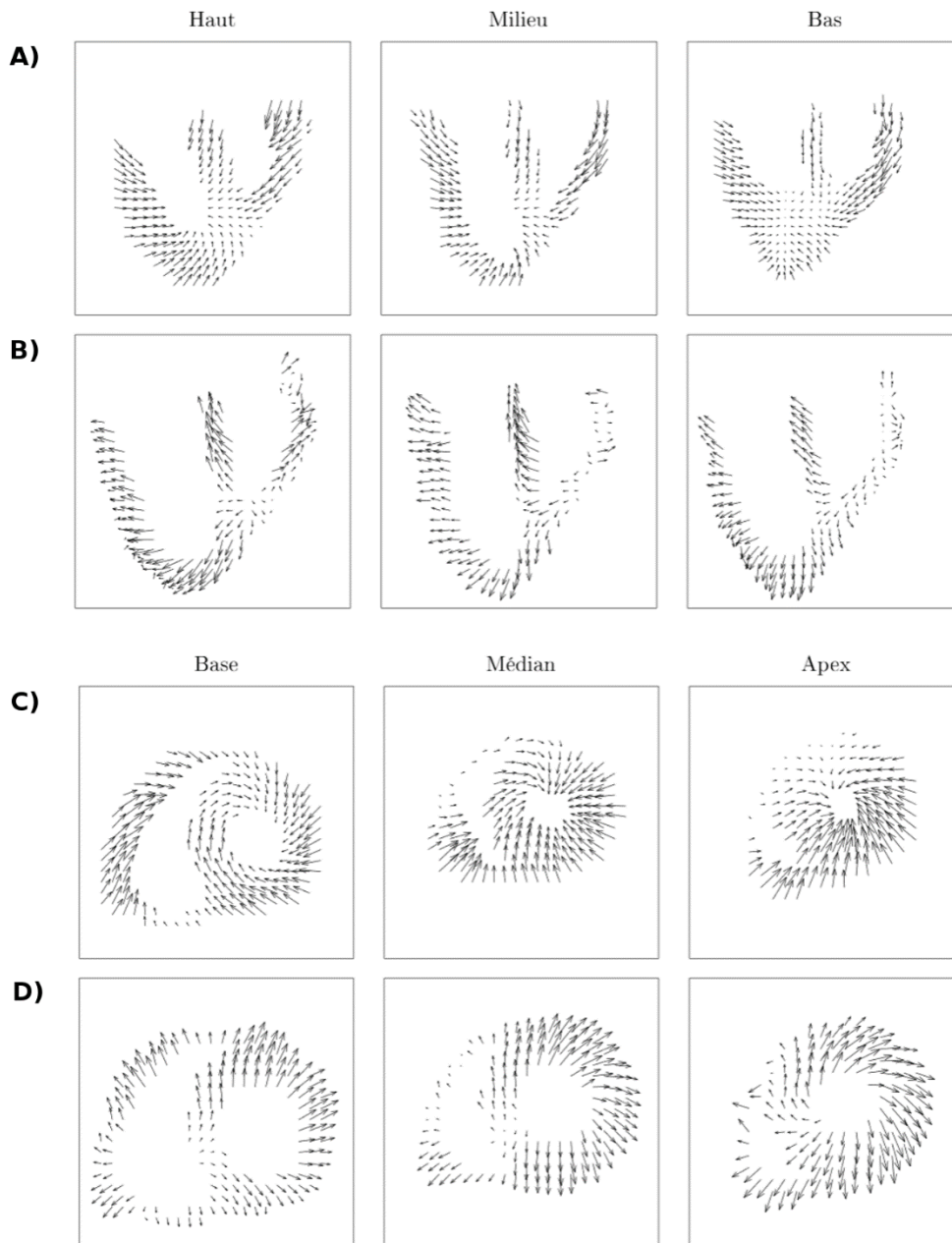


Figure 8: Exemple de champs de déplacement systolique (A et C) et diastolique (B et D) pour un volontaire sain d'après [Peti03] (Figure 5.15 et 5.16). A et B représentent les champs de déplacement sur une séquence petit axe et C et D sur une séquence grand axe.

1.1.2. PATHOLOGIES

Les maladies cardiovasculaires représentent la première cause de mortalité dans le monde. Effectivement, selon l'Organisation Mondiale de la Santé en 2012 (chiffres mis à jour en janvier 2015

sur le site internet de l'OMS¹) le nombre de décès lié à ces maladies est estimé à 17,5 millions, soit à 31% de la mortalité totale mondiale (dont 7,4 millions dû à une cardiopathie coronarienne). En France, elles représentent 32% des décès selon le Haut Comité de la Santé Publique. L'une des principales maladies cardiovasculaires concernent les artères coronaires. La coronaropathie est une maladie grave et sera dans les années à venir une des premières causes de mortalité dans le monde quel que soit le type de population.

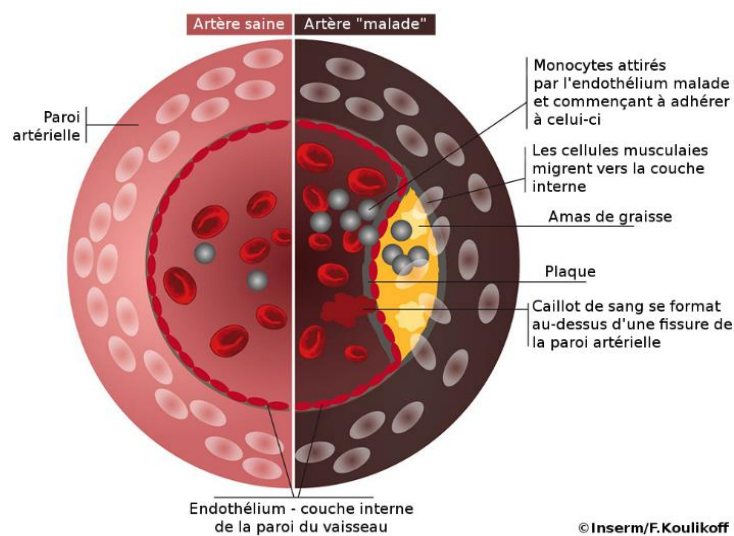


Figure 9: Coupe schématique d'une artère saine et d'une artère athérosclérose (Inserm, 2016).

L'une des causes principales de cette maladie est l'athérosclérose. Des plaques d'athérome s'accumulent sur les parois internes des vaisseaux coronariens [Hans05, SMKM05]. Elles sont constituées de corps gras (tel que le cholestérol) ou de calcium (Figure 9). Ce processus obstrue progressivement la lumière du vaisseau, diminuant le flux sanguin vers le myocarde ([Libb02] décrivant le processus de formation des plaques d'athérome). Or, ce sont ces artères qui permettent de « nourrir » le muscle cardiaque en lui fournissant l'O₂ et les nutriments nécessaires. La diminution du flux sanguin engendre un manque d'oxygène (ischémie) pour le myocarde et provoque alors la mort progressive ou totale d'une partie du myocarde (Figure 10). L'athérosclérose peut ainsi conduire à une insuffisance cardiaque, à une angine de poitrine, à un infarctus du myocarde ou encore à une mort subite.

¹ <http://www.who.int/mediacentre/factsheets/fs317/fr/>

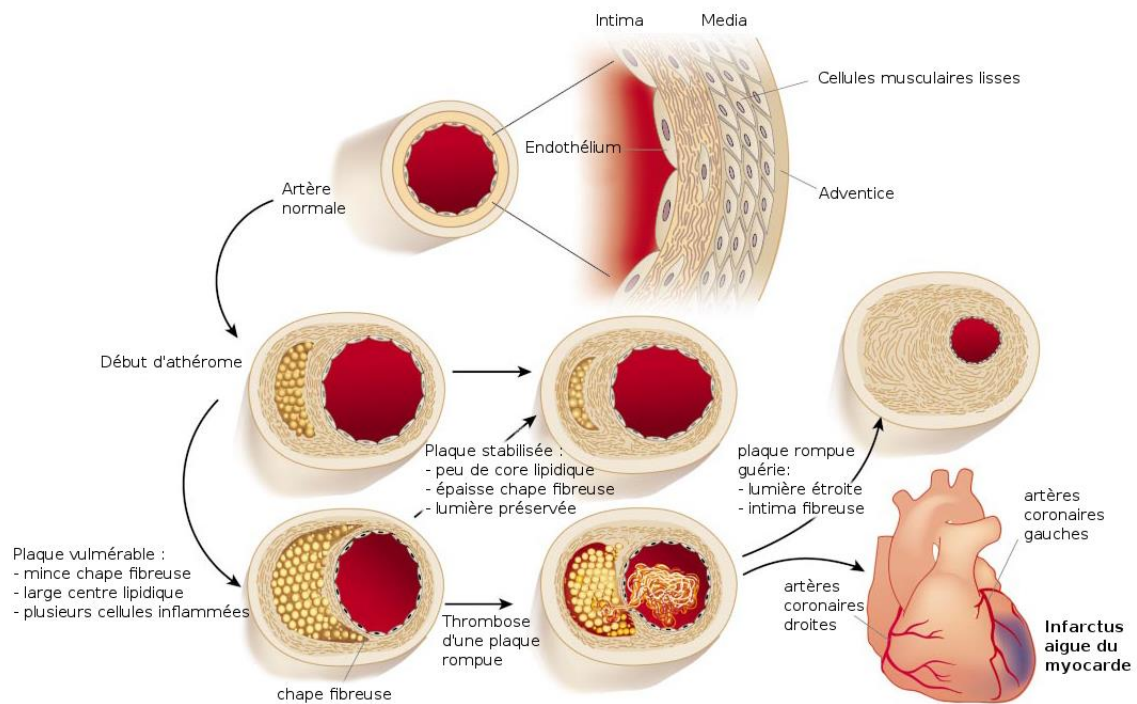


Figure 10: Schéma de l'évolution d'un athérome [Libb02].

1.1.3. DIAGNOSTICS

La phase diagnostique de coronaropathies est déterminante pour privilégier un type de traitement efficace. Grâce à l'évolution des nouvelles technologies et des différentes modalités d'acquisition, les techniques d'exploration par l'image (2D, 3D, 2D+T) occupent une place primordiale dans le diagnostic.

Cependant, le diagnostic lié à ces structures est très complexe et les différentes modalités d'acquisition ne permettent pas forcément d'observer toutes les caractéristiques de la lésion (précision de la nature, la localisation ou encore le degré de gravité). Ces difficultés sont dues en partie à la structure 3D des artères coronaires (fines et tortueuses) ainsi qu'aux mouvements du cœur non rigide et non linéaire.

Plusieurs examens complémentaires permettent de détecter une coronaropathie :

- examens biologiques,
- test d'effort,

- électrocardiogramme (ECG),
- échographie introvasculaire (IVUS),
- coronarographie,
- scintigraphie (imagerie nucléaire),
- angiographie par résonance magnétique (ARM),
- scanner (CT), etc.

La coronarographie (ou angiographie coronarienne à rayons X) est un examen médical invasif des coronaires opacifiées par un produit de contraste utilisant la technique de radiographie à rayons X produisant des images 2D obtenues par projection. La coronarographie s'effectue par cathétérisme, c'est à dire par l'insertion d'un cathéter soit au niveau de l'artère fémorale située au niveau de l'aîne, soit au niveau de l'artère radiale située au niveau du poignet. L'extrémité du cathéter est guidée, via l'aorte, à la naissance des artères coronaires. Un produit de contraste iodé est ensuite injecté dans les vaisseaux, permettant de les visualiser lors de l'acquisition RX. En effet, les vaisseaux sont aussi peu radio-opaques que les tissus mous avoisinants. Le cardiologue peut alors visionner, par le biais d'une série d'images 2D+T et d'une incidence donnée, les artères coronaires mises en évidence. Si l'examen est réalisé via une angiographie rotationnelle RX (voir section 1.1.5 et 1.2.3), les acquisitions des images 2D sont effectuées selon plusieurs incidences (Figure 11).

Cet examen permet ainsi de visualiser la lumière artérielle grâce au produit de contraste. Par conséquent, la morphologie des artères coronaires et leurs structures peuvent alors être observées. De plus, les plaques d'athéromes apparaissent comme des rétrécissements de lumière, cet examen permet ainsi de localiser la sténose. C'est pourquoi, la coronarographie permet un diagnostic fiable grâce à la réalisation d'un bilan anatomique de la lésion. À l'aide de la série d'images 2D acquises, une cartographie de cette lésion est effectuée le plus finement possible. Sa localisation est obtenue ainsi que son degré de gravité (degré de sténose et de calcification) ou le diamètre du vaisseau sténosé. Les caractéristiques de la plaque d'athérome, telles que sa composition, sa taille ou encore sa forme géométrique, sont également étudiées. Cependant, à cause de la nature 2D des images, le chirurgien doit être capable de visualiser mentalement en 3D la structure des artères coronaires du patient, pour effectuer correctement ce diagnostic. Ce bilan permet de choisir également quelle sorte de traitement peut être privilégié. Si l'intervention mini-invasive est envisagée, le choix du matériel (ballon, stent) et de l'endoprothèse (longueur, diamètre, etc.) doivent être déterminés.

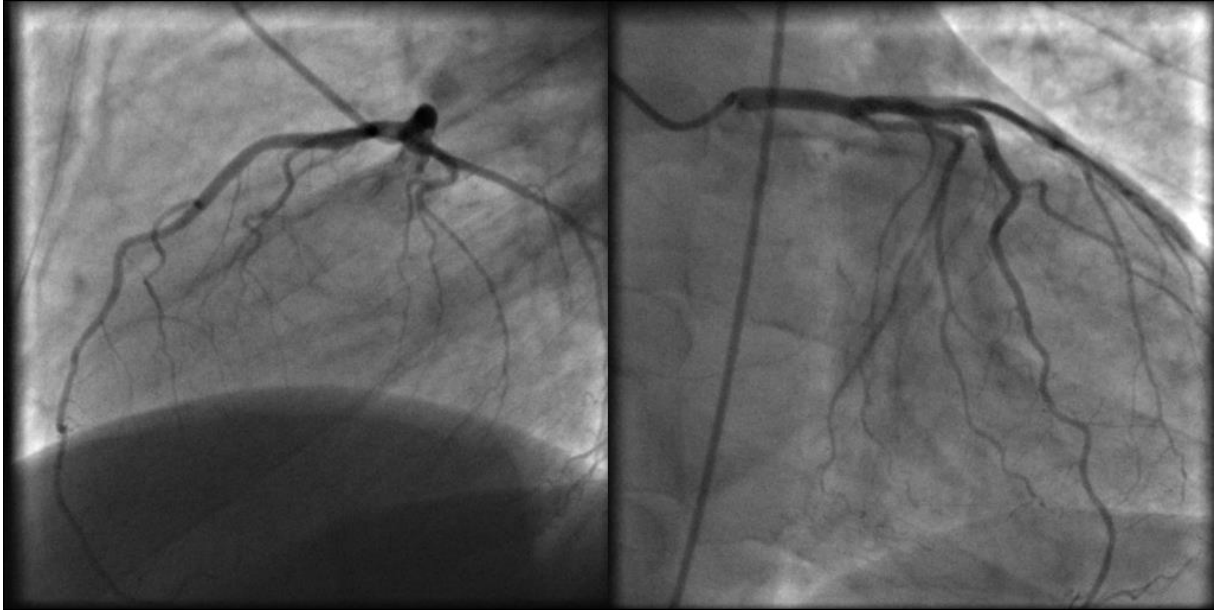


Figure 11: Exemples de projection 2D des artères coronaires sous deux incidences différentes réalisées pendant une coronarographie rotationnelle pour un même patient avec une seule injection de produit de contraste.

1.1.4. TRAITEMENTS

Le traitement d'une coronaropathie vise :

- l'allongement de la survie (réduction des événements cardiovasculaires graves et de la mortalité),
- le ralentissement de la progression de l'athérosclérose,
- l'amélioration de la qualité de vie (réduction du handicap fonctionnel),
- la prévention d'une thrombose et l'amélioration de l'irrigation du myocarde.

Le traitement de la coronaropathie a été privilégié suite au diagnostic. Plusieurs types de traitement peuvent être mis en œuvre :

- médicamenteux (bêtabloquants, inhibiteurs calciques, etc.),
- mini-invasifs (angioplastie, pose d'un stent) et/ou,
- chirurgicaux (pontage coronarien).

Une angioplastie coronaire peut être réalisée à la suite de la coronarographie. Le patient reste alors dans le même bloc opératoire pour ces deux phases consécutives. Cet examen permet de réaliser le traitement dans la continuité de la phase diagnostique. C'est d'ailleurs cette technique de

revascularisation minimalement invasive qui est privilégiée par rapport à une chirurgie plus lourde comme le pontage coronarien (préconisé dans des cas pathologiques plus avancés). Cette dernière présente néanmoins des risques plus graves pour le patient. Cette Angioplastie Coronaire Transluminale Percutanée (ACTP) est une technique guidée par l'image. Elle est réalisée à l'aide d'images angiographiques RX 2D acquises selon une incidence donnée tout au long du traitement.

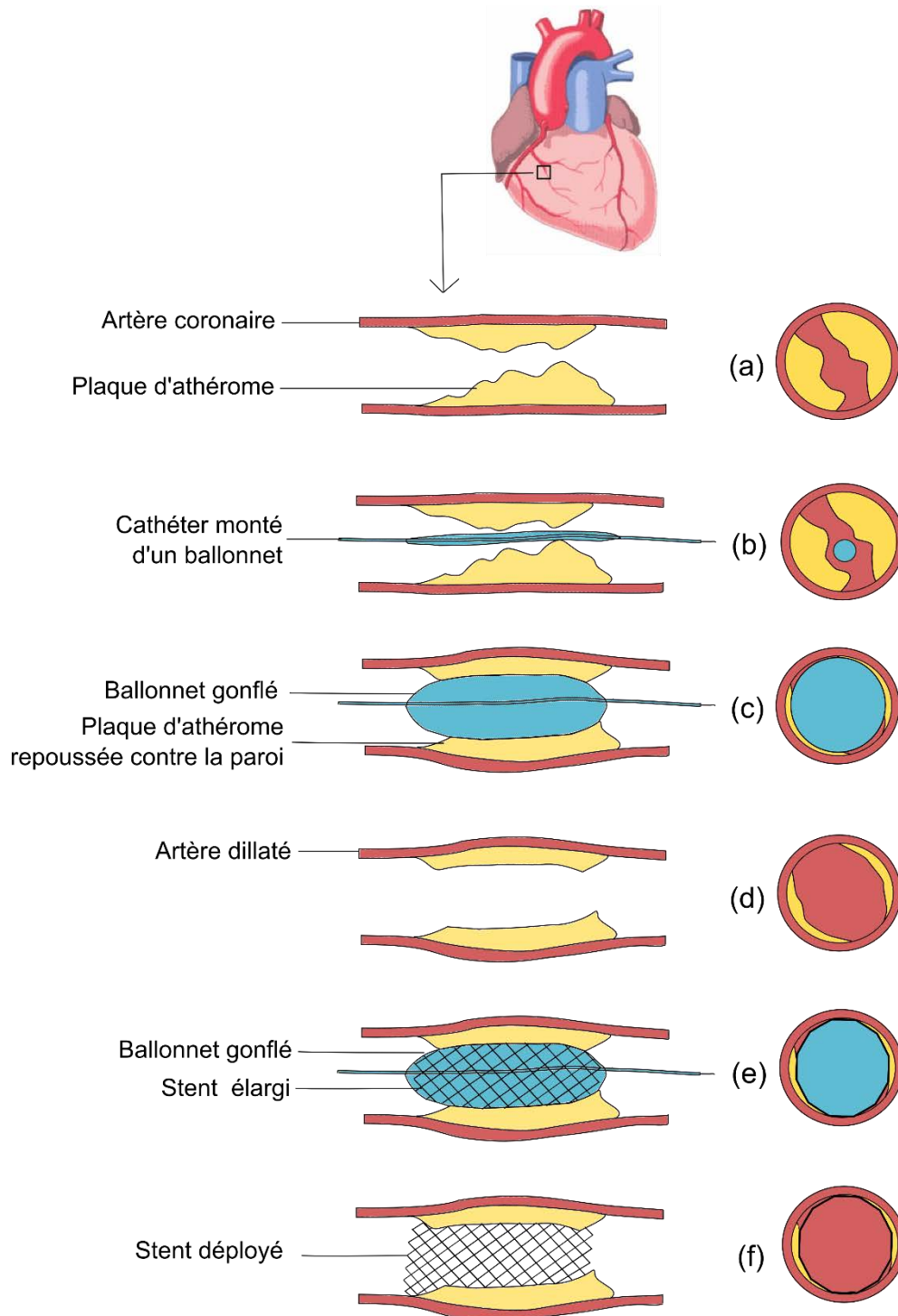


Figure 12: Schéma des étapes d'une angioplastie avec pose de stent.

Comme pour la coronarographie, un cathéter est inséré par voie percutanée au niveau de l'artère fémorale ou radiale. Comportant à son extrémité un ballonnet gonflable, le chirurgien grâce à la visualisation directe des projections RX 2D, l'introduit jusqu'au segment de l'artère coronaire sténosé (Figure 12-(a,b)). Le ballonnet est alors gonflé à plusieurs reprises et à pression variable, permettant alors de repousser la plaque d'athérome contre la paroi de l'artère (Figure 12-(c)). Cette action peut cependant fissurer les dépôts durs de ces plaques. L'artère est alors dilatée, revenant à un diamètre quasiment normal (Figure 12-(d)). La plupart de ces interventions est complétée par la pose d'un stent pour éviter une resténose de l'artère traitée. Cette prothèse métallique est déployée contre la paroi de l'artère à l'aide d'un autre ballonnet, renforçant l'action précédente (Figure 12-(e,f)). Le stent peut être recouvert d'un principe actif (stent actif). Le médicament est conçu pour se libérer progressivement dans l'artère prévenant une resténose.

1.1.5. MODALITE D'ACQUISITION

Comme expliqué précédemment, différentes techniques d'imagerie pour diagnostiquer les pathologies coronariennes existent. Elles peuvent être regroupées en deux catégories : les acquisitions dites invasives et celles réalisées de manière non invasives. Malgré le développement rapide des techniques non invasives tels que le scanner coronaire (CTA) ou l'imagerie par résonance magnétique (IRM), l'angiographie par rayons X (RX) reste la technique de référence pour l'imagerie des artères coronaires, en particulier pour la définition de l'anatomie coronaire. Nous nous focalisons sur ce système d'imagerie utilisé dans notre projet.

La coronarographie et l'angioplastie, utilisant cette même modalité d'acquisition, présentent plusieurs avantages. L'angiographie RX comme expliqué précédemment est exploitée aussi bien en phase diagnostique, avec l'observation des structures coronaires et de la lésion, qu'en phase interventionnelle durant, par exemple, la pose d'un stent. Si une angioplastie est préconisée suite à la coronarographie, elle est alors réalisée dans la continuité du diagnostic. Le patient ne change pas de salle puisqu'il se trouve déjà dans un bloc opératoire équipé d'un tel système de coronarographie. De plus, les résolutions spatiale et temporelle sont supérieures aux autres systèmes d'acquisitions d'images médicales. Une meilleure observation de ces structures complexes est alors envisageable. L'angiographie rotationnelle peut également être acquise en même temps que l'ECG permettant de synchroniser les différentes projections par rapport à une phase du cycle cardiaque (voir section 1.2.3).

Le système d'acquisition (angiographe) est composé d'une table sur laquelle est allongé le patient, d'un arceau (bras en forme de C) aux extrémités duquel se trouve la source des rayons X et le détecteur plan et d'un pivot posé au sol sur lequel est monté le bras. Cet ensemble offre aux cardiologues le choix de déterminer une incidence pour l'intervention grâce aux deux degrés de liberté en rotation pour l'arceau, aux différents degrés de liberté du pivot en fonction du type d'angiographe (Figure 14) ainsi qu'au trois degrés de liberté en translation de la table.

Les paramètres principaux définissant la configuration géométrique de l'arceau sont les suivants (Figure 13) :

- l'angle Cranial/Caudal noté **CRA/CAU** ou φ , définissant le plan de rotation du C-arm dans le plan sagittal, appelé également angle secondaire,
- l'angle antérieur oblique droit/antérieur oblique gauche noté **OAG/OAD** (LAO/RAO en anglais) ou θ , décrivant la course du C-arm dans le plan axial, appelé également angle primaire,
- la distance **SID** (Source – Intensifieur Distance) entre la source des rayons X et le détecteur plan,
- la distance **SOD** (Source – Objet distance) entre la source des rayons X et l'iso-centre, c'est-à-dire la zone corporelle du patient visée par l'angiographie.

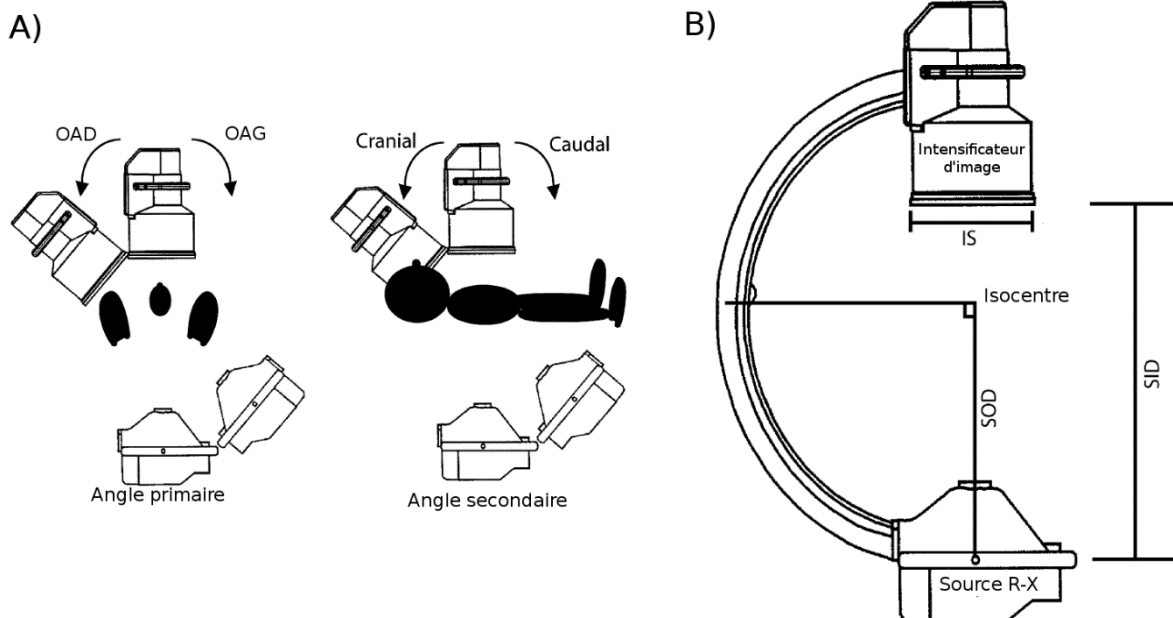


Figure 13: Paramètres géométriques du C-arm. A) Définition des angles OAG/OAD et CRAN/CAU. B) Schéma d'un C-arm avec définition des distances [SSRB05].

Deux modes d'imagerie angiographique existent:

- le mode **statique** (angiographie standard) qui réalise une série d'images 2D+T des artères coronaires selon une incidence choisie où les paramètres décrits précédemment restent constants,
- le système **rotationnel**, nommé *cardiac spin*, introduit au début des années 2000, qui réalise une série d'images 2D+T des artères coronaires selon plusieurs incidences. L'arceau décrit une rotation dans le plan axial. La valeur d'angle CRA/CAU reste constante et égale à 0° et la valeur d'angle LAO/RAO peut varier jusqu'à 200°. La distance SID et SOD reste constante durant la rotation.

Une classification synthétique des systèmes C-arm existants en fonction de leurs principales caractéristiques est décrite dans [Cast10]. En plus de la classification 2D et 3D de l'arceau, il différencie les systèmes possédant un intensificateur conventionnel pour l'acquisition d'images et les nouveaux systèmes avec détecteurs d'état solide (flat-panel) ou numériques, les systèmes fixes ou mobiles (Figure 14), ainsi que les C-arm mono ou biplan (i.e. un ou deux systèmes source-détecteur de rayons X).

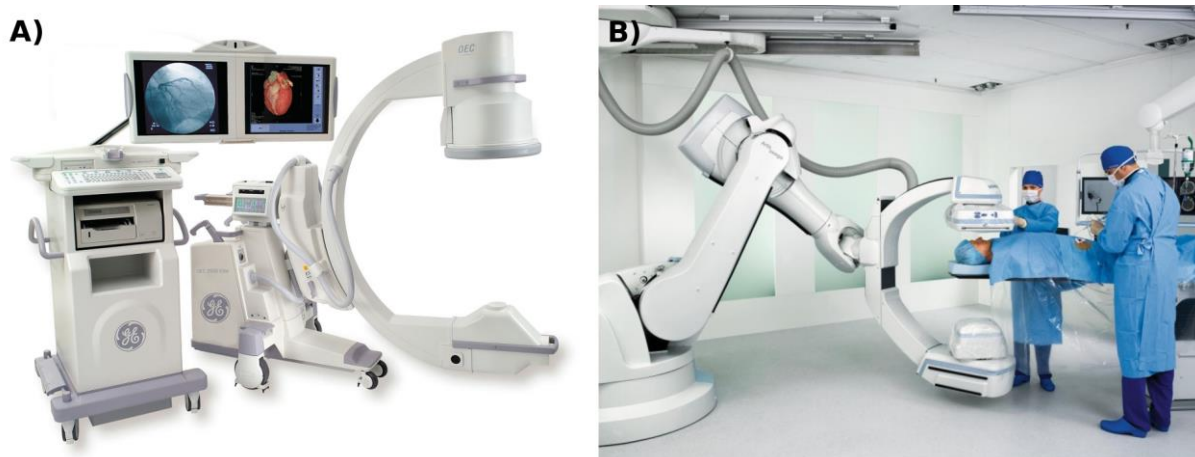


Figure 14 : Exemples de C-arm. A) C-arm mobile. B) C-arm motorisé. (<http://www3.gehealthcare.com> et <http://www.healthcare.siemens.com>)

1.2. MOTIVATION DE LA THESE

Au cours de cette partie, le projet global lié à l'utilisation de l'angiographie rotationnelle RX sera décrit pour pouvoir situer au mieux la problématique de la thèse. Les limitations et difficultés de l'angiographie RX allant de l'acquisition des données à l'interprétation des images projetées seront tout d'abord expliquées pour pouvoir ensuite aborder les solutions proposées à ces problèmes.

1.2.1. LIMITATIONS ET DIFFICULTES LIEES A L'ANGIOGRAPHIE RX

La coronarographie, quel que soit le type d'angiographe utilisé, permet de réaliser le diagnostic et le planning suivi de l'angioplastie pour la phase de traitement. Cependant, de part les caractéristiques de la structure des artères coronaires, ces interventions restent complexes et présentent un certain nombre de limitations.

Tout d'abord, la principale limitation de la coronarographie et de l'angioplastie concerne l'utilisation d'images de projection 2D, représentant les structures 3D complexes des coronaires, produites via l'angiographie statique ou rotationnelle RX. Cette dimension perdue rend l'interprétation complexe et peut entraîner des difficultés pour les différentes phases de diagnostic (localisation et caractérisation de la sténose), de planning (choix de ballon et stent, etc.) et d'intervention (montée du cathéter, déploiement du stent, etc.). Effectivement de par la projection de l'acquisition RX, les vaisseaux peuvent être superposés entre eux ou à d'autres structures environnantes et alors confondus (effet de superposition) ou déformés (effet de raccourcissement des longueurs et variation du facteur d'agrandissement). L'aspect morphologique des artères coronaires peut donc être faussé. En outre, chaque projection 2D peut être parasitée par les structures avoisinantes (vertèbres, cage thoracique, cathéter d'injection, poumons, etc.) et par un taux important de bruits d'acquisition pouvant ainsi masquer les informations des petits vaisseaux (coronaires). Ces images projetées ne permettent de visualiser que la lumière artérielle sans fournir d'information complémentaire sur la paroi artérielle.

Une autre limitation concerne le caractère invasif de ces examens. Il nécessite l'utilisation de produit de contraste iodé pour visualiser les coronaires. Le risque de complication lié à ce produit n'est pas à négliger (réaction allergique ou encore risque d'insuffisance rénale). De plus, l'angiographie utilise les rayons X entraînant une irradiation du patient et dans une autre mesure du personnel

hospitalier réalisant cet examen plusieurs fois par jour et durant de nombreuses années. La dose peut également augmenter en fonction du nombre d'incidences envisagées.

Dans le cas de l'angiographie standard, le cardiologue visualise ces images 2D sous une incidence choisie. A partir des angles de vue standard de la littérature et de ses propres connaissances [GCHM05, GRSM99, ZMDG76], il doit choisir une incidence dite **optimale** montrant le segment sténosé dans sa dimension la plus déployée et sans superposition avec d'autres segments pour ne pas fausser son diagnostic, le planning envisagé, ou l'intervention à cause de la nature 2D des images projetées. N'ayant pas de représentation 3D de l'arbre coronaire, le clinicien peut être amené malgré tout à réaliser plusieurs angiographies sous différents angles pour obtenir une vue correcte du segment sténosé ainsi qu'une meilleure représentation mentale 3D. Le nombre d'acquisitions d'images 2D+T sous différentes incidences, et donc le nombre d'injections de produit de contraste afin de visualiser les artères coronaires, peut alors se révéler élevés pour un patient si son anatomie coronaire s'avère complexe. Effectivement, le cardiologue, pour réaliser le diagnostic, doit avoir une idée précise de la structure 3D des artères coronaires. Or, comme expliqué dans la section 1.1.1, l'anatomie des artères coronaires est très variable inter-individu. La configuration géométrique des artères coronaires est très irrégulière. Avec l'âge, elles deviennent plus tortueuses et présentent des plaques d'athérome plus ou moins calcifiées. Les doses d'irradiation et d'utilisation de produit de contraste sont alors augmentées et peuvent être dangereuses pour l'organisme. Pour la coronarographie, cette difficulté est en partie contournée grâce à l'utilisation de l'angiographie rotationnelle (*cardiac spin*). Cette autre technique de radiologie permet d'obtenir des vues multiples d'un vaisseau, sous différents angles de vues, lors d'une même injection de produit de contraste [KuND02]. Une vision pseudo 3D est fournie au chirurgien permettant d'obtenir un diagnostic le plus fiable possible. Il doit cependant continuer à se représenter mentalement cette structure en 3D.

Une autre difficulté concernant cette intervention endovasculaire est liée aux mouvements complexes des artères coronaires ainsi qu'à leur faible diamètre. Comme ces dernières suivent le mouvement du cœur, la phase de diagnostic peut s'avérer complexe, de même que pour la phase d'intervention. Le déplacement du cathéter jusqu'à la zone à traiter est effectué dans une structure haute en mouvement, selon les contractions du cœur. De plus, ce déplacement s'appuie uniquement sur des observations 2D acquises pendant l'intervention et selon une incidence donnée. Le guidage du cathéter et la pose du stent peuvent alors s'avérer difficiles selon la situation de la zone de lésion (présence de bifurcation, état de la paroi, etc.). Lors de la montée du cathéter dans les artères coronaires, les plaques d'athérome ne doivent pas être heurtées au risque de se déchirer et de provoquer une embolie [ChCa00].

Avec ces limitations, la sténose peut être sous-estimée, comme seule la lumière des vaisseaux est mise en évidence, une analyse fiable de la plaque d'athérome (épaisseur, localisation des zones de calcification, etc.) n'est pas permise. De plus le choix du type de cathéter ou de l'endoprothèse peut être inexact. Toutes ces limitations peuvent donc conduire à des erreurs dans l'estimation des caractéristiques d'une lésion, à un choix inadapté des instruments de revascularisation, à un surdosage en rayons X, à des risques d'effets secondaires accrus suite à l'injection répétée de produit de contraste ou encore à un placement incorrect du stent sur la paroi.

1.2.2. PROJET

Un patient à forte suspicion de lésions coronaires est directement orienté vers la salle d'intervention pour y passer un examen coronarographique, puis dans la continuité, si cela s'avère nécessaire une angioplastie transluminale est réalisée. Ce type d'intervention est appliqué le plus souvent à l'inverse du pontage coronarien qui est une lourde opération chirurgicale préconisée dans les cas les plus graves. Dans la majorité des cas, les données préopératoires ne sont pas disponibles.

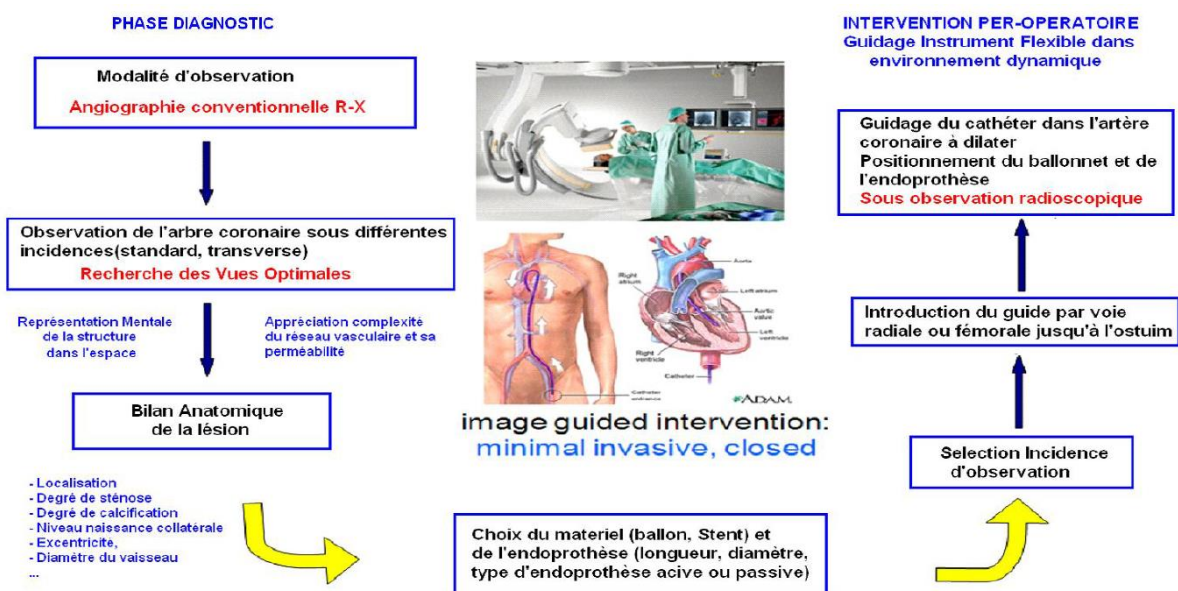


Figure 15: Etapes diagnostiques de l'atteinte coronarienne et procédure de revascularisation.

L'utilisation de l'angiographie rotationnelle à rayons X permet de développer de nouveaux outils pour assister le cardiologue aussi bien dans sa prise de décision de diagnostic ou de planning que dans le guidage du geste interventionnel. L'objectif est de minimiser les erreurs d'interprétation liées à la nature 2D des images et d'augmenter la précision et la sécurité du geste thérapeutique ainsi que de la fiabilité de l'intervention. A partir d'angiographies rotationnelles RX, ces outils concernent (Figure 15) :

- la reconstruction 3D de l'arbre coronaire pour chaque phase cardiaque,
- la caractérisation de la lésion,
- la recherche de l'incidence optimale,
- le guidage du cathéter dans un environnement dynamique.

1.2.3. RECONSTRUCTION 3D DES ARTERES CORONAIRES

Une modélisation 3D+T des arbres coronaires peut pallier les différentes limitations liées à l'angiographie RX. Cette modélisation 3D, construite en phase peropératoire à partir des observations radiographiques 2D acquises durant la phase diagnostique, permet de visualiser l'arbre coronaire 3D selon différentes phases cardiaques et sous différentes orientations contournant les principales limitations de l'examen. Elle entraîne entre autre une meilleure compréhension de la structure des artères coronaires, de la morphologie de la zone à traiter, c'est-à-dire la lésion. Le choix du dispositif à mettre en place et le geste du chirurgien seront facilités pour cette intervention [GVLW06].

Les artères coronaires peuvent être reconstruites à partir de plusieurs projections acquises lors de l'angiographie rotationnelle à rayons X selon une phase donnée pour une seule injection de produit de contraste. L'acquisition dynamique de ces images 2D est réalisée grâce à la rotation autour du patient de l'arceau radiologique. Ainsi, 80 à 150 projections des arbres coronaires droits et gauches sont acquises sur une durée de 4 à 5 cycles cardiaques correspondant à une seule injection de produit de contraste (Figure 16). Cette séquence de projections est exploitée pour réaliser une reconstruction 3D de l'arbre coronaire selon une phase donnée ou l'ensemble de la séquence, facilitant l'intervention du chirurgien cardiologue. Une des principales difficultés pour reconstruire en 3D les artères coronaires à partir de l'angiographie rotationnelle à rayons X vient du mouvement complexe du cœur. L'acquisition des projections 2D se déroule dans le temps. La structure étudiée subit les mouvements du cœur. Chaque projection représente les artères coronaires à différents instants du cycle cardiaque (4 ou 5 projections représentent la même phase du cycle cardiaque en fonction du nombre de cycles

présents), en plus d'être acquise selon une incidence différente. Une autre difficulté influençant la reconstruction vient de la finesse des structures ainsi que du bruit présent dans les images de projection. Cependant, cette acquisition rotationnelle est effectuée avec seulement une dose de produit de contraste limitant le nombre de cycle cardiaque où les structures des artères coronaires sont visibles dans les images de projection. Selon certains travaux [GAGM09, KuND02, MWMG04, RMNA04], l'utilisation de cette technique permettrait de réduire l'irradiation et le volume du produit de contraste tout en conservant la même précision du diagnostic que l'angiographie non-rotationnelle.

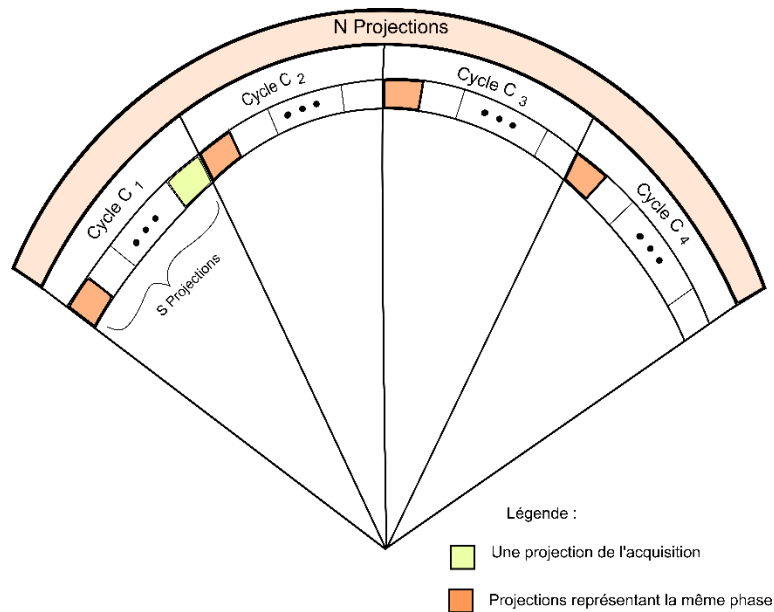


Figure 16 : Séquence de projections acquise sur une durée de 4 cycles cardiaques. Les projections (orange) représentant la même phase sont utilisées pour la reconstruction de l'arbre coronaire à cet instant t .

Différentes méthodes pour reconstruire un objet en mouvement à partir de projections 2D d'un capteur en mouvement existent dans la littérature. Deux approches sont généralement considérées [ÇGGF16, ScNG09].

La première envisage la reconstruction 3D à partir des projections associées à une même phase du cycle cardiaque [Ouki15]. L'arbre est reconstruit pour une phase donnée. Sur les projections représentant une même phase cardiaque (utilisation de l'ECG ou des informations de l'image), les artères sont quasiment à la même position mais observées sous des angles différents. Le choix des projections (4 à 5 généralement en fonction du nombre de cycles cardiaques exploitables) peut être laborieux à cause des différents mouvements subis par les artères coronaires. Ce sous échantillonnage

angulaire conduit à résoudre un problème inverse mal posé difficile à résoudre du fait de la complexité des images. La reconstruction 3D des artères coronaires peut être ainsi réalisée pour différentes phases du cycle cardiaque en sélectionnant différentes projections permettant ainsi d'obtenir une modélisation 3D pour chaque phase du cycle cardiaque.

Cependant le nombre de projections utilisées pour effectuer la reconstruction à un instant t peut varier en fonction des conditions et des caractéristiques de l'examen :

- l'acquisition des projections peut débuter sans que le produit de contraste soit présent dans tout le réseau des artères coronaires (présences de projections inexploitable),
- la durée d'un cycle cardiaque, lié à la fréquence cardiaque, peut varier d'un patient à un autre selon son âge, son stress, s'il vient d'avoir une attaque, s'il est sportif, etc.,
- le produit de contraste peut s'être dispersé dans l'organisme avant la fin de l'acquisition,
- la structure peut ne pas être centrée dans le champ d'observation et peut apparaître tronquée sur certaines incidences.

La deuxième approche passe par une estimation préalable du mouvement des artères coronaires sur l'ensemble du cycle cardiaque. Cette estimation est incorporée dans la procédure de reconstruction tomographique et permet d'exploiter toutes les projections (de 80 à 150 projections environ) pour un instant donné. Cette reconstruction compensée en mouvement consiste à projeter la matrice de mouvement estimée 3D+T dans l'espace 2D pour compenser la position des pixels sur chaque image de la séquence, puis de réaliser ensuite une reconstruction tomographique dans un cadre statique avec les points corrigés. Toutes les phases du cycle sont exploitées pour pouvoir déformer une reconstruction (celle de la phase statique) afin de la faire correspondre à toutes les autres phases cardiaques. Cette reconstruction s'appuie sur des méthodes itératives [BJGB09, BMVA06, HSDG08, Xie12] ou de rétroprojections filtrées [RLBP10].

Afin de simplifier le modèle, une segmentation préalable des structures coronaires est considérée. Elle permet de simplifier la recherche du mouvement aux seules structures d'intérêts et par la suite à se réduire à un problème de reconstruction épars en considérant un certain nombre de simplifications dans la résolution du problème inverse.

1.2.4. DONNEES : SEQUENCE 3D+T D'ARBRES CORONAIRES

Dans cette partie, l'obtention des données d'entrée de notre méthode de recherche d'incidence optimale globale, représentant une séquence dynamique complète de volumes d'artères coronaires, est décrite ainsi que leurs transformations en arbres à partir de leurs squelettes.

1.2.4.1. DEFINITIONS : NOTIONS DE GRAPHE ET D'ARBRE

Tout au long de ce manuscrit des notations provenant de la théorie des graphes seront utilisées. Nous allons définir brièvement ces notations. Pour une présentation plus complète des graphes voir [Berg73, Boll13, BoMu76, Pier00].

Soit $G = (V, E)$ un graphe non orienté où V est un ensemble de sommets ou de nœuds (*vertex* ou *node* en anglais) et E un ensemble d'arêtes (*edge* en anglais) représentant un ensemble de paires de nœuds (v_i, v_j) avec $v_i, v_j \in V$. La *taille* du graphe G correspond au cardinal de l'ensemble E , c'est-à-dire au nombre d'arêtes du graphe. L'*ordre* du graphe G correspond quant à lui au cardinal de l'ensemble V , c'est-à-dire au nombre de sommets du graphe.

Un graphe $G = (V, E)$ *orienté* représente un graphe G dont les arêtes sont orientées (sens bien précis). Les arêtes orientées sont appelées alors arcs.

Un *graphe simple (orienté ou non orienté)* $G = (V, E)$ représente un graphe pour lequel il existe au plus une arête liant deux sommets.

Un *graphe pondéré (orienté ou non orienté)* $G = (V, E)$ est un graphe dont les arêtes possèdent un poids, c'est-à-dire un coût de passage d'un sommet à un autre. Ce type de graphe est utilisé dans certains types de problèmes comme celui du voyageur de commerce.

Nous allons voir maintenant quelques caractéristiques des graphes.

Deux nœuds $v_i, v_j \in V$ du graphe $G = (V, E)$ sont dits *adjacents* si $\exists (v_i, v_j) \in E$. Pour les graphes non orientés, on peut parler également de voisins. Le *degré* d'un sommet représente le nombre d'arêtes entrant et sortant de ce sommet (en fonction de s'il est orienté ou non). Le degré d'un graphe correspond quant à lui au plus grand des degrés de ces nœuds.

Un *chemin* p (path en anglais) dans le graphe $G = (V, E)$ représente une suite $p = \{v_0, v_1, \dots, v_i, \dots, v_n\}$ tel $(v_i, v_{i+1}) \in E$ pour $0 \leq i \leq n$, c'est-à-dire que les nœuds de la suite p sont reliés par une arête (ils sont adjacents). Le nœud v_0 représente le début du chemin p (*source* en anglais) et le nœud v_n correspond à la fin du chemin p (*target* en anglais). $n + 1$ est appelé la longueur de p .

Un *cycle* du graphe non orienté $G = (V, E)$ représente un chemin $p = \{v_0, v_1, \dots, v_i, \dots, v_n\} \in G$ tel que $v_0 = v_n$. Pour un graphe orienté $G = (V, E)$, on parle de circuit.

Un graphe est dit *complet* $G = (V, E)$ si l'ensemble des nœuds de V sont reliés entre eux. Un graphe d'ordre n est appelé *n-clique*. Chaque sommet $v_i \in V$ est de degré $n - 1$. Une *clique* C d'un graphe $G = (V, E)$ représente un sous arbre complet de G . Un graphe peut avoir plusieurs cliques.

Un graphe $G = (V, E)$ est dit *connexe* si $\forall v_i, v_j \in V$, il existe un chemin reliant v_i et v_j . Le nombre de connexités du graphe G correspond au nombre de composantes connexes.

Deux graphes $G = (V, E)$ et $G' = (V', E')$ sont dit *isomorphes* s'ils possèdent le même nombre de nœuds connectés de la même façon, c'est-à-dire s'il existe une bijection entre les deux ensembles de nœuds V et V' telle que la paire de nœuds $(v_i, v_j) \in E$ si et seulement si $(f(v_i), f(v_j)) \in E'$.

Nous allons maintenant aborder quelques notions pour un type spécifique de graphe : les arbres.

Soit $G = (V, E)$ un graphe. Il est dit un **arbre** s'il est connexe et sans cycle. Le nœud $r \in V$ est une racine (*root* en anglais) si pour tout sommet $v_i \in V$ il existe un chemin reliant r et v_i . L'arbre noté $T = (V, E, r)$ est appelé *arbre enraciné de racine r* . La notion de racine implique celle d'arborescence, soit une hiérarchisation des nœuds de V . Dans notre cas, le choix de la racine introduit également une orientation implicite des arcs de l'arbre. Chaque arc s'oriente en respectant la profondeur des nœuds. La *profondeur* d'un nœud $v_i \in V$ correspond à son niveau dans l'arbre $T = (V, E, r)$, c'est-à-dire la longueur du plus court chemin le reliant à la racine r .

Nous allons maintenant décrire quelques caractéristiques de $T = (V, E, r)$ un arbre enraciné orienté de racine r .

Soit $e = (v_i, v_j) \in E$ un arc de T avec $v_i, v_j \in V$. v_i est appelé la source (source également en anglais) de e et v_j la destination (*target* en anglais) de e .

Soit $v_i \in V$ un nœud de l'arbre T avec $v_i \neq r$. Il existe un et un seul nœud $v_j \in V$ tel que $(v_j, v_i) \in E$. v_j est appelé le *père* de v_i et l'arc (v_j, v_i) représente l'arc entrant dans le nœud v_i . Seule la racine r ne possède pas de père. Les nœuds se situant sur le chemin $p(r, v_i)$ sont des nœuds parents de v_i .

Soit $v_i \in V$ un nœud de l'arbre T . L'ensemble des nœuds $fils(v_i) = \{v_j \in V \mid (v_i, v_j) \in E\}$ représente les *fils* du nœud v_i , ce sont les descendants directs.

Un nœud $v_i \in V$ est dit *feuille* (*leaf* en anglais) si $v_i \neq r$, s'il est de degré 1 et s'il n'a pas de descendant c'est-à-dire si $fils(v_i) = \emptyset$.

La *hauteur* d'un arbre $T = (V, E, r)$ représente la longueur du plus long chemin de la racine de l'arbre r jusqu'à une feuille.

Soit deux nœuds $v_i, v_j \in V$ de T n'ayant pas de relation père/fils. Ces deux nœuds peuvent être considérés comme *frère* (direct ou non). Il existe alors un nœud commun $v_k \in V$ tel que v_k soit un nœud parent à la fois de v_i et de v_j . Ce nœud parent identique est appelé le *plus proche parent commun* de v_i et de v_j .

1.2.4.2. OBTENTION DE LA SEQUENCE D'ARBRES CORONAIRES 3D

La reconstruction 3D des artères coronaires est un problème complexe et mené en parallèle également au sein l'équipe IMPACT du LTSI [Ouki15, Xie12]. Les résultats obtenus ne sont pas encore exploitables dans notre cas puisque nous souhaitons de nombreux volumes représentant différents instants du cycle cardiaque pour différents patients donnés. C'est pourquoi nous utilisons pour nos tests un modèle dynamique 3D de l'arbre artériel coronarien construit à partir de données patient acquises directement en 3D à partir d'un scanner GE 64 multi barrettes et d'une segmentation de volumes de la séquence par une méthode semi-automatique basée sur une technique de suivi des lignes centrales [GJBG10, LABF09, YBTS06]. Une séquence de 10 volumes binaires représentant l'arbre coronarien a ainsi été reconstruite tous les 10% de l'intervalle RR du signal ECG.

A partir de la reconstruction 3D ou modélisation 3D d'une phase donnée [ÇGGF16, LABF09], le squelette des artères coronaires de cette phase peut être calculé (squelettisation [LoBe04, Loho01]) (Figure 17 et Figure 18). L'extraction des lignes centrales [GVBL13] s'effectue branche par branche en remontant à partir de chaque extrémité jusqu'à la racine ou une ligne déjà parcourue (Figure 19). Ces

lignes centrales sont représentées sous forme de splines cubique d’Hermite utilisant l’interpolateur décrit par Kochanek et Bartels [KoBa84]. La spline de Kochanek module le comportement des tangentes aux n points de contrôle à interpoler avec une tension, une polarisation et un paramètre de continuité. Les lignes centrales, obtenues pour chaque branche peuvent former intuitivement une hiérarchie, c’est-à-dire un graphe, utile par exemple pour la mise en correspondance des différentes artères coronaires tout le long du cycle cardiaque.

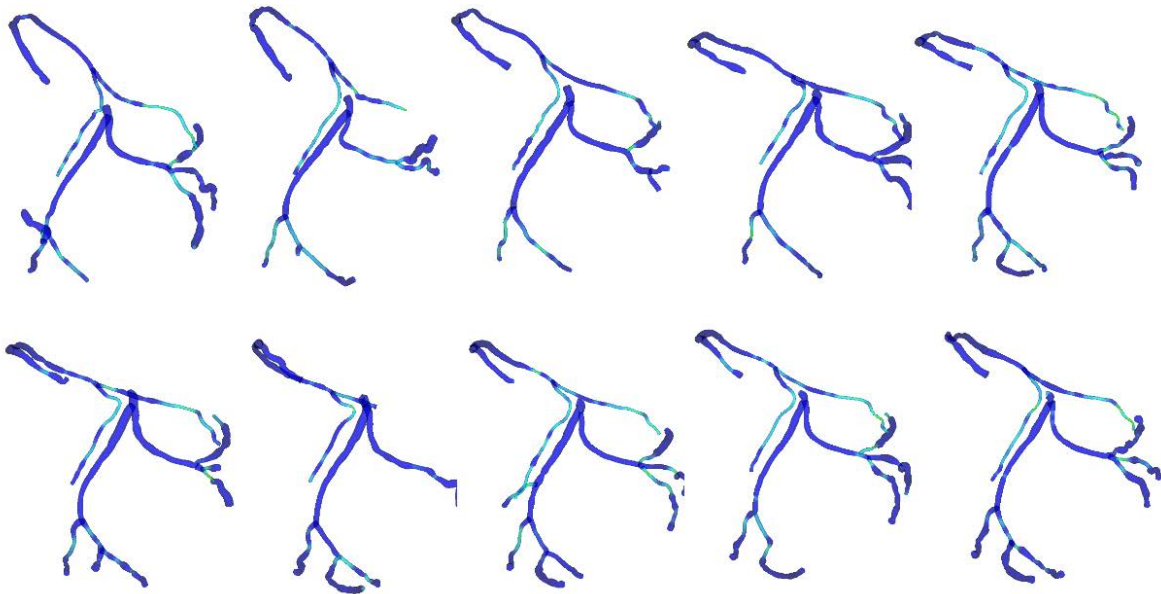


Figure 17 : Exemples de séquence d'arbres coronaires 3D droit et gauche représentant différents instants du cycle cardiaque obtenus après modélisation 3D.

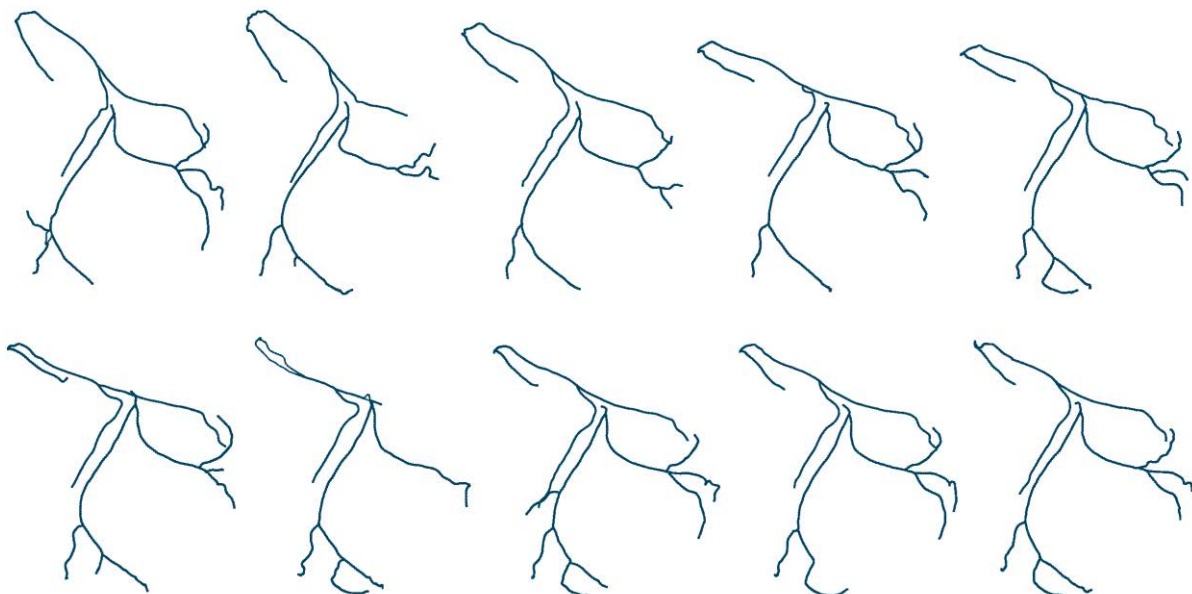


Figure 18: Lignes centrales de la séquence d'arbre coronaire 3D représentée dans la Figure 17.

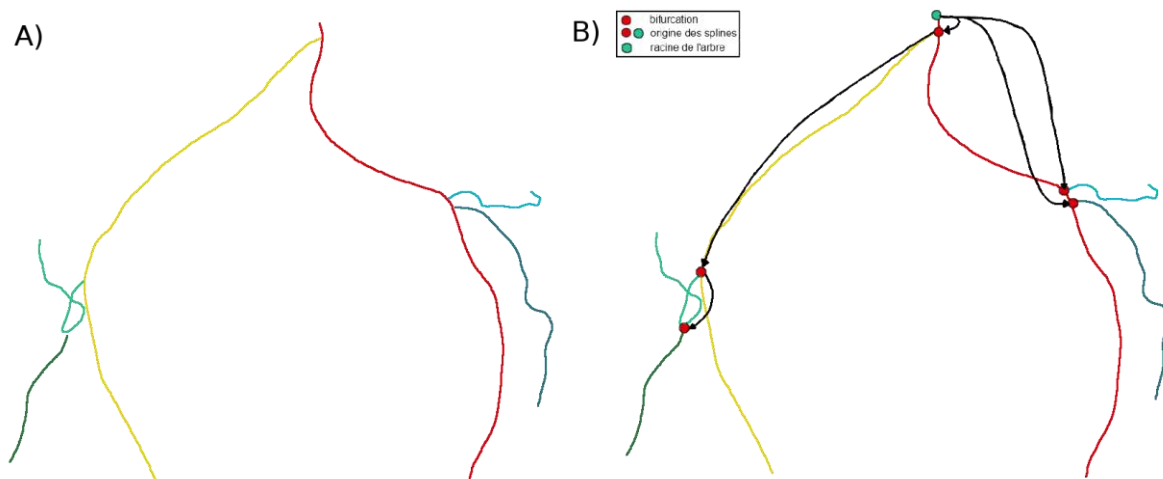


Figure 19 : A) Extraction des lignes centrales sous forme de spline de couleur différentes. B) Détection des bifurcations et construction de la hiérarchie du graphe.

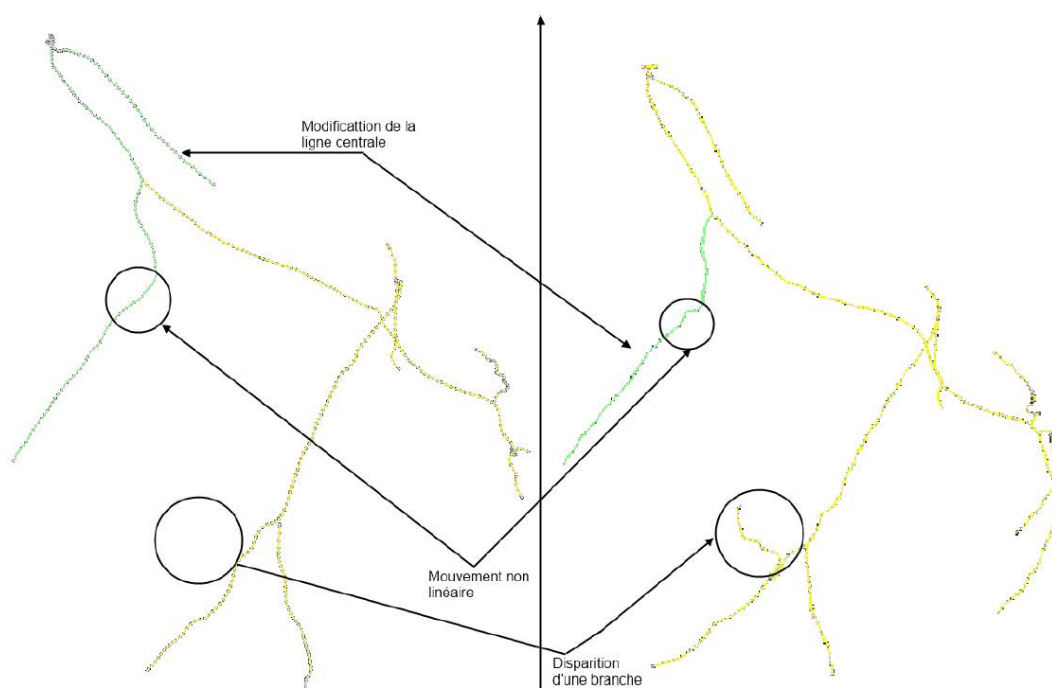


Figure 20 : Arbres coronaires non isomorphes : difficulté pour la mise en correspondance.

Entre chaque phase du cycle cardiaque, la reconstruction de l'arbre 3D et l'extraction des lignes centrales peuvent différer (Figure 20). En effet, les arbres coronaires 3D de la séquence sont **non-isomorphes**. Une branche peut effectivement disparaître, apparaître ou encore être de longueur ou de tortuosité différente selon la phase cardiaque. Ce problème rend alors difficile la mise en correspondance des artères pour chaque phase. Ces variations peuvent être dues à différentes difficultés liées aux images 2D acquises (présence de bruit, réduction de produit de contraste, superposition des structures, etc.), au mouvement du cœur, à la respiration du patient ou encore des différences liées à l'angle de vue, rendant ainsi la segmentation et la reconstruction 3D très difficile.

1.2.4.3. REPRESENTATION DES DONNEES SOUS FORME D'ARBRES

Les arbres coronaires 3D gauche et droit (volumes et lignes centrales) issus de reconstructions ou modélisations 3D tels que ceux issus d'angiographie rotationnelles RX [Ouki15, Xie12] correspondent à nos données d'entrée de notre programme comme expliqué précédemment. Pour chaque arbre gauche et droit, les squelettes des vaisseaux 3D d'une même phase cardiaque vont être représentés par un arbre orienté enraciné $T = (V, E, r)$.

Par la suite, nous prenons pour exemple les arbres coronaires gauches reconstruits sur plusieurs phases cardiaques. Soit $RS_{0,I-1} = \{R_0, \dots, R_i, \dots, R_{I-1}\}$ une séquence dynamique de I arbres coronaires gauches représentant successivement une phases t_i du cycle cardiaque. Chaque i coronaire gauche reconstruit contient des informations de volume R_i (mesh) et le squelette (lignes centrales). Basée sur les lignes centrales, une structure d'arbre orienté enraciné $T_i = (V_i, E_i, r_i)$ est extraite pour chaque volume R_i de la séquence $RS_{0,I-1}$ (Figure 21). La séquence $S_{0,I-1} = \{T_0, \dots, T_i, \dots, T_{I-1}\}$, contenant les structures hiérarchiques des lignes centrales des arbres, est ainsi obtenue. Nous garderons ces notations tout le long de ce manuscrit.

Les nœuds présents dans $V_i = \{v_i^1, \dots, v_i^{\alpha_i}, \dots, v_i^{n_i}\}$ correspondent aux points particuliers de la structure comme la racine de l'arbre r_i , les bifurcations ou les points terminaux des branches. $n_i = \text{Card}(V_i)$ représente le nombre de nœuds, soit l'ordre de T_i .

Les arcs appartenant à l'ensemble E_i représentent les chemins reliant deux nœuds entre eux, c'est-à-dire des portions de la ligne centrale entre le nœud source et le nœud de destination.

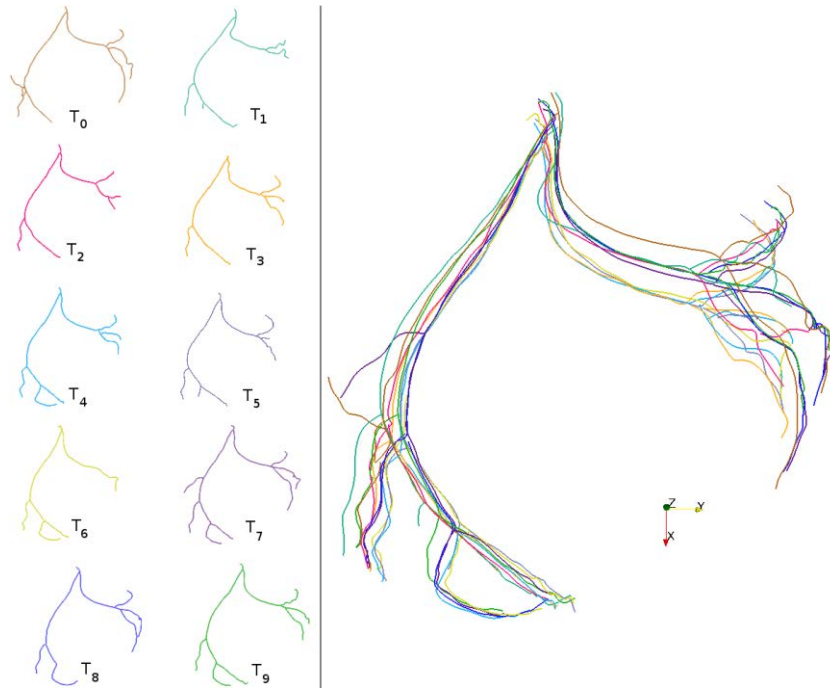


Figure 21 : Détails d'une séquence $S_{0,9}$ de 10 arbres coronaires gauches T_i (lignes centrales). L'image de droite nous montre la superposition de cette séquence permettant d'observer le mouvement des artères selon les phases du cycle cardiaque.

Détermination de la hiérarchie :

Dans un premier temps, pour créer les structures d'arbres à partir des lignes centrales extraites (représentées par des splines dans la Figure 19), les différentes bifurcations sont détectées afin de créer la hiérarchie de cet arbre. Une bifurcation représente le point de départ d'une spline qui peut être rattachée à une autre spline, si elle n'est pas la spline mère de l'arbre.

Pour pouvoir déterminer la hiérarchie entre les splines de l'arbre T_i (lignes centrales), une matrice de distance $distMap$ de taille m^2 est créée avec m le nombre de splines de l'arbre. Les éléments de cette matrice correspondent à la distance euclidienne la plus courte entre les points de rattachement de chaque spline et le « corps » des autres splines, c'est-à-dire les points des splines excluant leurs points de départ. L'Algorithm 1 décrit le remplissage de la matrice $distMap$. L'Algorithm 2 explique quant à lui les calculs de la distance intervenant dans le remplissage de cette même matrice. $distMap$ est une matrice de valeurs positives telle qu'à l'initialisation :

$$\forall k \in [1; m], distMap[k][k] = +\infty$$

Cette matrice est exploitée pour récupérer la hiérarchie du graphe. Effectivement, nous pouvons dire que la spline l est la plus proche du début de la spline k ($l \neq k$) avec lorsque l'équation

suivante est vérifiée : $distMap[k][l] = \min(distMap[k])$. Ces structures de données permettent alors de pouvoir récupérer le point de la spline l qui est connecté à la spline k et la distance qui les sépare.

Cette information nous permet de construire un vecteur de connectivité $CDistMap < int, Distance >$ de taille m correspondant au nombre de splines. Il contient l'indice d'une spline ainsi qu'une distance et est rempli selon la formule suivante (Algorithm 3) :

$$CDistMap[k] = \langle \underset{l}{\operatorname{argmin}}(distMap[k][l]), distMap[k][l] \rangle$$

Pour faciliter la compréhension, pour une spline quelconque k , le premier élément correspondant à l'indice d'une spline peut être récupéré de la manière suivante : $CDistMap[k].left$. La distance quant à elle, peut être récupérée et exploitée en utilisant : $CDistMap[k].right$. Grâce à ce vecteur $CDistMap$, la racine de l'arbre peut être obtenue facilement. Elle correspond à la spline k où la distance contenue dans le vecteur $CDistMap$ est la plus grande. La spline k doit donc vérifier l'équation suivante : $\underset{k}{\operatorname{argmax}}(CDistMap[k].right)$. La spline k (qui est différente de la spline racine) est la mère de la spline $j = CDistMap[k].left$.

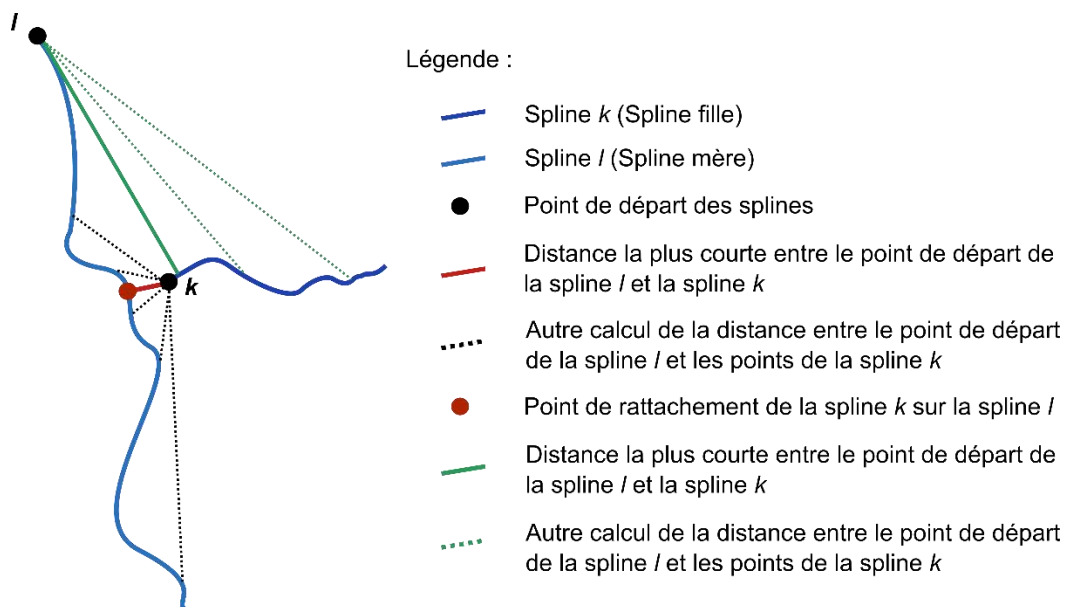


Figure 22 : Schéma expliquant le calcul de la hiérarchie des splines : calcul de la distance associée à la matrice $distMap$.

La Figure 22 donne un exemple de spline mère et fille ainsi qu'une idée du calcul de la distance associée à la matrice $distMap$. Sur cette figure, le segment rouge correspond à la distance minimale

entre ces deux splines avec la spline k comme référence. Le segment plein vert se réfère quant à lui également à la distance minimale entre ces deux splines mais avec la spline l comme référence. Ce sont ces distances qui sont stockées dans la matrice $DistMap$, puis dans le vecteur $CDistMap$. La distance verte est bien plus grande que la distance de couleur rouge, ce qui revient à dire que la spline l dans cet exemple est la mère de la spline k et dans ce cas également la racine de l'arbre.

La Figure 19-B) nous montre une hiérarchie de lignes centrales d'un arbre coronaire obtenue après l'application de l'algorithme.

Algorithm 1 : Détermination de la matrice de distance

Input : m le nombre de splines dans l'arbre coronaire. $spline[k]$ la spline d'index k .

Output : $distMap[m][m]$ la matrice de distance entre chaque spline.

```

1:  procedure COMPUTEDISTMAP
2:    for  $k \leftarrow 0, m$  do
3:      for  $l \leftarrow 0, m$  do
4:        if  $k == l$  then
5:          //La distance entre les splines n'est pas à calculer : valeur par défaut.
6:           $distMap[k][l] \leftarrow +\infty$ 
7:        else
8:          //Algorithme 2 : calcul de la distance avec la liste de points de chaque spline.
9:           $distMap[k][l] \leftarrow DISTANCE ( spline[k], spline[l] )$ 
10:       end if
11:     end for
12:   end for
13: end procedure

```

Algorithm 2 : Détermination de la distance euclidienne entre deux splines

Input : $spline1$ et $spline2$, deux listes de points de splines.

Output : $m_distance$ correspond à la distance minimale entre le début de la spline de référence et le point le plus proche de la spline à parcourir. $m_closestPoint$ correspond au point le plus proche de la spline à parcourir par rapport à la spline de référence.

```

1:  procedure DISTANCE (spline1, spline2)
2:      m_distance  $\leftarrow +\infty$ 
3:      //Récupération du point de départ de la spline de référence.
4:      point1  $\leftarrow$  spline1.getPoint(0)
5:      for p  $\leftarrow$  0, spline2.getNumberOfPoints() do
6:          point2  $\leftarrow$  spline2.getPoint(p)
7:          squareDistance  $\leftarrow$  distance2BetweenPoints(point2, point1)
8:          if squareDistance < m_distance then
9:              //mise à jour des informations pour trouver le point le plus proche de la spline de
10:             // référence ainsi que sa distance
11:             m_distance = squareDistance
12:             m_closestPoint = point2
13:         end if
14:     end for
15:     return m_distance
16: end procedure

```

Algorithm 3 : Calcul du vecteur de connectivité et de la racine de l'arbre

Input : *m* le nombre de splines dans l'arbre coronaire. *spline*[*k*] la spline d'index *k*.

distMap[*m*][*m*] la matrice de distance entre chaque spline.

Output : *CDistMap* < *int*, *Distance* > le vecteur de connectivité de taille *m*.

```

1:  procedure CONNECTIVITYDSTM
2:      distanceRoot  $\leftarrow$  0
3:      //Pour une spline k, nous recherchons la spline l la plus proche.
4:      for k  $\leftarrow$  0, m - 1 do
5:          bestSpline  $\leftarrow$  0
6:          lowDist  $\leftarrow +\infty$ 
7:          for l  $\leftarrow$  0, m - 1 do
8:              if distMap[k][l] < lowDist then
9:                  bestSpline = l
10:                 lowDist = distMap[k][l]
11:             end if
12:         end for

```

```

13 :      //Mise à jour du vecteur de connectivité.
14 :       $CDistMap[k] \leftarrow bestSpline, distMap[k][bestSpline]$ 
15 :      //Mise à jour de la racine de l'arbre correspondant à la plus grande distance trouvée.
16 :      if  $distMap[k][bestSpline] > distanceRoot$  then
17 :           $m\_root = k$ 
18 :           $distanceRoot = distMap[i][bestSpline]$ 
19 :      end if
20 :  end for
21 :  end procedure

```

Création de la structure d'arbre à partir de la hiérarchie :

Dans un deuxième temps, une fois la hiérarchie de l'arbre obtenue, la structure d'arbre est créée. De par l'extraction des lignes centrales et le type de données traitées, la structure d'arbre est tout de suite obtenue après application de notre algorithme. L'arbre est connexe et ne présente aucun cycle.

Comme expliqué précédemment, les nœuds du graphe représentent des points particuliers : le nœud racine, les nœuds de bifurcation et les nœuds terminaux (nœuds feuilles). Avec la détermination de la hiérarchie, ces points particuliers sont connus : la racine correspond au premier point de la spline racine, les nœuds de bifurcation correspondent à un point de la spline mère auquel la spline fille est rattachée (variable *m_closestPoint* de l'Algorithm 2) et les nœuds feuilles correspondent au dernier point des splines n'ayant pas de descendance. Ces nœuds possèdent, comme étiquette, un identifiant c'est-à-dire un numéro positif unique. Des informations complémentaires sont stockées dans ces nœuds : les coordonnées du point, ses caractéristiques (nœud racine, nœud feuille, nœud artificiel, etc.), ainsi que d'autres propriétés utiles pour le reste de notre projet décrites par la suite.

La hiérarchie entre ces nœuds est créée grâce à l'insertion des arcs. S'il existe une ligne centrale entre deux nœuds, un arc est ajouté. Des informations à propos des segments de spline (portions) sont stockées. Ces segments représentent une structure de données ayant comme donnée une liste de points (venant de la ligne centrale associée avec comme premier point le nœud source et comme dernier point le nœud de destination de l'arc) et leur rayon (correspondant à la lumière du vaisseau) associé ainsi que d'autres propriétés utilisées dans le reste de notre programme, décrites également par la suite. Ces segments peuvent être également représentés graphiquement sous forme de spline (Figure 23).

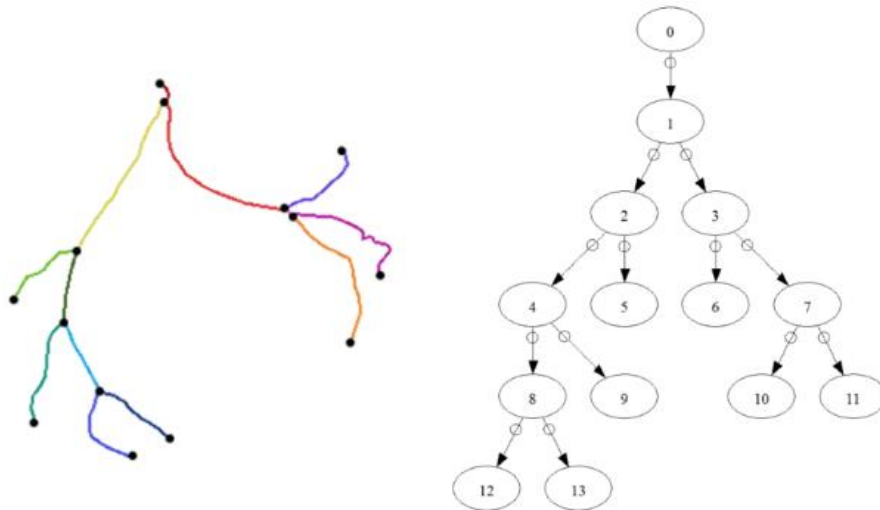


Figure 23 : Artère coronaire gauche représenté sous forme d'arbre. L'image de droite représente la structure d'arbre et l'image de gauche les différentes informations stockées (les points noirs correspondent aux nœuds du graphe et les segments de couleurs aux arcs)

Dans notre programme, chaque arbre est géré avec la bibliothèque *Boost*² [Schä14, SiLL01] (en C++) sous la forme d'une liste d'adjacence.

1.2.5. CONTRIBUTION

Notre problématique s'inscrit dans la continuité de la reconstruction 3D des artères coronaires pour le planning de l'angioplastie coronarienne. A partir de cette séquence dynamique 3D+T d'arbres coronaires, notre but est de déterminer l'incidence 2D dite optimale sous laquelle sera réalisée l'angioplastie. Comme expliqué précédemment dans les limitations liées à l'angiographie à rayons X, les procédures interventionnelles dépendent fortement du choix de cette incidence. Elle doit apporter un maximum de visibilité du segment vasculaire sténosé sur lequel le cardiologue doit établir un diagnostic puis intervenir, c'est-à-dire obtenir le segment avec notamment le moins de superpositions avec les structures avoisinantes et le plus déployé possible. Cette visibilité doit rester optimale tout le long du cycle cardiaque. L'incidence optimale doit alors prendre en compte le mouvement et les différentes déformations subies par les artères coronaires au cours des différentes phases cardiaques.

² <http://www.boost.org/>

Une méthode permettant de calculer de manière automatique cette incidence optimale dans un cas statique puis dans un cas dynamique a été mise en œuvre en utilisant une reconstruction 3D+T d'un arbre coronaire. Différentes étapes sont à considérer pour calculer la position angulaire optimale du C-arm :

- mise en correspondance spatio-temporelle des branches vasculaires de toute la séquence 3D+T,
- sélection interactive du segment vasculaire d'intérêt ou de la zone d'intérêt si présence d'une bifurcation,
- choix de la zone de recherche pour l'angle de vue optimale (uniquement sur le segment d'intérêt pour le déploiement du stent, sur l'ensemble du chemin depuis la racine de l'arbre jusqu'au segment d'intérêt pour le guidage des outils, etc.),
- simulation de la projection de chaque volume 3D dans un plan 2D selon une orientation compatible avec les couples d'angles d'acquisition du C-arm,
- calcul pour chaque orientation et chaque phase du cycle cardiaque de différentes métriques (ratio de raccourcissement 3D/2D du segment d'intérêt, ratio de superposition avec d'autres segments vasculaires puis également avec lui-même, ratio d'ouverture de l'angle de bifurcation),
- sélection sous critère des incidences optimales et affichage du résultat pour validation par le cardiologue.

Pour réaliser ces différentes étapes nous sommes partis de la séquence dynamique 3D+T obtenue après reconstruction ou modélisation 3D. Comme expliqué dans la partie précédente, pour chaque volume binaire de coronaire 3D correspondant à une phase donnée, une structure d'arbre issue des lignes centrales a été extraite. Sur la séquence 3D+T, les arbres obtenus sont alors non-isomorphes. Cette contrainte, liée à la reconstruction, doit être prise en compte dans les différentes étapes de la détermination de l'angle de vue optimale. Notre programme développé en langage C++, propose une interface graphique pour réaliser toutes ces étapes basées sur la bibliothèque Qt³. L'outil développé est alors facile d'utilisation avec un temps de calcul raisonnable et donc utilisable en routine clinique.

Le manuscrit décrit les différentes étapes décrites ci-dessus. Ainsi le chapitre 2 propose un état de l'art des différentes méthodes mises en œuvre dans notre programme correspondant aux deux principales étapes : la mise en correspondance et la recherche de l'incidence optimale.

³ www.qt.io

Le chapitre 3 décrit tout d'abord la mise en correspondance entre deux arbres successifs, c'est-à-dire proches au niveau des phases du cycle cardiaque avec un déplacement et une déformation faibles entre ces deux arbres (deux phases successives). Cette mise en correspondance est réalisée à partir des lignes centrales et prend en compte la hiérarchie de notre structure. Pour rappel, ces deux arbres sont non-isomorphes à cause de leurs différences liées à leur reconstruction ou segmentation. Des mesures de similarité utilisant à la fois des caractéristiques hiérarchiques et géométriques des arbres sont étudiées. Afin de prendre en compte le côté non-isomorphe des arbres, nous ajoutons la notion de nœud artificiel. Des nœuds artificiels sont ajoutés dans les arbres permettant de mettre en correspondance des portions de branches lorsqu'une bifurcation est manquante ou lors d'une différence de longueur de branches terminales. Le chapitre termine sur la mise en correspondance de l'ensemble de la séquence 3D grâce à la construction d'un arbre le plus complet possible : l'arbre mosaïque. Cet arbre mosaïque, construit principalement à partir de la fusion d'arbres d'unions successifs rassemble tous les nœuds et les segments présents dans au moins une phase de la séquence. L'arbre mosaïque est complet. Nous émettons effectivement l'hypothèse qu'un arbre d'une phase donnée n'est qu'une vue partielle de l'arbre. L'arbre mosaïque remet également en question la véracité des précédentes mises en correspondance et prend en considération toutes les différences topologiques entre les arbres de la séquence.

Le chapitre 4 explique le principe de la recherche de l'angle de vue optimale en prenant en compte le mouvement des artères coronaires par rapport au cycle cardiaque grâce à la mise en correspondance réalisée précédemment. Les métriques telles que le ratio de raccourcissement ou de chevauchement sont décrites. De même, plusieurs zones de recherche sont prises en compte : un angle de vue optimale est alors déterminé pour le segment d'intérêt pour le déploiement d'un stent, ou alors pour le guidage des outils jusqu'à la zone de la lésion.

Le chapitre 5 décrit les résultats obtenus au cours des différentes étapes. Les méthodes de mise en correspondance à partir de différentes mesures de similarité (ainsi que leurs combinaisons) sont évaluées puis comparées afin d'obtenir la meilleure mise en correspondance. Les résultats obtenus pour la détermination de l'incidence optimale selon les deux critères d'optimalités (a) segment d'intérêt ou bifurcation seule, et b) chemin depuis la racine jusqu'à la zone sténosée) sont également proposés à partir de données réelles et simulées.

CHAPITRE 2

ÉTAT DE L'ART

SOMMAIRE

2.1.	Appariement d'arbres vasculaires.....	40
2.1.1.	Algorithme exact	41
2.1.2.	Algorithme inexact	41
2.1.3.	Applications aux artères coronaires.....	46
2.2.	Problème de la clique maximum (MCP).....	47
2.2.1.	Algorithmes exacts	48
2.2.2.	Approche heuristique.....	50
2.2.3.	Maximum weight clique problem (MWCP).....	50
2.3.	Recherche d'incidences optimales.....	52

De nombreuses applications dans divers domaines tels que la biologie [Myer94, NeWu70, YaSz07, ZaBV09] ou la reconnaissance vocale [RaJu93] utilisent la mise en correspondance d'une séquence de points. La mise en correspondance est également utilisée dans la reconnaissance de caractères [FSFR13, PISr00, SuYa98], la comparaison et l'indexation de fichiers dans des bases de données [Heck78, SLBK04]. Dans le domaine de l'imagerie numérique, des travaux de mise en correspondance de points ont été menés en compression d'images [LuSz97], reconnaissance de formes [BuAl83, BuCs95] et plus généralement en analyse d'images [BuBü93, CTFC98, HrMc90]. En synthèse d'images, des algorithmes de programmation dynamique ont par exemple été utilisés pour le mélange de postures et pour l'adaptation et la reconnaissance de mouvement représentés par des séquences temporelles [SSGD03].

Dans le domaine médical, la mise en correspondance est utilisée majoritairement pour le recalage d'images [MTLP12, SRLG11]. Traditionnellement, des méthodes de mise en correspondance de points [Pinh13] (*point cloud matching*), de courbes (*string matching* [Step94] par exemple ou *Iterative closest courbe* – ICC [Bens15]) ou de graphe [BuMe97, Bunk00, CFSV04, LiRi13] (*graph matching*) sont utilisées. La première catégorie contient des méthodes d'appariements très connues comme l'algorithme de RANSAC [FiBo81] (*Random Sample Consensus*), de CPD [MyXu10] (*Coherent Point Drift*), d'ICP [BeMc92] (*Iterative Closest Point*) ou encore de TPS [ChRa03] (*Thin-plate Spline*). Cependant, utiliser ce genre d'approche reviendrait à ne pas utiliser toutes les informations mises à notre disposition (pas seulement les informations géométriques), comme la structure hiérarchique des arbres vasculaire par exemple. C'est pourquoi nous nous focalisons sur la catégorie de méthode de mise en correspondance de graphe.

2.1. APPARIEMENT D'ARBRES VASCULAIRES

Les différents états de l'art récent dans le problème de la mise en correspondance de graphe [Bunk00, LiRi13, Wu10] nous montrent que la classification de ces différents algorithmes est complexe à cause d'une grande diversité existante des classes. Certains papiers se focalisent sur des applications spécifiques comme Fan [Fan12] pour l'analyse des réseaux sociaux. Nous reprenons la classification utilisée par Bengoetxea [Beng02], Conte et al. [CFSV04] et Oyarzun Laura [Oyar15] qui peut être utilisée dans le domaine de l'imagerie médicale. Les méthodes de mise en correspondance sont séparées en deux catégories : les algorithmes de mise en correspondance dit « exacts » et « inexacts » basés sur la théorie des graphes.

2.1.1. ALGORITHME EXACT

La première catégorie consiste à déterminer l'appariement entre deux arbres isomorphes. Ces deux arbres doivent alors posséder la même structure (nombre identique de nœuds et d'arêtes, préservation de la hiérarchie, etc.) [Vali02]. Néanmoins, dans le cas de l'imagerie médicale, les arbres sont très rarement identiques. Ces derniers peuvent présenter plusieurs modifications. Une branche entière ou une partie de branche peut être manquante, une branche supplémentaire peut apparaître ou encore une trifurcation dans un arbre peut devenir une double bifurcation dans un autre. Toutes ces variations peuvent être dues aux erreurs de prétraitements (segmentation 3D, reconstruction 3D, etc.) de la structure à partir des images acquises (2D ou 3D). Elles sont liées en partie à la présence de bruit dans les images, à la superposition des structures et la non-homogénéité du produit de contraste ou encore aux mouvements des différents organes du patient (cœur, respiration, etc.). L'utilisation de ces algorithmes pas assez flexible n'est donc pas appropriée dans notre cas. Le problème de sous-graphes isomorphes est moins restrictif que le cas précédent [MeBu98, PeSZ99, Ullm76]. Les arbres ne doivent pas être forcément similaires : seul le plus grand sous arbre commun à ces deux arbres est identifié.

2.1.2. ALGORITHME INEXACT

La catégorie des algorithmes inexacts peut ainsi pallier dans la majorité des cas ces problèmes. Cette catégorie peut être également divisée en plusieurs types de technique (voir Figure 24 correspondant à la Figure 2.12 de [Oyar15]). Dans le domaine des images médicales, la majorité des méthodes est située dans les techniques basées sur les « arbres de recherche » ou la « *théorie probabiliste* ». Cependant de nouvelles méthodes utilisent des techniques évolutives telles que des algorithmes génétiques ou des algorithmes à estimation de distribution.

Nous nous sommes focalisés au cours de cette thèse dans les techniques basées « arbres de recherche » particulièrement développées dans le domaine de l'imagerie médicale. D'après Oyarzun Laura [Oyar15], Bengoetxea [Beng02] et Conte et al. [CFSV04] nous pouvons diviser cette technique en quatre groupes.

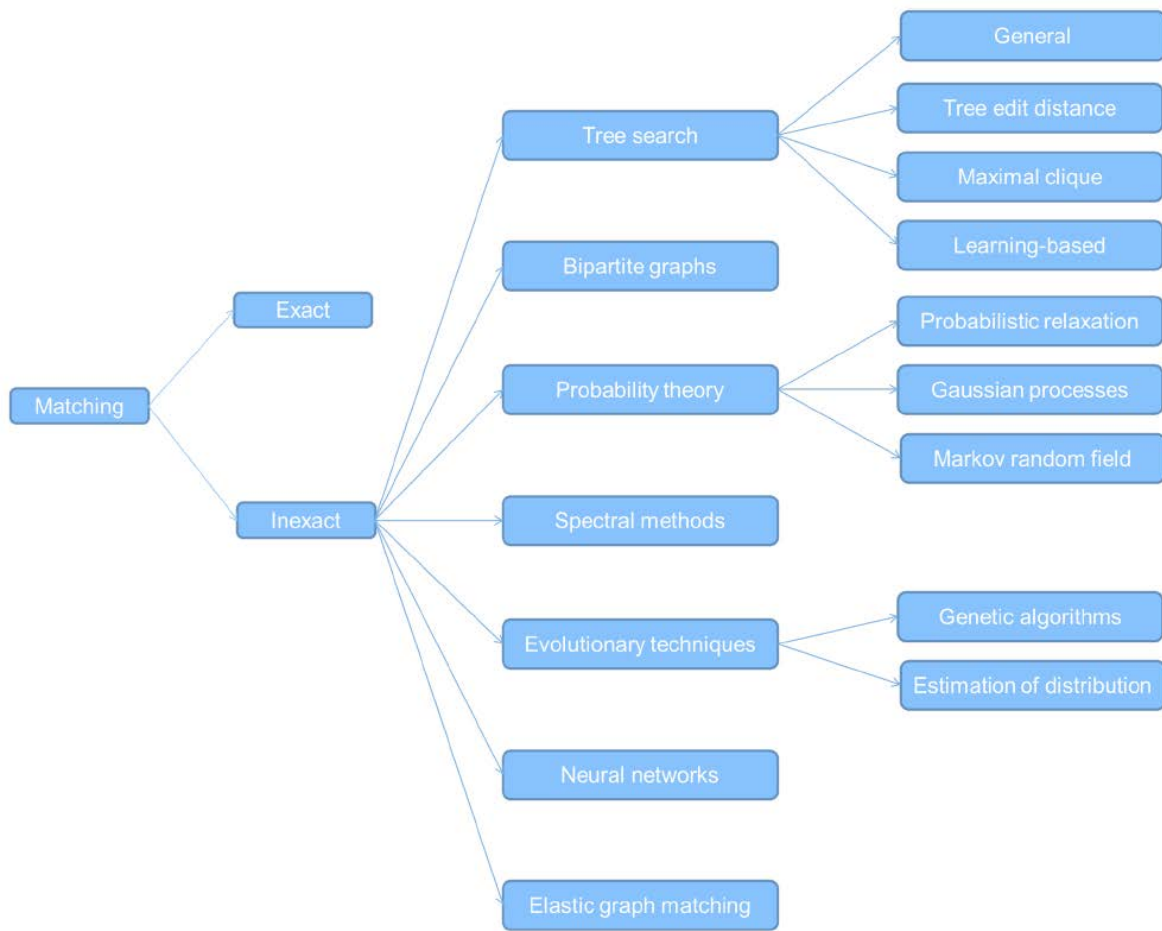


Figure 24 : Classification des méthodes d'appariement de graphe selon Oyarzun Laura [Oyar15].

Le premier groupe contient des méthodes générales, ne pouvant être classées dans les autres groupes. Charnoz et al. [CAMS05, Char07] propose une méthode basée sur certaines hypothèses pour mettre en correspondance deux modèles représentant le même arbre vasculaire, lui permettant d'étudier les déformations du foie ou le suivi de tumeur. Les différences de forme et de topologie sont prises en compte dans son algorithme. Le processus commence par mettre en correspondance les racines. L'algorithme recherche donc localement les prochaines correspondances dans les sous-arbres des nœuds analysés. Pour cela, deux étapes sont effectuées. Tout d'abord, l'algorithme recherche un appariement entre les arêtes sortantes du nœud. Ces derniers sont ensuite analysés pour trouver les prochains nœuds pouvant être appariés. Ces étapes permettent d'étoffer progressivement un ensemble de solutions pertinentes mais nombreuses. C'est pourquoi pour chacune de ces hypothèses, une mesure de qualité y est associée permettant de les ordonner. Seules les solutions favorisant les bons appariements, c'est-à-dire deux chemins géométriquement similaires, sont alors conservées. A chaque étape, la viabilité des solutions est évaluée permettant d'en conserver un nombre restreint. La

mesure de qualité utilisée se traduit par l'emploi d'une fonction de coût (pour chaque appariement et pour la solution globale). Plus la valeur de ce coût sera faible, plus la similitude entre les nœuds et les arêtes sera élevée. Ces fonctions de coûts sont basées sur les attributs des arcs et des nœuds (longueurs et orientations des chemins, rayons des branches, angle formé par deux chemins, etc.). Cette approche donne des bons résultats sur des données simulées d'arbre vasculaires du foie (issu du Visible Man) ayant subi diverses transformations (suppression de branches, déformation géométrique, etc.). Elle donne également des résultats prometteurs sur des données réelles mais avec des arbres possédant moins de nœuds (95% de nœuds corrects mis en correspondance).

Lohe et al [LKZP07] quant à eux proposent une méthode hiérarchique basée sur la construction d'un arbre de recherche symétrique pour la mise en correspondance d'arbres bronchiques. Contrairement à l'algorithme précédent qui calculait les appariements des nœuds et des arcs séparément, cette méthode combine ces deux associations en une seule étape. La méthode commence par mettre en correspondance les deux racines. A chaque niveau l de l'arbre de recherche, un nœud de chaque arbre d'entrée est traité. Tout d'abord, l'algorithme effectue une sélection sur tous les couples de nœuds pouvant être associés (dont un des nœuds se trouve être un des deux traités). Si le couple de nœuds forme un appariement possible, c'est-à-dire si une liste de contraintes de cohérence est respecté (relation d'héritage, différence entre les longueurs ou les volumes des branches, etc.), il est ajouté au niveau l de l'arbre de recherche. Une fois seulement, les mesures de similitudes sont calculées pour ce couple de nœuds. Le résultat est ensuite accumulé avec celui de son père dans l'arbre de recherche. Deux métriques sont utilisées : la longueur du chemin entre les deux nœuds et leur première mise en correspondance commune (souvent la racine), et l'angle entre les deux chemins et leurs dernières branches communes. Lorsque l'arbre de recherche est construit, la plus grande mesure de similarité accumulée des feuilles est examinée. Une dernière étape de backtracking à partir de ce nœud jusqu'à la racine de l'arbre de recherche nous donne la liste des nœuds mis en correspondance. Cette méthode, évaluée sur onze patients (humain et cochon) ainsi que sur deux organes, donne un pourcentage d'erreur de 19,1%. Un des avantages de cette méthode est qu'elle n'est pas restreinte à un seul type d'application médicale et que d'autres métriques relatives à notre cas peuvent être facilement ajoutées.

Bülow et al. [BLWH06] propose une méthode de labélisation d'arbres bronchiques. Ils comparent deux approches basées sur les positions des points des lignes centrales utilisant des caractéristiques différentes : la forme 3D des branches ou les moments statistiques sur la distribution locale des points. 69% d'appariement correct est obtenu avec la première approche contre 40% avec la dernière.

Kaftan et al. [KKNN06] propose une méthode basée non plus sur la théorie des graphes avec une recherche de mise en correspondance de nœuds ou d'arc, mais sur l'appariement des lignes centrales

de la racine jusqu'aux feuilles des arbres. Cette technique, comparant leur silhouette, n'est alors que peu affectée par les branches manquantes ou les fausses branches. Trois mesures de similarités sont utilisées dans un cas général pour comparer deux chemins : la distance, l'angle (différence moyenne des directions des deux chemins) et la variance de la distance. Cette méthode ne donne pas cependant une correspondance « one to one ». Une étape de prétraitement pour exclure les correspondances « many to one » doit être préalablement effectuée. Cette méthode donne des résultats corrects pour le cas de la mise en correspondance des arbres des bronches : testée sur 10 patients, soit sur 702 chemins, 87 % y sont correctement appariés.

Le deuxième groupe concerne les méthodes par édition de graphes [Bunk97, Selk77, ShZh90, WaFi74]. Grâce à plusieurs opérations d'édition tels que la suppression et l'insertion de nœuds ou d'arêtes, l'isomorphisme entre graphes peut être atteint. A chacune de ces modifications est associé un coût d'édition. La minimisation de cette fonction de coût permet d'obtenir la mise en correspondance optimale entre les graphes. Par la suite, des fonctions de coût assimilant les attributs des nœuds et des arêtes ont été introduites [SeKK04, Tai79].

Tang et Chung [TaCh06] propose un algorithme basé sur la minimisation de la distance d'édition entre deux arbres vasculaires cérébraux permettant de déterminer la localisation d'anévrisme. Les arbres sont extraits à partir d'angiographie rotationnelle RX. Des fonctions de coût basées sur le rayon, l'orientation et la longueur des branches sont utilisées. Cet algorithme est testé sur deux données réelles d'arbres vasculaires cérébraux ayant plus de 30 branches, et semble donner des résultats corrects pour les branches principales. Cependant, seule une vérification visuelle est effectuée. La méthode ne contient pas d'évaluation quantitative.

Ces différentes méthodes d'édition de graphes sont de complexités élevées (NP-difficile pour la plupart des cas), leurs temps d'exécutions sont exponentiels [GXTL10]. Cette complexité peut être réduite en ne considérant que certaines catégories d'arbres (cas de graphe où le nombre de sommet est faible par exemple). Autrement, d'autres méthodes, à complexité moins élevée, ont été proposées comme « tree alignment », « tree inclusion » et « constrained tree-edit distance » [Bill03, Bill05]. Un des désavantages de celles-ci est qu'elles n'acceptent pas toutes les erreurs possibles d'extraction : les méthodes de « tree inclusion » ne tolèrent les branches incorrectes uniquement dans un seul des deux arbres.

Le troisième groupe contient des méthodes basé sur la recherche de la clique maximum à l'aide de la construction d'un graphe d'association. La section 2.2 décrit le problème de la clique maximum ainsi que différents algorithmes pouvant être utilisés afin d'extraire la clique maximum du graphe d'association.

Tschirren et al. [TMPH05] propose une méthode pour mettre en correspondance puis labelliser les arbres bronchiques. Celle-ci est décomposée en plusieurs étapes permettant de réduire le temps de calcul. La première étape consiste à enlever les branches parasites pour réduire la taille des arbres puis à les déplacer dans le même repère de coordonnées en utilisant un recalage rigide. Ensuite, la méthode détermine la mise en correspondance optimale grâce à la recherche de la clique maximum du graphe d'association construit à partir de l'algorithme décrit par Pelillo et al. [PeSZ99]. Pour construire le graphe d'association entre deux arbres, plusieurs métriques basées sur les attributs des nœuds sont utilisées telles que la distance euclidienne, la distance topologique ou encore la relation hiérarchique entre eux. La clique maximum est quant à elle déterminée à partir de l'algorithme publié par Carraghan and Pardalos [CaPa90]. Cette méthode donne des résultats corrects pour le type d'arbre traité et tolère, grâce au seuil utilisé pour la distance topologique, les fausses branches qui n'ont pas été supprimées lors du prétraitement. Cependant, elle est sensible aux autres erreurs dues en partie au bruit. L'algorithme est très rapide (temps d'exécution de 1 à 3 secondes) pour mettre en correspondance des arbres contenant environ 300 nœuds et présente un pourcentage d'erreur de 7,1%.

Une autre approche basée également sur les graphes d'association est celle utilisée par Metzen et al. [MKSZ09]. L'une des principales différences avec les algorithmes de ce type est l'utilisation de métriques basées non seulement sur les caractéristiques des nœuds des arbres, mais également sur celles des lignes centrales, comme les méthodes présentée par Bülow et al. [BLWH06] et par Kaftan et al. [KKNN06]. Dans l'algorithme de Metzen et al. [MKSZ09], ces chemins sont représentés sous la forme de polyligne normalisée, permettant de comparer les lignes centrales indépendamment de leurs longueurs, de leurs orientations ou de leurs positions. Deux étapes sont nécessaires à la construction du graphe d'association basé également sur l'algorithme de Pelillo et al. [PeSZ99]. Tout d'abord, l'ajout des nœuds potentiels au graphe est basé sur des contraintes unaires entre les nœuds des deux arbres ou le chemin entre la racine des arbres et eux même (aire minimale entre les polygones, direction des polygones, coordonnées euclidienne des nœuds, niveau de l'arbre, etc.). Une fois cette sélection effectuée, les nœuds du graphe d'association sont ensuite reliés s'ils sont cohérents entre eux. Cela est déterminé par le calcul de plusieurs contraintes binaires (direction des polygones, courbure moyenne et longueurs des chemins, etc.) basé sur la comparaison des deux chemins représentés par deux nœuds du graphe. Une fois le graphe d'association construit, les nœuds mis en correspondance sont déterminés grâce à la recherche de la clique maximum, basée sur l'heuristique de Pelillo et al. [Peli99] déterminée à partir du théorème de Motzkin-Straus [MoSt65]. Cette méthode, à cause du nombre important de paramètres, peut ne pas sélectionner tous les nœuds appariés correctement. Pour prendre quand même en considération ces nœuds, une étape consistant à augmenter la taille de la clique est réalisée. Cette approche avec un temps d'exécution d'environ 288 secondes est évaluée

sur une paire d'arbres vasculaires issus du foie et des poumons. Elle donne de bons résultats : environ 92,9% des nœuds sont correctement appariés.

Haris et al. [HEMP99] propose une méthode pour labéliser les artères coronaires en utilisant les cliques maximales au lieu des cliques maximum. Un modèle 3D est généré afin de le projeter en 2D sous la même perspective que l'image angiographique. Selon les experts, la méthode obtient 78,7% d'artères correctement labélisées.

Le quatrième groupe (*Learning-based approach*) est composé de méthodes basées sur la génération d'un modèle après observation de données.

Graham et Higgins [Grah06, GrHi06a, GrHi06b] propose un algorithme de programmation dynamique prenant en compte les différentes modifications apportées aux arbres entre deux acquisitions. Cette méthode permet d'obtenir un arbre commun entre deux arbres grâce à une série de primitives (liées aux différentes déformations possibles trouvées entre les arbres tels qu'une branche manquante). Cet ensemble de primitives permet de déterminer les appariements valides. L'arbre commun permet de mettre en correspondance facilement ces deux structures. Des mesures de similarités basées à la fois sur les attributs des nœuds et des branches sont utilisées : longueurs et directions des branches, coordonnées euclidienne des nœuds, etc. La méthode a été évaluée pour quatre patients et possède un faible temps d'exécution (5 secondes pour apparié 321 et 131 nœuds respectivement).

Oyarzun Laura et Drechsler [OyDr11] propose une méthode basée sur l'algorithme de Graham et Higgins [Grah06, GrHi06a, GrHi06b]. Plusieurs changements y sont apportés pour correspondre à leur problématique. Essentiellement, les mesures de similarités sont modifiées et d'autres primitives déterminant l'arbre commun sont ajoutées.

2.1.3. APPLICATIONS AUX ARTERES CORONAIRES

Notre problématique concerne la mise en correspondance des artères coronaires en 3D sur différentes phases du cycle cardiaque. Des travaux sur les artères coronaires utilisent une mise en correspondance soit pour un recalage 3D/2D ou 2D/2D [BeMV15, Bens15, BMSG14, Gu05, HEMP99, RiSC12, SPSK15, SRLG11, TLGD05, ZhQi11]. Ils ne répondent cependant pas totalement à notre problématique d'appariement sur une séquence 3D+T des artères coronaires. Les arbres extraits possèdent peu de nœuds (entre 10 et 20) et un faible niveau hiérarchique (en moyenne 5), permettant ainsi d'appliquer les différentes méthodes proposées ci-dessus. Cependant, ces structures sont fines, de l'ordre de quelques millimètres, posant problème lors de l'extraction des structures dans le cas

d'une angiographie RX. D'une reconstruction 3D à une autre, les branches finales subissent beaucoup de modifications : certaines sont entièrement ou en partie manquantes, alors que d'autres incorrectes apparaissent. Les méthodes utilisées doivent alors prendre en compte ces différents cas. De plus, avec le mouvement lié à la respiration du patient et du cœur, les branches subissent des déformations non linéaires supplémentaires et différentes pour chaque patient. Deux branches semblables d'une phase à une autre ne présentent pas la même courbure ou les mêmes coordonnées par exemple, rendant difficile le choix des mesures de similarités. Cependant, d'une phase du cycle cardiaque à une autre, les déformations sont minimales. La mise en correspondance peut alors être appliquée plus facilement pour deux phases proches, représentant un instant semblable du cycle cardiaque.

Nous voulons en plus un algorithme qui peut être utilisé avec une structure d'arbre enraciné dans un cas 3D et permettant d'obtenir une mise en correspondance « one to one ».

Avec toutes ces caractéristiques, nous avons décidé de tester la méthode générique proposée par Lohe et al. [LKZP07] pour pouvoir l'adapter à notre problème en y ajoutant des mesures de similarités cohérentes. Cette méthode est rapide d'exécution avec des résultats corrects. Une méthode plus spécifique utilisant un graphe d'association est aussi testée, basée à la fois sur celle de Metzen et al. [MKSZ09] et celle de Tschirren et al. [TMPH05]. La construction du graphe d'association repose principalement sur la première méthode à cause des différentes mesures de similarité calculées : les attributs des nœuds et des lignes centrales sont tous les deux pris en compte.

2.2. PROBLEME DE LA CLIQUE MAXIMUM (MCP)

Certaines méthodes de mise en correspondance d'arbres basées sur la construction d'un graphe d'association nécessitent par la suite de rechercher dans ce dernier le plus grand sous arbre complet de cardinalité maximum [MKSZ09, TMPH05]. Ce problème revient à déterminer la clique maximum de ce graphe. Par définition, une clique représente un sous graphe complet, c'est-à-dire un sous graphe dont tous les vertices sont adjacents. Une clique est dite **maximale** s'il n'existe pas dans le graphe une clique de taille supérieure la contenant. Une clique est dite **maximum** si sa cardinalité est supérieure aux autres cliques du graphe. Une clique maximum est également une clique maximale. La Figure 25 schématise ces différentes notions. La recherche de la clique maximum est un problème NP-difficile [AMCG99, Karp72]. Dans le reste de cette section, nous considérons le graphe $G = (V, E)$ avec V l'ensemble des vertices et E l'ensemble des arcs de G , C une clique du graphe G , c'est-à-dire un sous ensemble de V ainsi que $\omega(G)$ l'ordre de la clique maximum de G , c'est-à-dire le nombre de vertices qu'elle contient.

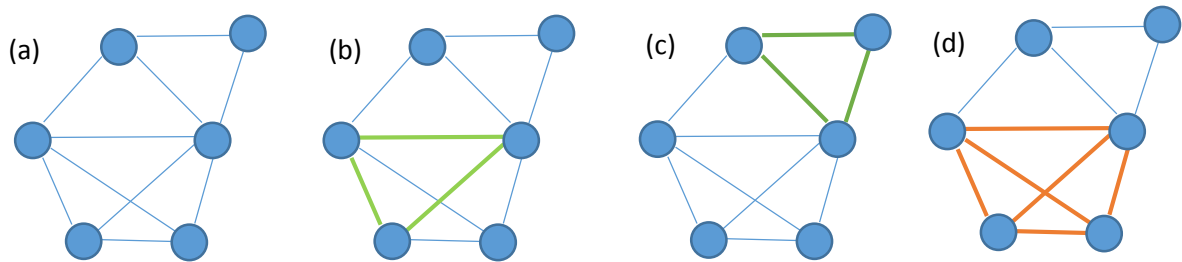


Figure 25 : (a) un graphe. (b) une clique en vert d'ordre 3. (c) une clique maximale d'ordre 3. (d) une clique maximum d'ordre 4.

Le problème de clique maximum est utilisé dans divers domaines allant de la bio-informatique [MaAY10], à l'économie [BoBP06] ou encore à l'analyse des réseaux sociaux [BaBH11, PaYB12] et est également relié à plusieurs problèmes d'optimisation tels que le « *graph clustering* » [Scha07], « *graph vertex coloring* » [ChHW87, WuHa12] ou encore « *optimal winner determination* » [WuHa15a].

Afin de résoudre ce problème, plusieurs méthodes ont été développées [BBPP99, PaXu94, WuHa15b]. Elles sont divisées en deux principales catégories : les méthodes dites exactes et les méthodes avec une approche heuristique. Les méthodes exactes sont principalement basées sur l'algorithme de Séparation – Évaluation (*Branch and Bound*) utilisées pour résoudre les problèmes d'optimisation combinatoire. Leur principal avantage est d'obtenir une solution optimale mais leur temps de calcul est généralement trop important, excepté pour les problèmes avec un graphe de taille limitée. C'est pourquoi, les méthodes basées sur une approche heuristique ont été développées pour les problèmes de grandes tailles afin d'obtenir une solution correcte dans un temps d'exécution raisonnable. Bien évidemment, ces deux types de méthodes peuvent être combinés. Nous allons maintenant présenter succinctement quelques algorithmes de référence.

2.2.1. ALGORITHMES EXACTS

L'algorithme de base des méthodes exactes a été développé par Carraghan and Pardalos [CaPa90]. Cet algorithme a pour avantage d'être simple à comprendre et à implémenter en plus d'être efficace en termes de résultat. Nous reprenons les notations utilisées dans [CaPa90].

La méthode commence par trier par ordre croissant les vertices du graphe $G = (V, E)$ avec $V = \{v_1, v_2, \dots, v_n\}$ en fonction de leur degré. v_1 a donc un degré plus faible que v_2 . Ce tri de vertices permet de réduire le temps d'exécution lorsque le graphe possède de nombreuses arêtes. L'algorithme parcourt les vertices ordonnés du graphe au fur et à mesure et recherche la clique maximum. Il

commence par rechercher la plus grande clique contenant le premier vertex v_1 dans G . Puis il essaye de trouver une plus grande clique contenant le deuxième vertex v_2 dans $G - \{v_1\}$ et ainsi de suite. Plusieurs conditions sont utilisées afin de réduire l'espace de recherche (*pruning techniques*). Pour cela, Carraghan and Pardalos [CaPa90] introduit la notion de profondeur (*depth*). L'algorithme, commençant par étudier le vertex v_1 , considère à *depth 2* tous les vertices adjacents à ce vertex (trié également en fonction de leur degré). A *depth 3*, tous les vertices présents à *depth 2* et adjacents au premier vertex de *depth 2* sont considérés. Un parcours en profondeur est alors réalisé afin de trouver la clique maximum. Lorsque le niveau maximum de *depth* ne peut plus être développé pour une clique étudié, l'algorithme remonte au niveau de *depth* précédent et ainsi de suite. Une condition en fonction du niveau de profondeur et du nombre de vertex adjacent dans le *depth* courante permet d'éviter de parcourir toutes les différentes possibilités. Soit v_{d_i} le vertex concerné pour la clique à *depth* d et à l'étape i . Nous avons donc la profondeur (*depth*) $d : v_{d_1}, \dots, v_{d_i}, \dots, v_{d_m}$. Si $d + (m - i) \leq$ *taille de la meilleure clique courante (CBC)*, l'algorithme arrête de rechercher à ce niveau car la plus grande clique pouvant être obtenue en développant v_{d_i} sera de taille inférieure (ou identique) à CBC.

Différentes méthodes sont dérivées de cet algorithme comme ceux proposés par Östergård [Öste02] (*sub-clique problem*), Tomita and Seki [ToSe03] (*subgraphs coloring*), Segundo et al. [SMRH13] (*subgraphs coloring*) ou encore Li and Quan [LiQu10] et Maslov et al. [MaBP14] (*MaxSAT*). Prosser [Pros12] propose un étude comparative de plusieurs algorithmes exacts ainsi qu'un bref historique de ces algorithmes. Wu and Hao [WuHa15b] propose quant à eux également une description et un commentaire sur les performances de ces différentes méthodes. Leur tableau 1 regroupe les principaux algorithmes exacts pour le problème de la clique maximum.

Nous allons maintenant décrire brièvement l'algorithme Östergård [Öste02] qui est une amélioration de [CaPa90]. Plusieurs modifications ont été apportées.

Dans un premier temps, l'ordre de recherche est inversé. Nous reprenons également dans ce paragraphe les notations utilisées dans [Öste02]. Soit $S_i = \{v_i, v_{i+1}, \dots, v_n\}$ avec v_n le dernier vertex du graphe G . L'algorithme commence par considérer les cliques dans l'ensemble $S_n = \{v_n\}$. La plus grande clique courante est à ce stade $CBC = \{v_n\}$. Ensuite, l'algorithme recherche s'il existe une clique plus grande dans l'ensemble $S_{n-1} = \{v_{n-1}, v_n\}$, et ainsi de suite.

Dans un deuxième temps, les techniques de *pruning* utilisées diffèrent. La taille de la plus grande clique trouvée dans S_i est noté $c(i)$. Au début de l'algorithme, nous avons donc $c(n) = 1$ et la taille de la clique maximum du graphe sera donnée par $c(1)$. Pour $1 \leq i \leq n - 1$, nous avons donc soit

$c(i) = c(i + 1)$ ou $c(i) = c(i + 1) + 1$. Dans le premier cas, la plus grande clique de S_i est identique à celle trouvée à l'étape précédente dans S_{i+1} . L'algorithme n'a pas trouvé de clique incluant le vertex v_i . A l'inverse, dans le deuxième cas, une clique incluant v_i à été trouvée. La nouvelle *CBC* est de taille supérieure d'un vertex à celle trouvée dans S_{i+1} . Cette notion de taille est utilisée dans les nouvelles techniques de *prunning* proposées. La notion de profondeur (*depth*) est également utilisée comme dans [CaPa90]. Östergård cherche à savoir si le vertex considéré v_i peut devenir le $(j + 1)^{\text{ème}}$ vertex dans la clique qui est en train d'être formée (qui est composée pour l'instant de j vertices). Il utilise la condition suivante pour stopper la recherche à un niveau de profondeur d donnée : $j + c(i) \leq s$. Notons que j est également égal à $d - 1$. Ce programme en modifiant l'inégalité peut être utilisée pour trouver toutes les cliques maximum d'un graphe.

2.2.2. APPROCHE HEURISTIQUE

Différentes approches heuristiques ont été utilisées dans le problème de la clique maximum. Un état de l'art détaillé de ses méthodes peuvent être trouvé dans Wu and Hao [WuHa15b], Bomze et al. [BBPP99], Butenko [Bute03] et Pardalos and Xue [PaXu94]. Des algorithmes gloutons (*greedy algorithms*) ont été développés afin d'extraire la clique maximum comme le propose par exemple Jagota et al [JaSa01], Busygin [Busy06] et Grosso et al. [GrLC04]. D'autres méthodes proposent d'utiliser des algorithmes évolutionnaires (Guturu and Dantu [GuDa08]) couplés avec des algorithmes génétiques (recherche tabou, Friden et al [FrHW89], Gendreau et al. [GeSS93]) ou des algorithmes de type colonies de fourmi (Solnon and Frenet [SoFe06]). De nombreux algorithmes utilisant la recherche locale (*local search*) comme méthode d'optimisation ont également été développés au cours de ses dernières années (Pullan et al. [PuHo06], Pullan et al. [PuMB11]). Wu and Hao [WuHa15b] proposent une description détaillée de ce dernier type d'algorithme, ainsi qu'un tableau (tableau 2 [WuHa15b]) récapitulatif des principaux algorithmes heuristiques utilisés dans le problème de la clique maximum.

2.2.3. MAXIMUM WEIGHT CLIQUE PROBLEM (MWCP)

Dans notre problématique de mise en correspondance d'arbres vasculaires, nous ne voulons obtenir qu'une solution, c'est à dire qu'une seule clique maximum. Or les précédents algorithmes présentés peuvent donner plusieurs cliques maximums comme résultat si le graphe en possède

plusieurs (Figure 26). C'est pourquoi nous allons utiliser des algorithmes de clique maximum pondérée (*maximum weight clique, MWCP*). Ces algorithmes peuvent être séparés en deux catégories : ceux utilisant les pondérations sur les vertices (*maximum vertex weight clique, MVWCP*) et ceux utilisant les pondérations sur les arêtes (*maximum edge weight clique, MEWCP*).

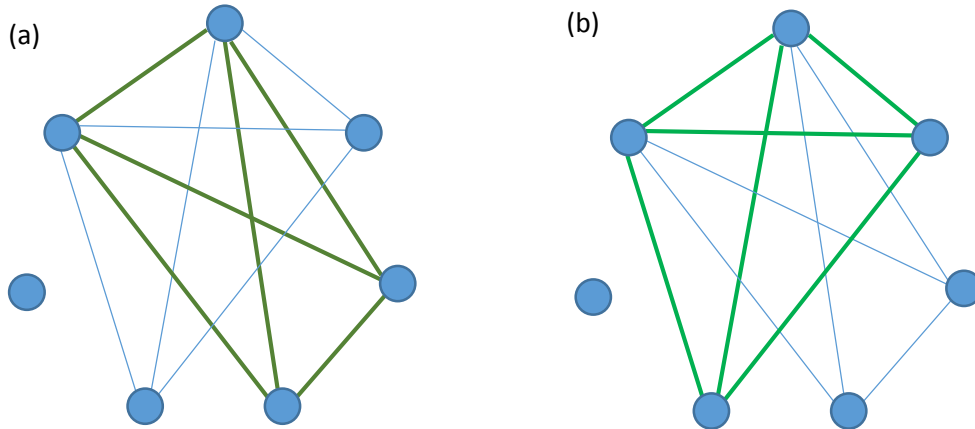


Figure 26 : Exemple de cliques maximums multiples dans un graphe.

Maximum vertex weight clique problem (MVWCP)

Dans le problème de mise en correspondance, nous rappelons que la clique maximum C doit être extraite dans un graphe d'association $G_{i,i+1} = (V_{i,i+1}, E_{i,i+1})$. Les nœuds de ce graphe d'association (ensemble $V_{i,i+1}$) correspondent aux couples de nœuds $v_{i,i+1}^{\alpha_i, \alpha_{i+1}} = (v_i^{\alpha_i}, v_{i+1}^{\alpha_{i+1}})$ potentiellement appariables entre deux arbres T_i et T_{i+1} avec $v_i^{\alpha_i} \in T_i$, $v_{i+1}^{\alpha_{i+1}} \in T_{i+1}$ et $v_{i,i+1}^{\alpha_i, \alpha_{i+1}} \in G_{i,i+1}$. Ces couples de nœuds sont sélectionnés et ajoutés grâce à des métriques (contraintes unaires pour l'algorithme de Metzen et al. [MKSZ09] par exemple). Ce groupe de métriques est normalisé dans l'intervalle $[0; 1]$ (voir le chapitre 3 pour une explication détaillée). Chaque nœud possède donc une valeur dans cet intervalle que l'on peut considérer comme un poids $w(v_{i,i+1}^{\alpha_i, \alpha_{i+1}})$. Plus ce poids est proche de 1, plus les nœuds de ce couple sont similaires. Le poids d'une clique maximum $W(C)$ est défini par la somme de tous les poids de ces vertices. Le problème MVWCP revient à déterminer la clique maximum ayant le poids le plus important parmi toutes les autres présentes dans le graphe. Le problème de la clique maximum (MCP) peut être considéré comme un cas spécial de MVWCP lorsque chaque vertex a un poids égal à 1. Plusieurs algorithmes exacts tel que Östergård [Öste99], Kumlander [Kuml04], Warren and Hicks [WaHi07] propose de résoudre ce problème. De même, plusieurs approches heuristiques ont été développées (Pullan [Pull08], Mannino [Mann99], Wu et al. [WuHG12], Benlic and Hao [BeHa13], Singh and Gupta [SiGu06]).

Maximum edge weight clique problem (MEWCP)

Le problème de la clique maximum peut également utiliser des pondérations sur les arêtes du graphe. Dans notre cas, les arêtes sont ajoutées dans le graphe d'association si les deux chemins entre les deux nœuds sont considérés comme similaires (voir chapitre 3 pour plus de détail). De la même manière que pour les vertices, la similarité entre les chemins est déterminée grâce à un ensemble de mesure de similarités (contraintes binaires pour l'algorithme de Metzen et al. [MKSZ09]) normalisées dans l'intervalle $[0; 1]$. Cette valeur peut également être utilisée comme poids sur les arêtes du graphe. Le problème MEWCP revient donc à déterminer la clique maximum ayant la somme la plus importante des poids des arêtes. Ce problème est également connu sous différents noms tels que « maximum diversity » (Marti et al. [MGDP13]), « remote-clique » (Chandra et al. [ChHa01]) ou encore « maximum edge-weight subgraph » (Macambira [Maca02]). De nombreuses méthodes exactes existent pour la résolution de ce problème (Park et al. [PaLP96], Marti et al. [MaGD10]), ainsi que des méthodes heuristiques (Palubeckis [Palu07], Wang et al. [WHGL14], Silva et al. [SAOM07]).

Dans notre problématique, nous pouvons utiliser les deux techniques pour rechercher la clique maximum pondérée (maximisation des poids des vertices ou des arêtes) comme expliqué précédemment. Nous avons choisi d'utiliser l'algorithme exact proposé par Östergård [Öste99] qui donne de très bon résultat sur les graphes possédant un faible nombre de vertices avec un faible temps d'exécution. De plus, cet algorithme est simple à implémenter et à adapter à notre structure (graphe d'association avec des informations sur les vertices et les arêtes).

2.3. RECHERCHE D'INCIDENCES OPTIMALES

Le choix des vues angiographiques pour les interventions coronariennes percutanées est une étape importante pour le guidage, le déploiement et le positionnement du sent comme expliqué dans le chapitre 1. Ces angles de vues doivent montrer la zone d'intérêt (segment sténosé) avec le moins de superposition et le plus déployé possible.

Dans les années 1990, quelques méthodes pour déterminer l'angle de vue optimale ont été mises en œuvre [ChCa98, DuRG94, SAHN98]. De nos jours, la majorité des algorithmes développés utilisent les deux critères proposés par Chen et al. [ChCa98] pour calculer le taux de raccourcissement (*foreshortening rate*) et le taux de chevauchement (*overlap rate*) pour un segment sélectionné. Le

premier critère est calculé à partir des différentes projections de la ligne centre du segment d'intérêt (vaisseau sténosé). Il permet d'obtenir une visualisation déployée du segment de l'artère coronaire. Le deuxième critère mesure le taux de superposition du vaisseau sténosé avec les autres branches de l'arbre coronaire. Le segment doit être le plus isolé pour permettre au chirurgien de procéder à son intervention dans les meilleures conditions possibles. Une visualisation des différents résultats est également proposée sous la forme d'une carte de vue optimale (*optimal view map*). Une carte pour chaque critère peut être également réalisée indiquant les valeurs obtenues pour chaque projection correspondant à un couple d'angle possible du C-arm.

Green et al. [GCHM05] à partir de la méthode de Chen et al. [ChCa98] compare sur plusieurs données (156 segments issu de 149 patients) les résultats des angles de vues optimaux obtenus par les médecins et par l'algorithme développé. Les vues sélectionnées par les opérateurs possèdent plus de raccourcissement sur les vaisseaux d'intérêts que celles proposées par la méthode de Chen et al. [ChCa98].

Nous avons choisi de distinguer deux approches différentes : les méthodes nécessitant la création d'une base de données de patient pour déterminer des angles de vues optimaux dits universels en fonction des branches vasculaires et les méthodes spécifiques au patient.

Nous allons commencer par décrire les différentes méthodes existantes de la dernière approche : les méthodes spécifiques au patient [ChCa98, KMJK08, WKCC02, WKCC03, WYCA14]. Nous nous intéresserons plus particulièrement aux méthodes qui diffèrent de celle proposée par Chen et al. [ChCa98].

Wink et al. [WKCC02, WKCC03] proposent une méthode basée sur les deux critères de Chen et al. [ChCa98]. Cependant, pour avoir une orientation optimale du C-arm, le segment d'intérêt doit être le moins raccourci possible et le moins superposé avec les structures avoisinantes et ce durant tout le cycle cardiaque. Pour pouvoir prendre en compte le mouvement des artères coronaires durant le cycle cardiaque, tous les arbres à différentes phases sont modélisés dans un seul et même volume. Cela permet de calculer la carte contenant les taux de chevauchement grâce à l'intersection de ce volume avec la modélisation du segment d'intérêt. Pour une implémentation plus efficace en temps de calcul, le taux de chevauchement est calculé uniquement pour les angles de vue ayant un faible taux de raccourcissement.

Chen et al [CLZW14] propose également plus récemment un algorithme déterminant l'angle de vue optimale pour un segment sur l'ensemble du cycle cardiaque à partir de données 4D MSCT. Les deux mêmes critères (raccourcissement et chevauchement) sont utilisés. Cependant, le processus est

différent de celui de Wink et al. [WKCC02, WKCC03]. La carte optimale est calculée pour chaque volume (segment d'intérêt et les autres vaisseaux) de la séquence en ne gardant que les angles de vue avec moins de 2% de raccourcissement et aucun chevauchement. L'intersection de toutes les cartes optimales est par la suite considérée comme la carte de vue optimale pour le cycle cardiaque entier. Cependant, il n'existe pas de mise en correspondance entre les différents segments des différents volumes. Dans leur test (une séquence de huit volumes), le segment d'intérêt d'une longueur de 15 voxels, est choisi près d'une bifurcation. De plus leur méthode ne prend pas en compte la nature non-isomorphe des arbres coronaires 3D de la séquence.

Kitslaar et al. [KMJK08] propose une méthode permettant de visualiser les bifurcations des artères coronaires de façon optimale. Comme pour les autres méthodes, un volume 3D des artères est disponible pour un instant donné (segmenté à partir d'un scanner). Le critère utilisé pour calculer le taux de raccourcissement est identique à celui proposé par Chen et al. [ChCa98]. Celui-ci est donné par le rapport de la longueur de la ligne centrale 2D du segment d'intérêt projeté selon une direction et la longueur de la ligne centrale du segment 3D. De même, Kitslaar et al. [KMJK08] réutilise le critère concernant le taux de chevauchement entre le segment d'intérêt (ici la bifurcation) et le reste de l'arbre coronaire. Comme Wink et al. [WKCC02, WKCC03], ce taux de chevauchement est calculé en exploitant le stencil buffer de OpenGL [SSKL13]. Celui-ci permet de calculer l'intersection entre deux volumes en comptant le nombre de pixels couverts par les deux volumes lors de la rasterisation. Kitslaar et al. [KMJK08] introduit un troisième critère permettant de calculer le taux de chevauchement propre à la bifurcation. Ce critère est appelé « *internal overlap* » et est utilisé pour empêcher la bifurcation de se superposer avec elle-même permettant une meilleure visualisation sur les projections angiographiques. Les trois branches de la bifurcation sont séparées et le taux de chevauchement interne entre ces trois segments est calculé également à l'aide de stencil buffer. Les trois cartes, représentant les taux de raccourcissement, de chevauchement et de chevauchement interne, sont calculées selon toutes les orientations possibles et sont ensuite combinées en une seule carte.

Tu et al. [THKW11, TJHO12, Tu12] propose une nouvelle méthode permettant de déterminer des angles de vues optimales pour un segment d'intérêt ou d'une bifurcation sans avoir besoin de reconstruire tout l'arbre 3D. Juste la portion de branche sélectionnée est reconstruite. Après calcul du taux de raccourcissement, cinq propositions d'angles avec une différence d'au moins 15 degré sont sélectionnés. La prédiction de chevauchement sur ces vues est réalisée via l'image angiographique projetée.

Plus récemment, Wang et al. [WYCA14] ont proposé une méthode déterminant l'angle optimal pour un segment d'intérêt (« *interesting vessel segment* » - IVS) ou une bifurcation (« *interesting vessel*

bifurcation » - IVB). De nouveaux critères ont été intégrés dans la recherche de l'angle de vue optimale pour ces deux types. Tout d'abord pour le segment d'intérêt, trois critères sont calculés : « *foreshortening rate* », « *stenosis rate* », et « *overlap rate* ». Un nouveau critère est donc utilisé concernant la forme et le degré de la sténose. L'angle de vue optimale d'une sténose considère que sa projection doit montrer le moins de surface possible du vaisseau (minimisation) pour pouvoir visualiser au mieux la sténose. Ensuite, pour les bifurcations (IVB), trois critères sont également utilisés : « *foreshortening rate* », « *overlap rate* » et « *adjacent spacing rate* ». Comme pour Kitslaar et al. [KMJK08], deux types de chevauchement sont utilisés : un global et un interne. De plus, le nouveau critère (« *adjacent spacing rate* ») utilise l'espacement entre les deux branches filles de la bifurcation grâce à un calcul d'aire (à maximiser pour obtenir l'angle de vue optimale). En plus de nouveaux critères, Wang et al. [WYCA14] permet de calculer l'incidence optimale sur de multiples segments ou bifurcations.

Garcia et al. [GMCK09] et Bi et al. [BiGS10] présentent des méthodes de calculs de l'angle de vue optimale sur un panel de patients afin de déterminer de façon générale les angles de vues optimaux. Ces méthodes sont globalement similaires à celle présentée ci-dessus : elles s'appuient sur la création de cartes de raccourcissement et de chevauchement pour un ensemble de patient en fonction des angles de vues. Les méthodes utilisées pour la création des volumes 3D et le calcul des incidences sont similaires à celle présentée précédemment (taux de raccourcissement et de chevauchement). Les cartes sont ensuite fusionnées pour obtenir une carte globale indiquant la zone contenant l'angle de vue optimale pour ces patients. Garcia et al. [GMCK09], à partir de 137 patients, proposent une carte globale pour les différentes branches vasculaires anatomiques. Bi et al. [BiGS10] travaillent plus particulièrement sur les veines coronaires (14 patients), mais sa méthode est aussi réutilisable pour les artères coronaires. Il propose également d'obtenir des angles de vues optimaux pour plusieurs segments d'intérêts (c'est-à-dire un chemin) grâce à la fusion des cartes obtenues pour les différents segments constituant le chemin. Afin de calculer les cartes d'incidences optimales, les vues utilisées pour générer tous les modèles 3D représentent la même phase du cycle cardiaque pour ne pas être influencé par les mouvements des coronaires.

Ces deux dernières méthodes [BiGS10, GMCK09] non spécifique au patient ne sont efficace que si la carte de vue optimale statique est construite à partir d'une large population. Cependant, elles peuvent être trop générique pour un patient donné même si Garcia et al. [GMCK09] propose une méthode prenant en compte la variabilité anatomique inter-patient (dominance droite et gauche). De plus, ces méthodes considèrent que les artères coronaires des différents patients sont appariées

(Garcia et al. [GMCK09] calcule les cartes de vues optimales générales pour différentes branches anatomiques). Cette mise en correspondance est effectuée manuellement. Comme nous nous focalisons sur une structure dynamique ayant un mouvement complexe, l'angle de vue optimale devrait être calculé sur l'ensemble du cycle cardiaque. Seulement quelques méthode (Wink et al. [WKCC02, WKCC03] et Chen et al. [CLZW14]) prennent en compte le mouvement des artères coronaires. A notre connaissance, aucune méthode de détermination de l'angle de vue optimale n'utilise une méthode automatique pour la mise en correspondance des artères coronaires sur l'ensemble du cycle cardiaque. En outre, aucune méthode ne propose de déterminer l'angle de vue optimale pour un segment et une bifurcation d'intérêt sur l'ensemble du cycle cardiaque avec mise en correspondance de cet objet sur les différentes phases.

Nous avons alors choisi de développer une méthode patient-spécifique afin de déterminer l'angle de vue optimale sur l'ensemble du cycle cardiaque grâce à notre algorithme de mise en correspondance 3D+T décrit dans le chapitre 3. Comme expliqué dans le chapitre 4, nous avons calculé la carte de vue optimale pour différents objets d'intérêt : un segment, de multiples segments ou une bifurcation. La carte optimale permet de déterminer un angle de vue optimale soit pour déployer un stent au niveau d'un segment ou d'une bifurcation sténosés (angle local) ou soit pour guider le cathéter ainsi que d'autres outils dans les vaisseaux, par exemple de la première branche des artères coronaires à la zone de lésion (angle global).

CHAPITRE 3

MISE EN CORRESPONDANCE

D'ARTERES CORONAIRES

SOMMAIRE

3.1.	Matching of two vascular trees.....	60
3.1.1.	Hierarchical tree approach.....	61
3.1.2.	Association graph approach.....	63
3.1.2.1.	Association graph building.....	63
3.1.2.2.	Maximum clique extraction.....	66
3.1.3.	Adaptation to our coronary matching problem.....	67
3.1.3.1.	Similarity measure choice.....	67
3.1.3.2.	Artificial node insertion in coronary trees.....	71
3.2.	Matching of the coronary tree sequence.....	75
3.2.1.	Mosaic-based matching of coronary tree sequence.....	76
3.2.2.	Characteristics of merging and union nodes.....	78
3.2.3.	Building of the union tree $UT_{i,i+1}$ from the matching result $M_{i,i+1}$ between T_i and T_{i+1} ..	80
3.2.4.	Building of the mosaic tree $US_{0,i+1}$ of the tree sequence $S_{0,i+1}$	83
3.2.4.1.	Node merging from $US_{0,i}$ and $UT_{i,i+1}$ related by their common phase i in $US_{0,i+1}$	85
3.2.4.2.	Insertion in $US_{0,i+1}$ of not merged node from $US_{0,i}$ and $UT_{i,i+1}$	85
3.2.4.3.	Merging of newly inserted union nodes with an existing node of $US_{0,i+1}$	86
3.2.4.4.	Correction of the newly inserted node from $UT_{i,i+1}$ with artificial node insertion in $US_{0,i-1}$	87
3.2.4.5.	Last tree T_{i+1} correction thanks to supplementary information from $US_{0,i}$	89
3.3.	Conclusion	89

Dans le cadre du planning d'une angioplastie, nous recherchons à suivre la zone sténosée ou le chemin du guide dans les artères coronaires en mouvement afin d'obtenir une incidence optimale prenant en compte tout le cycle cardiaque. Ainsi, nous proposons de mettre en correspondance chaque segment de l'arbre coronaire non isomorphe dans l'ensemble de la séquence 3D+T.

Ce chapitre présente une nouvelle approche d'appariement des arbres vasculaires et plus particulièrement les artères coronaires. Comme expliqué dans le chapitre 2, de par les caractéristiques des arbres coronaires de la séquence, nous avons choisi d'implémenter deux méthodes de référence et de les adapter à notre problématique. La première méthode de référence, proposée par Lohe et al. [LKZP07], est une approche hiérarchique basée sur la construction d'un arbre de recherche symétrique. La deuxième méthode de référence, basée sur la construction d'un graphe d'association et l'extraction de la clique maximum, est proposée par Metzen et al. [MKSZ09] ainsi que par Tschirren et al. [TMPH05].

Dans un premier temps, nous appliquons ses deux méthodes de référence à deux arbres coronaires T_i et T_{i+1} issus deux phases successives du cycle cardiaque obtenus à partir de CT (section 1). Après la description de ces deux méthodes (section 3.1.1 pour la première méthode [LKZP07] et section 3.1.2 pour la deuxième approche [MKSZ09, TMPH05]), nous les avons adaptées à nos données non isomorphes en choisissant tout d'abord de nouvelles mesures de similarité (section 3.1.3). Le seuillage de la première mesure permet de réduire le nombre de cas à traiter par la suite, c'est-à-dire en sélectionnant une partie des paires de nœuds potentiellement appariés. Nous voulons que cette étape soit rapide en temps d'exécution et qu'elle réduise considérablement l'espace de recherche. C'est pourquoi, nous utilisons les coordonnées spatiales stockées dans chaque nœud d'arbre pour définir une région d'intérêt autour d'un nœud donné, sélectionnant ainsi les nœuds potentiellement semblables dans les autres phases cardiaques. Les autres mesures seuillées sont importantes : elles définissent la mise en correspondance entre les nœuds sélectionnés précédemment parmi les meilleures. Ces mesures doivent être exécutées dans un temps raisonnable et être robustes. Nous avons choisi d'utiliser des mesures géométriques basées sur les lignes centrales permettant de prendre en compte le mouvement non rigide des artères coronaires ainsi que leurs différences topologiques. Trois mesures (position, tangente et direction) à partir desquelles une distance est calculée basée sur l'algorithme Dynamic Time Warping [KePa01] sont proposées pour déterminer la similarité entre les chemins (c'est-à-dire entre deux paires de nœuds sélectionnées).

La mise en correspondance entre deux arbres contient à ce moment-là des couples de nœuds qui sont vraiment similaires par rapport aux différentes mesures de similarité précédentes. La précision et la qualité de l'appariement des nœuds sont pour nous plus importants que le nombre de

nœuds mis en correspondance. En effet, augmenter le nombre d'appariements (c'est-à-dire proposer un seuillage moins strict des mesures de similarité) risquerait d'augmenter le nombre de nœuds incorrectement mis en correspondance. La qualité et la précision des processus suivants, correspondant à la mise en correspondance de toute la séquence et à la détermination de l'angle de vue optimale, dépendant de cette méthode peuvent alors être affectées. Nous avons donc un faible nombre de nœuds mis en correspondance par rapport à l'ensemble de nœuds existant dans les deux arbres. Effectivement, à cause des erreurs d'extraction ou de reconstruction (arbres non-isomorphes) certains nœuds de bifurcations ou nœuds feuilles ne sont pas mis en correspondance. Afin d'obtenir un appariement optimal et plus complet, nous introduisons une nouvelle étape consistant à insérer des nœuds artificiels dans les arbres correspondant à ces nœuds non appariés (section 3.1.3.2). Les deux arbres mis en correspondance sont ainsi complétés par des informations supplémentaires associés aux nœuds artificiels. En plus de la hiérarchie et du résultat de la précédente mis en correspondance entre les deux arbres successifs, des mesures de similarités sont utilisées afin de déterminer quand insérer un nœud artificiel. Nous avons choisi d'utiliser les trois mêmes mesures (coordonnées, tangente et direction à partir desquelles la distance euclidienne est calculée par l'algorithme Dynamic Time Warping [KePa01]) que précédemment afin de prendre en compte la différence de distance et de tortuosité entre les deux arbres à apparier. Une étape de vérification de la hiérarchie et de correction de conflit est insérée pour chaque nœud artificiel créé rendant ainsi le programme plus robuste.

Cette méthode d'appariement deux à deux n'était cependant pas suffisamment robuste pour l'appliquer directement à l'ensemble des arbres de la séquence $S_{0,I-1} = \{T_0, \dots, T_i, T_{i+1}, \dots, T_{I-1}\}$. Elle prend en compte uniquement les différences de topologie entre arbres successifs et non celles de l'ensemble des arbres de la séquence. C'est pourquoi dans un deuxième temps, nous proposons dans la section 3.2 une nouvelle méthode globale appariant l'ensemble des arbres de la séquence. Pour cela, l'algorithme proposé construit un arbre complet, nommé arbre mosaïque, fusionnant toutes les informations des arbres de la séquence ainsi que leurs appariements segments par segments.

L'algorithme de mise en correspondance entre deux arbres T_i et T_{i+1} devient une étape préliminaire de notre algorithme d'appariement globale. A partir du résultat de cette mise en correspondance corrigée et des deux arbres T_i et T_{i+1} complétés et corrigés, un arbre d'union $UT_{i,i+1}$ est construit (section 3.2.3). Dans cet arbre d'union, une étiquette est définie pour chaque nœud en fonction de son statut : soit le nœud correspond à un couple de nœuds appariés soit il représente un nœud seul sans correspondance.

Ensuite, à partir de cet arbre d'union, un arbre mosaïque temporaire $US_{0,i+1}$ est construit par la fusion d'arbres d'union deux à deux jusqu'à la fin de la séquence et l'obtention de l'arbre mosaïque global $US_{0,I-1}$ (section 3.2.4). L'arbre mosaïque temporaire $US_{0,i+1}$ représente la fusion du précédent arbre mosaïque $US_{0,i}$ et de l'arbre union $UT_{i,i+1}$ correspondant aux deux arbres appariés T_i et T_{i+1} . Premièrement, l'arbre mosaïque $US_{0,i+1}$ est initialisé avec les nœuds de $US_{0,i}$. Deuxièmement, tous les nœuds de $UT_{i,i+1}$ non présents dans $US_{0,i+1}$ y sont ajoutés (c'est-à-dire les nœuds de l'arbre T_{i+1}). Les nœuds dans $US_{0,i+1}$ représentant le même nœud dans différents cycles cardiaques sont fusionnés. Quant aux autres nœuds, pour compléter la mise en correspondance globale, des nœuds artificiels sont recherchés dans les différentes phases de $S_{0,i+1}$. A chaque étape, chaque appariement est vérifié lors de la fusion. Les erreurs de hiérarchie, de conflit entre nœuds ou d'insertion de nœuds artificiels apparaissant avec ce supplément d'information sont corrigées. Ainsi, les arbres de la séquence $S_{0,i+1}$ sont complétés et corrigés par les étapes précédentes liées à la construction de l'arbre mosaïque $US_{0,I-1}$. De ce fait, le résultat de la mise en correspondance entre deux arbres devient de plus en plus robuste lors de la progression de celle-ci en fonction de l'avancée du traitement de la séquence. En effet, l'arbre T_i a été complété par les nœuds artificiels, correspondants à des informations supplémentaires, présents dans les autres arbres de la séquence $S_{0,i}$ obtenu lors des appariements précédents. Au final, le nombre de nœuds des arbres modifiés T_i de la séquence $S_{0,I-1}$ double par rapport aux arbres T_i initiaux.

Ce chapitre, rédigé en anglais, est tiré de la synthèse de deux articles soumis [FeNT16a, FeNT16b] décrivant les différentes étapes de mise en correspondance d'artères coronaires. Le premier article concerne la mise en correspondance de deux arbres représentant deux phases successives (section 3.1) et le deuxième article correspond à une amélioration de cette méthode par rapport à notre problématique, c'est-à-dire la mise en correspondance complète de l'ensemble de la séquence d'arbres coronaires 3D (section 3.2).

3.1. MATCHING OF TWO VASCULAR TREES

Firstly in this section, the two reference inexact tree matching algorithms based on a hierarchical tree approach [LKZP07] and an association graph approach [MKSZ09, TMPH05] are described. To adapt these matching approaches to our coronary tree structure, the geometric measures and graph characteristics are next modified. Moreover, a new step based on the artificial

node insertion is proposed to take into account the topology variation between trees. An additional hierarchical correction step is also proposed to avoid conflicts because of artificial node insertion.

3.1.1. HIERARCHICAL TREE APPROACH

Lohe et al. [LKZP07] proposed a symmetric tree search algorithm using the hierarchical tree structure. Let be $TS_{i,i+1} = (V_{TS_{i,i+1}}, E_{TS_{i,i+1}})$ the search tree which corresponds to an intermediate data structure created from the two consecutive trees T_i and T_{i+1} . $V_{TS_{i,i+1}}$ represents the node set and $E_{TS_{i,i+1}}$ the edge set. Each node $v_{TS_{i,i+1}}^{\alpha_i, \alpha_{i+1}} \in V_{TS_{i,i+1}}$ of this search tree represents a pair of the two tree nodes $v_i^{\alpha_i}$ and $v_{i+1}^{\alpha_{i+1}}$ respectively from V_i and V_{i+1} . To build the search tree $TS_{i,i+1} = (V_{TS_{i,i+1}}, E_{TS_{i,i+1}})$, two metric sets are used: a consistency constraint set CC and a similarity measure set SM .

- 1) **Consistency constraint** $cc^\lambda(v_i^{\alpha_i}, v_{i+1}^{\alpha_{i+1}}) \rightarrow [true, false] \in CC$ which decides if two nodes $v_i^{\alpha_i} \in T_i$ and $v_{i+1}^{\alpha_{i+1}} \in T_{i+1}$ represent a valid matched node pair. This first constraint set selects the most relevant assignments $v_{TS_{i,i+1}}^{\alpha_i, \alpha_{i+1}} = (v_i^{\alpha_i}, v_{i+1}^{\alpha_{i+1}}) \in V_{TS_{i,i+1}}$ among all possible cases and inserts them in the search tree. Node selection from two consecutive trees can be based on various characteristics described by Lohe et al. [LKZP07]. The inheritance relationship between two nodes must be equal. The leaf nodes mustn't be matched. The length difference of the two paths from their root to their current node of the two consecutive trees must be smaller than a given threshold depending on the maximum path length. The difference of 3D lumen volume between the two branches must also be smaller than another threshold.
- 2) **Similarity measure** $sm^\mu(v_{TS_{i,i+1}}^{\alpha_i, \alpha_{i+1}}) \in SM$ which evaluates the matching quality of the most previously relevant assignments $v_{TS_{i,i+1}}^{\alpha_i, \alpha_{i+1}} = (v_i^{\alpha_i}, v_{i+1}^{\alpha_{i+1}}) \in V_{TS_{i,i+1}}$. This second metric set is used to determine the set of best assignments among all relevant assignments. Firstly, Lohe et al. [LKZP07] compare the lengths of the two paths $p_i(r_i, v_i^{\alpha_i})$ and $p_{i+1}(r_{i+1}, v_{i+1}^{\alpha_{i+1}})$ from the root node r_i or r_{i+1} to the current node $v_i^{\alpha_i}$ or $v_{i+1}^{\alpha_{i+1}}$ in the two trees. Secondly, they compare the angles between two other paths from the nearest matched parent node in the search tree to the current node from the two trees T_i and T_{i+1} . Each similarity measure $sm^\mu(v_{TS_{i,i+1}}^{\alpha_i, \alpha_{i+1}})$ is normalized in the range [0,1] by the function $2 \cdot \frac{1-sm^\mu}{s(sm^\mu)-1} - 1$, with $s(sm^\mu)$ the threshold of the similarity measure sm^μ . This normalization allows combining the μ similarity measures

with their weight factor $w(sm^\mu)$. For each search tree node $v_{TS_{i,i+1}}^{\alpha_i, \alpha_{i+1}}$, this combination result $SM(v_{TS_{i,i+1}}^{\alpha_i, \alpha_{i+1}}) = \sum_{sm^\mu \in SM} [sm^\mu(v_{TS_{i,i+1}}^{\alpha_i, \alpha_{i+1}}) \times w(sm^\mu)]$ is computed. Then, this value is included to the cumulated sum of similarity results of its parent node $SM(mp(v_{TS_{i,i+1}}^{\alpha_i, \alpha_{i+1}}))$. This final result $SM(v_{TS_{i,i+1}}^{\alpha_i, \alpha_{i+1}}) + SM(mp(v_{TS_{i,i+1}}^{\alpha_i, \alpha_{i+1}}))$ is saved in the node $v_{TS_{i,i+1}}^{\alpha_i, \alpha_{i+1}}$.

After the construction of the search tree (see the Algorithm1 in [LKZP07]), the leaf node $v_{TS_{i,i+1}}^{\alpha_i, \alpha_{i+1}}$ having the maximum value of cumulated similarity measures $SM(v_{TS_{i,i+1}}^{\alpha_i, \alpha_{i+1}})$ is looked in the search tree $TS_{i,i+1}$. Backtracking from this leaf node to the root gives the best tree matching M for this algorithm.

The main advantage of the algorithm of Lohe et al. [LKZP07] is its adaptability. Thanks to this hierarchical approach, each computed similarity measure sm^μ depends on the previously matched vertices. The consistency constraint cc^λ and the similarity measure sm^μ can be chosen according to our application. Indeed, most of metrics described in [LKZP07] are not suitable for our real data due to the nature of the coronary arteries and in particular the non-isomorphic tree structure and the non-rigid coronary movement. For instance, topology differences can involve a non-identical inheritance relationship between two nodes or an inaccuracy concerning the branch radii. Besides, with these tortuous arteries and their non-rigid motion, the global angle-based measure is difficult to compute. Nevertheless, without this metric, the algorithm is not enough robust and can arise false node matching if only the path length comparison is used. Two paths, which have the same length, may have few similarity i.e. not the same spatial coordinates and directions of trees.

Despite previous advantages, this general algorithm is not optimal for a specific application such as coronary matching. Another drawback is that the performance depends on the tree size (number of edges, hierarchical level, etc.). This algorithm is able to find the optimal matching for small trees i.e. having a low hierarchical level and a low number of vertices, but if the trees have more than hundred vertices, with the exponential complexity, the execution time is very long. Another version of this algorithm, with a reduction of the search space, allows performing on larger trees. In our case, the first version is enough since the coronary trees have only few vertices (lower than 100).

3.1.2. ASSOCIATION GRAPH APPROACH

Metzen et al. [MKSZ07, MKSZ09] and Tschirren et al. [TMPH05] proposed to match two vessel trees by an association graph approach derived from the Pelillo et al.'s algorithm [PeSZ99]. Based on these two methods, we propose an association graph algorithm to determine the matching of two successive coronary trees. The association graph $G_{i,i+1} = (V_{G_{i,i+1}}, E_{G_{i,i+1}})$ is an intermediate data structure created from the two trees T_i and T_{i+1} . $V_{G_{i,i+1}}$ and $E_{G_{i,i+1}}$ are respectively the node set and the edge set. Several successive steps are needed to obtain the best matching $M_{i,i+1}$ of these two consecutive trees (Figure 27).

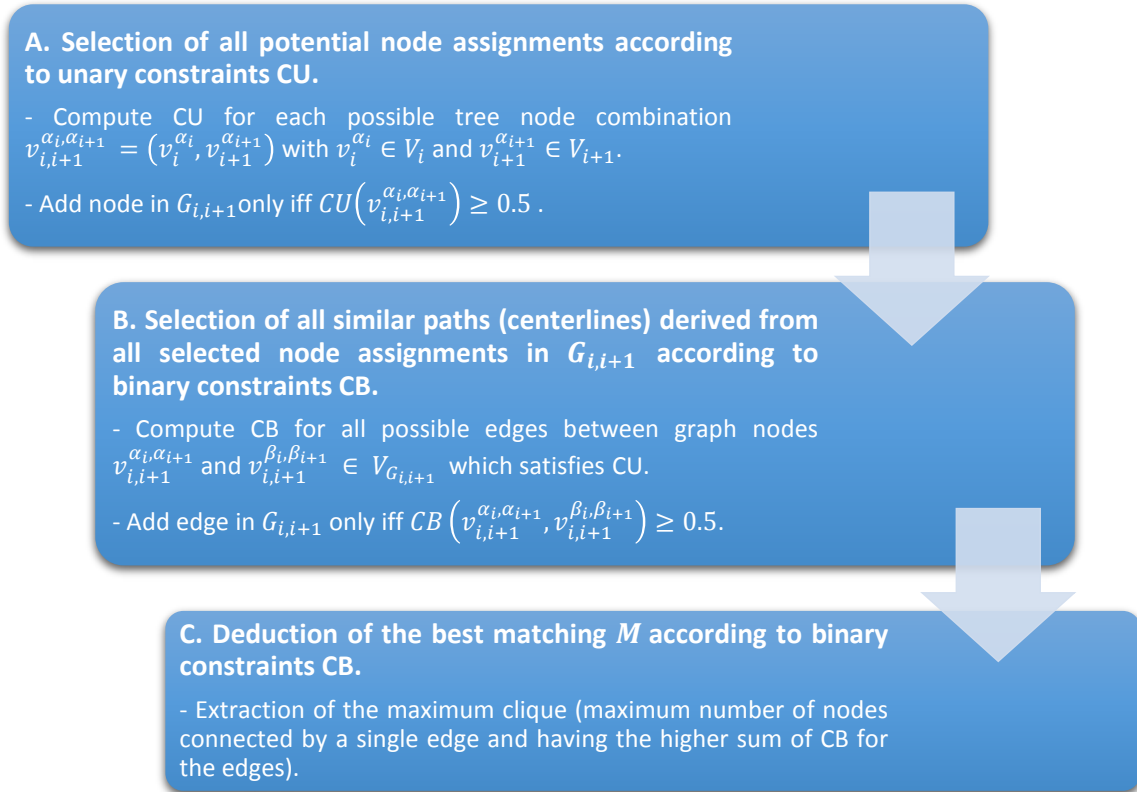


Figure 27 : Three main steps to obtain the matching result $M_{i,i+1}$ with the association graph approach based on Metzen et al 's algorithm [MKSZ09].

3.1.2.1. ASSOCIATION GRAPH BUILDING

First, the matching algorithm starts with the building of the association graph $G_{i,i+1}$ in two steps based on unary constraint set CU , i.e. measures used to determine node similarities (Figure 27.A)

and binary constraint set CB , i.e. measures used to determine similarity between centerlines (Figure 27.B).

UNARY CONSTRAINTS CU

The unary constraints CU are primarily based on local node properties. They can relate to the hierarchical level or the spatial coordinates of each node, the length or mean radius between the current node and the root node or the size of its subtree [MKSZ09, TMPH05]. However, these node-to-node similarity measures can be penalized by errors related to the 3D tree extraction (segmentation or reconstruction). The level of two corresponding nodes can be different between trees and two corresponding branches can have different lengths. Thus, other reliable measures based on the centerline characteristics can be used. Metzen et al. [MKSZ09] used the normalized path between the root and a given node, called also normalized polyline. The polyline similarity is evaluated on the basis of geometric criteria such as the area difference of the curvatures or the deviation of the tangents at each point of the polylines. The geometric normalization allow to obtain a measure which is independent of the orientation, the length and the absolute position of the polylines. Even if these constraints give a high rate of correct assignments, they have a major drawback: the execution time can increase significantly [MKSZ09]. Furthermore, two normalized polylines can have similar properties while the centerlines, without the geometric normalization, can have much dissimilarities such as significant differences of length or orientation.

BINARY CONSTRAINTS CB

Basic binary constraints CB are based on the difference of Euclidean or topological distance [MKSZ09, TMPH05] between two nodes in the association graph (i.e. two extracted paths). However, these constraints, as node-to-node similarity measures, can be affected by the same extraction errors. Binary constraints can also use the normalized polylines such as the comparison of their directions or the area between them [MKSZ09]. However, as this kind of constraints can be too expensive, the comparison of the path length and the path curvature (path length divided by the Euclidean distance between the source and the target of the path) are a good alternative: acceptable results almost equivalent to those obtained by polylines are obtained with faster execution time. As the similarity measures used in the Lohe et al.'s algorithm (section 3.1.1), we added three binary constraints based on the Dynamic Time Warping [KePa01] to measure the degree of similarity between two paths.

GRAPH BUILDING

Firstly, this graph matching is carried out by minimizing a cost function from unary and binary constraints of nodes.

The first step inserts nodes $v_{i,i+1}^{\alpha_i, \alpha_{i+1}} = (v_i^{\alpha_i}, v_{i+1}^{\alpha_{i+1}})$ which are composed of two tree nodes $v_i^{\alpha_i} \in V_i$ and $v_{i+1}^{\alpha_{i+1}} \in V_{i+1}$ in the association graph $G_{i,i+1}$ (Equation (1)).

$$\begin{aligned} & \mathbf{V}_{G_{i,i+1}} \\ &= \{ \mathbf{v}_{i,i+1}^{\alpha_i, \alpha_{i+1}} = (v_i^{\alpha_i}, v_{i+1}^{\alpha_{i+1}}) \in V_i \times V_{i+1} \mid \sum_{cu^\gamma \in CU} w(cu^\gamma) \times cu^\gamma(\mathbf{v}_{i,i+1}^{\alpha_i, \alpha_{i+1}}) \geq 0.5 \} \end{aligned} \quad (1)$$

with $cu^\gamma(\mathbf{v}_{i,i+1}^{\alpha_i, \alpha_{i+1}}) \in [0, 1]$, $w(cu^\gamma) \in [0, 1]$, $\sum_\gamma w(cu^\gamma) = 1$

A first selection is based on the unary constraint set CU: only the pair $v_{i,i+1}^{\alpha_i, \alpha_{i+1}}$ considered as a potential assignment is added. The result of each unary constraint cu^γ is normalized in the range $[0, 1]$ through the function $\frac{1-cu^\gamma}{2^{s(cu^\gamma)-1}}$, with $s(cu^\gamma)$ the selectivity parameter (threshold). The constraint cu^γ equals to 1 indicates a strong similarity. Several unary constraints can be combined with the weight factor $w(cu^\gamma)$. To consider the node $v_{i,i+1}^{\alpha_i, \alpha_{i+1}}$ as a potential assignment, the weighted sum of all unary constraints must be greater or equal to 0.5.

The second step inserts edges in the association graph $G_{i,i+1}$ between two nodes $v_{i,i+1}^{\alpha_i, \alpha_{i+1}} = (v_i^{\alpha_i}, v_{i+1}^{\alpha_{i+1}}) \in V_{G_{i,i+1}}$ and $v_{i,i+1}^{\beta_i, \beta_{i+1}} = (v_i^{\beta_i}, v_{i+1}^{\beta_{i+1}}) \in V_{G_{i,i+1}}$ with $v_i^{\alpha_i}, v_i^{\beta_i} \in V_i$ and $v_{i+1}^{\alpha_{i+1}}, v_{i+1}^{\beta_{i+1}} \in V_{i+1}$ and $1 \leq \alpha_i \leq n_i$, $1 \leq \alpha_{i+1} \leq n_{i+1}$ and $1 \leq \beta_i \leq n_i$, $1 \leq \beta_{i+1} \leq n_{i+1}$ (Equation (2)).

$$\begin{aligned} & \mathbf{E}_{G_{i,i+1}} \\ &= \{ (v_{i,i+1}^{\alpha_i, \alpha_{i+1}}, v_{i,i+1}^{\beta_i, \beta_{i+1}}) \in V_{G_{i,i+1}} \times V_{G_{i,i+1}} \mid \sum_{cb^\delta \in CB} w(cb^\delta) \times cb^\delta(v_{i,i+1}^{\alpha_i, \alpha_{i+1}}, v_{i,i+1}^{\beta_i, \beta_{i+1}}) \geq 0.5 \} \end{aligned} \quad (2)$$

with $cb^\delta(v_{i,i+1}^{\alpha_i, \alpha_{i+1}}, v_{i,i+1}^{\beta_i, \beta_{i+1}}) \in [0, 1]$, $w(cb^\delta) \in [0, 1]$, $\sum_\delta w(cb^\delta) = 1$

The edge between $v_{i,i+1}^{\alpha_i, \alpha_{i+1}}$ and $v_{i,i+1}^{\beta_i, \beta_{i+1}}$ is added in the edge set $E_{G_{i,i+1}}$ if the two paths $p_i(v_i^{\alpha_i}, v_i^{\beta_i}) \in T_i$ and $p_{i+1}(v_{i+1}^{\alpha_{i+1}}, v_{i+1}^{\beta_{i+1}}) \in T_{i+1}$ derived from these two nodes are considered as similar according to the set of binary constraints CB. Similarly to unary constraints, each δ binary constraint $cb^\delta(v_{i,i+1}^{\alpha_i, \alpha_{i+1}}, v_{i,i+1}^{\beta_i, \beta_{i+1}})$ is normalized in the range $[0, 1]$ by the same normalization function

and the selectivity parameter $s(cb^\delta)$, and next weighted by the factor $w(cb^\delta)$. If the weighted sum of the binary constraint set is greater or equal to 0.5 for the two potential assignments $(v_{i,i+1}^{\alpha_i, \alpha_{i+1}}, v_{i,i+1}^{\beta_i, \beta_{i+1}})$, the two corresponding paths are considered as similar and an edge is added between them.

3.1.2.2. MAXIMUM CLIQUE EXTRACTION

Afterwards, when the association graph $G_{i,i+1}$ is established, the complete sub-graph having maximum cardinality must be detected. This matching problem corresponds to search the maximum clique C_{max} into the association graph $G_{i,i+1}$, i.e. the maximum number of nodes that are connected by an edge and corresponds to a maximal sub-graph isomorphism. The best matching $M_{i,i+1}$ is deduced from this maximum clique C_{max} : each node of $M_{i,i+1}$ corresponds to a pair of matched nodes in the two trees T_i and T_{i+1} (Figure 27.C).

The detection of the maximum clique is a NP-hard problem [Karp72]. Many methods apply some heuristics to find a good approximation of the maximum clique such as Metzen et al.'s algorithm [MKSZ09]. They chose to adopt heuristics proposed by Pelillo [Peli02] based on the Motzkin-Straus theorem [MoSt65] and conclude with a step which completes the matching with further correct assignments. Indeed, some correct matched nodes in the association graph, which are not present in the maximum clique C_{max} , might be connected to almost all nodes of this clique. This heuristic helps to take into account the few missing edges to add the correct nodes in C_{max} . Other methods used exact algorithms to detect the maximum clique in a graph such as Tschirren et al.'s algorithm [TMPH05]. Even if more efficient algorithms are known for some special cases, they used the method proposed by Carraghan et Pardalos [CaPa90] which performs well for their small graph problem.

In our case, the association graph $G_{i,i+1}$ has also few nodes (less than 20 nodes) because of the selection based on unary constraints CU . We chose to adopt the Östergard's algorithm [Öste02] which is a fast algorithm based on the Carraghan et Pardalos' method [CaPa90] for different graph sizes and gives good result for small graphs (less than thousand nodes). If the Östergard's algorithm [Öste02] is preserved, all maximum cliques are found. A simple extension is used: only one maximum clique, with the higher sum of binary constraints for all of these edges, is retained.

As for Lohe et al.'s algorithm, an advantage of this approach is that the unary and binary constraints used to build the association graph can be adapted to our coronary tree application.

Moreover, using a geometric feature set can take into account the differences of vascular extraction. However, this method presents many parameters which have to be fixed empirically (i.e. selectivity parameters and weight factors) and the best matching is only specific for this constraint set. If the constraints are too restrictive, some correct node assignments can be missed. Conversely, if the constraints are too extensive, the number of wrong node assignments might increase.

3.1.3. ADAPTATION TO OUR CORONARY MATCHING PROBLEM

Several modifications have been applied to adapt the two reference described algorithms [LKZP07, MKSZ09] to our coronary tree matching problem. The first concerns the new specific node-to-node metrics and associated thresholds used in the two algorithms. The second is to add a new process to take into account the topology variation between vascular trees by adding artificial nodes which correspond to bifurcation or terminal nodes absent in one tree and present in another. Moreover, with the artificial nodes insertion, several hierarchical conflicts can appear. We propose to check and correct the matching result at different steps: firstly from the maximum clique result and secondly when artificial nodes are inserted. The matching result is corrected and also the two input trees T_i and T_{i+1} .

3.1.3.1. SIMILARITY MEASURE CHOICE

Consistency constraints CC and unary constraints CU have the same aim in the two algorithms [LKZP07, MKSZ09, TMPH05]. These node-to-node constraints are used to reduce the hypothesis number in the search tree $TS_{i,i+1}$ or the node number (i.e. the promising assignments) in the association graph $G_{i,i+1}$. As a reminder, we wish for a fast execution time for this step and a substantial decrease of potential assignment number. Therefore, we used the spatial coordinates of node as similarity measure in order to determine a region of interest around the considered node (Figure 28-A)). The Euclidean distance d_e between two node coordinates in different trees can be used since coronary trees are extracted from the same acquisition of the same patient: trees are in the same coordinate space. Moreover, between two coronary trees which represent two successive phases of cardiac cycle, the movement can be considered low.

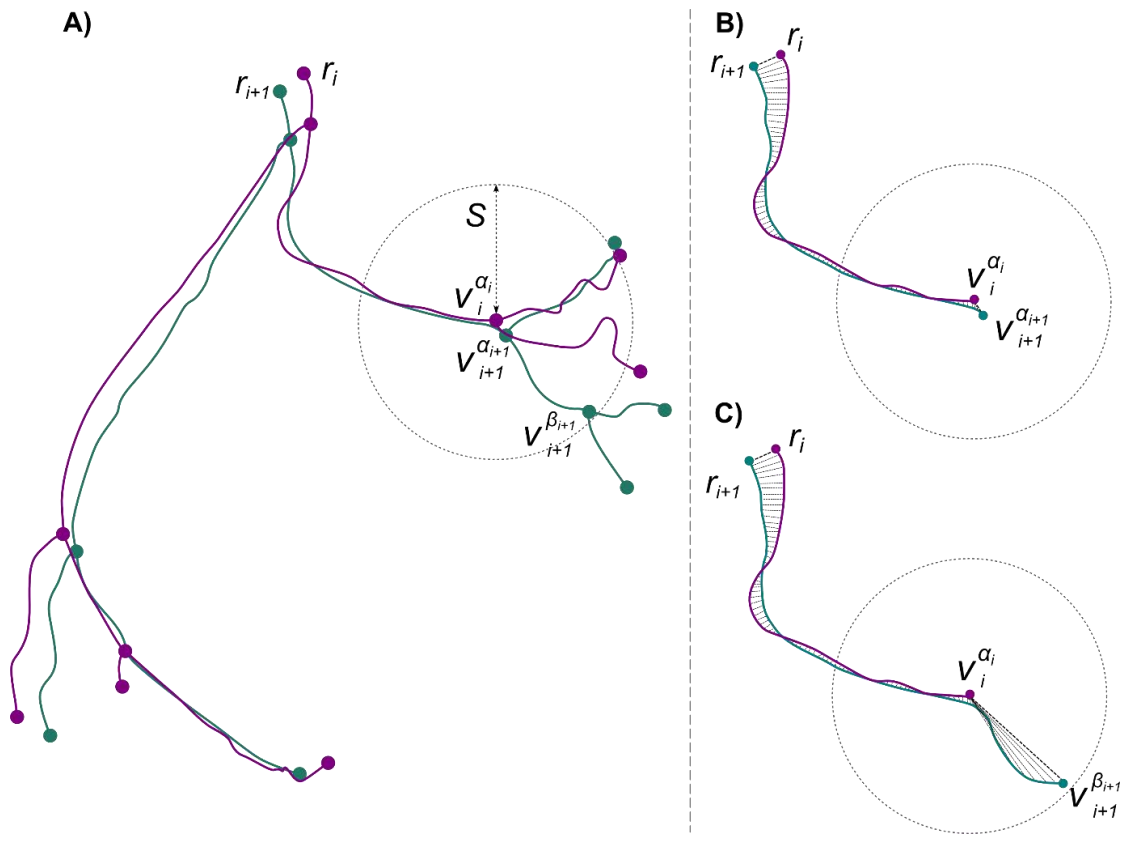


Figure 28 : Representation of the similarity measures used to match two nearby coronary trees. A) First selection: the current node $v_i^{\alpha_i}$ can be assigned with the nodes $v_{i+1}^{\alpha_{i+1}}$ and $v_{i+1}^{\beta_{i+1}}$. S represents the diameter selected to determine the region of interest (threshold value associated to the constraint). B) Application of the DTW's algorithm between two paths to evaluate the similarity between the two nodes $v_i^{\alpha_i}$ and $v_{i+1}^{\alpha_{i+1}}$ and for C) respectively for the two nodes $v_i^{\alpha_i}$ and $v_{i+1}^{\beta_{i+1}}$. The green path is longer than the current purple path: the result of their matching will be penalized.

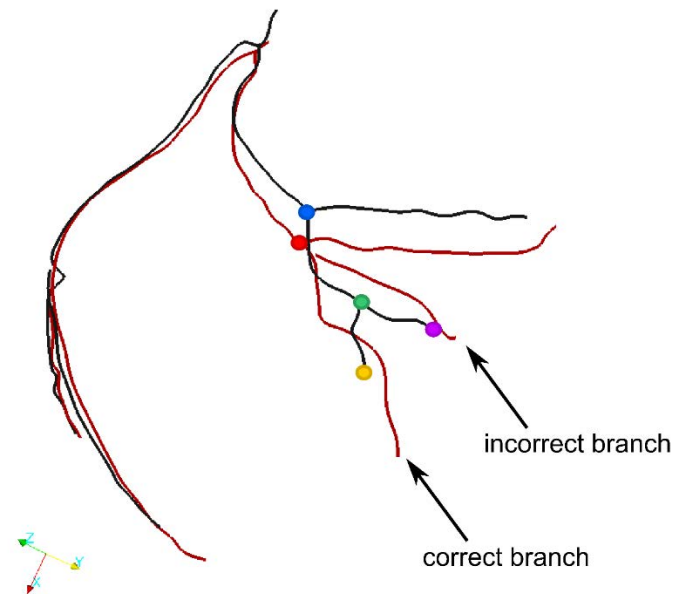


Figure 29: Example of wrong branch assignment by the computation of the Euclidean distance of only node coordinate by DTW. The dark branch between the blue node and the purple node is matched with the incorrect red branch. Normally, this is the dark branch between the blue node and the yellow node which must be matched with the correct red branch.

Similarity measures SM or binary constraints CB influence significantly the pair-wise matching results. The choice of these node-to-node metrics is important. They must be the most efficient as possible with a correct execution time. Geometrical features based on the tree centerlines are used without computing their normalized polylines.

We proposed other metrics to determine the similarity of the two paths $p_i(v_i^{\alpha_i}, v_i^{\beta_i}) \in T_i$ and $p_{i+1}(v_{i+1}^{\alpha_{i+1}}, v_{i+1}^{\beta_{i+1}}) \in T_{i+1}$ based on the distance computation thanks to Dynamic Time Warping's algorithm (DTW) [KePa01, KeRa05, RaKe05, SaCh07, SaCh78] (Figure 28). These three new node-to-node metrics (coordinates, tangent and direction) allow taking into account non-rigid movements of the coronary arteries (Figure 29). Each path is considered as a sequence of points. Let be $p_i(v_i^{\alpha_i}, v_i^{\beta_i}) = \{m_i^1, \dots, m_i^{\rho_i}, \dots, m_i^{\mu_i}\}$ and $p_{i+1}(v_{i+1}^{\alpha_{i+1}}, v_{i+1}^{\beta_{i+1}}) = \{m_{i+1}^1, \dots, m_{i+1}^{\rho_{i+1}}, \dots, m_{i+1}^{\mu_{i+1}}\}$ the point list of the two paths (centerlines), with respectively $m_i^1 = v_i^{\alpha_i}$ and $m_{i+1}^1 = v_{i+1}^{\alpha_{i+1}}$ the source nodes of the paths, and $m_i^{\mu_i} = v_i^{\beta_i}$ and $m_{i+1}^{\mu_{i+1}} = v_{i+1}^{\beta_{i+1}}$ the target nodes of the paths. μ_i and μ_{i+1} represent respectively the number of points of the paths p_i and p_{i+1} . Three attributes were associated to each point $m_i^{\rho_i}$ (and respectively $m_{i+1}^{\rho_{i+1}}$) of the two paths: the spatial coordinate $\overrightarrow{sc}_i^{\rho_i}$, the tangent $\overrightarrow{tg}_i^{\rho_i}$, and the direction $\overrightarrow{d}_i^{\rho_i}$ (and respectively $\overrightarrow{sc}_{i+1}^{\rho_{i+1}}$, $\overrightarrow{tg}_{i+1}^{\rho_{i+1}}$ and $\overrightarrow{d}_{i+1}^{\rho_{i+1}}$).

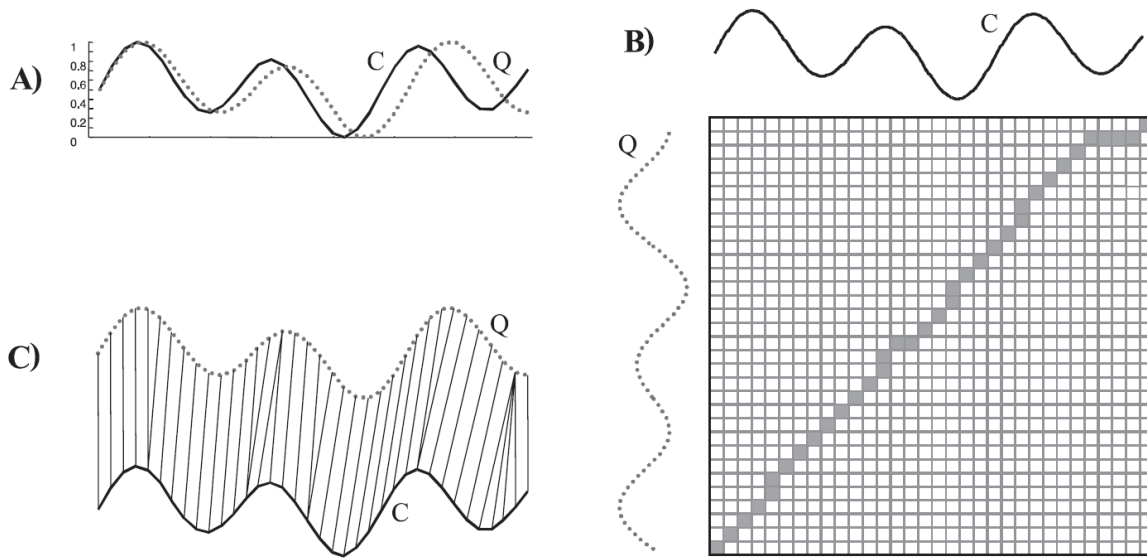


Figure 30: Illustration of the Dynamic Time Warping's algorithm from [KeRa05] (Fig. 3). A) The two sequences C and Q are similar but out of phase. B) Distance matrix between the two sequences. The optimal warping path (point by point optimal matching) is shown in solid squares. C) Alignment result from the distance matrix.

A multi parametric distance matrix D is build. An example of distance matrix between two similar sequences is shown in the Figure 30. For each element $D[m_i^{\rho_i}, m_{i+1}^{\rho_{i+1}}]$ of D , three node-to-node

criteria are computed based on the three point attributes and next cumulated with the minimum results of its neighbors (see Equation (3)). We choose to use the step patterns described in the Figure 31 and the equation (3). The three criteria, from which is computed the Euclidean distance by the DTW algorithm, correspond to the Euclidean distance $dist(\overrightarrow{sc_i^{\rho_i}}, \overrightarrow{sc_{i+1}^{\rho_{i+1}}})$, between two point coordinates $m_i^{\rho_i}$ and $m_{i+1}^{\rho_{i+1}}$ of the two paths, the angle between the two tangents $angle(\overrightarrow{tg_i^{\rho_i}}, \overrightarrow{tg_{i+1}^{\rho_{i+1}}})$ and the angle between the two directions $angle(\overrightarrow{d_i^{\rho_i}}, \overrightarrow{d_{i+1}^{\rho_{i+1}}})$.

$$D[m_i^{\rho_i}, m_{i+1}^{\rho_{i+1}}] = \left\{ dist(\overrightarrow{sc_i^{\rho_i}}, \overrightarrow{sc_{i+1}^{\rho_{i+1}}}), angle(\overrightarrow{tg_i^{\rho_i}}, \overrightarrow{tg_{i+1}^{\rho_{i+1}}}), angle(\overrightarrow{d_i^{\rho_i}}, \overrightarrow{d_{i+1}^{\rho_{i+1}}}) \right\} + \min(D[m_i^{\rho_i-1}, m_{i+1}^{\rho_{i+1}}], D[m_i^{\rho_i}, m_{i+1}^{\rho_{i+1}-1}], D[m_i^{\rho_i-1}, m_{i+1}^{\rho_{i+1}-1}]) \quad (3)$$

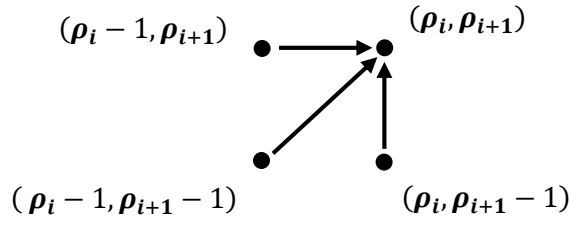


Figure 31: Step pattern used by the Dynamic Time Warping's algorithm to compute the distance matrix.

The distance matrix D is entirely built: all points of the two paths are considered. We don't use any global path constraint (such as the Sakoe-Chiba band or the Itakura Parallelogram [KePa01, KeRa05, Müll07, SaCh78]) to reduce the search path. Moreover, in the case of binary constraints or similarity measure, the boundary conditions of the Dynamic Time Warping is taken into account. This condition forces the match between the first points of the two paths and also the last points of the paths (in the Figure 30, the warping path begins and finishes in diagonally opposite corner cells of the matrix). Consequently, the algorithm is not stopped when the last point of the shortest sequence is reached. If the algorithm stops at this moment, the matching result will be skewed. Indeed, if the two paths have a similar curvature and spatial coordinates but a large difference path length, they can be completely matched when they must be only partially matched. So, we take into account the difference between path lengths by penalizing the result: the DTW result $D[m_i^{\rho_i}, m_{i+1}^{\rho_{i+1}}]$ is normalized by the minimum length between these two paths (Figure 28). Kaftan et al. [KKN06] used a similar constraint (point by point feature for the comparison of paths): the sum of the squared distance

between each point of the first path p_i and its closest point in the second path p_{i+1} is computed. Then, to normalize this metric, this sum is divided by the number of points μ_i of the first path p_i .

3.1.3.2. ARTIFICIAL NODE INSERTION IN CORONARY TREES

Using the two state-of-the-art published algorithms (sections 3.1.1 and 3.1.2) and thanks to measures adapted to coronary trees, the best matching result $M_{i,i+1}$ is only composed by pair set of tree nodes which are very similar based on the choice of constraints and their measure threshold value. The matching accuracy is more important than the matched node number. Increasing their number can increase wrong assignments which might decrease the matching quality and consequently also affect the quality of various next processes from these matching results (such as the matching of the whole tree sequence and next the determination of the optimal view over cardiac cycle). As a result, the number of matched nodes is small and covers only a small subset of tree nodes. Some bifurcations and leaf nodes are not matched because of the extraction variations.

Our goal is to find the optimal matching $M_{i,i+1}^*$ of similar branches minimizing wrong assignments to determine more precisely the area of the lesion and to facilitate the cardiac catheterizations through branches. That is why artificial nodes a_{i+1}^ϕ corresponding to existing points of branches in T_i (bifurcation or terminal nodes) are added in the other tree T_{i+1} to improve this coronary matching. The two input trees of the sequence T_i and T_{i+1} become more complete after this step and shared their additional information. In result, they have more nodes with these artificial nodes.

The Algorithm 4 shows the different steps for the artificial node insertion. We try to associate previously unmatched nodes corresponding to the start of spurious branches in the reference tree T_i and unmatched leaf nodes with an artificial node a_{i+1}^ϕ which corresponds to a point of one centerline in the tree T_{i+1} (function *artificialNodesInsertion()* in lines 1 and 4 of the Algorithm 4). For each unmatched node v_i^ϕ in the reference tree T_i , the algorithm begins by searching its nearest matched parent node $mp(v_i^\phi)$ and its nearest matched children node $mc(v_i^\phi)$ (respectively functions *nearestMatchedParentPairNode()* and *nearestMatchedChildrenPairNode()* in lines 6 and 7 of the Algorithm 4). Let be respectively $mp(a_{i+1}^\phi)$ and $mc(a_{i+1}^\phi)$ their matching node in the other tree T_{i+1} . Thus, the path $p_i(mp(v_i^\phi), v_i^\phi)$ is deduced in the tree T_i . We looked for a similar path p_{i+1} to p_i in

the other tree T_{i+1} to find if the centerline point corresponding to the artificial node a_{i+1}^ϕ exists (Figure 32).

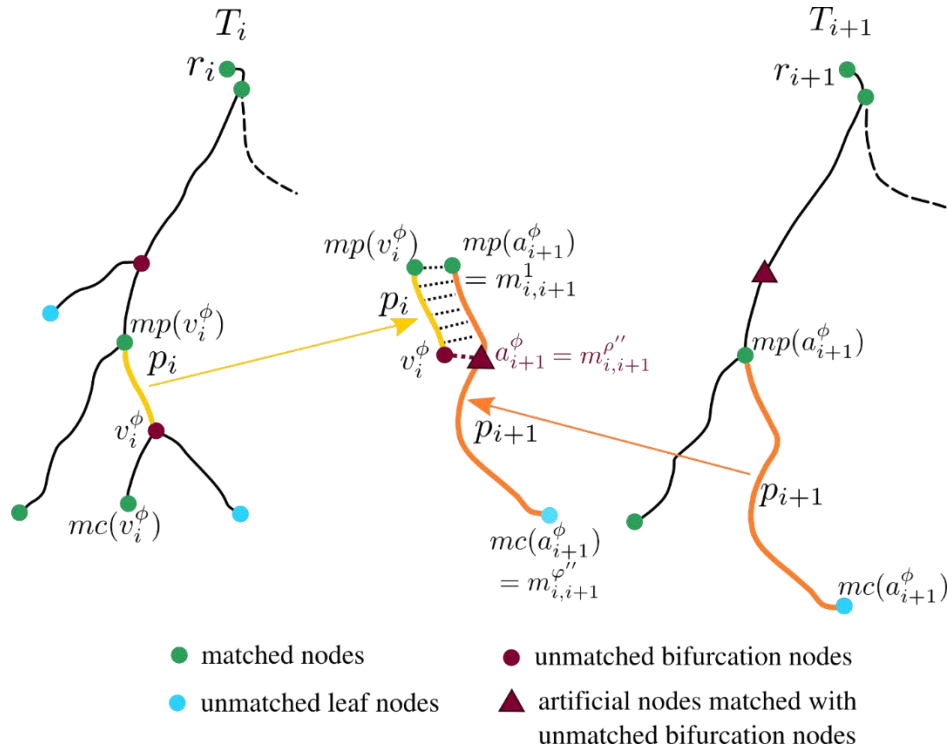


Figure 32: Illustration of artificial node insertion a_{i+1}^ϕ associated to an unmatched bifurcation node v_i^ϕ .

Two possible cases must be considered. In the first scenario, the matched children pair $(mc(v_i^\phi), mc(a_{i+1}^\phi))$ exists (condition block beginning at line 12 of the Algorithm 4). The branch having the point which corresponds to the artificial node a_{i+1}^ϕ is found. This branch corresponds to the path p_{i+1} with $mp(a_{i+1}^\phi)$ as source node and $mc(a_{i+1}^\phi)$ as target node.

In the second scenario, the matched children pair $(mc(v_i^\phi), mc(a_{i+1}^\phi))$ doesn't exist (condition block beginning at line 15 of the Algorithm 4). For example, it's the case for all unmatched leaf nodes. We have to search the similar path to p_i among all paths beginning by the node $mp(a_{i+1}^\phi)$ in T_{i+1} (line 16 of the Algorithm 4). There are as many leaf nodes l_{i+1}^ψ from the subtree induced by the node $mp(a_{i+1}^\phi)$ as possible paths. To determine the similar path p_{i+1} , the DTW's algorithm coupled with the three new node-to-node metrics is applied between p_i and each ψ possible paths (line 18 of the Algorithm 4). The same measures described in the previous section 3.1.3 are used: the node coordinates, the local tangent and the local direction. However, contrarily to the binary constraint or

similarity measure case, the boundary constraint is not used. Indeed, the difference of path lengths are taken into account. The algorithm is stopped when the last point in p_i , corresponding to the unmatched node v_i^ϕ , is reached. Only the first points of the two paths p_i and p_{i+1} are forced to match (and not the last points). Another modification of the Dynamic Time Warping concerns the second path p_{i+1} . We search a similar path to p_i to determine if an artificial node can be inserted. So, a translation is performed on all second path points to superimpose the first point of the two paths p_i and p_{i+1} (translation equal to the vector between $mp(v_i^\phi)$ and $mp(a_{i+1}^\phi)$). The similar path p_{i+1} is found if next two conditions are respected:

- its final minimum accumulated distance must be smaller than the DTW's results of the three criteria for the other possible paths (condition block in line 19 of the Algorithm 4),
- its DTW's result of the three criteria divided by the length of the path p_i must be smaller than an associated threshold which determines the path similarity (condition block line 25 of the Algorithm 4).

If these two conditions are not validated, there are none similar path to p_i in the tree T_{i+1} and no artificial node are inserted to be matched with v_i^ϕ (line 26 of the Algorithm 4).

For the two scenarii, we have determined the similar bit of the path p_{i+1} to p_i and also the point index ρ_{i+1} in this path p_{i+1} corresponding to the unmatched node v_i^ϕ thanks to the DTW's algorithm (Figure 32). This point $m_{i+1}^{\rho_{i+1}} \in p_{i+1}$ is inserted as an artificial node a_{i+1}^ϕ in the tree T_{i+1} , and the tree structure is updated, e.g. an edge is splitted (function *addArtificialNode()* in the line 29 of the Algorithm 4). The artificial node a_{i+1}^ϕ is matched with the bifurcation node v_i^ϕ . This pair (v_i^ϕ, a_{i+1}^ϕ) is inserted in the best matching $M_{i,i+1}^*$ (function *addMatchingPairNode()* in the line 30 of the Algorithm 4).

A last checking and correction step is performed (function *verificationHierarchy()* in line 31 of the Algorithm 4). With the new node pair inserted in $M_{i,i+1}^*$, hierarchical conflicts can arise. It can be the case with two near bifurcations for example. We detect conflicts cause by a matched node pair in $M_{i,i+1}^*$ and search if the matching result $M_{i,i+1}^*$ can be modified to solve it. If better assignment is found thanks to the Dynamic Time Warping, the matching result $M_{i,i+1}^*$ is corrected and the two trees T_i and T_{i+1} updated. If no correction is found, the matched node pair which causes the conflict is erased of the matched result $M_{i,i+1}^*$. If this matched node pair has an artificial node, this node is deleting in its corresponding tree.

For these two trees, the same step will be performing with the other tree T_{i+1} as reference (function *artificialNodesInsertion()* in line 2 of the Algorithm 4).

Algorithm 4: Artificial node insertion.

Input: two coronary trees T_i and T_{i+1} at successive phases t_i and t_{i+1} and the best matching $M_{i,i+1}$

Output: optimal matching $M^*_{i,i+1}$ and the modified and corrected trees T_i and T_{i+1} .

```
1:  ( $M^*_{i,i+1}, T_i, T_{i+1}$ ) = artificialNodesInsertion ( $T_i, T_{i+1}, M_{i,i+1}$ )
2:  ( $M^*_{i,i+1}, T_{i+1}, T_i$ ) = artificialNodesInsertion ( $T_{i+1}, T_i, M^*_{i,i+1}$ )
3:  return ( $M^*_{i,i+1}, T_i, T_{i+1}$ )

4:  function artificialNodesInsertion ( $T_i, T_{i+1}, M_{i,i+1}$ )
5:    for each unmatched bifurcation and leaf node  $v_i^\phi$  in  $T_i$  do
6:      ( $mp(v_i^\phi), mp(a_{i+1}^\phi)$ ) = nearestMatchedParentPairNode( $v_i^\phi$ )
7:      ( $mc(v_i^\phi), mc(a_{i+1}^\phi)$ ) = nearestMatchedChildrenPairNode( $v_i^\phi$ )
8:       $p_i$  = buildPath( $mp(v_i^\phi), v_i^\phi, T_i$ )
9:       $p_{i+1}$  = null
10:      $min\_dtw\_distance$  =  $+\infty$ 
11:      $\rho''$  = 0
12:     if the children node  $mc(a_{i+1}^\phi)$  exists then
13:        $p_{i+1}$  = buildPath( $mp(a_{i+1}^\phi), mc(a_{i+1}^\phi), T_{i+1}$ )
14:       ( $min\_dtw\_distance, \rho_{i+1}$ ) = DTW ( $p_i, p_{i+1}$ )
15:     else
16:       for each leaf node  $l_{i+1}^\psi$  of the sub-tree induced by the node  $mp(a_{i+1}^\phi)$  do
17:          $path$  = buildPath( $mp(a_{i+1}^\phi), l_{i+1}^\psi$ )
18:         ( $dtw\_distance, index$ ) = DTW ( $p_i, path$ )
19:         if  $dtw\_distance \leq min\_dtw\_distance$  then
20:            $min\_dtw\_distance$  =  $dtw\_distance$ 
21:            $p_{i+1}$  =  $path$ 
22:            $\rho_{i+1}$  =  $index$ 
23:         end if
24:       end for
```

```

25:         if (min_dtw_distance/ length(pi)) ≤ threshold then
26:             exit
27:         end if
28:     end if
29:     (aφi+1, Ti+1) = addArtificialNode(pi+1, ρi+1, Ti+1)
30:     Mi,i+1 = addMatchingPairNode (vφi, aφi+1, Mi,i+1)
31:     (Mi,i+1, Ti, Ti+1) = verificationHierarchy(Mi,i+1, Ti, Ti+1)
32: end for
33: return (Mi,i+1, Ti, Ti+1)
34: end function

```

3.2. MATCHING OF THE CORONARY TREE SEQUENCE

To deal with our final problem, the estimation of the optimal view angle for coronary treatment, the matching must process the whole temporal tree sequence. Moreover, this mosaic-based matching algorithm must respect some constraints due to artificial node presence. Indeed, coronary trees of the same sequence are non-isomorph and the insertion of artificial nodes in trees allows solving some problems such as the bifurcation missing, branch missing and the difference terminal branch lengths. These nodes have to be labeled in each tree of the sequence and not only for consecutive trees. They complete and correct the tree sequence thanks to mosaic tree building.

In this section, a global matching algorithm of the whole coronary tree sequence is proposed based on the complete mosaic tree building (Figure 33). Image mosaicking is an important and classical problem in the fields of medical image processing [AMFE07, CCMM06, CSRT02]. This approach is used to merge a set of images registered as precisely as possible into a single composite image as seamlessly as possible. In retinal application, an image sequence with different partial view angles of the same patient can be registered to construct a mosaic image giving a larger view of the retina. In our paper, the mosaic tree represents the merging of successive pair-wise union trees. The first step corresponds to our previous pair-wise tree matching method which matches only two successive trees. This proposed mosaic-based matching is more complete and accurate than the pair-wise algorithm and takes into account the topology variation of all trees in the whole sequence. Moreover, the algorithm

checks the matching exactness and corrects potential matching errors thanks to the additional information gradually obtained during the mosaic tree building process.

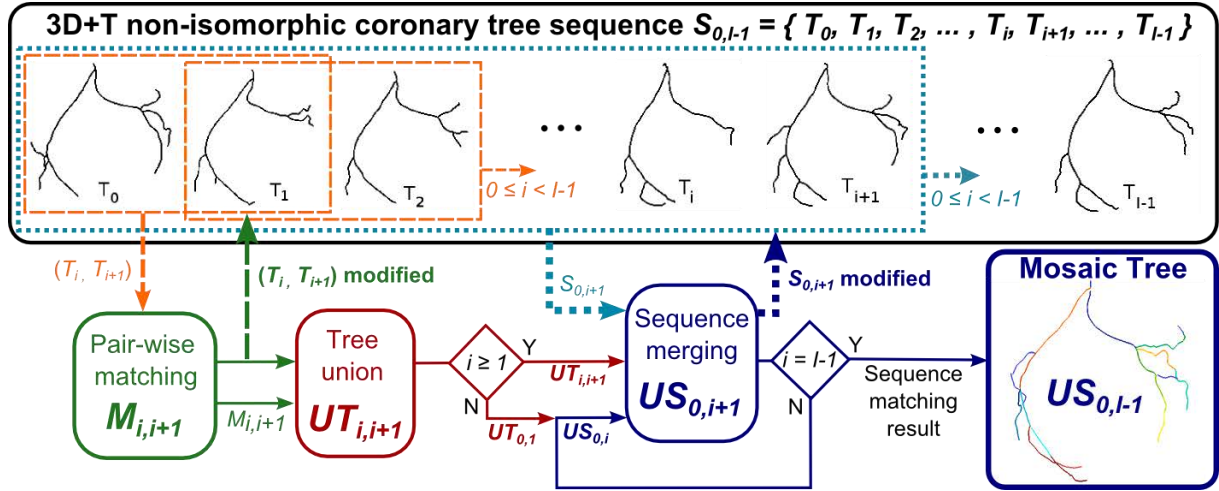


Figure 33: Global mosaic-based matching algorithm for a 3D+T coronary tree sequence.

In this section, we begin to explain the global mosaic-based method. Next, the subsection 3.2.3 presents the pair-wise union tree building $UT_{i,i+1}$ between two successive trees T_i and T_{i+1} and subsection 3.2.4 the mosaic tree building $US_{0,l-1}$.

3.2.1. MOSAIC-BASED MATCHING OF CORONARY TREE SEQUENCE

The pair-wise matching algorithm of the two successive trees T_i and T_{i+1} presented in the previous section 1 with the artificial node insertion and correction step becomes a preliminary step (function *MatchingTwoTrees()* of the Algorithm 5). Consequently, the two trees T_i and T_{i+1} of the sequence $S_{0,l-1}$ are completed with artificial nodes and corrected (Figure 34-A)).

With the corrected pair-wise matching result $M_{i,i+1}$ containing artificial nodes, a union tree $UT_{i,i+1}$ is built from the two modified and matched successive trees T_i and T_{i+1} (Equation (4) and Figure 34-B)).

$$UT_{i,i+1} = \mathbf{BuildUnionTwoTrees}(T_i, T_{i+1}, M_{i,i+1}) \quad (4)$$

In this tree union, a label is defined for each union node: either a union node corresponding to a matched node pair from $M_{i,i+1}$ or a union node representing a single node of T_i or T_{i+1} which have no correspondence due to the non-isomorphic nature of trees (different branch lengths, bifurcation absence, etc.) (line 3 of the **Algorithm 5** and function *BuildUnionTree()* in the **Algorithm 6**).

After that, a mosaic tree $US_{0,i+1}$ is built (equation (5)) until the sequence end and the obtaining of the global mosaic tree $US_{0,I-1}$ which represents the complete matched tree of the whole tree sequence $S_{0,I-1} = \{T_0, \dots, T_i, \dots, T_{I-1}\}$ (see **Algorithm 5** from line 4 to line 8 with the function *ComputeMosaicTree()*). The mosaic tree $US_{0,i+1}$ represents the merging of the previous mosaic tree $US_{0,i}$ and the union tree $UT_{i,i+1}$ (equation (5) and Figure 34-C).

$$\begin{aligned} (US_{0,i+1}, S_{0,i+1}) &= \text{ComputeMosaicTree}(US_{0,i}, UT_{i,i+1}) \\ &\text{with } US_{0,1} = UT_{0,1} \text{ and } i \in [0, I - 1] \end{aligned} \quad (5)$$

Firstly, the mosaic tree $US_{0,i+1}$ is initialized by the previous computed mosaic tree $US_{0,i}$. Secondly, all nodes in $UT_{i,i+1}$ which are not present in the mosaic tree $US_{0,i+1}$ (i.e. nodes of the tree T_{i+1}) are added in $US_{0,i+1}$. Some merging nodes of $US_{0,i+1}$ which represent the same matched tree node in different cardiac phases are merged. For other nodes, to complete the global matching, artificial nodes are searched in each phase of $S_{0,i+1}$. Moreover, at each step, the matching rightness are checked during the tree merging of $US_{0,i}$ and $UT_{i,i+1}$ (merging result corresponds to $US_{0,i+1}$) and corrected if some errors appears with the additional information contained in T_{i+1} such as hierarchical errors or artificial node position errors. Tree sequence $S_{0,i+1}$ are consequently corrected and completed by artificial nodes with these iterative steps during the building of the mosaic tree $US_{0,i+1}$.

For a pair-wise matching result $M_{i,i+1}$ between two trees T_i and T_{i+1} , the tree T_i has been completed with artificial nodes obtained from previous pair-wise matching which correspond to additional information present in the previous processed trees of the sequence $S_{0,i}$. Therefore, the matching result $M_{i,i+1}$ is gradually more robust with the sequence matching progress since T_i is completed from $US_{0,i}$. Indeed, the node number of the last modified tree T_{I-1} has doubled compared to the node number of the original tree T_{I-1} , as for the other trees of the sequence $S_{0,I-1}$.

Algorithm 5: Mosaic-based matching algorithm of the whole tree sequence (equation (4) and (5))

Input : The sequence of coronary trees $S_{0,I-1} = \{T_0, \dots, T_i, \dots, T_{I-1}\}$

Output : The final mosaic tree $US_{0,I-1}$ with the corrected tree sequence $S_{0,I-1}$

```

1:  for each tree  $T_i$  with  $0 \leq i < I - 1$  do
2:       $(M_{i,i+1}, T_i, T_{i+1}) = \text{MatchingTwoTrees}(T_i, T_{i+1})$  //See section 3.1.3
3:       $UT_{i,i+1} = \text{BuildUnionTwoTrees}(T_i, T_{i+1}, M_{i,i+1})$  //See Algorithm 6 (section 3.2.3)
4:      if  $i == 1$  then //See Algorithm 7 (section 3.2.4)
5:           $(US_{0,2}, S_{0,2}) = \text{ComputeMosaicTree}(UT_{0,1}, UT_{1,2})$ 
6:      else if  $i \geq 2$  then //See Algorithm 7 (section 3.2.4)
7:           $(US_{0,i+1}, S_{0,i+1}) = \text{ComputeMosaicTree}(US_{0,i}, UT_{i,i+1})$ 
8:      end if
9:  end for each
10: return  $US_{0,I-1}$ 

```

In the next section, we explain the structures of mosaic tree $US_{0,i+1}$ and union tree $UT_{i,i+1}$ in particular their node characteristics. Then, we describe more precisely the union process of two successive trees $UT_{i,i+1}$ based on their corrected matching result $M_{i,i+1}$. We finish by the description of the complete matching algorithm of the whole vascular tree sequence $S_{0,I-1}$ based on the global mosaic tree building $US_{0,I-1}$ from unions of previously matched trees and the different matching checks and associated corrections.

3.2.2. CHARACTERISTICS OF MERGING AND UNION NODES

First, some characteristics of merging nodes $V_{US_{0,i+1}}$ of the mosaic tree $US_{0,i+1}$ and respectively the union nodes $V_{UT_{i,i+1}}$ of $UT_{i,i+1}$ are described to understand how the complete matching of $S_{0,I-1}$ is realized. Let be $V_{US_{0,i+1}} = \{v_{US_{0,i+1}}^1, \dots, v_{US_{0,i+1}}^\alpha, \dots, v_{US_{0,i+1}}^{n_{US_{0,i+1}}}\}$ the merging node set of the mosaic tree $US_{0,i+1}$ with $n_{US_{0,i+1}}$ their node number and $V_{UT_{i,i+1}} = \{v_{UT_{i,i+1}}^1, \dots, v_{UT_{i,i+1}}^\alpha, \dots, v_{UT_{i,i+1}}^{n_{UT_{i,i+1}}}\}$ the union node set of the union tree $UT_{i,i+1}$ from pair-wise matching $M_{i,i+1}$. Each merging node $v_{US_{0,i+1}}^\alpha$ has a list $L(v_{US_{0,i+1}}^\alpha) = \{v_0^{\alpha_0}, \dots, v_i^{\alpha_i}, v_{i+1}^{\alpha_{i+1}}\}$ of matched

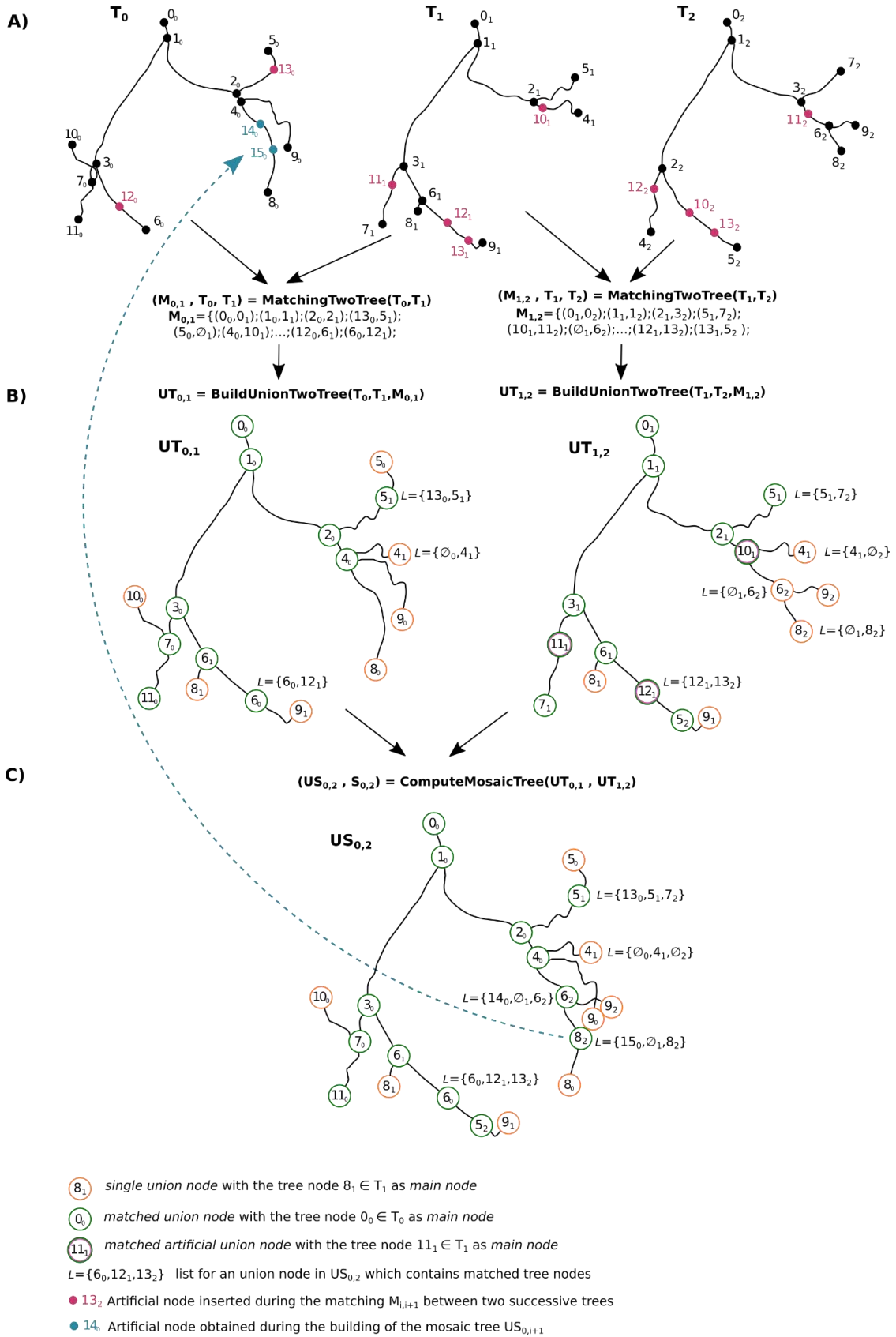


Figure 34: Example of a built mosaic tree from the tree sequence $S_{0,2}$. A) Tree sequence $S_{0,2} = \{T_0, T_1, T_2\}$ with artificial nodes (pink and blue) inserted during the matching process. B) Building of the two union trees $UT_{0,1}$ and $UT_{1,2}$ from respectively the two trees T_0, T_1 and T_1, T_2 . C) Built mosaic tree $US_{0,2}$ from the sequence $S_{0,2}$ derived from $UT_{0,1}$ and $UT_{1,2}$.

tree nodes. This node list contains at least one to ideally $i + 1$ matched nodes (so only one or none matched node of each tree T_i of the sequence $S_{0,i+1}$). Each matched node in this list corresponds to the same node at different cardiac phases. Each union node $v_{UT_{i,i+1}}^\alpha$ has a list $L(v_{UT_{i,i+1}}^\alpha) = \{v_i^{\alpha_i}, v_{i+1}^{\alpha_{i+1}}\}$ of one single node or ideally two matched tree nodes in each tree T_i and T_{i+1} .

This matched node list allows determining some merging and union node characteristics (Figure 34):

- A single merging node (and respectively a single union node for $UT_{i,i+1}$) is a merging node $v_{US_{0,i+1}}^\alpha$ which have only one node $v_i^{\alpha_i} \in T_i$ in the list $L(v_{US_{0,i+1}}^\alpha) = \{\emptyset_0, \dots, \emptyset_{i-1}, v_i^{\alpha_i}, \emptyset_{i+1}\}$. A single node corresponds to a node which hasn't been matched with another node among an intermediary tree sequence $S_{0,i+1}$. Single nodes are represented with an orange circle in the Figure 34.
- A matched merging node (and respectively a matched union node for $UT_{i,i+1}$) is a merging node $v_{US_{0,i+1}}^\alpha$ which have at least two nodes in the list $L(v_{US_{0,i+1}}^\alpha)$. Nodes in this list $L(v_{US_{0,i+1}}^\alpha)$ represent the same matched node among the tree sequence $S_{0,i+1}$. Matched nodes are represented with a green circle in the Figure 34.
- An artificial union node in $UT_{i,i+1}$ is a single union node or a matched union node which have in the list only artificial nodes. A single merging node or a matched merging node of $US_{0,i+1}$ cannot have only artificial nodes. Artificial nodes are represented with a pink circle and green circle in the Figure 34.
- A main node of a merging node $v_{US_{0,i+1}}^\alpha$ and respectively of a union node $v_{UT_{i,i+1}}^\alpha$ represents a node in the list $L(v_{US_{0,i+1}}^\alpha) = \{v_0^{\alpha_0}, \dots, v_i^{\alpha_i}, v_{i+1}^{\alpha_{i+1}}\}$. This main node corresponds to the first found non-artificial node in the phase order of the sequence $S_{0,i+1}$. In the case of an artificial union node $v_{UT_{i,i+1}}^\alpha$ of a union tree $UT_{i,i+1}$, the main node must be the first artificial node.

3.2.3. BUILDING OF THE UNION TREE $UT_{i,i+1}$ FROM THE MATCHING RESULT $M_{i,i+1}$ BETWEEN T_i AND T_{i+1}

As explained in the **Algorithm 5**, when the matching result $M_{i,i+1}$ between T_i and T_{i+1} is obtained, the temporary union tree $UT_{i,i+1}$ is built (Figure 34-B)). This union process is described in the **Algorithm 6**. The function *BuildUnionOfTwoTrees()* begins with the intersection of the two trees T_i and T_{i+1} . Union nodes corresponding to the previously matched node pair in $M_{i,i+1}$ are added in $UT_{i,i+1}$ (block line 1 to 9 in the **Algorithm 6** with the function *insertUnionNode()*). In this step, each

union node $v_{UT_{i,i+1}}^\alpha$ have in its list $L(v_{UT_{i,i+1}}^\alpha) = \{v_i^{\alpha_i}, v_{i+1}^{\alpha_{i+1}}\}$ the matched node pair $(v_i^{\alpha_i}, v_{i+1}^{\alpha_{i+1}}) \in M_{i,i+1}$ between T_i and T_{i+1} . Moreover, the label of each node e.g. *matched union node* and *artificial union node* (at this step there is no *single union node*) and the *main node* are updated. Next, the other nodes of T_i and T_{i+1} which have no node correspondence are inserted in the union tree as *single union node* (block line 10 to 19 in the **Algorithm 6** with the function *insertUnionNodeAsSingleNode()*).

When all nodes of the trees T_i and T_{i+1} are added in the temporary union tree $UT_{i,i+1}$, edges are inserted (block line 20 to 27 in the **Algorithm 6** with the function *insertEdge()*). Each edge corresponds to the centerline data between two nodes. We have to determine firstly which union nodes of $UT_{i,i+1}$ have a father-son connection and secondly in which tree (T_i or T_{i+1}) is the centerline data between these two linked union nodes. For each union node pair $(v_{UT_{i,i+1}}^\alpha, v_{UT_{i,i+1}}^\beta)$ with $\alpha \neq \beta$ and α and $\beta \in n_{UT_{i,i+1}}$, the edge insertion between them depends on the *main node* of the target node $v_{UT_{i,i+1}}^\beta$. For example, we consider that its *main node* $v_i^{\beta_i}$ is in the tree T_i . We search if the union node $v_{UT_{i,i+1}}^\alpha$ (representing the edge source node) has in its list $L(v_{UT_{i,i+1}}^\alpha)$ a tree node corresponding to the father of $v_i^{\beta_i}$ in the same tree T_i . If the father of $v_i^{\beta_i}$ is found, an edge in the union tree $UT_{i,i+1}$ is inserted between $v_{UT_{i,i+1}}^\alpha$ and $v_{UT_{i,i+1}}^\beta$. This new edge has the same centerline data as the one of the tree T_i with the father of $v_i^{\beta_i}$ as source node and $v_i^{\beta_i}$ as target node.

Algorithm 6: function *BuildUnionOfTwoTrees()*

Input: T_i and T_{i+1} the two consecutive coronary trees to match, $M_{i,i+1}$ the matching result between these two trees.

Output: $UT_{i,i+1}$ the tree corresponding to the union between the two matched input trees T_i and T_{i+1} .

```

1:  function BUILDUNIONOFTWOTREES
2:      //First step: add in  $UT_{i,i+1}$  union node considered as matched union node
3:      for each pair of matched nodes  $(v_i^{\alpha_i}, v_{i+1}^{\alpha_{i+1}})$  in  $M_{i,i+1}$  do
4:          if  $v_i^{\alpha_i}$  is an artificial node and  $v_{i+1}^{\alpha_{i+1}}$  is not an artificial node then
5:              //  $v_{i+1}^{\alpha_{i+1}}$  become the main node of the union node newly inserted. A union node  $v_{UT_{i,i+1}}^\alpha$ 
              // with  $L(v_{UT_{i,i+1}}^\alpha) = \{v_i^{\alpha_i}, v_{i+1}^{\alpha_{i+1}}\}$  is created.

```

```

6:          $UT_{i,i+1} = insertUnionNode(v_{i+1}^{\alpha_{i+1}}, v_i^{\alpha_i})$ 
7:     else
8:         //  $v_i^{\alpha_i}$  is the main node.  $v_i^{\alpha_i}$  and  $v_{i+1}^{\alpha_{i+1}}$  can be artificial nodes.
9:          $UT_{i,i+1} = insertUnionNode(v_i^{\alpha_i}, v_{i+1}^{\alpha_{i+1}})$ 
10:    end if
11: end for each
12: //Second step: adding union node considered as single node in  $UT_{i,i+1}$ 
13: for each tree node  $v_i^{\alpha_i}$  of  $T_i$  do
14:     if  $v_i^{\alpha_i}$  is not found in the list  $L(v_{UT_{i,i+1}}^{\alpha})$  of all union nodes  $v_{UT_{i,i+1}}^{\alpha}$  already inserted then
15:          $UT_{i,i+1} = insertUnionNodeAsSingleNode(v_i^{\alpha_i}, \emptyset_{i+1})$ 
16:     end if
17: end for each
18: for each tree node  $v_{i+1}^{\alpha_{i+1}}$  of  $T_{i+1}$  do
19:     if  $v_{i+1}^{\alpha_{i+1}}$  is not found in the list  $L(v_{UT_{i,i+1}}^{\alpha})$  of all union node  $v_{UT_{i,i+1}}^{\alpha}$  already inserted then
20:          $UT_{i,i+1} = insertUnionNodeAsSingleNode(\emptyset_i, v_{i+1}^{\alpha_{i+1}})$ 
21:     end if
22: end for each
23: //Third step: adding edge between union nodes in  $UT_{i,i+1}$ 
24: for each union node  $v_{UT_{i,i+1}}^{\alpha}$  of  $UT_{i,i+1}$  do
25:     for each union node  $v_{UT_{i,i+1}}^{\beta}$  of  $UT_{i,i+1}$  do
26:         if  $\alpha \neq \beta$  and the main node of  $v_{UT_{i,i+1}}^{\alpha}$  is the children of a node in  $L(v_{UT_{i,i+1}}^{\beta})$  in the
same view then
27:             //same edge between the main node of  $v_{UT_{i,i+1}}^{\alpha}$  and its father in the union tree
 $UT_{i,i+1}$ .
28:              $UT_{i,i+1} = insertEdge(v_{UT_{i,i+1}}^{\beta}, v_{UT_{i,i+1}}^{\alpha})$ 
29:         end if
30:     end for each
31: end for each

```

```

32:   return  $UT_{i,i+1}$ 
33: end function

```

3.2.4. BUILDING OF THE MOSAIC TREE $US_{0,i+1}$ OF THE TREE SEQUENCE $S_{0,i+1}$

When the two first consecutive union trees $UT_{0,1}$ and $UT_{1,2}$ are built, the **Algorithm 5** calls the method *ComputeMosaicTree()* (line 5) described in the **Algorithm 7**. Firstly, this algorithm merges the two first union trees $UT_{0,1}$ and $UT_{1,2}$ to obtain the first temporary mosaic tree $US_{0,i+1}$ ($US_{0,2}$) of the sequence $S_{i+1} = \{T_0, T_i, T_{i+1}\}$ with $i = 1$ (Figure 34-C)). Secondly, the following calls of this function (line 7 of the **Algorithm 5**) merge the current mosaic tree $US_{0,i}$ and the temporary consecutive union tree $UT_{i,i+1}$ with $i \geq 2$. Now in the paper, we consider that we build the next mosaic tree $US_{0,i+1}$ from $US_{0,i}$ and $UT_{i,i+1}$ corresponding to the (5).

Algorithm 7: function *ComputeMosaicTree()*

Input: $UT_{i-1,i}$ for the first function call represents the temporary union tree of T_{i-1} (i.e. T_0) and T_i (i.e. T_1) or $US_{0,i}$ for the other function calls represents the mosaic tree (see Equation (5)) for the matched tree sequence $S_{0,i} = \{T_0, T_1, \dots, T_i\}$; $UT_{i,i+1}$ is the temporary union tree from pair-wise matching result $M_{i,i+1}$ between two consecutive phases T_i and T_{i+1} .

Output: $US_{0,i+1}$ the mosaic tree for the tree sequence $S_{i+1} = \{T_0, T_1, \dots, T_i, T_{i+1}\}$.

```

1:  function COMPUTEMOSAICTREE
2:       $US_{0,i+1} = \text{MERGINGNODES}(US_{0,i}, UT_{i,i+1})$ 
3:       $US_{0,i+1} = \text{MERGINGNODES}(UT_{i,i+1}, US_{0,i})$  // used for the single node.
4:       $US_{0,i+1} = \text{AddEdge}()$  //Insert edges between merged nodes in the mosaic tree
5:      for each union node  $v_{US_{0,i}}^\alpha$  not analyzed in  $US_{0,i}$  do
6:           $US_{0,i+1} = \text{addMergingNodeAndEdge}(v_{US_{0,i}}^\alpha)$  //add same merging node  $v_{US_{0,i}}^\alpha$  in
7:           $US_{0,i+1}$  (special cases already considered)
8:      end for each
9:      for each union node  $v_{UT_{i,i+1}}^\alpha$  not analyzed in  $UT_{i,i+1}$  do
10:          $US_{0,i+1} = \text{addMergingNodeAndEdge}(v_{UT_{i,i+1}}^\alpha)$ 

```

```

10:         isMerged = isSameNode ( $v_{UT_{i,i+1}}^\alpha$ ) //search if the node can be matched with another
node previously inserted in  $US_{0,i+1}$ 
11:         if isMerged == false then
12:             isSamePoint( $v_{UT_{i,i+1}}^\alpha$ ) //search if the node can be matched with a point of a
centerline in previous trees (artificial node can be inserted)
13:         end if
14:     end for each
15:     searchArtificialNodeLastPhase( $US_{0,i}, UT_{i,i+1}$ ) //Add artificial node in the last phase
 $T_{i+1}$  which can be matched with previous node in  $UT_{0,i+1}$ .
16:     return  $US_{0,i+1}$ 
17: end function
18:
19: function MERGINGNODES( $US_{0,i}, UT_{i,i+1}$ )
20:     for each node  $v_{US_{0,i}}^\alpha$  in  $US_{0,i}$  which are not analyzed do
21:         for each node  $v_{UT_{i,i+1}}^\alpha$  in  $UT_{i,i+1}$  which are not analyzed do
22:             isRelated = isRelatedNode ( $v_{US_{0,i}}^\alpha, v_{UT_{i,i+1}}^\alpha, i$ )
23:             if isRelated == true then
24:                 isArtificial = isArtificialNode ( $v_{UT_{i,i+1}}^\alpha$ )
25:                 isSimilarPath = true
26:                 if isArtificial == true and the  $v_{UT_{i,i+1}}^\alpha$  is not a single node then
27:                     isSimilarPath = isArtificialSimilarPath ( $v_{US_{0,i}}^\alpha, v_{UT_{i,i+1}}^\alpha, i$ )
28:                 end if
29:                 if isSimilarPath == true then
30:                      $US_{0,i+1} = \text{mergeNodes}(v_{US_{0,i}}^\alpha, v_{UT_{i,i+1}}^\alpha, i)$  //merge nodes  $v_{US_{0,i}}^\alpha$  and  $v_{UT_{i,i+1}}^\alpha$  in the
mosaic tree
31:                 end if
32:             end if
33:         end for each
34:     end for each
35:     return  $US_{0,i+1}$ 
36: end function

```

3.2.4.1. NODE MERGING FROM $US_{0,i}$ AND $UT_{i,i+1}$ RELATED BY THEIR COMMON PHASE i IN $US_{0,i+1}$

Firstly in the **Algorithm 7**, i.e. the building of the mosaic tree $US_{0,i+1}$, we search which union nodes can be merged (line 1, 2 and block function *NodeMerging ()* from line 16 to 32). A union node $v_{US_{0,i}}^\alpha$ of $US_{0,i}$ and another node $v_{UT_{i,i+1}}^\alpha$ of $UT_{i,i+1}$ can be merged if they are related in their common phase i , in other word, they must have in their node list L the same node $v_i^{\alpha_i}$ (function *isRelatedNode()* line 19 of the **Algorithm 7**). We have one special case: if the second union node $v_{UT_{i,i+1}}^\alpha$ is an *artificial union node* (function *isArtificialNode()* line 21 of the **Algorithm 7**): before merging the two union nodes, we check the rightness of the last artificial node position of $L(v_{UT_{i,i+1}}^\alpha)$ in the last tree T_{i+1} . In this way, with the Dynamic Time Warping's algorithm, we evaluate for each node in the list $L(v_{US_{0,i}}^\alpha)$, if the path from their root node to them are similar to the path from r_{i+1} to the artificial node of T_{i+1} (function *isArtificialSimilarPath()* line 24 of the **Algorithm 7**). After merging all union nodes (function *mergeNode()* line 27 of the **Algorithm 7**), edges are added (function *AddEdge()* line 3 in the **Algorithm 7**).

3.2.4.2. INSERTION IN $US_{0,i+1}$ OF NOT MERGED NODE FROM $US_{0,i}$ AND $UT_{i,i+1}$

Secondly, some union nodes in $US_{0,i}$ and $UT_{i,i+1}$ are not inserted in $US_{0,i+1}$. We begin to insert in $US_{0,i+1}$ the union nodes which didn't be analyzed in $US_{0,i}$ (loop corresponding to the line block 4 to 6 in **Algorithm 7**). With this first mosaic tree, we have no additional correction steps: they have been already considered in the building of the previous mosaic tree $US_{0,i}$ (recursive algorithm).

After, we insert in $US_{0,i+1}$ the union nodes in $UT_{i,i+1}$ which are not analyzed (loop corresponding to the line block 7 to 13 in the **Algorithm 7** with the function *addUnionNodeAndEdge()*). These nodes in $UT_{i,i+1}$ must have in their list $L(v_{UT_{i,i+1}}^\alpha) = \{\emptyset_i, v_{i+1}^{\alpha_{i+1}}\}$ i.e. only one node from the last considered tree T_{i+1} . Indeed, these nodes $v_{UT_{i,i+1}}^\alpha$ cannot have in their list $L(v_{UT_{i,i+1}}^\alpha)$ a node belonging to the tree T_i , since all nodes from it have been already inserted in the mosaic tree $US_{0,i+1}$ with the previous steps. The union nodes $v_{US_{0,i+1}}^\alpha$ newly inserted in $US_{0,i+1}$ haven't any matched nodes

in the previous trees among T_0 to T_i . Some important merging and correction steps must be taken into account.

3.2.4.3. MERGING OF NEWLY INSERTED UNION NODES WITH AN EXISTING NODE OF $US_{0,i+1}$

In the first merging step (Figure 35), we search if the new union node $v_{US_{0,i+1}}^\alpha$ can be merged with another union node $v_{US_{0,i+1}}^\beta$ in $US_{0,i+1}$ (function *isSameNode()* line 9 in the **Algorithm 7**).

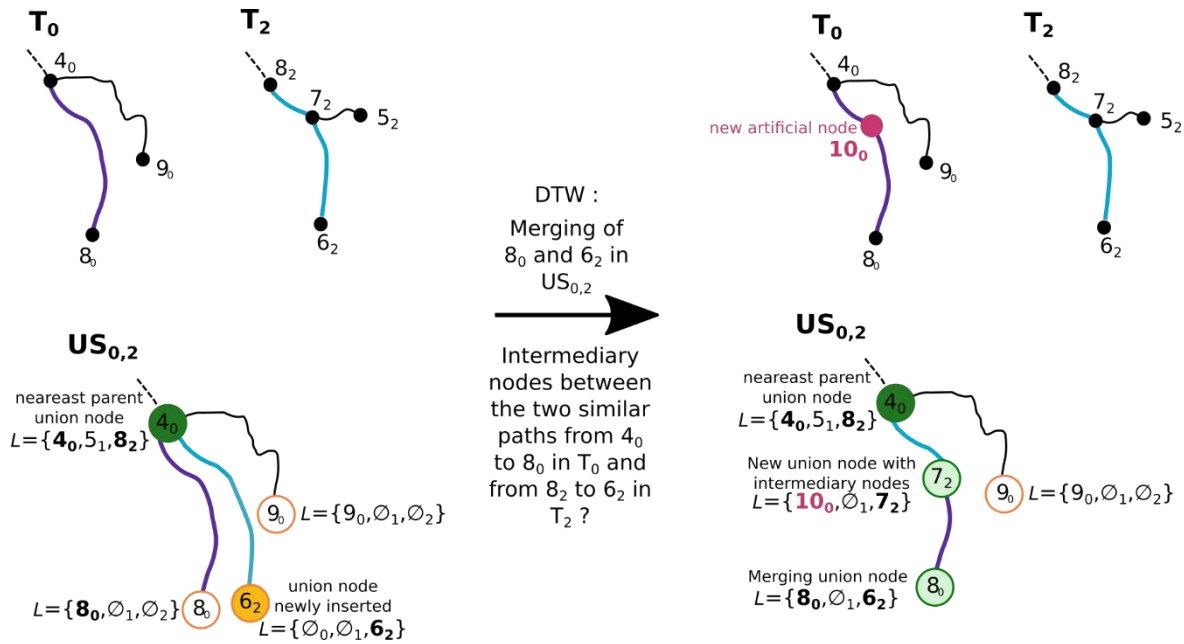


Figure 35: First correction step: the single union node with 6₂ as main node is newly inserted, we search if it can be matched with another union node in previous phases. After the Dynamic Time Warping (DTW) computation based on Euclidean distance of node coordinates, tangent and direction, the two paths in blue (from main nodes 4₀ and 6₂) and in purple (from main nodes 4₀ to 8₀) are considered as similar, so nodes 8₀ and 6₂ must be merged. The algorithm searches if there are intermediary nodes between the two similar paths: in the example the node 7₂ in the tree T₂. As previously, we search if this node can be matched with another node. In this example, an artificial node 10₀ is inserted in T₀ to match with 7₂.

To reduce the search space, we tested only the union nodes $v_{US_{0,i+1}}^\beta$ which didn't have in their list $L(v_{US_{0,i+1}}^\beta)$ node of the two trees T_i and T_{i+1} because of the previously building of the mosaic tree $US_{0,i}$. With the selected union nodes $v_{US_{0,i+1}}^\beta$ in $UT_{0,i+1}$, we search thanks to the Dynamic Time

Warping algorithm [KePa01, RaKe05] if all nodes in the list $L(v_{US_{0,i+1}}^\beta) = \{v_0^{\alpha_0}, \dots, v_{i-1}^{\alpha_{i-1}}, \emptyset_i, \emptyset_{i+1}\}$ are similar to the nodes in the newly inserted node list $L(v_{US_{0,i+1}}^\alpha)$. If the mean of DTW's result for the different nodes in the range [0-1] is lower than the value 0.5, we consider that the two union nodes represent the same matched node in different cardiac phases. Before merging them, some checks are realized about parent and children hierarchy. Hierarchical conflicts are detected in trees (from T_0 to T_{i-1}) which have a node in the list $L(v_{US_{0,i+1}}^\beta)$. If one conflict is detected, the two union nodes are not merged.

3.2.4.4. CORRECTION OF THE NEWLY INSERTED NODE FROM $UT_{i,i+1}$ WITH ARTIFICIAL NODE INSERTION IN $US_{0,i-1}$

The second correction step (Figure 36) is considered only if the first case is invalid (function *isSamePoint()* line 11 in the **Algorithm 7**).

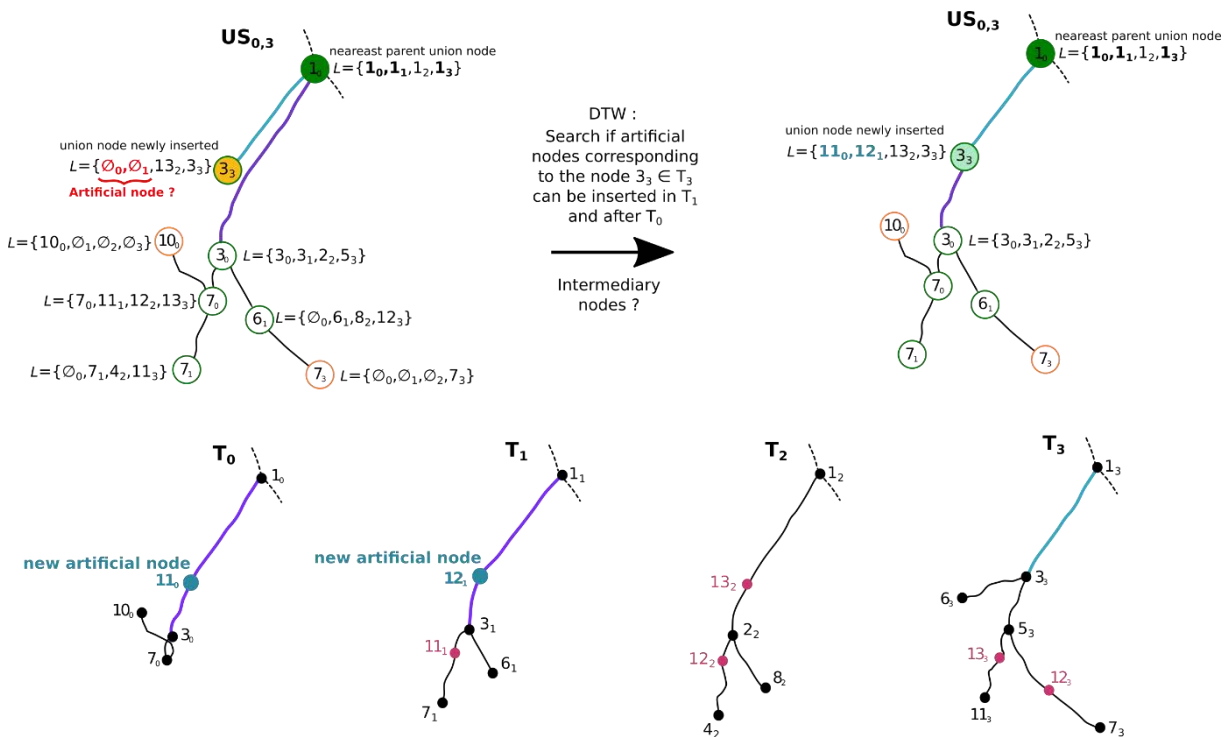


Figure 36: Second correction step: we search if an artificial node can be added in previous tree to complete the matching result. The algorithm takes into account the intermediary nodes between the two similar paths determined by DTW before the insertion of artificial nodes.

After searching if a merging node of $US_{0,i+1}$ represents the same matched node of another merging node, we evaluate if this last inserted merging node $v_{US_{0,i+1}}^\alpha$ can be matched with a centerline point in previous trees $S_{0,i-1}$. So we didn't want to merge two merging nodes but to complete the list $L(v_{US_{0,i+1}}^\alpha) = \{\emptyset_0, \dots, \emptyset_{i-1}, \emptyset_i, v_{i+1}^{\alpha_{i+1}}\}$ with the insertion of artificial nodes in previous trees of $S_{0,i-1}$ which are themselves completed (artificial nodes in blue in the Figure 34). As previously, we don't process the trees T_i and T_{i+1} . In addition, to reduce the execution time, in the same way as the artificial node insertion for the matching between two trees, all the centerlines are not considered to evaluate if one point of them can become an artificial node. Only the paths beginning by the *nearest common parent node* $v_{US_{0,i+1}}^\beta$ of the new merging node $v_{US_{0,i+1}}^\alpha$ are evaluated. The *nearest common parent node* must have in its list $L(v_{US_{0,i+1}}^\beta)$ nodes from the last considered tree T_{i+1} and also from the previous tree belonging to $S_{0,i-1}$ where we want insert artificial node. By default, the *nearest common parent node* is the root of the mosaic tree. The Dynamic Time Warping's algorithm determined the best similar path to the one described in T_{i+1} from the *nearest common parent node* $v_{US_{0,i+1}}^\beta$ to the new merging node $v_{US_{0,i+1}}^\alpha$. To determine the best similar path we must take into account if the new merging node $v_{US_{0,i+1}}^\alpha$ has children or not.

We begin with the first case with children. If the merging node $v_{US_{0,i+1}}^\alpha$ has more than one children merging node with a node from the tree T_{i+1} in their list, we search if conflicts due to parent or child nodes are present. If a conflict is detected, we resolve it: the matching can be modified and consequently the mosaic tree $US_{0,i+1}$ and so each tree of the sequence $S_{0,i+1}$. For example, the algorithm can detect wrong assignments due to a bifurcation absence near another one in previous trees. After, we search a children merging node of $v_{US_{0,i+1}}^\alpha$ which have in its list a node in the tree belonging to $S_{0,i-1}$ where we want insert an artificial node. The best similar path is determined: the source node is the *nearest common parent node* $v_{US_{0,i+1}}^\beta$ and the target corresponds to the merging children node found previously.

In the second case, the new merging node $v_{US_{0,i+1}}^\alpha$ has no child. To determine the target node of best similar path, we look only the children merging nodes of *nearest common parent node* $v_{US_{0,i+1}}^\beta$ which have in their list L a node corresponding to a leaf in the tree where we want insert an artificial node. For each of these children merging nodes, the parent and child hierarchy are checked in each tree. Among the children merging nodes with no conflict, thanks to the DTW's algorithm, the node corresponding to the target of the best similar path is determined.

When the best similar path is found in the current tree, an artificial node is added in this tree. We process to a last checking. The artificial node can correspond to an existing node: this matched node pair can be forgotten in the first step of the algorithm ($M_{i,i+1}$ the matching result of the two successive tree T_i and T_{i+1}). If it's the case, the matching result in the mosaic tree $US_{0,i+1}$ is corrected and each tree of the sequence $S_{0,i+1}$ updated. Besides, the algorithm takes into account intermediate union nodes in the two similar paths (see the intermediary node in Figure 35 and Figure 36). Therefore, some additional artificial nodes can be inserted in different trees according to the node including in the list L of the intermediate union nodes.

3.2.4.5. LAST TREE T_{i+1} CORRECTION THANKS TO SUPPLEMENTARY INFORMATION FROM $US_{0,i}$

Once all nodes of previous mosaic tree $US_{0,i}$ and union tree $UT_{i,i+1}$ are added in the new mosaic tree $US_{0,i+1}$ and their corresponding edges too, one last step must be realized (function *searchArtificialNodeLastView()* line 14 in the **Algorithm 7**). Indeed, there is some merging nodes $v_{US_{0,i+1}}^\alpha$ in $US_{0,i+1}$ from $US_{0,i}$ which haven't in their list any matched node from the two last trees T_{i+1} and T_i . That's why, for each union node in that case, we try to find an artificial node in the tree T_{i+1} to match it with all other nodes of theirs list $L(v_{US_{0,i+1}}^\alpha) = \{v_0^{\alpha_0}, \dots, v_{i-1}^{\alpha_{i-1}}, \emptyset_i, \emptyset_{i+1}\}$.

3.3. CONCLUSION

La mise en correspondance entre deux arbres vasculaires est un problème complexe. Une séquence d'arbres représentant la même structure anatomique pour un patient donné doit être mise en correspondance afin de prendre en compte à la fois les différents mouvements du cœur et les différences de topologie entre les arbres correspondant à différentes phases cardiaques dus à la variabilité de la reconstruction ou de la segmentation 3D.

Ce chapitre propose donc une méthode de mise en correspondance complète d'une séquence dynamique d'arbres coronaires non-isomorphes. De par les mesures de similarité utilisées et les nouvelles étapes proposées, notre approche permet d'obtenir un nombre de nœuds de bifurcation et de feuilles appariés plus important sur l'ensemble de la séquence par rapport aux données de départ.

L'algorithme exploite en effet toutes les informations des arbres de la séquence et s'autocorrige au fur et à mesure de la progression du traitement de la séquence.

Cette méthode apparie les nœuds des arbres de la séquence entre eux. A partir de ces nœuds, la correspondance entre les différents segments de la toute séquence est déduite pour permettre d'utiliser ce résultat lors de la recherche de l'incidence optimale. En effet, dans le prochain chapitre portant sur la recherche de l'angle de vue optimale, nous allons voir comment cette mise en correspondance des arbres sur l'ensemble de la séquence est utilisée. Elle permet d'obtenir un vue angiographique optimale pour le segment ou la bifurcation d'intérêt sur l'ensemble du cycle cardiaque et cela de manière automatique.

CHAPITRE 4

RECHERCHE D'INCIDENCES OPTIMALES

SOMMAIRE

4.1.	Presentation of the global algorithm	94
4.2.	Object of interest.....	96
4.2.1.	Vascular edges (segment).....	96
4.2.2.	Bifurcations.....	97
4.3.	Selection criteria.....	97
4.3.1.	Foreshortening	99
4.3.2.	External overlap.....	100
4.3.3.	Internal overlap	101
4.3.4.	Bifurcation opening angle	101
4.4.	Optimal angiographic view determination	102
4.4.1.	Segment optimal view	103
4.4.2.	Bifurcation optimal view	107
4.4.3.	Time complexity analysis.....	110
4.5.	Conclusion	111

L'arbre coronaire mosaïque d'une phase donnée, représentant une structure arborescente complète obtenue par la mise en correspondance de tous les arbres non-isomorphes de la séquence 3D+T, va nous permettre de déterminer l'angle de la vue optimale d'un objet d'intérêt (segment unique, segment multiple, bifurcation sténosée) sur l'ensemble du cycle cardiaque. Comme expliqué dans le chapitre 2, aucune méthode de détermination d'incidence optimale ne propose de méthode automatique de mise en correspondance de branches. De plus, notre méthode propose un vue de projection angiographique optimale pour différents objets d'intérêts. Effectivement, l'utilisateur par l'intermédiaire de l'interface graphique (GUI en anglais) de notre application peut sélectionner un segment ou une bifurcation comme zone de lésion afin de pouvoir déployer au mieux un stent. Il peut également choisir des segments multiples comme objets d'intérêts (par exemple de la racine de l'arbre coronaire à la zone de lésion) afin d'obtenir un angle de vue permettant à la fois de guider le cathéter de la racine jusqu'au déploiement du stent au niveau de la zone sténosée.

Au cours de ce chapitre, nous décrivons dans un premier temps les différents critères permettant d'obtenir les angles de vues optimales pour les différents objets d'intérêt et pour tout le cycle cardiaque. Le premier critère correspond au taux de **raccourcissement** (*foreshortening rate*) qui est souvent sous-évalué par les cardiologues [GCHM05] (artères coronaires très tortueuses). L'objectif est d'obtenir un angle de vue où l'objet sélectionné est le plus déployé possible. Les deux critères suivants sont liés au chevauchement de l'objet d'intérêt. Deux types de chevauchement sont définis : le chevauchement de l'objet avec le reste de l'arbre coronaire appelé **chevauchement externe** (*external overlap rate*) et le **chevauchement interne** (*internal overlap rate*) qui correspond au chevauchement de l'objet avec lui-même dans le cas d'une bifurcation. Effectivement, nous considérons que la bifurcation est composée de trois segments : la branche principale (branche mère) puis les deux branches filles. Le chevauchement interne prend en compte la superposition de ces trois branches entre elles. Ces deux critères de chevauchement nous permettent d'obtenir une vue angiographique de l'objet d'intérêt projeté avec le moins de superposition possible (meilleure visibilité). Le dernier critère concerne uniquement les bifurcations et est complémentaire aux autres. Il s'agit du taux d'**ouverture de l'angle** de la bifurcation (*opening bifurcation rate*). Plus l'angle entre les deux branches filles de la bifurcation est important, meilleur sera l'angle de vue ce qui permettra de déployer au mieux le stent dans cette zone.

Les deux premiers critères (raccourcissement et chevauchement externe) sont utilisés pour calculer l'angle de vue optimale pour les segments. Les quatre critères sont quant à eux utilisés pour le cas des bifurcations. La deuxième partie de ce chapitre explique comment ces différents critères

sont combinés entre eux en prenant en compte les différentes phases du cycle cardiaque. Prenons le cas le plus complexe, celui d'une bifurcation. L'utilisateur choisit via l'interface graphique la bifurcation comportant la sténose. Notre programme, grâce à l'algorithme précédent de mise en correspondance, reconnaît la bifurcation sur l'ensemble des arbres de la séquence. De part la nature non isomorphe des arbres coronaires, la bifurcation sélectionnée peut ne pas être présente sur tous les arbres de la séquence. L'algorithme proposé utilise uniquement les arbres possédant la bifurcation entière appariée (c'est-à-dire que les trois branches de la bifurcation doivent être toutes mises en correspondance).

Dans un premier temps, les taux de raccourcissement et d'angle d'ouverture de la bifurcation sont calculés pour chaque orientation possible du C-arm correspondant à la première phase. Deux cartes sont ainsi obtenues avec en abscisse les angles CRA/CAU allant de -30° à 30° et en ordonnée les angles OAD/OAG allant de -90° à 90° avec un pas de 1° . Les valeurs obtenues pour les deux cartes sont situées dans l'intervalle $[0,1]$. Une valeur forte correspond à un angle de vue plus approprié pour l'intervention (peu de raccourcissement ou un grand angle d'ouverture par exemple). Ces deux critères, basés sur la projection des points de la ligne centrale, sont calculés simultanément. La carte de raccourcissement et la carte d'angle d'ouverture de la bifurcation pour la première phase sont ensuite seuillées pour supprimer les angles les plus mauvais (seuil de 40%). La même démarche est ensuite réalisée pour les autres phases où la bifurcation est appariée. Cependant, les taux sont calculés uniquement pour les valeurs retenues précédemment. Une fois les cartes obtenues pour chaque phase, la carte globale des deux critères (raccourcissement et angle d'ouverture) est calculée pour l'ensemble du cycle cardiaque. Pour chaque orientation retenue, la valeur minimum de la séquence est conservée. L'utilisation du minimum, et non la moyenne ou la médiane, permet d'obtenir un angle de vue optimale sur toute la séquence prenant ainsi en compte les différences de topologies entre les arbres.

Dans un deuxième temps, les cartes des deux critères de chevauchement interne et externe sont calculées puis seuillées phase par phase où la bifurcation est présente et pour les orientations retenues à l'étape précédente (lors du calcul des taux de raccourcissement et d'angle d'ouverture). Ces cartes se basent sur la projection des volumes de la bifurcation (séparée en trois branches) et du reste de l'arbre coronaire. La carte globale du critère de chevauchement externe et celle du chevauchement interne pour l'ensemble du cycle cardiaque sont obtenues de la même façon que précédemment avec l'utilisation de la valeur minimum entre les cartes.

Une fois les cartes globales pour chaque critère obtenues, celle-ci sont fusionnées pour obtenir la carte globale de l'angle de vue optimale. Une simple combinaison linéaire de poids identique est

utilisée (moyenne des cartes). Effectivement, nous considérons que chaque critère possède la même importance avec 50% de critères spécifiques à la bifurcation (chevauchement interne et angle d'ouverture). Au final, une carte optimale, prenant en compte les différences de topologie des arbres de la séquence 3D+T et le mouvement du cycle cardiaque, est proposée au cardiologue. Pour réduire le nombre de choix, 5% ou 2% des meilleures incidences sont proposées ainsi que la valeur maximum représentant l'angle de vue optimale pour la bifurcation sélectionnée.

La détermination de l'angle de vue optimale pour un segment d'intérêt unique ou un segment multiple (concaténation de plusieurs segments) est effectuée de la même manière. En effet, les cartes de raccourcissement sont calculées puis seuillées phase par phase, puis les cartes de chevauchement externes sont obtenues seulement dans les régions où le taux de raccourcissement est considéré comme suffisant (par le seuil). Concernant les cartes optimales obtenues pour des segments multiples (segments de la racine jusqu'à la zone de lésion sélectionnée), nous vérifions également si le résultat est précis et cohérent grâce à la fusion des cartes des différents segments composant ce segment multiple. Deux tests de fusion sont considérés : chaque carte a le même poids ou bien le poids d'un segment est pondéré par rapport à sa distance au segment sténosé ainsi que par sa longueur.

Ce chapitre rédigé également en anglais est issu de deux articles : un publié démontrant la faisabilité d'une pré-mise en correspondance sur un segment d'intérêt [FSNB13] et un article en cours de soumission.

4.1. PRESENTATION OF THE GLOBAL ALGORITHM

In this chapter, we choose to develop a patient-specific method to determine the optimal viewing angle for the whole cardiac cycle thanks to our mosaic-based tree matching algorithm of the whole 3D+T coronary sequence (section 3.2) as proposed in the section 2.3. Moreover, the optimal view map can be computed for different objects of interest: a single segment, a multiple segment and a bifurcation. The map of the different objects of interest can be used to choose an optimal view to deploy the stent (stenosis segment or bifurcation) or a global optimal gantry orientation for the catheter guidance in the vessel (i.e. from the first coronary artery vessel to the stenosis localization).

The Figure 37 summarizes our algorithm which determines the optimal viewing angle of an object of interest during the whole cardiac cycle. Our method begins by matching the whole coronary 3D+T sequence. Each coronary segment can be followed in the different cardiac phases. Afterward, for a

matched and selected object (single segment, bifurcation and multiple segment) the optimal view map on the whole cardiac cycle is determined under criteria (foreshortening, external and internal overlap, and bifurcation opening rates). These different criteria are described and next, the global algorithm is proposed.

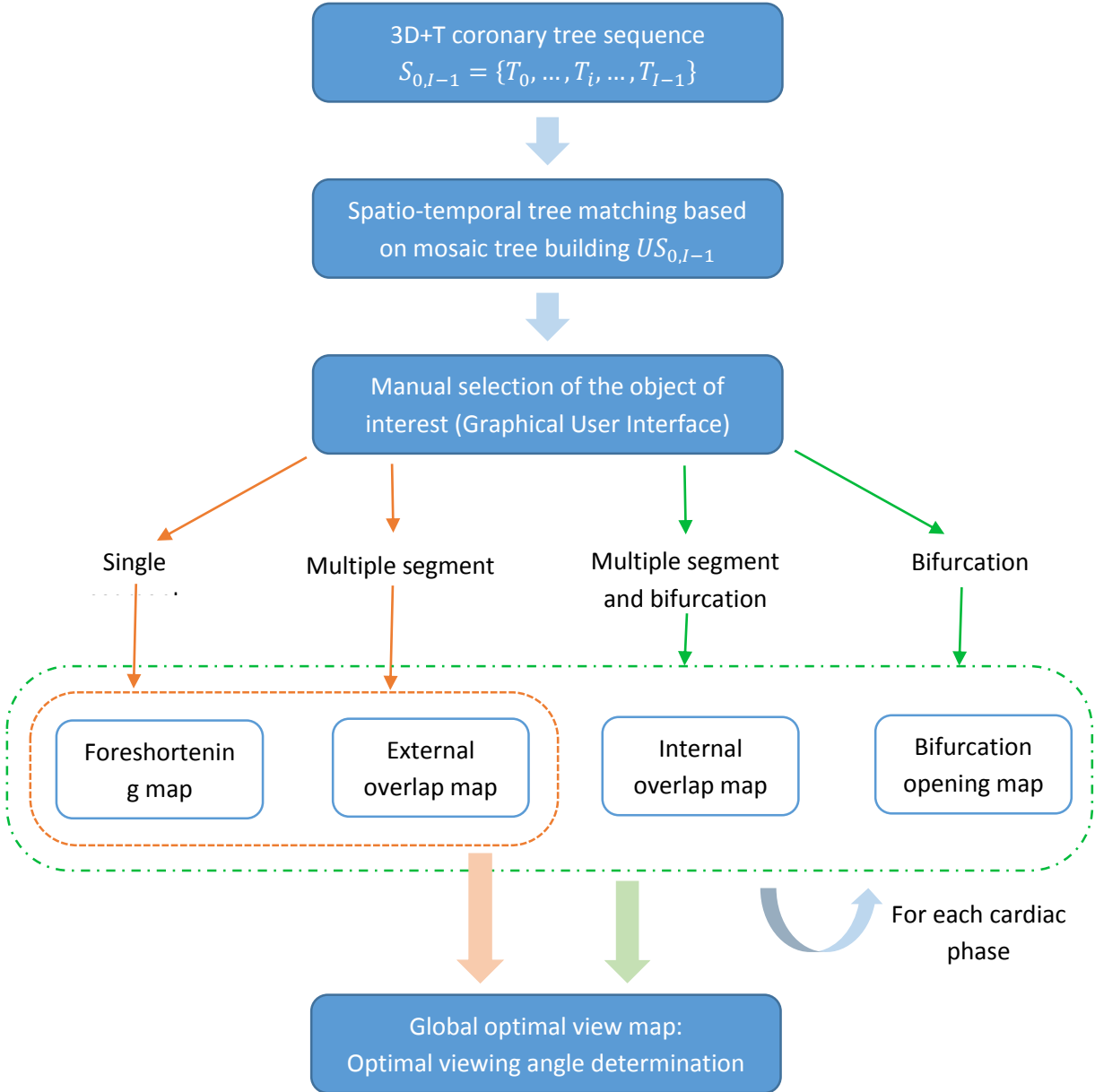


Figure 37: summary steps of the optimal viewing angle determination.

4.2. OBJECT OF INTEREST

After the matching of the 3D+T coronary sequence have been performed, the user selects with the graphical user interface (GUI) the object of interest. If the physician needs an optimal angiographic projection view to guide the catheter in this dynamic structure from the root to the stenosis location, he should select the multiple vascular edge (*multiple segment* in Figure 37). Conversely, if he wants an optimal view used only to deploy the stent, the object of interest must correspond to one single edge (*segment* in Figure 37). Moreover, if the stenosis is located in a bifurcation area, the user can select 1) *Bifurcation* or 2) *Multiple segment and bifurcation* as object of interest (Figure 37).

4.2.1. VASCULAR EDGES (SEGMENT)

Let be $L_{0,Q-1} = \{l_0, \dots, l_q, \dots, l_{Q-1}\}$ the different matched centerline portions in the coronary tree sequence $S_{0,I-1}$, with Q the number of matched centerlines in the whole cardiac cycle. $l_q = \{m_q^1, \dots, m_q^{\rho_q}, \dots, m_q^{\mu_q}\}$ represents the matched segments of interest selected by the user (e.g. single edge, multiple edge or edge part) in one cardiac phase, i.e. a list of μ_q points (Figure 38-A). The different centerlines l_q in $L_{0,Q-1}$ represent the same matched segment. They have the same label after the mosaic-based tree matching algorithm. Thus, we have $1 \leq Q - 1 \leq I - 1$. If $Q - 1 = 1$, there is only a selected segment in one cardiac phase. Conversely, if $Q - 1 = I - 1$, there is a matched segment in all cardiac phases. If the user selects a matched edge $e_i^{\alpha_i} = \{m_i^1, \dots, m_i^{\rho_i}, \dots, m_i^{\mu_i}\}$ in a t_i cardiac phase, the point list of the edge $e_i^{\alpha_i}$ are the same as in l_q . However, if a multiple edge is selected by the user, the list of points of each edge of the multiple edge is concatenated to obtain l_q .

The different matched centerlines are used to compute the foreshortening and the bifurcation opening rates. To compute the two overlap rates, the volume information corresponding to meshes are needed. Let be $B_{0,Q-1} = \{b_0, \dots, b_q, \dots, b_{Q-1}\}$ the matched volume of the selected vessel, with as previously Q the number of matched 3D vessels in the cardiac cycle. b_q and l_q represent respectively the meshes and the centerlines of the selected volume at the same cardiac phase. $\overline{B_{0,Q-1}} = \{\overline{b_0}, \dots, \overline{b_q}, \dots, \overline{b_{Q-1}}\}$ represents the other vessels in the corresponding trees.

4.2.2. BIFURCATIONS

As explained by Kitslaar et al. [KMJK08], the selected bifurcations $Bif_{0,Q-1} = \{bif_0, \dots, bif_q, \dots, bif_{Q-1}\}$ with Q the number of matched bifurcations in the cardiac cycle is separated in three vessels (Figure 38-B). The first vessel corresponds to the parent. For a bifurcation bif_q , the parent centerline is written l_q^{father} and the associated volume information b_q^{father} . The two other vessels represent the two children branches (respectively named l_q^{child1} and l_q^{child2} for the centerline information and b_q^{child1} and b_q^{child2} for the volume information). If the chosen object of interest corresponds to *multiple segment and bifurcation*, the additional information is concatenated in the centerline or the volume information (for instance l_q^{father} and b_q^{father} if the multiple segment corresponds to the tree segment root until the bifurcation) such as previously.

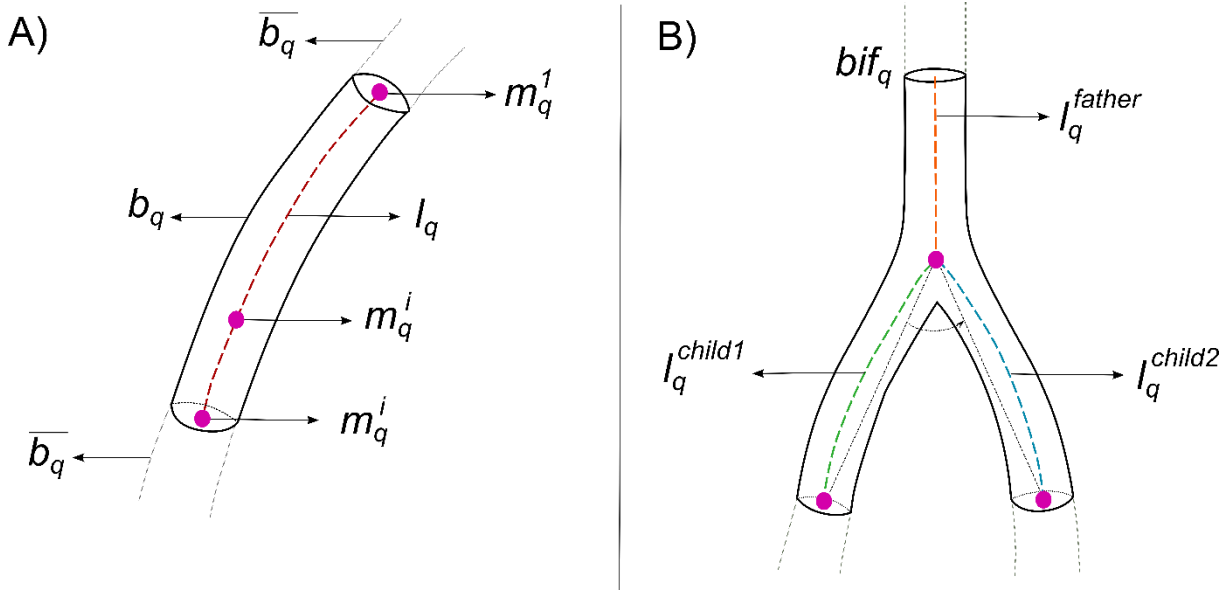


Figure 38: Schema and notation for a segment (A) and a bifurcation (B) of interest.

4.3. SELECTION CRITERIA

When the object of interest is selected by the user, the different criteria must be computed. In this paper, we use four criteria (Figure 37). The first criterion corresponds to the vascular foreshortening. The vessel foreshortening rate, as shown in Green et al. [GCHM05], is underappreciated by the

physicians in the angiographic projecting views contrary to the vessel overlap rate which is more easily identified. In the selected gantry orientation, the object of interest must be the most deployed as possible (Figure 39). The two next criteria concern the structure overlap. The selected 2D view must showing the object of interest with the least overlap with the other cardiac vessels (external overlap rate) but also with himself (internal overlap rate). The last criterion is related to the bifurcation opening angle. If the stenosis is in a bifurcation area, this region should to be perpendicular to the X-ray source. This bifurcation angle must be maximizing (Figure 38-B).

These different criteria are computed for all possible gantry orientation (θ_j, φ_k) . These view angles depend on the angiograph. That's why we put the RAO and LAO angulation limited to 90° and the CRAN and CAU angulation restricted to 30° (equation (6)). In this paper, we choose an angular step of 1° for our evaluation.

$$(\theta_j, \varphi_k) \in \mathbb{R}^2 \text{ with } \theta_j \in [-90^\circ; 90^\circ] \text{ and } \varphi_k \in [-30^\circ; 30^\circ] \tag{6}$$

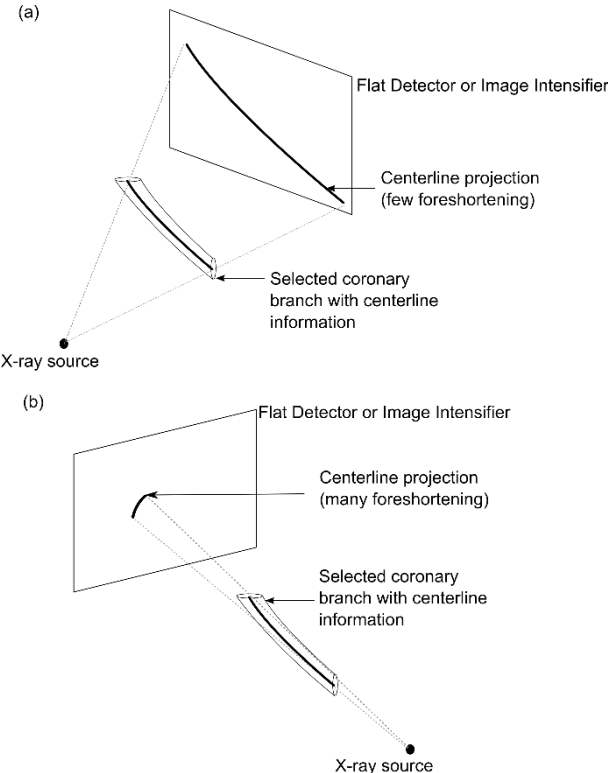


Figure 39: schematic representation of two foreshortening rates for two different viewing angles (θ_j, φ_k) . (a) The gantry orientation corresponds to a high foreshortening rate and it better than (b) which represent a bad foreshortening rate.

4.3.1. FORESHORTENING

Let be $f_q(\vec{p}, \theta_j, \varphi_k, l_q)$ the foreshortening rate for a t_i cardiac phase of the matched and selected centerline l_q . The vessel foreshortening for a cardiac phase is normally determined by comparing the length of the 3D selected centerline l_q to the length of the projected vessel centerline for a given gantry orientation (θ_j, φ_k) [BiGS10, CTFC98]. However, we choose to use not the real 3D length of the centerline but the maximum length found of this projected centerline. This estimation also offers pair of angles with the lowest foreshortening among given selected gantry orientation. The foreshortening rate for a cardiac phase is computed as follows:

$$f_q(\vec{p}, \theta_j, \varphi_k, l_q) = \frac{\text{length}_q(\vec{p}, \theta_j, \varphi_k, l_q)}{\max_{(\vec{p}, \theta_j, \varphi_k)}(\text{length}_q(\vec{p}, \theta_j, \varphi_k, l_q))} * 100\%, \quad (7)$$

$$\text{with } \text{length}_q(\vec{p}, \theta_j, \varphi_k, l_q) = \sum_{\rho_q=1}^{\varphi_q-1} d_e\left(P_{\vec{p}, \theta_j, \varphi_k}(m_q^{\rho_q}), P_{\vec{p}, \theta_j, \varphi_k}(m_q^{\rho_q+1})\right)$$

where $P_{\vec{p}, \theta_j, \varphi_k}(m_q^{\rho_q})$ is the forward projection of the centerline point $m_q^{\rho_q} \in l_q$ according to the pair of angles (θ_j, φ_k) and the projection direction \vec{p} . $d_e(\)$ represents the Euclidean distance between two points (here, two 2D projected points).

The equation (7) can be also used for multiple centerline segments and consequently a selected bifurcation: the length of each projected segment are added for a given orientation and next divided as previously by the computed maximum length.

When the foreshortening rate is computed for each cardiac phase which have an existing matched segment l_q , the foreshortening rate for all cardiac cycles $F(\vec{p}, \theta_j, \varphi_k, L_{0,Q-1})$ can be calculated (equation (8)).

The chosen angiographic view must be optimal for all cardiac phases. Thus, we don't use the mean or median of foreshortening rates for a given orientation which is not representative, but we select the minimum value. With our optimisation process, if the minimum value is too low for a given orientation in only one phase, this orientation should not be retained as possible optimal viewing angle.

$$F(\vec{p}, \theta_j, \varphi_k, L_{0,Q-1}) = \min_q \left(f_q(\vec{p}, \theta_j, \varphi_k, l_q) \right) \quad (8)$$

4.3.2. EXTERNAL OVERLAP

As explained previously, the vessel overlap happens when a ray intersects at least two vessels, i.e. when two or more segments are projected into the same pixel in the 2D angiographic projection image. In this paper, two different kinds of vessel overlaps are considered: the *external overlap* and the *internal overlap*.

The first overlap criteria, the *external overlap*, is defined when the selected vessel is superposed with other vascular structures in 2D projections for a given gantry orientation (θ_j, φ_k) . Let be $o_{ext_q}(\vec{p}, \theta_j, \varphi_k, b_q)$ the external overlap rate of the matched selected vessel b_q (meshes) for a t_i cardiac phase and a given orientation (θ_j, φ_k) . This rate is computed as follows:

$$o_{ext_q}(\vec{p}, \theta_j, \varphi_k, b_q) = \frac{\cap \left(Area \left(P_{\vec{p}, \theta_j, \varphi_k} (b_q) \right), Area \left(P_{\vec{p}, \theta_j, \varphi_k} (\overline{b_q}) \right) \right)}{Area \left(P_{\vec{p}, \theta_j, \varphi_k} (b_q) \right)} * 100\%, \quad (9)$$

where $P_{\vec{p}, \theta_j, \varphi_k} (b_q)$ is the projection of the points which composes the selected matched volume $b_q \in B_{0, Q-1}$ (meshes) according to the projection direction \vec{p} and the pair of angles (θ_j, φ_k) . The projected points for a volume are next triangulated to obtain a 2D surface. $\overline{b_q}$ corresponds to the other vessels of the coronary tree. $Area()$ represents the 2D area of the projected volume in parenthesis.

As in Chen et al. [CTFC98], Kitslaar et al. [KMJK08], Bi et al. [BiGS10] and Wang et al. [WYCA14], the equation (9) counts the pixels of the superposed projection areas and to divide it by the pixel number of the projected selected vessel in the 2D projection image.

To take into account the coronary movement and the non-isomorphic vascular structure, the external overlap of the whole cardiac cycle for a given orientation corresponds to the minimum value of external overlap $o_{ext_q}(\vec{p}, \theta_j, \varphi_k, b_q)$ of the matching structure b_q for each existing cardiac phase (equation (10)). Contrarily to the foreshortening rate $f_q(\vec{p}, \theta_j, \varphi_k, l_q)$ (equation (7)), the order of the value is inverted: a value near 1 corresponds to a high external overlap rate, the segment has a lot of superposition with the other vascular structures. To combine all criteria in the next section 4.4, we take as external overlap rate: $1 - o_{ext_q}(\vec{p}, \theta_j, \varphi_k, b_q)$.

$$O_{ext}(\vec{p}, \theta_j, \varphi_k, B_{0, Q-1}) = \min_q \left(1 - o_{ext_q}(\vec{p}, \theta_j, \varphi_k, b_q) \right) \quad (10)$$

4.3.3. INTERNAL OVERLAP

The second overlap criterion, the *internal overlap*, is only computed if a bifurcation bif_q is selected. This rate calculates the overlap between the bifurcation with itself. As explained previously and by Kitslaar et al. [KMJK08], the bifurcation is divided into three segments. To compute this percentage for a given orientation and each phase, only the bifurcation is projected in the 2D image. The calculating principle of the internal overlap rate $o_{int_q}(\vec{p}, \theta_j, \varphi_k, bif_q)$ is identical to the external overlap rate $o_{ext_q}(\vec{p}, \theta_j, \varphi_k, bif_q)$ (equation (11)). The internal overlap rate for a given phase corresponds to the number of pixels in the intersection of the three branches divided by the pixel number of the union of these three branches.

$$o_{int_q}(\vec{p}, \theta_j, \varphi_k, bif_q) \quad (11)$$

$$= \frac{\cap \left(Area \left(P_{\vec{p}, \theta_j, \varphi_k} (b_q^{father}) \right), Area \left(P_{\vec{p}, \theta_j, \varphi_k} (b_q^{child1}) \right), Area \left(P_{\vec{p}, \theta_j, \varphi_k} (b_q^{child2}) \right) \right)}{\cup \left(Area \left(P_{\vec{p}, \theta_j, \varphi_k} (b_q^{father}) \right), Area \left(P_{\vec{p}, \theta_j, \varphi_k} (b_q^{child1}) \right), Area \left(P_{\vec{p}, \theta_j, \varphi_k} (b_q^{child2}) \right) \right)}$$

* 100%,

with $P_{\vec{p}, \theta_j, \varphi_k} (b_q^{father})$ the projection of the mesh b_q^{father} according to the projection direction \vec{p} and the angles (θ_j, φ_k) . $Area(\)$ represents the area of the projection volume after triangulation.

As the other criteria, the internal overlap rate is computed for the whole cardiac cycle (equation (12)). As for the external overlap rate, a value near 1 corresponds to a bad value: the bifurcation has a lot superposition with itself. Thus, to combine all criteria in the next section 4.4, we take as internal overlap rate: $1 - o_{int_q}(\vec{p}, \theta_j, \varphi_k, bif_q)$.

$$O_{int}(\vec{p}, \theta_j, \varphi_k, Bif_{0,Q-1}) = \min \left(1 - o_{int_q}(\vec{p}, \theta_j, \varphi_k, bif_q) \right) \quad (12)$$

4.3.4. BIFURCATION OPENING ANGLE

The last criterion is also used when a bifurcation is selected. As previously, the bifurcation is also split into three segments. However, only centerlines $(l_q^{father}, l_q^{child1}, l_q^{child2})$ are used and

especially three points: the bifurcation point and the last point (target) of the two children segments. The *bifurcation opening* criterion corresponds to maximize the angle represented by the three projected points. This angle for a given orientation is computed thanks to the scalar product. The equation (13) gives the *bifurcation opening* angle for a given gantry orientation.

$$bo_q(\vec{p}, \theta_j, \varphi_k, bif_q) = \frac{angle_q(\vec{p}, \theta_j, \varphi_k, bif_q)}{\max_{(\theta_j, \varphi_k)} angle_q(\vec{p}, \theta_j, \varphi_k, bif_q)} * 100\%, \quad (13)$$

$$\text{With } angle_q(\vec{p}, \theta_j, \varphi_k, bif_q) = \cos^{-1} \left(\frac{\overline{OA} \cdot \overline{OB}}{\|OA\| \times \|OB\|} \right)$$

As previously, the *bifurcation opening* rate is computed for each cardiac phase (equation (14)) with the minimum value of the different maps which is kept.

$$BO(\vec{p}, \theta_j, \varphi_k, Bif_{0,Q-1}) = \min_q (a_q(\vec{p}, \theta_j, \varphi_k, bif_q)) \quad (14)$$

4.4. OPTIMAL ANGIOGRAPHIC VIEW DETERMINATION

As explained in Figure 37, the criterion choice depends on the object of interest selected by the user. For a segment, the foreshortening and external overlap criteria are computed and for a bifurcation, the four proposed criteria are used to determine optimal viewing angles. In this section, we explain how the different criteria are combined to optimize the optimal viewing angle determination process. Indeed, if criteria are computed for each possible gantry orientation and for each cardiac phase, the execution time becomes too high (if the coronary tree sequence has ten cardiac phases, 110410 projections ($181 \times 61 \times 10$, see equation (6) with the interval of the primary and secondary angles, the angular step of 1° and the 10 phases) have to be computed and evaluated). The global algorithm and the optimisation process are slightly different according to the type of the object of interest using different criteria (foreshortening, external and internal overlap and bifurcation opening angle).

4.4.1. SEGMENT OPTIMAL VIEW

In this section, we explain the algorithm used to determine optimal viewing angle for a matched selected segment $L_{0,Q-1} = \{l_0, \dots, l_q, \dots, l_{Q-1}\}$. The algorithm can be divided in three steps (Algorithm 8).

Firstly, from line 8 to line 38 in the Algorithm 8, the foreshortening rate is computed for each matched l_q and for each gantry orientation (equation (7)). However, after foreshortening map f_q building for each q phase, a threshold is performed to remove angles with worse foreshortening values (block line 25 to 28 Algorithm 8). Indeed, we want an angiographic view which must be optimal for all cardiac cycles. Thus, in the algorithm, a binary mask (with same dimension as maps) is created. It's updated after each map building: for each thresholded foreshortening rate, the corresponding value in the mask becomes false and for the following maps, these angles are not anymore considered. Next, the global foreshortening map F for $L_{0,Q-1}$ is computed with the minimum value for each selected gantry orientation in the mask (equation (8)). To reduce the projection number and consequently the execution time of the algorithm, we perform another threshold, more selective than the previous, on the global foreshortening map F (block line 31 to 35 in the Algorithm 8).

Secondly, from line 40 to 61 in the Algorithm 8, the external overlap rate is calculated for each projected vessel $B_{0,Q-1} = \{b_0, \dots, b_q, \dots, b_{Q-1}\}$ and for the retained gantry orientation i.e. angles with true value in the mask (equation (9)). As previously, a threshold is performed on each map o_{ext_q} to remove incorrect angles (block line 51 to 55 in the Algorithm 8). Contrary to the foreshortening step, no selection is made after the global external overlap map O_{ext} building (equation (10)).

The last step, from line 63 to 76 in the Algorithm 8, consists to compute the optimal view map OV thanks to the linear combination of the foreshortening F and external overlap O_{ext} maps for each selected gantry orientation (line 56 in the Algorithm 8). The maximum value of the optimal view map gives the optimal viewing angle of the segment in the whole cardiac cycle.

Algorithm 8: Optimal view algorithm for a matched segment of interest, function *OptimalViewMapSegment()*.

Input: the matched centerline of the selected segment $L_{0,Q-1} = \{l_0, \dots, l_q, \dots, l_{Q-1}\}$. $B_{0,Q-1} = \{b_0, \dots, b_q, \dots, b_{Q-1}\}$ the matched volume of the selected vessel. $\overline{B_{0,Q-1}} = \{\overline{b_0}, \dots, \overline{b_q}, \dots, \overline{b_{Q-1}}\}$ the other vessels in the corresponding trees. *thresholdSupp* is the threshold used to suppress worse

couples of angle and *thresholdForeshortening*, the threshold used to select gantry orientation with best foreshortening rate.

Output: The optimal view map *OV* and consequently the optimal viewing angles $(\theta_j, \varphi_k)_{max}$.

```

//Initialization of Foreshortening map F, External Overlap map O_ext and Optimal View map OV
1: //for each possible gantry orientation with  $\theta_j \in [-90^\circ, 90^\circ]$  and  $\varphi_k \in [-30^\circ, 30^\circ]$  :
2:  $F[\theta_j][\varphi_k], O_{ext}[\theta_j][\varphi_k], OV[\theta_j][\varphi_k] \leftarrow 0$ 
//Initialization of temporary Foreshortening and External overlap maps for each gantry
3: //orientation  $(\theta_j, \varphi_k)$  and each matched phases  $q$  with  $0 \leq q \leq Q - 1$ :
4:  $f[\theta_j][\varphi_k][q], o_{ext}[\theta_j][\varphi_k][q] \leftarrow 0$ 
//Initialization of the mask used to select correct gantry orientation thanks to the different
5: //thresholds:
6:  $mask[\theta_j][\varphi_k] \leftarrow 1$ 
7:
8: //First step: compute Foreshortening map F.
9: for each matched selected segment  $l_q$  in  $L_{0,Q-1}$  do
10:    $length_{max} \leftarrow 0$ 
11:   for each pair of angles  $(\theta_j, \varphi_k)$  with  $\theta_j \in [-90^\circ, 90^\circ]$  and  $\varphi_k \in [-30^\circ, 30^\circ]$  do
12:     if  $mask[\theta_j][\varphi_k] == 1$  then
13:       //Projection of  $l_q$  according to  $(\theta_j, \varphi_k)$  and the projection direction  $\vec{p}$ :
14:        $P_{\vec{p}, \theta_j, \varphi_k}(l_q) \leftarrow computeProjectionPoints(\vec{p}, \theta_j, \varphi_k, l_q)$ 
15:       //Compute the length of the projected segment ( $length_q(\vec{p}, \theta_j, \varphi_k, l_q)$ ) in equation
16:       // (7):
17:        $f[\theta_j][\varphi_k][q] \leftarrow compute2DLength(P_{\vec{p}, \theta_j, \varphi_k}(l_q))$ 
18:       //Update the computed maximum length:
19:        $length_{max} \leftarrow updateMaxLength(f[\theta_j][\varphi_k][q])$ 
20:     end if
21:   end for each
22: //Normalisation of each  $f[\theta_j][\varphi_k][q]$  by the maximum projected length:
23: for each pair of angles  $(\theta_j, \varphi_k)$  where  $mask[\theta_j][\varphi_k] == 1$  do
24:    $f[\theta_j][\varphi_k][q] \leftarrow f[\theta_j][\varphi_k][q] / length_{max}$  //  $f_q(\vec{p}, \theta_j, \varphi_k, l_q)$  in equation (7).
25:    $F[\theta_j][\varphi_k] \leftarrow updateMinForeshorteningValue(f[\theta_j][\varphi_k][q])$  //equation (8)
26:   if  $f[\theta_j][\varphi_k][q] < thresholdSupp$  then

```

```

26:     mask[θj][φk] ← 0 //this gantry orientation is unselected: bad value for one phase.
27:     F[θj][φk] ← 0
28:   end if
29:   if q == Q - 1 then //if last matched view for the segment lq
30:     //Threshold foreshortening used to reduce execution time and selected better angles:
31:     if F[θj][φk] < thresholdForeshortening then
32:       mask[θj][φk] ← 0
33:       F[θj][φk] ← 0
34:     end if
35:   end
36: end for each
37: end for each
38:
39: //Second step: compute External Overlap map Oext.
40: for each matched selected segment bq in B0,Q-1 do
41:   for each pair of angles (θj, φk) with θj ∈ [-90°, 90°] and φk ∈ [-30°, 30°] do
42:     if mask[θj][φk] == 1 then
43:       //Projection of bq and  $\overline{b_q}$  according to (θj, φk) and the projection direction  $\vec{p}$ :
44:       P $\vec{p}, \theta_j, \varphi_k$ (bq) ← computeProjectionVolume( $\vec{p}$ , θj, φk, bq)
45:       P $\vec{p}, \theta_j, \varphi_k$ ( $\overline{b_q}$ ) ← computeProjectionVolume( $\vec{p}$ , θj, φk,  $\overline{b_q}$ )
46:       //Compute external overlap criteria: number of superimposed pixels between
47:       //P $\vec{p}, \theta_j, \varphi_k$ (bq) and P $\vec{p}, \theta_j, \varphi_k$ ( $\overline{b_q}$ ) divided by the pixel number of P $\vec{p}, \theta_j, \varphi_k$ (bq):
48:       oext[θj][φk][q] ← externalOverlap(P $\vec{p}, \theta_j, \varphi_k$ (bq), P $\vec{p}, \theta_j, \varphi_k$ ( $\overline{b_q}$ )) //equation (9)
49:       Oext[θj][φk] ← updateMinExtOverlapValue(oext[θj][φk][q]) //equation (10)
50:       //Suppression of gantry orientation with worse values for one phase:
51:       if oext[θj][φk][q] < thresholdSupp then
52:         mask[θj][φk] ← 0 //this gantry orientation is unselected: bad value for one phase.
53:         Oext[θj][φk] ← 0
54:       end if
55:     end if
56:   end for each
57: end for each

```

```

58:
59: //Third step: compute Optimal view map OV with linear combination between the two criteria.
60:  $OV_{max} \leftarrow 0$  //Initialization of the optimal viewing angle value
61: for each pair of angles  $(\theta_j, \varphi_k)$  with  $\theta_j \in [-90^\circ, 90^\circ]$  and  $\varphi_k \in [-30^\circ, 30^\circ]$  do
62:   if  $mask[\theta_j][\varphi_k] == 1$  then
63:      $OV[\theta_j][\varphi_k] \leftarrow F[\theta_j][\varphi_k] * 0.5 + O_{ext}[\theta_j][\varphi_k] * 0.5$ 
64:     if  $OV[\theta_j][\varphi_k] > OV_{max}$  then
65:        $OV_{max} \leftarrow OV[\theta_j][\varphi_k]$ 
66:        $(\theta_j, \varphi_k)_{max} \leftarrow (\theta_j, \varphi_k)$ 
67:     end if
68:   end if
69: end for each
75: return  $(OV, (\theta_j, \varphi_k)_{max})$ 

```

This algorithm can be used for a single or multiple segment (in our case, a multiple segment corresponds to successive segments from the tree root to the selected segment). In the section 5.3.3, we evaluate if the optimal view map for a multiple segment is equivalent to a linear combination between the optimal view maps of the different single segments (belonging to the multiple segment). The result may be different. Indeed, the multiple segment should have less overlap with the other structures. The different single segments are concatenate into one object. One at a time, each single segment can overlap the other single segments. Moreover, the number of phases, where the segment is matched, can be different in the multiple segments and in the separated segments. Indeed, the different parent segments of the selected segment can be present in more phases (e.g. the root segment is present in each phase).

We consider that each optimal view map of the independent single segments doesn't have the same influence on the result. To obtain an optimal viewing angle which can be used both for the catheter guidance and the stent deployment, we choose two criteria to determine the weight of each map. The first criterion is about the hierarchical distance of each segment compared to the selected segment (stenotic segment). More the segment is near the selected segment, more its weight must be high. Furthermore, the physician should have more information about segments which are deeper than the first segment in the coronary tree (root and direct children for example). The difficulty to guide the catheter is more significant for segment with higher depth than for the root segment. The second criterion concerns the length of each segment. The weight of a longer segment is higher than

the weight of a small segment. A segment smaller than the others is considered as less important. The physician spends less time in this small segment during the catheter guidance. To determine the weight according to the length criterion for each segment, its length is divided by the total length of all segments. The weight related to the hierarchical level is computed in the same way: the level of the target node of the segment is divided by the sum of all segment level of the tree. The final weight factor corresponds to the mean of the two preview weights (length and hierarchical level).

4.4.2. BIFURCATION OPTIMAL VIEW

The algorithm to determine optimal viewing angle for a bifurcation is slightly different. Indeed, it uses the four constraints as described previously (Algorithm 9). The first part of the algorithm begins by calculating the foreshortening and the bifurcation opening angle rates in the same time as the two equations (7) and (13) (line block 8 to 35 in the Algorithm 9). They both use the centerline projection. The same mask created as previously is updated at each built map f_q and bo_q thanks to the threshold value (line block 30 to 33 in the Algorithm 9). When the thresholded foreshortening f_q and bifurcation opening bo_q maps are calculated for each phase q , their global map F and BO is obtained (equations (8) and (14)). Contrary to the algorithm created for a segment, no other threshold is performed. The second part of the algorithm (line block 37 to 58 in the Algorithm 9) consists to build the internal O_{int} and external overlap O_{ext} maps in the same way as previously (successive threshold on the internal o_{int_q} and external o_{ext_q} overlap map for each phase) (equations (9) to (12)). The last part of the Algorithm 9, from line 60 to line 73, computes the optimal view map OV thanks to a linear combination with same weight between the four criteria. The maximum value of the optimal view map gives the optimal viewing angle for the selected bifurcation.

Algorithm 9: Optimal view algorithm for a matched bifurcation of interest, function *OptimalViewMapBifurcation()*.

Input: the matched centerline of the selected segment $L_{0,Q-1} = \{l_0, \dots, l_q, \dots, l_{Q-1}\}$. $B_{0,Q-1} = \{B_q^{father}, B_q^{child1}, B_q^{child2}\}$ the matched volume of the selected bifurcation with $B_q^{father} = \{b_0^{father}, \dots, b_q^{father}, \dots, b_{Q-1}^{father}\}$, $B_q^{child1} = \{b_0^{child1}, \dots, b_q^{child1}, \dots, b_{Q-1}^{child1}\}$ and $B_q^{child2} = \{b_0^{child2}, \dots, b_q^{child2}, \dots, b_{Q-1}^{child2}\}$ the three matched bifurcation branches. $\overline{B_{0,Q-1}} = \{\overline{b_0}, \dots, \overline{b_q}, \dots, \overline{b_{Q-1}}\}$ the other vessels in the corresponding trees. *thresholdSupp* is the threshold used to suppress worse couples of angles.

Output: The optimal view map OV and consequently the optimal viewing angles $(\theta_j, \varphi_k)_{max}$.

```

//Initialization of Foreshortening map F, Bifurcation Opening map BO, Internal Overlap map
1: //Oint, External Overlap map Oext and Optimal View map OV for each possible gantry
//orientation with  $\theta_j \in [-90^\circ, 90^\circ]$  and  $\varphi_k \in [-30^\circ, 30^\circ]$  :
2:  $F[\theta_j][\varphi_k], BO[\theta_j][\varphi_k], O_{int}[\theta_j][\varphi_k], O_{ext}[\theta_j][\varphi_k], OV[\theta_j][\varphi_k] \leftarrow 0$ 
//Initialization of temporary Foreshortening, Bifurcation Opening, Internal and External
3: //overlap maps for each gantry orientation  $(\theta_j, \varphi_k)$  and each matched phases  $q$  with  $0 \leq q \leq$ 
// $Q - 1$ :
4:  $f[\theta_j][\varphi_k][q], bo[\theta_j][\varphi_k][q], o_{int}[\theta_j][\varphi_k][q], o_{ext}[\theta_j][\varphi_k][q] \leftarrow 0$ 
5: //Initialization of the mask used to select correct gantry orientation thanks to threshold:
6:  $mask[\theta_j][\varphi_k] \leftarrow 1$ 
7:
8: //First step: compute Foreshortening map F and Bifurcation Opening map BO.
9: for each matched selected segment  $l_q$  in  $L_{0,Q-1}$  do
10:    $length_{max}, angle_{max} \leftarrow 0$ 
11:   for each pair of angles  $(\theta_j, \varphi_k)$  with  $\theta_j \in [-90^\circ, 90^\circ]$  and  $\varphi_k \in [-30^\circ, 30^\circ]$  do
12:     if  $mask[\theta_j][\varphi_k] == 1$  then
13:       //Projection of  $l_q$  according to  $(\theta_j, \varphi_k)$  and the projection direction  $\vec{p}$ :
14:        $P_{\vec{p}, \theta_j, \varphi_k}(l_q) \leftarrow computeProjectionPoints(\vec{p}, \theta_j, \varphi_k, l_q)$ 
15:       //Compute the length of the projected bifurcation ( $length_q(\vec{p}, \theta_j, \varphi_k, l_q)$  in equation
//((7)):
16:        $f[\theta_j][\varphi_k][q] \leftarrow compute2DLength(P_{\vec{p}, \theta_j, \varphi_k}(l_q))$ 
17:       //Compute the bifurcation angle ( $angle_q(\vec{p}, \theta_j, \varphi_k, l_q)$  in equation (13)):
18:        $bo[\theta_j][\varphi_k][q] \leftarrow computeAngle(P_{\vec{p}, \theta_j, \varphi_k}(l_q))$ 
19:       //Update the maximum length and computed maximum angle:
20:        $length_{max} \leftarrow updateMaxLength(f[\theta_j][\varphi_k][q])$ 
21:        $angle_{max} \leftarrow updateMaxAngle(bo[\theta_j][\varphi_k][q])$ 
22:     end if
23:   end for each
24:   //Normalisation of each  $f[\theta_j][\varphi_k][q]$  by the maximum projected length and next each
// $bo[\theta_j][\varphi_k][q]$  by the maximum bifurcation angle:
25:   for each pair of angles  $(\theta_j, \varphi_k)$  where  $mask[\theta_j][\varphi_k] == 1$  do
26:      $f[\theta_j][\varphi_k][q] \leftarrow f[\theta_j][\varphi_k][q]/length_{max}$            // $f_q(\vec{p}, \theta_j, \varphi_k, l_q)$  in equation (7).

```

```

27:    $F[\theta_j][\varphi_k] \leftarrow \text{updateMinForeshorteningValue}(f[\theta_j][\varphi_k][q])$  //equation (8)
28:    $bo[\theta_j][\varphi_k][q] \leftarrow ob[\theta_j][\varphi_k][q]/\text{angle}_{max}$  //  $bo_q(\vec{p}, \theta_j, \varphi_k, l_q)$  in equation (13)
29:    $BO[\theta_j][\varphi_k] \leftarrow \text{updateMinOpeningBifValue}(bo[\theta_j][\varphi_k][q])$  //equation (14)
30:   if ( $f[\theta_j][\varphi_k][q]$  or  $bo[\theta_j][\varphi_k][q]$ ) <  $\text{thresholdSupp}$  then
31:      $mask[\theta_j][\varphi_k] \leftarrow 0$  //this gantry orientation is unselected: worse values for one
//phase.
32:      $F[\theta_j][\varphi_k], BO[\theta_j][\varphi_k] \leftarrow 0$ 
33:   end if
34: end for each
35: end for each
36:
37: //Second step: compute External  $O_{ext}$  and Internal Overlap  $O_{int}$  maps.
38: for each matched selected bifurcation  $b_q = (b_q^{father}, b_q^{child1}, b_q^{child2})$  in  $B_{0,Q-1}$  do
39:   for each pair of angles  $(\theta_j, \varphi_k)$  with  $\theta_j \in [-90^\circ, 90^\circ]$  and  $\varphi_k \in [-30^\circ, 30^\circ]$  do
40:     if  $mask[\theta_j][\varphi_k] == 1$  then
41:       //Projection of  $b_q$  and  $\overline{b_q}$  according to  $(\theta_j, \varphi_k)$  and the projection direction  $\vec{p}$ :
42:        $P_{\vec{p}, \theta_j, \varphi_k}(b_q) \leftarrow \text{computeThreeProjectionVolumes}(\vec{p}, \theta_j, \varphi_k, b_q)$ 
43:        $P_{\vec{p}, \theta_j, \varphi_k}(\overline{b_q}) \leftarrow \text{computeProjectionVolume}(\vec{p}, \theta_j, \varphi_k, \overline{b_q})$ 
44:       //Compute external overlap criteria: number of superimposed pixels between
// $P_{\vec{p}, \theta_j, \varphi_k}(b_q)$  and  $P_{\vec{p}, \theta_j, \varphi_k}(\overline{b_q})$  divided by the pixel number of  $P_{\vec{p}, \theta_j, \varphi_k}(b_q)$ :
45:        $o_{ext}[\theta_j][\varphi_k][q] \leftarrow \text{externalOverlap}(P_{\vec{p}, \theta_j, \varphi_k}(b_q), P_{\vec{p}, \theta_j, \varphi_k}(\overline{b_q}))$  //equation (9)
46:        $O_{ext}[\theta_j][\varphi_k] \leftarrow \text{updateMinExtOverlapValue}(o_{ext}[\theta_j][\varphi_k][q])$  //equation (10)
//Compute internal overlap criteria: number of superimposed pixels between the
47: //three projected bifurcation branches  $P_{\vec{p}, \theta_j, \varphi_k}(b_q^{father}), P_{\vec{p}, \theta_j, \varphi_k}(b_q^{child1})$  and
// $P_{\vec{p}, \theta_j, \varphi_k}(b_q^{child2})$  divided by the pixel number of  $P_{\vec{p}, \theta_j, \varphi_k}(b_q)$  :
49:        $o_{int}[\theta_j][\varphi_k][q] \leftarrow \text{internalOverlap}(P_{\vec{p}, \theta_j, \varphi_k}(b_q^{father}), P_{\vec{p}, \theta_j, \varphi_k}(b_q^{child1}),$ 
 $P_{\vec{p}, \theta_j, \varphi_k}(b_q^{child2}), P_{\vec{p}, \theta_j, \varphi_k}(\overline{b_q}))$  //equation (11)
50:        $O_{int}[\theta_j][\varphi_k] \leftarrow \text{updateMinIntOverlapValue}(o_{int}[\theta_j][\varphi_k][q])$  //equation (12)
51:       //Suppression of gantry orientation with worse values for one phase:
52:       if ( $o_{ext}[\theta_j][\varphi_k][q]$  or  $o_{int}[\theta_j][\varphi_k][q]$ ) <  $\text{thresholdSupp}$  then

```

```

53:          $mask[\theta_j][\varphi_k] \leftarrow 0$  //this gantry orientation is unselected: bad values for one
        //phase.
54:          $O_{ext}[\theta_j][\varphi_k], O_{int}[\theta_j][\varphi_k] \leftarrow 0$ 
55:     end if
56: end if
57: end for each
58: end for each
59:
60: //Third step: compute Optimal view map OV: linear combination of the four criteria.
61:  $OV_{max} \leftarrow 0$  //Initialization of the value of the optimal viewing angle.
62: for each pair of angles  $(\theta_j, \varphi_k)$  with  $\theta_j \in [-90^\circ, 90^\circ]$  and  $\varphi_k \in [-30^\circ, 30^\circ]$  do
63:     if  $mask[\theta_j][\varphi_k] == 1$  then
64:          $OV[\theta_j][\varphi_k] \leftarrow F[\theta_j][\varphi_k] * 0.25 + BO[\theta_j][\varphi_k] * 0.25 + O_{ext}[\theta_j][\varphi_k] * 0.25 +$ 
         $O_{int}[\theta_j][\varphi_k] * 0.25$ 
65:         if  $OV[\theta_j][\varphi_k] > OV_{max}$  then
66:              $OV_{max} \leftarrow OV[\theta_j][\varphi_k]$ 
67:              $(\theta_j, \varphi_k)_{max} \leftarrow (\theta_j, \varphi_k)$ 
68:         end if
69:     end if
70: end for each
71: return  $(OV, (\theta_j, \varphi_k)_{max})$ 

```

4.4.3. TIME COMPLEXITY ANALYSIS

In this subsection, the theoretical computational cost of our algorithm is analysed. For the two objects of interest (segment and bifurcation), the complexity of the algorithm depends on different parameters. Firstly, the size of the different maps must be taking into account: $nb_{(\theta_j, \varphi_k)}$ corresponds to the number of all evaluated gantry orientations (equation (6)). For each possible gantry orientation, the projection of the object (centerline or volume) is performed with the different criteria. The complexity of the centerline projection depends on the point number of this centerline such as for the foreshortening and bifurcation angle opening rates.

$$O_S = \left(nb_{(\theta_j, \varphi_k)} * (pt c_1 + c_2 + c_3 + c_4) \right) * Q \quad (15)$$

In the equation (15), pt represents the number of points in the centerline. c_1 corresponds to the cost of the projection of a point, c_2 the cost of a foreshortening rate, c_3 the cost of the projection of the volume and c_4 the cost of the external overlap rate. We consider that in this equation no threshold is used. Otherwise the external overlap is not computed $nb_{(\theta_j, \varphi_k)}$ times. For a bifurcation, we add the cost of the bifurcation opening angle and internal overlap rates.

4.5. CONCLUSION

La recherche d'incidence optimale est une étape clef pour le déroulement d'une angioplastie. Le cardiologue recherche un angle de projection où la zone sténosée est visualisée sans superposition et la plus déployée possible sur l'ensemble du cycle cardiaque. C'est pourquoi dans ce chapitre, nous avons proposé un nouvel algorithme pour déterminer l'incidence optimale d'un objet d'intérêt préalablement sélectionné. Nous proposons ainsi par le biais d'une carte optimale la meilleure ainsi que les 5% ou 2% meilleures orientations possibles du C-arm pour un objet sur l'ensemble du cycle cardiaque grâce à un ensemble de critères. Cet objet peut être un segment unique, un segment multiple (chemin) ou une bifurcation en fonction du type de sténose et du besoin du cardiologue (guidage du cathéter et/ou déploiement du stent). Quatre critères ont ainsi été développés : le raccourcissement et le chevauchement externe pour les segments et en plus le chevauchement interne et l'angle d'ouverture pour les bifurcations.

L'angle de vue optimale, calculé pour chaque phase du cycle cardiaque, prend en compte le mouvement des artères coronaires. Effectivement, l'algorithme de mise en correspondance proposé dans le chapitre précédent a été appliqué pour suivre l'objet d'intérêt tout le long du cycle cardiaque. De plus, grâce au choix pertinent du minimum des cartes des phases pour calculer la carte globale correspondant au cycle cardiaque entier, les variations topologiques des arbres de la séquence sont également prises en compte.

Le prochain chapitre propose les évaluations des différentes méthodes de mise en correspondance et de recherche d'incidence optimale à l'aide de données réelles et simulées.

CHAPITRE 5

RESULTATS

SOMMAIRE

5.1.	Pair-wise matching of successive trees	115
5.1.1.	Evaluation of the hierarchical tree method	115
5.1.2.	Evaluation of the association graph method	119
5.1.2.1.	Unary constraints	119
5.1.2.2.	Binary constraints.....	121
5.1.2.3.	Pair-wise matching algorithm.....	125
5.1.3.	Comparison of the two methods.....	126
5.1.4.	Matching evaluation with artificial nodes.....	127
5.1.5.	Result synthesis	131
5.2.	Mosaic-based matching of the coronary sequence	132
5.2.1.	Evaluation of real sequence	134
5.2.2.	Evaluation from simulated sequences	136
5.2.3.	Result synthesis	137
5.3.	Optimal viewing angle results	140
5.3.1.	Result from segment of interest	140
5.3.2.	Results for bifurcation of interest	142
5.3.3.	Results for multiple segments of interest	145
5.3.4.	Result synthesis	150
5.4.	Conclusion	151

Dans ce chapitre nous évaluons et comparons les différentes méthodes présentées dans les chapitres précédents. La séquence d'arbres coronaires 3D d'un patient décrite dans le chapitre 1 nous sert de référence pour les différents tests. Nos différents algorithmes et les métriques ont été évalués sur celle-ci. Nos méthodes sont également évaluées à partir de données simulées en prenant la séquence de données réelles comme référence (Annexe A). Comme pour les chapitre 3 et 4, les résultats de la mise en correspondance et de la recherche d'incidence optimale sont issus d'articles publiés [FeNT15, FNLT15, FSNB13] ou soumis [FeNT16b, FeNT16a].

La première partie se focalise uniquement sur les résultats liés à la mise en correspondance de deux arbres coronaires correspondant à deux phases consécutives (section 3.1.3). Les différents algorithmes proposés dans le chapitre 3 ainsi que les mesures de similarité associées sont évalués. L'apport de notre nouvelle méthode d'appariement de deux arbres vasculaires avec insertion de nœuds artificiels, les nouvelles mesures de similarité et l'étape de correction hiérarchique est ainsi mis en avant.

La deuxième partie évalue quant à elle l'algorithme proposé pour la mise en correspondance de l'ensemble de la séquence d'arbres coronaires basé sur la construction d'un arbre mosaïque (section 3.2). Pour rappel, ce nouvel algorithme prend en compte les différences topologiques de tous les arbres vasculaires de la séquence. Il s'autocorrige pendant le processus de construction de l'arbre mosaïque afin de résoudre les conflits hiérarchiques liés à l'insertion de nœuds artificiels et aux informations topologiques supplémentaires obtenues au fur et à mesure du traitement de la séquence.

La dernière partie du chapitre se focalise sur les résultats concernant la détermination de l'angle de vue optimale. Les algorithmes proposés dans le chapitre 4 permettent d'obtenir l'incidence optimale pour un segment unique, une bifurcation ou un segment multiple donné pour tout le cycle cardiaque tout en prenant en compte les différences topologiques entre les arbres de la séquence.

Tous ces algorithmes ont été développés dans le langage *C++* avec l'aide des bibliothèques *Qt*⁴ et *VTK*⁵ pour l'interface graphique (GUI) et *Boost*⁶ pour créer et gérer les structures d'arbre et leur traitement parallèle (threads). L'annexe A décrit brièvement l'interface graphique associée à notre application d'aide au diagnostic et de planning développée au cours de cette thèse. Elle permet entre autres de charger des données, créer des données simulées à partir de données réelles, utiliser les différents algorithmes de mise en correspondance et de choisir leurs mesures de similarités associées et de calculer enfin l'angle de vue optimale. Pour information, tous les tests de cette thèse ont été

⁴ <http://www.qt.io/>

⁵ <http://www.vtk.org/>

⁶ <http://www.boost.org/>

réalisés sur un ordinateur sous Windows 7 x64 bits comme système d'exploitation et équipé d'un processeur de 2,26 GHz Intel Xeon CPU E5520 et 8 Go de RAM.

5.1. PAIR-WISE MATCHING OF SUCCESSIVE TREES

In this first section, the different matching algorithms of two successive phases proposed in the chapter 3 are evaluated and compared. A manual classification is performed for each node to evaluate the algorithm and the node-to-node constraint efficiency. In our case, there are 6 possible classifications for one given node:

- correctly matched nodes (CM),
- correct nodes without assignment e.g. a leaf node of a spurious branch (CMWA),
- correctly matched bifurcation nodes with artificial nodes (CBAN),
- correctly matched leaf nodes with artificial nodes (CLAN),
- unmatched nodes (UM), and
- incorrectly matched nodes (IM).

This chapter presents a comparison of the two adapted state-of-the-art inexact tree matching algorithms [LKZP07, MKSZ09, TMPH05] using the node-to-node constraints described in sections 3.1.1 and 3.1.3. Beforehand, we explain in sections 5 and 5.1.2 how these two algorithms are evaluated and the determination of the best constraint combination. Next in section 5.1.4, the contribution of the new pair-wise matching method with the insertion of artificial nodes based on new node-to-node constraints is shown and compared to the others algorithms.

5.1.1. EVALUATION OF THE HIERARCHICAL TREE METHOD

In the section 3.1.1, describing Lohe et al.'s method [LKZP07], two sets of metrics have been highlighting. We evaluate some metrics adapted to our coronary problem for the inexact tree matching proposed by Lohe et al. [LKZP07] and new metrics based on the Dynamic Time Warping's algorithm [KePa01, RaKe05] that we proposed in the section 3.1.3 to obtain the combination which gives the best matching result.

In the evaluation, we analyze first two consistency constraints cc^λ which permit to select potential nodes $v_i^{\alpha_i} \in V_i$ and $v_{i+1}^{\alpha_{i+1}} \in V_{i+1}$ and then to add these nodes $v_{TS_{i,i+1}}^{\alpha_i, \alpha_{i+1}} = (v_i^{\alpha_i}, v_{i+1}^{\alpha_{i+1}})$ in the search tree $TS_{i,i+1}$:

- cc^1 : the difference between the two path lengths $p_i(r_i, v_i^{\alpha_i}) \in T_i$ and $p_{i+1}(r_{i+1}, v_{i+1}^{\alpha_{i+1}}) \in T_{i+1}$,
- cc^2 : The Euclidean distance between the couple of potential node coordinates $v_i^{\alpha_i}$ and $v_{i+1}^{\alpha_{i+1}}$.

Five similarity measures sm^μ used to determine the matching quality of the potential nodes $v_{TS_{i,i+1}}^{\alpha_i, \alpha_{i+1}} = (v_i^{\alpha_i}, v_{i+1}^{\alpha_{i+1}})$ in the search tree are also evaluated:

- sm^1 : the comparison of the two path lengths $p_i(r_i, v_i^{\alpha_i}) \in T_i$ and $p_{i+1}(r_{i+1}, v_{i+1}^{\alpha_{i+1}}) \in T_{i+1}$,
- sm^2 : the nearness of these paths with the DTW's algorithm measuring the Euclidean distance of node coordinates,
- sm^3 : the mean path curvature comparison.
- sm^4 : the local comparison of the path curvature with the DTW's algorithm based on the Euclidean distance of tangent in each centerline point,
- sm^5 : the local comparison of the path curvature with the DTW's algorithm based on the Euclidean distance of direction.

To determine the combination of metric set which gives the best results, several evaluations have been realized with different threshold values and weight factors from the nine matchings between successive trees of the real coronary tree sequence $S_{0,9}$.

First, each possible combination of a consistency constraint cc^λ and a similarity measure sm^μ are tested. Thresholds of consistency constraints $s(cc^\lambda)$ and on similarity measures $s(sm^\mu)$ are empirically chosen and their values are kept identical for each test (i.e. the nine matchings). The Table 1 shows the best matching results obtained for each evaluated combination. We noted that the best threshold values of the two consistency constraints are the same for all similarity measures such as the best threshold values of the five tested similarity measures. Moreover, each best proposed metric configuration has incorrectly matched nodes (IM). The second consistency constraint cc^2 combined with each similarity measure gives better matching results than the first constraint cc^1 . Better node classification is obtained with the second sm^2 , the third sm^3 and the fifth sm^5 similarity measures.

79.6% of nodes are correctly matched (i.e. classified) and their execution time is really fast (de 2 à 311 ms).

Table 1: Comparison of different consistency constraints cc^λ and similarity measures sm^μ for the nine matchings with different threshold values $s(cc^\lambda)$ and $s(sm^\mu)$. CM is the number of correctly matched nodes, CMWA is the number of correctly matched nodes without assignment, IM corresponds to the incorrectly matched nodes and UM to the unmatched nodes. The distribution percentage of each classification is given in brackets. The last column is the execution time in ms for the nine matchings.

cc^λ	$s(cc^\lambda)$	sm^μ	$s(sm^\mu)$	CM (%)	CMWA (%)	IM (%)	UM (%)	Runtime in ms
cc^1	7.5	sm^1	1.5	120 (61.2)	32 (16.3)	12 (6.2)	32 (16.3)	1
cc^1	7.5	sm^2	5	120 (61.2)	30 (15.3)	18 (9.2)	28 (14.3)	269
cc^1	7.5	sm^3	1.3	118 (60.2)	29 (14.8)	22 (11.2)	27 (13.8)	28
cc^1	7.5	sm^4	15	116 (59.2)	30 (15.3)	22 (11.2)	28 (14.3)	1147
cc^1	7.5	sm^5	20	118 (60.2)	30 (15.3)	20 (10.2)	28 (14.3)	1140
cc^2	10	sm^1	1.5	122 (62.2)	32 (16.3)	12 (6.12)	30 (15.3)	2
cc^2	10	sm^2	5	124 (63.3)	32 (16.3)	10 (5.1)	30 (15.3)	68
cc^2	10	sm^3	1.3	124 (63.3)	32 (16.3)	10 (5.1)	30 (15.3)	2
cc^2	10	sm^4	15	122 (62.2)	32 (16.3)	12 (6.2)	30 (15.3)	344
cc^2	10	sm^5	20	124 (63.3)	32 (16.3)	10 (5.1)	30 (15.3)	311

The Table 2 shows the matching result for each similarity measure and the combination of the two consistency constraints cc^1 and cc^2 . The better metric thresholds described previously in Table 1 are used to determine if this combination of consistency constraints give better matching results. The results are not improved: the number of correctly matched nodes is lower than the result obtained with only the second consistency constraint. However, for the second and fifth similarity measures, the number of incorrect nodes decreases slightly.

Table 2: Matching results for different similarity measures sm^μ with the combination of the two consistency constraints cc^1 and cc^2 and their threshold $s(cc^1) = 7.5$ and $s(cc^2) = 10$ obtain in relation to the Table 1. CM is the number of correctly matched nodes, CMWA is the number of correctly matched nodes without assignment, IM corresponds to the incorrectly matched nodes and UM to the unmatched nodes.

sm^μ	$s(sm^\mu)$	CM (%)	CMWA (%)	IM (%)	UM (%)	Runtime in ms
sm^1	1.5	116 (59.2)	32 (16.3)	12 (6.2)	36 (18.3)	17
sm^2	5	120 (61.2)	32 (16.3)	8 (4.2)	36 (18.3)	107
sm^3	1.3	118 (60.2)	32 (16.3)	10 (5.1)	36 (18.3)	15
sm^4	15	116 (59.2)	32 (16.3)	12 (6.2)	36 (18.3)	322
sm^5	20	118 (60.2)	32 (16.3)	10 (5.1)	36 (18.3)	314

The Table 3 shows the better matching results with the combination of node-to-node similarity measures. With the previous results, we choose to evaluate these different combinations with only

the second consistency constraint cc^2 which gives better node classification. With this real data, these combinations didn't perform better. However, similar results are obtained with metrics combinations such as sm^2 and sm^4 which provide additional information with an assessment both in the nearness and curvature of the paths.

Table 3 : Matching results for different combinations of similarity measures sm^u and their associated thresholds $s(sm^u)$ obtained with the consistency constraint cc^2 and its associated threshold $s(cc^2) = 10$. This constraint gives better result according to the Table 1 and Table 2. CM is the number of correctly matched nodes, CMWA is the number of correctly matched nodes without assignment, IM corresponds to the incorrectly matched nodes and UM to the unmatched nodes.

Configuration	CM (%)	CMWA (%)	IM (%)	UM (%)
$s(sm^1) = 1.5 ; s(sm^2) = 5$	122 (62.2)	32 (16.3)	12 (6.2)	30 (15.3)
$s(sm^1) = 1.5 ; s(sm^3) = 1.3$	122 (62.2)	32 (16.3)	12 (6.2)	30 (15.3)
$s(sm^1) = 1.5 ; s(sm^4) = 15$	122 (62.2)	32 (16.3)	12 (6.2)	30 (15.3)
$s(sm^1) = 1.5 ; s(sm^5) = 20$	122 (62.2)	32 (16.3)	12 (6.2)	30 (15.3)
$s(sm^2) = 5 ; s(sm^3) = 1.3$	124 (63.3)	32 (16.3)	10 (5.1)	30 (15.3)
$s(sm^2) = 5 ; s(sm^4) = 15$	124 (63.3)	32 (16.3)	10 (5.1)	30 (15.3)
$s(sm^2) = 5 ; s(sm^5) = 20$	124 (63.3)	32 (16.3)	10 (5.1)	30 (15.3)
$s(sm^3) = 1.3 ; s(sm^4) = 15$	124 (63.3)	32 (16.3)	10 (5.1)	30 (15.3)
$s(sm^3) = 1.3 ; s(sm^5) = 20$	124 (63.3)	32 (16.3)	10 (5.1)	30 (15.3)
$s(sm^4) = 15 ; s(sm^5) = 20$	122 (62.2)	32 (16.3)	12 (6.2)	30 (15.3)
$s(sm^2) = 5 ; s(sm^3) = 1.3 ; s(sm^5) = 20$	124 (63.3)	32 (16.3)	12 (6.2)	30 (15.3)
$s(sm^2) = 5 ; s(sm^4) = 15 ; s(sm^5) = 20$	124 (63.3)	32 (16.3)	12 (6.2)	30 (15.3)

We evaluate the matching algorithm proposed by Lohe et al. [LKZP07] with different metrics adapted to non-isomorph vascular data. The better results for the nine matchings are obtained with the Euclidean distance of consistency constraint and with similarity measures using the nearness and curvature between the paths. Principally, this metrics are proposed in the section 3.1.3. However, the number of incorrect matched nodes (IM) and unmatched nodes (UM) are still high (about 20%).

We will now evaluate the algorithm based on the association graph building derived from Metzzen et al. [MKSZ09] and Tschirren et al. [TMPH05] methods. Different kind of node-to-node similarity measures (called unary and binary constraints) are evaluated and compared to them proposed in the section 3.1.3.

5.1.2. EVALUATION OF THE ASSOCIATION GRAPH METHOD

The matching algorithm based on the association graph building $G_{i,i+1}$ and the maximum clique search C_{max} used two sets of metrics as explained previously in the section 3.1.2. The efficiency of the unary and binary constraints must be shown separately in order to obtain the best matching of two trees T_i and T_{i+1} .

5.1.2.1. UNARY CONSTRAINTS

First, the efficiency of the unary constraints cu^Y is analyzed. With this step, we only want to select all potential assignments $v_{i,i+1}^{\alpha_i, \alpha_{i+1}} = (v_i^{\alpha_i}, v_{i+1}^{\alpha_{i+1}})$ with $v_i^{\alpha_i} \in V_i$ and $v_{i+1}^{\alpha_{i+1}} \in V_{i+1}$ which must be present in the maximum clique C_{max} . These nodes, representing the best matching M (ground truth), must be in the association graph. Four unary constraints are evaluated:

- cu^1 : the comparison of the spatial coordinates of the two potential nodes $v_i^{\alpha_i} \in V_i$ and $v_{i+1}^{\alpha_{i+1}} \in V_{i+1}$,
- cu^2 : the comparison of path lengths $p_i(r_i, v_i^{\alpha_i})$ and $p_{i+1}(r_{i+1}, v_{i+1}^{\alpha_{i+1}})$,
- cu^3 : the minimum area between the two normalized polylines defined by the two paths p_i and p_{i+1} ,
- cu^4 : the comparison of the global curvature between the two paths p_i and p_{i+1} .

Moreover, their selectivity parameters (threshold associated to the constraints) are empirically chosen and identical for each matching test. To determine the best configuration, we check if all node pairs which must be matched are selected (without the pair of root nodes which is automatically matched). The number of all selected nodes in the association graph $G_{i,i+1}$ (i.e. the order of the association graph for the nine matchings) is also examined. More the node pair selection is significant, more the search space of the next steps is reduced decreasing consequently the execution time to the determination of the best matching. By the way, the execution time is also an important parameter and should be fast for this first step of the matching algorithm.

In Table 4, we compare these four unary constraints with three different values of associated selectivity parameter among all tested values. The chosen threshold values were the most representatives and also quite different. For each matching test, we have:

- The number of node pairs which must be in the association graph. In the ground truth, we have 53 node pairs that must be matched. More this number is near 53, better is the unary constraint.
- The number of node pairs in the association graph evaluating the selection criterion of the constraint. The percentage corresponding to the number of all possible pairs of nodes, is also given. More this value is low, more the selectivity is high.
- The execution time of the constraint in ms.

The best results was obtained with the first constraint cu^1 : all pairs of nodes which must be matched are selected and a significant selection is performed (i.e. a reduction of about 92.1% of the association graph order). Besides, the execution time is really fast contrary to the third unary constraints cu^3 using area between normalized polylines.

Table 4 : Comparison of different unary constraint results cu^y with different associated selectivity parameter values $s(cu^y)$ for the nine matchings.

cu^y	$s(cu^y)$	Number of pairs of matched nodes present in $G_{i,i+1}$ (%)	Number of potential nodes selected in $G_{i,i+1}$ (%)	Runtime (ms)
cu^1	15	53(100.0)	95(11)	2
cu^1	10	53 (100.0)	68 (7.9)	2
cu^1	9	52 (98.1)	65 (7.5)	2
cu^2	1.5	52 (98.1)	432 (49.9)	3
cu^2	1.3	50 (94.3)	298 (34.4)	3
cu^2	1.1	44 (83.0)	117 (13.5)	3
cu^3	0.3	52 (98.1)	807 (93.3)	72255
cu^3	0.1	49 (92.4)	415 (48)	70969
cu^3	0.05	43 (81.3)	108 (12.5)	68682
cu^4	1.5	51 (96.2)	823 (95.1)	26
cu^4	1.3	51 (96.2)	727 (84)	32
cu^4	1.1	45(84.9)	414 (47.9)	33

We have evaluated each unary constraint separately and found their best associated selectivity parameter value. Now, we will combine the different constraints to determine if the matching efficiency can be improved. We test each possible combination between two and four constraints with different weight factor.

Table 5 shows the best unary constraint combination obtained and their associated weight factor. Each combination in this table increases the selection of matched nodes. The number of selected potential nodes (about 9.2%) is slightly higher but closed to the better results in Table 4 (7.9% for the constraint cu^1 with the value 10 as selectivity parameter).

Table 5 : Example of the better combination of unary constraints cu^y for the nine matchings according to a weight factor.

Combination of unary constraints	Number of pairs of matched nodes present in $G_{i,i+1}$ (%)	Number of potential nodes selected in $G_{i,i+1}$ (%)	Runtime (ms)
$cu^1 * 0.75 + cu^2 * 0.25$	53 (100.0)	80 (9.2)	3
$cu^1 * 0.75 + cu^3 * 0.25$	53 (100.0)	81 (9.4)	69451
$cu^1 * 0.75 + cu^4 * 0.25$	53 (100.0)	82 (9.5)	3
$cu^1 * 0.33 + cu^2 * 0.33 + cu^3 * 0.33$	53 (100.0)	260 (30.1)	69156
$cu^1 * 0.33 + cu^2 * 0.33 + cu^4 * 0.33$	53 (100.0)	276 (31.9)	3
$cu^1 * 0.25 + cu^2 * 0.25 + cu^3 * 0.25 + cu^4 * 0.25$	53 (100.0)	435 (50.3)	69228

The unary constraint combination don't give better result compared to the single use of the first unary constraint cu^1 with $s(cu^1) = 10$. This unary constraint corresponds to our different criteria of our matching algorithm (section 3.1.3).

5.1.2.2. BINARY CONSTRAINTS

In this section, the efficiency of the proposed binary constraints cb^δ in section 3.1.2 and 3.1.3 is analyzed. To compare each binary constraint, all possible pairs of nodes are included in the association graph. The selection of nodes performed during the first step of the algorithm (i.e. unary constraints) is not taken into account.

Let be $v_{i,i+1}^{\alpha_i, \alpha_{i+1}} = (v_i^{\alpha_i}, v_{i+1}^{\alpha_{i+1}}) \in V_{G_{i,i+1}}$ and $v_{i,i+1}^{\beta_i, \beta_{i+1}} = (v_i^{\beta_i}, v_{i+1}^{\beta_{i+1}}) \in V_{G_{i,i+1}}$ two potential assignments in the association graph. To remain, the binary constraint is used to determine if two paths $p_i(v_i^{\alpha_i}, v_i^{\beta_i}) \in T_i$ and $p_{i+1}(v_{i+1}^{\alpha_{i+1}}, v_{i+1}^{\beta_{i+1}}) \in T_{i+1}$ can be considered as similar to insert next an edge between this two nodes $v_{i,i+1}^{\alpha_i, \alpha_{i+1}}$ and $v_{i,i+1}^{\beta_i, \beta_{i+1}}$ in the association graph $G_{i,i+1}$. We choose to test six binary constraints as proposed by Metzen et al. [MKSZ09], Tschirren et al. [TMPH05] and also from the section 3.1.3:

- cb^1 : the comparison of path lengths $p_i(v_i^{\alpha_i}, v_i^{\beta_i}) \in T_i$ and $p_{i+1}(v_{i+1}^{\alpha_{i+1}}, v_{i+1}^{\beta_{i+1}}) \in T_{i+1}$,
- cb^2 : the curvature comparison between the two paths p_i and p_{i+1} ,
- cb^3 : the minimum area between the two normalized polylines derived by the two paths p_i and p_{i+1} ,

- cb^4 : the minimum of accumulated distance between the two paths p_i and p_{i+1} derived from the DTW's algorithm,
- cb^5 : the local comparison of the path curvatures with the DTW's algorithm based on the tangent,
- cb^6 : the local comparison of the path curvatures based on the direction with the DTW's algorithm.

In the same way as unary constraints, each binary constraint is evaluated with different values of the associated selectivity parameters (empirically chosen). Significant results with the sum of matched node number for the nine matchings are shown in Figure 40. For a matching test, the node classification results are shown with their number and percentage. For this step, four classifications are considered: correctly matched nodes (CM), correct node without assignment (CMWA), incorrectly matched nodes (IM) and unmatched nodes (UM). Best results are obtained with the fourth constraint cb^4 , i.e. the minimum distance between paths. This constraint has a reasonable execution time (about 90 sec.), even if it's not the faster (see Table 6). However, this runtime doesn't take into account the first step with the selection of potential matched nodes thanks to the unary constraints. Indeed, with a lower node number, this step should decrease this time value (see section 5.1.2.3). The third binary constraint cb^3 with $s(cb^3) = 0.1$ have only six incorrect matched nodes (IM). However, the number of correctly matched nodes (CM) is lower than for the fourth constraint and more than 20% of nodes are unmatched (UM). In addition, the execution time is really long.

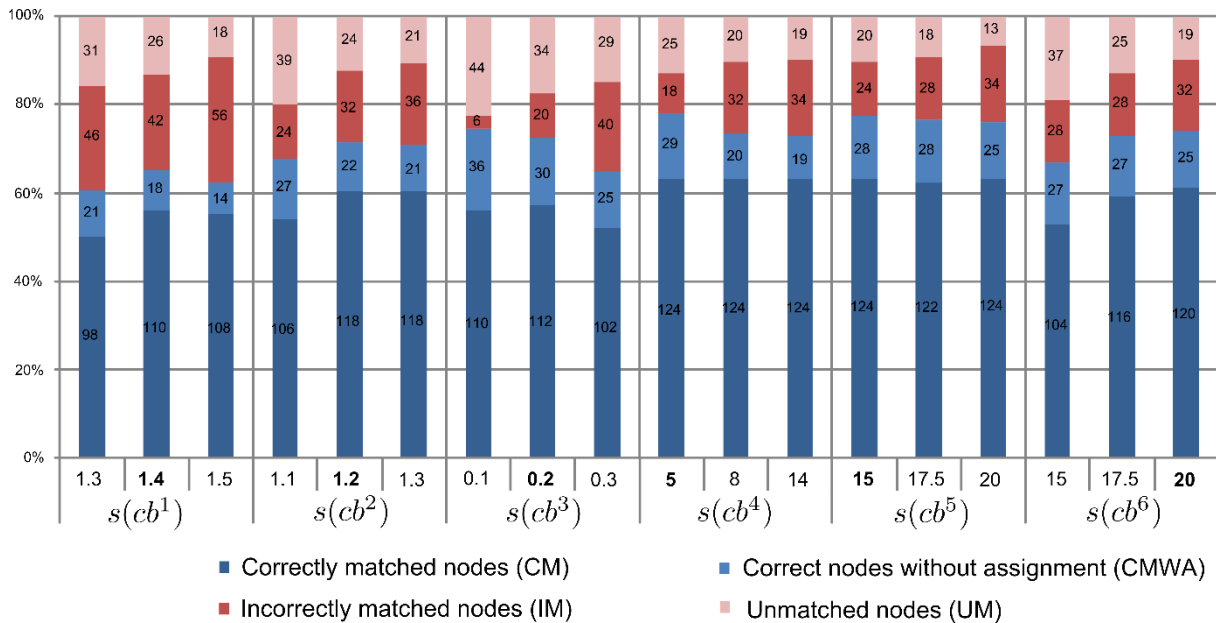


Figure 40 : Comparison of binary constraints cb^δ for the nine matching tests with different values of selectivity parameters $s(cb^\delta)$. The better selectivity parameter for each constraint are in bold.

Table 6: Runtime in ms of binary constraints cb^δ with different selectivity parameter values $s(cb^\delta)$ for the nine matchings.

cb^δ	$s(cb^\delta)$	Runtime constraint (ms)	Runtime clique maximum (ms)
cb^1	1,3	869	183
cb^1	1,4	853	307
cb^1	1,5	1047	634
cb^2	1,1	825	105
cb^2	1,2	818	237
cb^2	1,3	790	465
cb^3	0,1	25343295	47
cb^3	0,2	24981644	136
cb^3	0,3	25460346	1076
cb^4	5	90953	27
cb^4	8	88104	37
cb^4	14	89225	84
cb^5	15	93541	8
cb^5	17.5	98950	14
cb^5	20	96013	15
cb^6	15	92100	6
cb^6	17.5	95668	10
cb^6	20	92542	13

After testing each binary constraint separately, we want to determine if a combination of them can improve matching results, i.e. increasing the number of all correctly matched nodes (CM + CMWA) while decreasing the number of incorrect matched nodes (IM + UM). For each binary constraint, the selectivity parameter is determined according to the best matching results described in Figure 40. The correctly matched node number have to be higher than the other tests and next the incorrectly matched node number must be the lowest. However, best combination may be found without these best similarity parameters previously obtained. Only cb^5 and cb^6 represent the same measure unit (mean angle), that's why we choose to test also the same threshold value for them. To combine these binary constraints, several tests are realized with different weights. The Figure 41 shows the best binary constraint combination obtained with their associated weight factor. 12 representative tests have been selected:

- **test 1** : $cb^1 + cb^2$ with $s(cb^1) = 1.4$; $w(cb^1) = 0.75$ and $s(cb^2) = 1.2$; $w(cb^2) = 0.25$,
- **test 2** : $cb^1 + cb^4$ with $s(cb^1) = 1.4$; $w(cb^1) = 0.25$ and $s(cb^4) = 5$; $w(cb^4) = 0.75$,
- **test 3** : $cb^2 + cb^4$ with $s(cb^2) = 1.2$; $w(cb^2) = 0.25$ and $s(cb^4) = 5$; $w(cb^4) = 0.75$,
- **test 4** : $cb^3 + cb^4$ with $s(cb^3) = 0.2$; $w(cb^3) = 0.25$ and $s(cb^4) = 5$; $w(cb^4) = 0.75$,
- **test 5** : $cb^4 + cb^5$ with $s(cb^4) = 5$; $w(cb^4) = 0.5$ and $s(cb^5) = 15$; $w(cb^5) = 0.5$,
- **test 6** : $cb^4 + cb^6$ with $s(cb^4) = 5$; $w(cb^4) = 0.5$ and $s(cb^6) = 20$; $w(cb^6) = 0.5$,
- **test 7** : $cb^4 + cb^6$ with $s(cb^4) = 5$; $w(cb^4) = 0.5$ and $s(cb^6) = 15$; $w(cb^6) = 0.5$,

- **test 8** : $cb^1 + cb^2 + cb^4$ with $s(cb^1) = 1.4$; $w(cb^1) = 1/3$ and $s(cb^2) = 1.2$; $w(cb^2) = 1/3$ and $s(cb^4) = 5$; $w(cb^4) = 1/3$,
- **test 9** : $cb^4 + cb^5 + cb^6$ with $s(cb^4) = 5$; $w(cb^4) = 1/3$ and $s(cb^5) = 15$; $w(cb^5) = 1/3$ and $s(cb^6) = 20$; $w(cb^6) = 1/3$,
- **test 10** : $cb^4 + cb^5 + cb^6$ with $s(cb^4) = 5$; $w(cb^4) = 1/3$ and $s(cb^5) = 15$; $w(cb^5) = 1/3$ and $s(cb^6) = 15$; $w(cb^6) = 1/3$,
- **test 11** : $cb^1 + cb^2 + cb^3 + cb^4$ with $s(cb^1) = 1.4$; $w(cb^1) = 0.25$ and $s(cb^2) = 1.2$; $w(cb^2) = 0.25$ and $s(cb^3) = 0.2$; $w(cb^3) = 0.25$ and $s(cb^4) = 5$; $w(cb^4) = 0.25$,
- **test 12** : $cb^1 + cb^2 + cb^4 + cb^5 + cb^6$ with $s(cb^1) = 1.4$; $w(cb^1) = 0.2$ and $s(cb^2) = 1.2$; $w(cb^2) = 0.2$ and $s(cb^4) = 5$; $w(cb^4) = 0.2$ and $s(cb^5) = 15$; $w(cb^5) = 0.2$ and $s(cb^6) = 15$; $w(cb^6) = 0.2$.

As shown in the Figure 41, without taking into account the first step of the algorithm (selection of potential matched nodes thanks to unary constraints), the matching results are not better than those for the fourth binary constraint cb^4 alone and have consequently a higher runtime. Only the combination of cb^4 and cb^6 gives the same result.

We can see that the combination of the three binary constraints cb^4 , cb^5 and cb^6 , which are proposed in the section 3.1.3, obtains the same correctly matched node number. These three constraints give correct matching results without the first selected step. In the next section, we will compare the fourth constraint with the combination of the three last binary constraints during the complete matching algorithm (with the unary constraint).

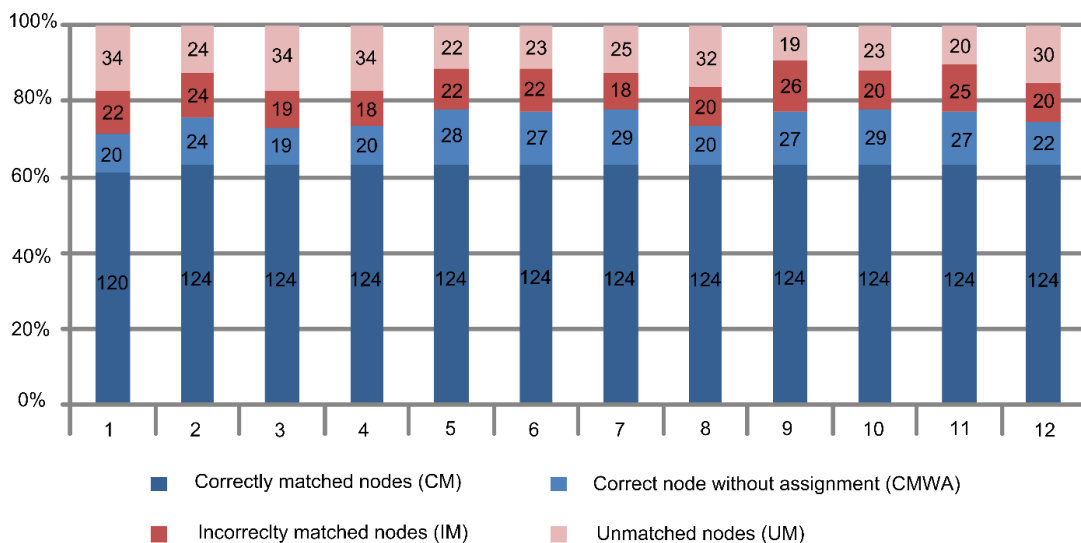


Figure 41 : Different combinations of binary constraints cb^δ for the nine matchings.

5.1.2.3. PAIR-WISE MATCHING ALGORITHM

We have evaluated the two kind of constraints separately. In this subsection, we applied the complete pair-wise matching algorithm with the better constraints found previously. The unary constraint is based on the Euclidean distance of the node coordinates (cu^1) with a selectivity parameters equal to 10 (section 5.1.1). In the evaluation of binary constraints (section 5.1.2.2), we found that the fourth constraint (nearness of the path based on the Dynamic Time Warping) give slightly better result than the others. In the section 3.1.3, we proposed another similarity measure also based on the DTW (coordinates, tangent and direction). Previously, we have shown that the combination of these three constraints gives good results without the selection step using unary constraints. We choose to compare the pair-wise matching algorithm with these two propositions:

- **algorithm A1** with the unary constraint cu^1 and $s(cu^1) = 10$ and the binary constraint cb^4 with $s(cb^4) = 5$,
- **algorithm A2** with the unary constraint cu^1 and $s(cu^1) = 10$ and the combination of three binary constraints $s(cb^4) = 5$, $s(cb^5) = 15$ and $s(cb^6) = 15$.

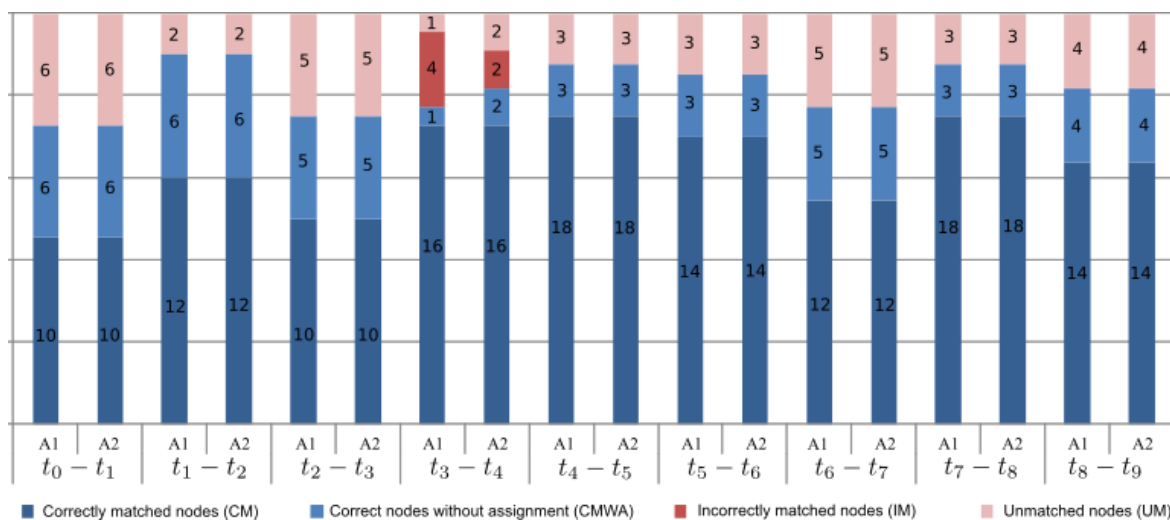


Figure 42: Comparison of constraints for the association graph algorithm. A1 corresponds to the algorithm with $s(cu^1) = 10$ and $s(cb^4) = 5$ for the nine matchings. A2 is based on the combination of the three binary constraints with $s(cu^1) = 10$ and $s(cb^4) = 5$, $s(cb^5) = 15$ and $s(cb^6) = 15$ for the nine matchings.

The Figure 42 shows the matching results for these two algorithms A1 and A2 for the nine matching tests. The number of correctly matched nodes (CM) is exactly the same and only one test

differs (phases t_3 and t_4). The number of incorrectly matched nodes (IM) decreases slightly with the combination of the three binary constraints while previously the single binary constraint cb^4 given better result in the section 5.1.2.2. Thus, the first selection step with the unary constraint is really important.

5.1.3. COMPARISON OF THE TWO METHODS

As shown in previous section, best results for the two coronary tree matching algorithms are obtained with two kinds of metrics: first, the Euclidean distance between the tree node coordinates to select potential assignments and second, the three new criteria between path points (the coordinates, the tangent and the local direction) from which is computed the Euclidean distance by the DTW's algorithm to measure the path similarity.

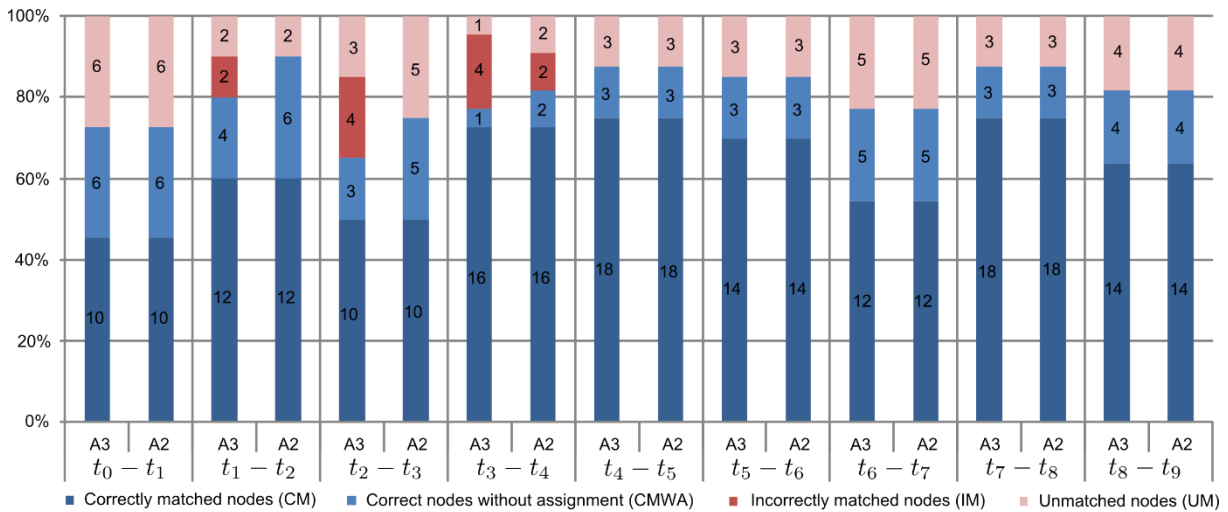


Figure 43: Comparison between Lohe et al.'s algorithm (A3) with $s(cc^2) = 10$ and $s(sm^2) = 5$, $s(sm^4) = 15$ and $s(sm^5) = 15$ and the algorithm based on Metzzen et al. and Tschirren et al. (A2) with $s(cu^1) = 10$ and $s(cb^4) = 5$, $s(cb^5) = 15$ and $s(cb^6) = 15$ for the nine matchings.

The Figure 43 explains the nine matching results obtained with Lohe et al.'s algorithm [LKZP07] and the association graph approach [MKSZ09, TMPH05] adapted with the two previous kind of metrics. The selectivity parameter's value is chosen according to the section 5 and 5.1.2. For the first **algorithm A3**, the best configuration is obtained with $s(cc^2) = 10$ for the consistency constraint and $s(sm^2) = 5$, $s(sm^4) = 15$ and $s(sm^5) = 15$ for the similarity measure. For the second **algorithm A2**, we take

$s(cu^1) = 10$ for the unary constraint and $s(cb^4) = 5$, $s(cb^5) = 15$ and $s(cb^6) = 15$ for binary constraints. The number of correctly matched nodes is exactly the same and the number of correct nodes without assignment, practically the same. This number is slightly lower than thus of Lohe et al.'s algorithm [LKZP07]. The major difference between these two best combinations is the number of false nodes which is higher for the first algorithm. Even if its execution time is a bit faster than the association graph's algorithm (see Table 7), we prioritize the quality of matching result with a low number of incorrectly matched nodes. This result is consistent with the paper of Metzen et al. [MKSZ09]. Indeed, with the same database, they have better results (i.e. a great number of correctly matched nodes) than the algorithm proposed by Lohe et al. [LKZP07].

Table 7: Comparison of the execution time (in ms) for four algorithms.

Algorithm	Phases $t_0 - t_1$	Phases $t_1 - t_2$	Phases $t_2 - t_3$	Phases $t_3 - t_4$	Phases $t_4 - t_5$	Phases $t_5 - t_6$	Phases $t_6 - t_7$	Phases $t_7 - t_8$	Phases $t_8 - t_9$
Lohe et al. [LKZP07] (A3)	20	61	36	197	157	23	19	88	48
Association graph [MKSZ09, TMPH05] (A2)	46	93	91	312	243	92	33	246	135
Association graph with artificial nodes (A4/A5)	147	217	236	1003	727	342	165	433	374

5.1.4. MATCHING EVALUATION WITH ARTIFICIAL NODES

The association graph approach based on Metzen et al.'s and Tschirren et al.'s algorithm [MKSZ09, TMPH05] with the best metric combination shown in section 5.1.2, gives better matching result than the other algorithm [LKZP07]. Thus, we choose to evaluate the step of artificial node insertion with this configuration. The Figure 44 shows the comparison of node classification between the algorithm without artificial node (A2) and with artificial nodes (A4 and A5). When the artificial nodes are added in trees, several node-to-node constraints are computed for the DTW's algorithm to determine if two branches are similar (section 3.1.3.2). These constraints are the same as the binary

constraints. In the **algorithm A4**, we choose to use only one constraint: the Euclidean distance of node coordinates cb^4 , with the same selectivity parameter $s(cb^4) = 5$. The **algorithm A5** (our new proposed algorithm) used a combination of the three node-to-node metrics used in the DTW's algorithm (exactly identical as binary constraints): the node coordinates cb^4 , the tangent cb^5 and the direction cb^6 with $s(cb^4) = 5, s(cb^5) = 15$ and $s(cb^6) = 15$ with $w(cb^4) = 0.5, w(cb^5) = 0.25$ and $w(cb^6) = 0.25$.

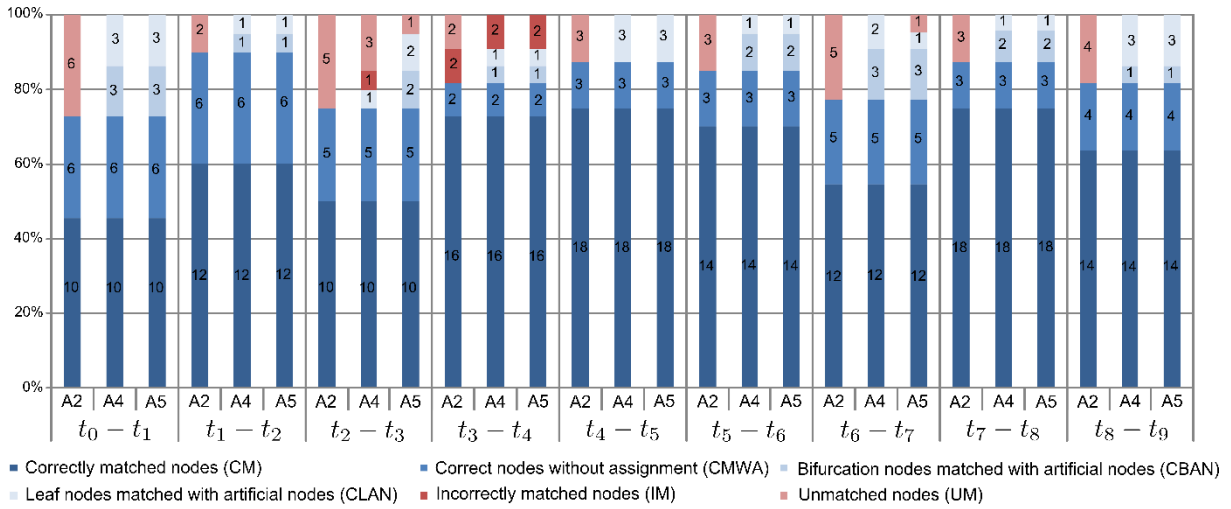


Figure 44: Comparison between our association graph algorithm without artificial nodes (A2) and with artificial nodes (A4 and A5). Constraints and associated selectivity parameters used to determine if two branches are similar are fixed for the nine tree matchings. A4 uses only the fourth constraint with $s(cb^4) = 5$. A5 uses a combination of node-to-node constraints: $s(cb^4) = 5, s(cb^5) = 15$ and $s(cb^6) = 15$ with $w(cb^4) = 0.5, w(cb^5) = 0.25$ and $w(cb^6) = 0.25$.

The number of correctly matched nodes is not modified, and false nodes present before this new step have the same classification too. They are many unmatched nodes (about 16.8%) without insertion of artificial nodes (A2). However, with insertion of artificial nodes, a major part of missing nodes is correctly matched. With the algorithm A4, 14.9% of nodes are supplementary correctly matched and classified against 15.8% for the algorithm A5. Consequently, the number of unmatched nodes decreases. The combination of metrics in the DTW's algorithm (A5) gives a better classification, so better matching results than only one constraint (A4). Indeed, as this last algorithm increases slightly the number of incorrectly matched nodes. In addition, the execution time of algorithms with artificial node insertion is still fast (see Table 7).

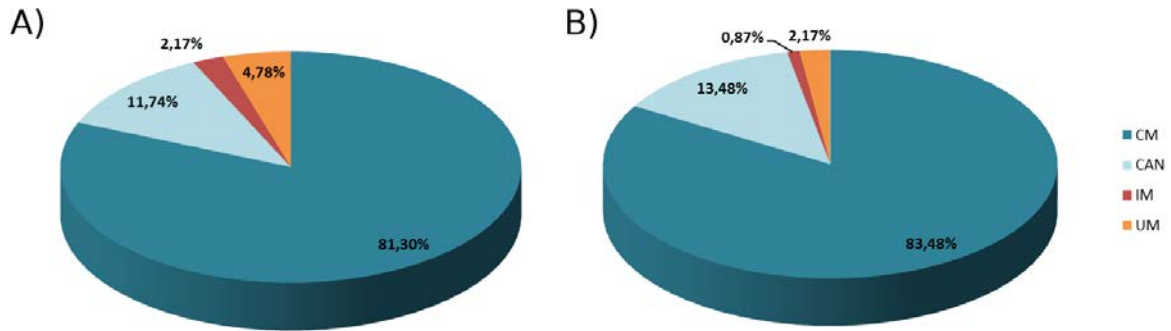


Figure 45: Comparison of node distribution with the matching between two successive trees of the sequence $S_{0,9}$. A) corresponds to the matching result with only one criteria, node coordinates, based on the DTW's algorithm as the binary constraint and the artificial node insertion constraint. B) corresponds to our new algorithm with the combination of the three criteria based on the DTW's algorithm and the hierarchy correction step (A5). We have the same node number in the two cases. CM corresponds to the correctly matched nodes. CAN represents the correct matched node with an artificial nodes (CBAN+CLAN). IM corresponds to the incorrectly matched nodes and UM the unmatched nodes.

The Figure 45 proposed a comparison of the nine matching results from the real sequence between the algorithm which used only one binary constraint based on the node coordinates and the same constraint in the artificial node insertion, and the algorithm A5 which used the combination of the three node-to-node constraints based on the Dynamic Time Warping. With these three new criteria used as the binary constraints and artificial node insertion step, we can see that the number of incorrectly matched node decreases (1.3%) such as the number of unmatched nodes (2.61%).

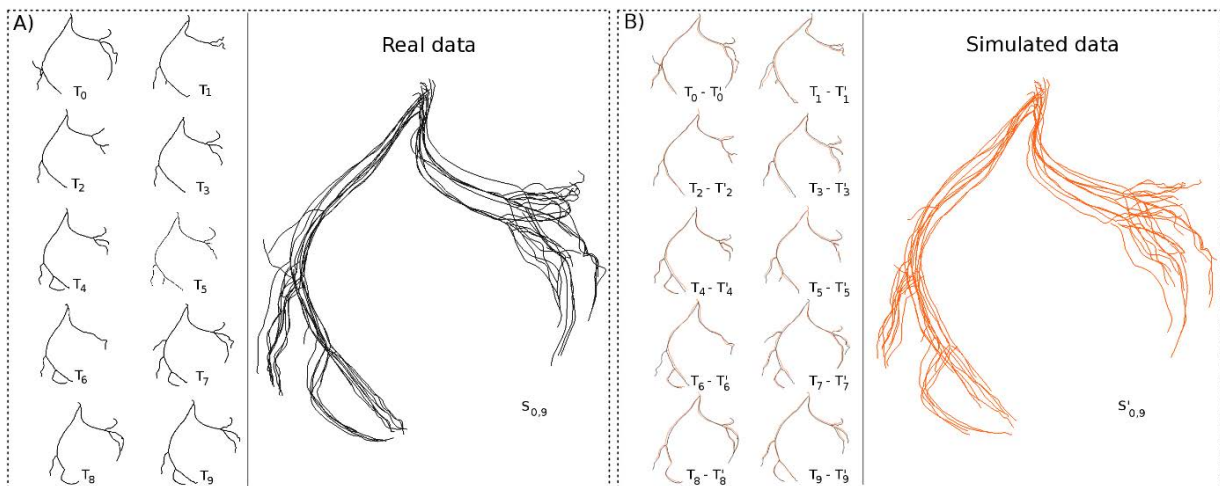


Figure 46: A) Real coronary tree sequence $S_{0,9}$ of 10 extracted left coronary trees T_i at different t_i cardiac phase and B) Simulated tree sequence $S'_{0,9}$ from modification of $S_{0,9}$.

From our real data sequence $S_{0,9} = \{T_0, \dots, T_i, \dots, T_9\}$, a new simulated tree sequence $S'_{0,9} = \{T'_0, \dots, T'_i, \dots, T'_9\}$ is created with our GUI (Graphical User Interface) (Figure 46 and Annexe A). The synthetic coronary tree T'_i represents the tree T_i with reductions of terminal branch length, modifications of the spatial coordinates of bifurcation and leaf nodes (after the first bifurcation) and modifications of the branch tortuosity.

Our proposed matching algorithm is next evaluated with combination of the real tree sequence and the new simulated tree sequence. The 36 sequence matching evaluations keep the same constraints and parameter's value as in the algorithm (A5) which gives better results. The Euclidean distance between node coordinates is used as unary constraint and the three node-to-node metrics (node coordinates, tangent and direction) based on the Dynamic Time Warping algorithm are used both as binary constraints and artificial node insertion constraint. The Figure 47 shows the node distribution for 36 matching results. The percentage represents the node number's mean for each category (CM, CBAN, CLAN, etc.) of the 36 tests. We obtain only 1.80% of errors (incorrectly matched and missing nodes). The details for the different node classification are described in the Figure 48. Unmatched nodes only come from artificial nodes corresponding to existing leaf nodes.

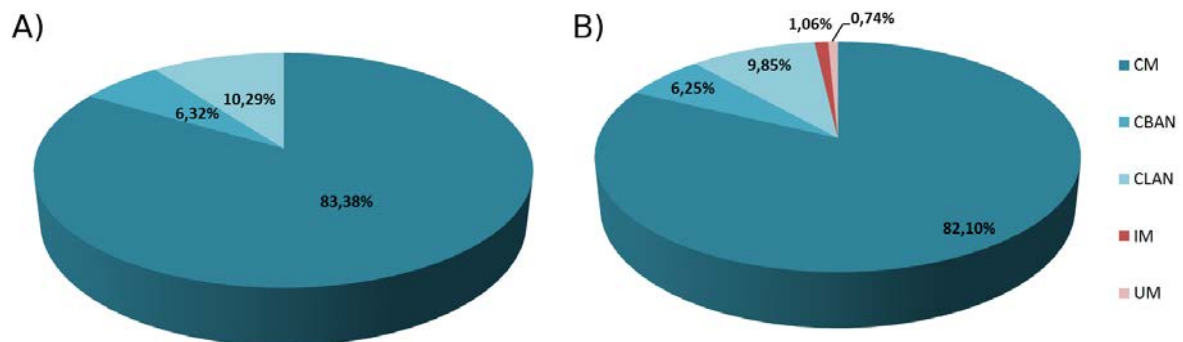


Figure 47: Node distribution in percentage for the 36 matching tests (mean of the total number for each classification). A) represents the real distribution (ground truth) and B) the node distribution after the matching of two successive trees (means of the 36 tests). CM corresponds to the correctly matched nodes, CBAN and CLAN respectively to correct matched bifurcation and leaf nodes with an artificial node. IM represents the incorrect matched nodes and UM the unmatched nodes.

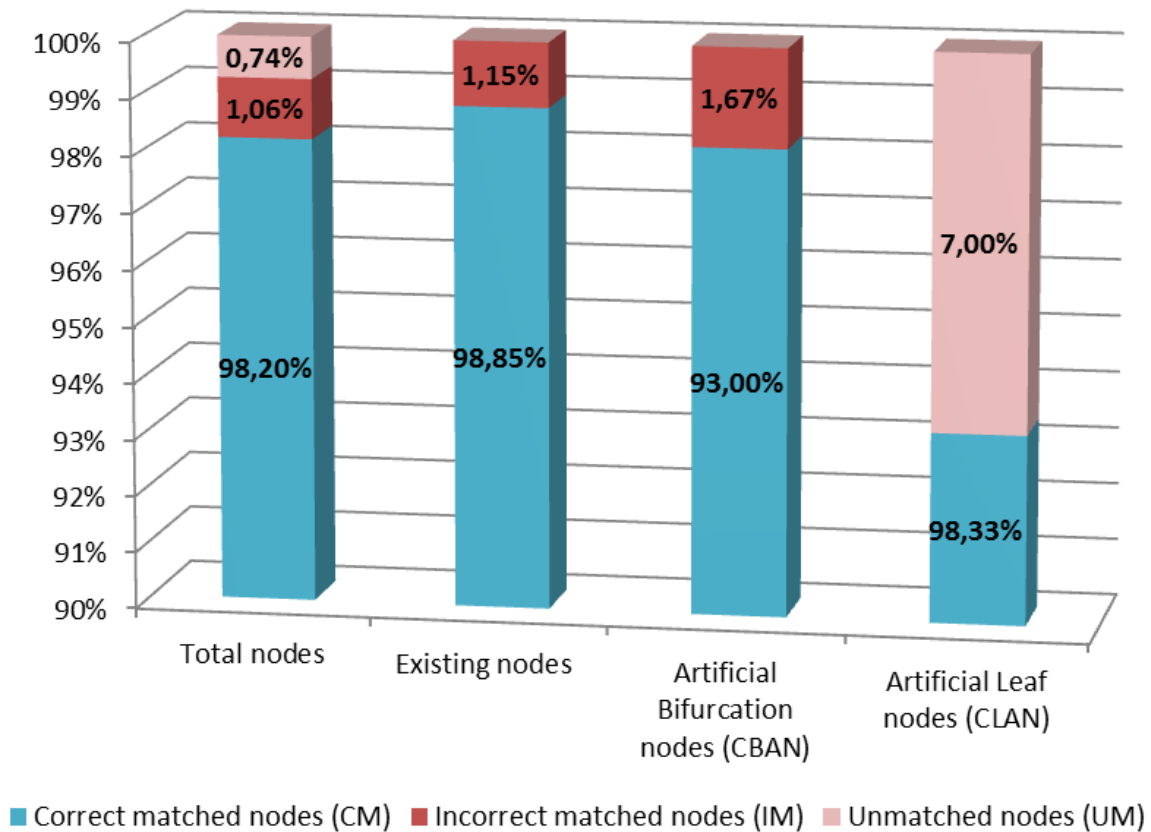


Figure 48: Node classification details for the 36 matching tests.

5.1.5. RESULT SYNTHESIS

To match two vascular tree structures, the choice of similarity measures is decisive. The compared algorithms in the previous chapters used the tree characteristics through their hierarchical structure and additional node-to-node constraints based on geometrical features. This association of nodes and paths improves the matching results in both algorithms. The best results are obtained with four proposed constraints with a fast execution time. The first constraint based on the Euclidean distance of node coordinates selects potential node assignments and so reduces significantly the number of potential matched node pairs. Latest three node-to-node constraints determine the quality of the kept potential assignments with the DTW's algorithm (node coordinates, tangent and direction). With the best configuration, four associated selectivity parameters (of node-to-node similarity

measures) and their thresholds must be empirically determined. During our tests in real data and simulated data, the same values are kept and the matching results are promising.

The best matching result, obtained with the association graph's approach [MKSZ09, TMPH05], have few incorrectly matched nodes but a higher number of unmatched nodes (16,8%). That is why we add a new step, the insertion of artificial nodes which improves the matching results without increasing significantly the execution time and influencing already false assignments. Indeed, the number of correctly matched nodes increases with this step (15.8% of nodes are supplementary correctly matched and classified) whereas the number of missing nodes decreases consequently. Nonetheless, some nodes are still unmatched. Moreover, the choice of last thresholds used to determine if two branches are similar is really important. If this threshold value is too large, some artificial nodes can be incorrectly inserted and wrong assignment deduced. Conversely, if the value is too strict, the similarity between branches cannot be found and the artificial node are not inserted in the tree. We choose to keep the same constraints based on the DTW's algorithm with the same values than the selectivity parameters associated with the binary constraints to reduce the number of parameters to evaluate.

5.2. MOSAIC-BASED MATCHING OF THE CORONARY SEQUENCE

To evaluate the mosaic-based matching algorithm of the whole coronary tree sequence a manually classification is also created for each node. Compared to the previous section, 9 possible classifications for a given node of a tree are proposed:

- 1) correctly existing matched node (CM),
- 2) correct artificial node corresponding to a bifurcation node (CBAN),
- 3) correct artificial node corresponding to a leaf node (CLAN),
- 4) unmatched existing node (UM),
- 5) artificial node corresponding to a bifurcation node not found (UBAN),
- 6) artificial node corresponding to a leaf node not found (ULAN),
- 7) incorrectly existing matched node (IM),
- 8) incorrect artificial node corresponding to a bifurcation node (IBAN) and
- 9) incorrect artificial node corresponding to a leaf node (ILAN).

The different constraints and their thresholds used during the matching process are the same for all tests. To remember, in our previous pair-wise matching algorithm (see section 3.1), the Euclidean distance between node coordinates is chosen as unary constraint (nodes similarities) with 10 as threshold value. Three node-to-node criteria (coordinates, tangent, direction distance) based on the Dynamic Time Warping's algorithm are used as binary constraints (centerline similarities) with respectively 5, 10 and 10 the threshold values and 0.5, 0.25 and 0.25 the weights of the three criteria. The same three criteria (and their thresholds and weights) are also used to insert artificial nodes.

Compared to our previous pair-wise matching approach, each tree of the coronary tree sequence contains more artificial nodes especially artificial nodes which corresponds to a leaf node in other tree sequence. Indeed, to perform the global matching we take into account all nodes of the tree sequence and not only nodes belonging to the next phase. The different information of additional branches or different branch lengths for one tree of the sequence affect the processing of all the next trees of the sequence. Artificial nodes are added in each tree to consider this extra information. Consequently, our global matching algorithm for the whole sequence is more accurate than the previous one (pair-wise matching in section 3.1).

The global matching method is evaluated with 42 sequences of 10 coronary trees which represent different combinations between the real data sequence $S_{0,9}$ and the simulated data sequence $S'_{0,9}$ (see section 5.1.4 and Figure 46). The global matching test from real coronary data $S_{0,9}$ is taken as reference to evaluate the following different combinations:

- 1) 10 sequences of 10 left coronary trees with only one synthetic tree among $S'_{0,9}$ and nine real trees from $S_{0,9}$,
- 2) 9 sequences of 10 left coronary trees with two successive synthetic trees among $S'_{0,9}$ and eight real trees from $S_{0,9}$,
- 3) 8 sequences of 10 left coronary trees with three successive synthetic trees among $S'_{0,9}$ and nine seven trees from $S_{0,9}$,
- 4) 7 sequences of 10 left coronary trees with four successive synthetic trees among $S'_{0,9}$ and six real trees from $S_{0,9}$,
- 5) 6 sequences of 10 left coronary trees with five successive synthetic trees among $S'_{0,9}$ and five real trees from $S_{0,9}$, and
- 6) 1 sequence of 10 left coronary trees with all synthetic trees of $S'_{0,9}$.

The Figure 49-A) shows the global node distribution in percent for the 42 tree sequences which corresponds to 8524 nodes to match in totality (203 nodes in mean for one sequence).

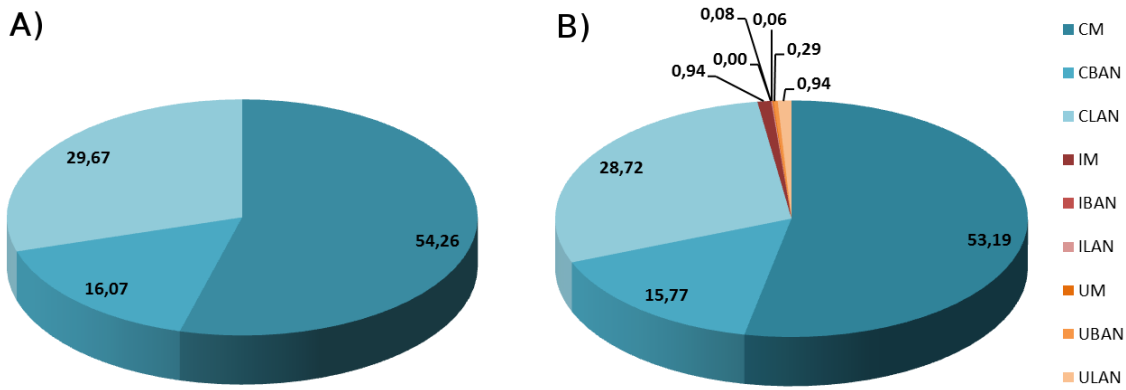


Figure 49: Node distribution in percent for the 42 tree sequences. A) represents the real distribution and B) the node distribution after our matching algorithm.

5.2.1. EVALUATION OF REAL SEQUENCE

The Figure 50 and Figure 51 represent the matching results of the real data sequence $S_{0,9} = \{T_0, \dots, T_i, \dots, T_9\}$. From the node matching result of the coronary tree sequence, each similar segment in the whole tree sequence is deduced and has the same ID.

In the Figure 50-A), the edge matching is shown with the superimposition of the whole tree sequence $S_{0,9}$. Similar edges with the same ID in the different trees have the same color. The Figure 50-B) shows the mosaic tree $US_{0,9}$ which is built from the first tree T_0 of the whole sequence $S_{0,9}$ completed with other segments present in the next trees of $S_{1,9}$.

The node classification for each tree of the sequence is described in the Figure 51. There is only one mistake in one tree T_6 which corresponds to a wrong matching of an existing leaf node and consequently to an unmatched artificial node (see Figure 56 case C). Although artificial node number is higher than in our previous pair-wise matching approach (section 5.1.4), the error percent are lower: 1.04% of incorrect nodes versus 2.04%.

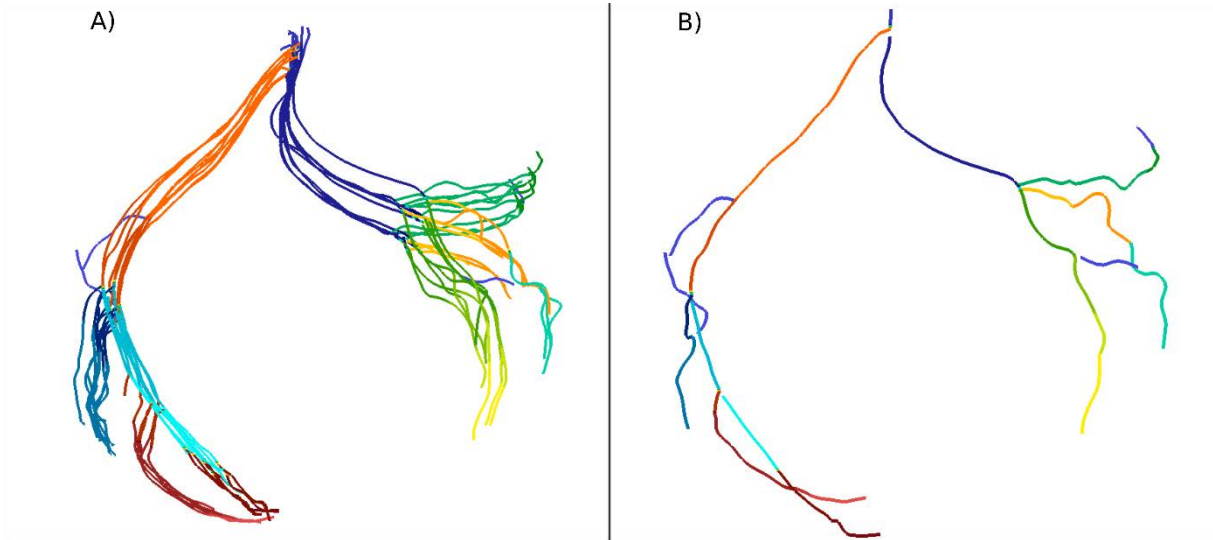


Figure 50: Matching result of the tree sequence $S_{0,9}$. A) Superimposition of the tree sequence with the matched edges represented in the same color. B) The global mosaic tree $US_{0,9}$ of the sequence $S_{0,9}$ based on the tree T_0 completed with information from $S_{1,9}$. The same colors are used for the edge matching.

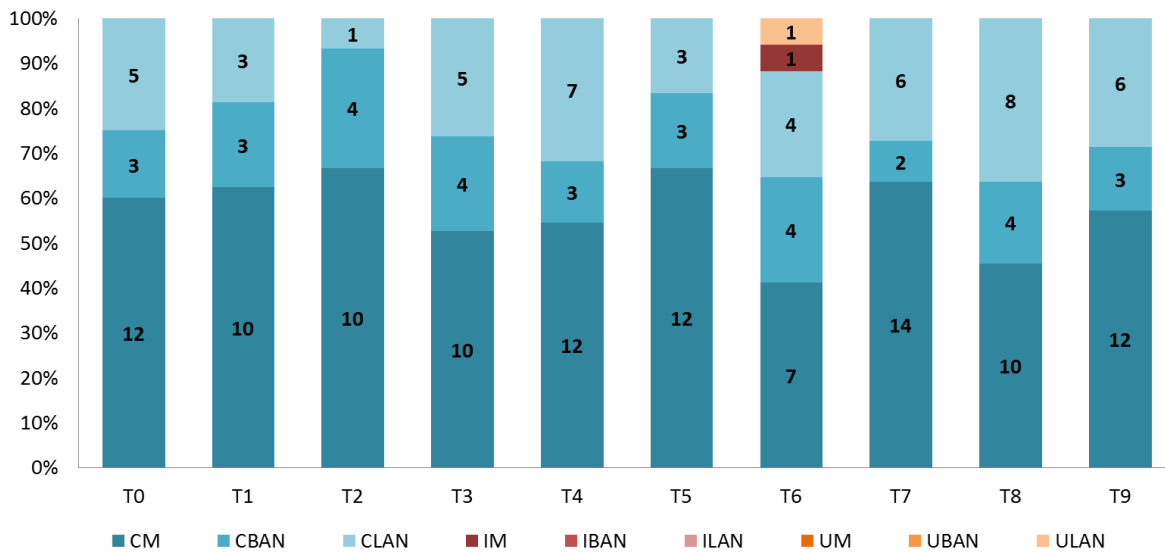


Figure 51: Result of the matched sequence $S_{0,9}$ with node classification for each tree.

The Figure 52 compares the node distribution of matching results obtained with our previous pair-wise algorithm between two successive trees of the real data sequence $S_{0,9}$ (9 matchings) and our new global mosaic-based matching algorithm. We can see that our mosaic-based algorithm detects more artificial nodes (42.19% of the total node numbers which have to be matched versus only 11.74%).

Moreover, although the total node number is higher than in with artificial nodes, we have less wrong and missing assignments mainly for the leaf nodes.

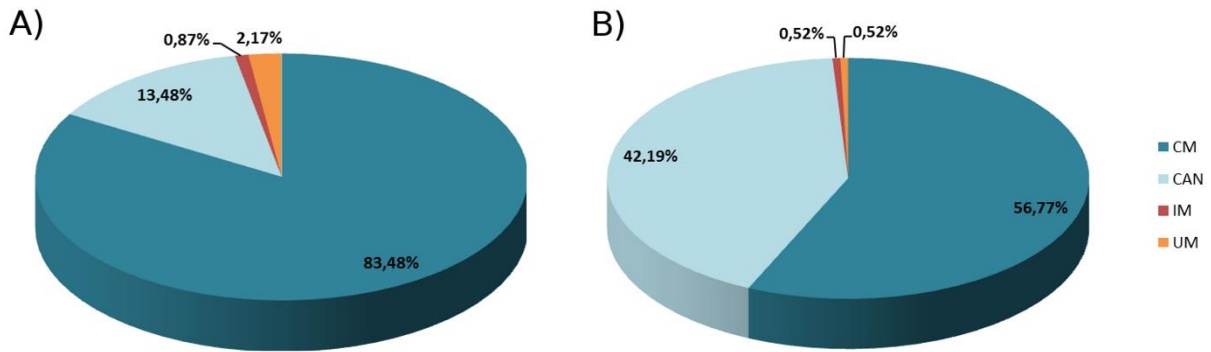


Figure 52: A) Node distribution in percent for the 9 matchings of two successive trees of the sequence $S_{0,9}$ computed in section 5.1. B) Node distribution in percent for the global matching algorithm of the sequence $S_{0,9}$ with mosaic tree building. In this figure, CAN represents all correct artificial nodes (CBAN + CLAN), IM and UM respectively the total incorrect matched node number and the total unmatched node number.

5.2.2. EVALUATION FROM SIMULATED SEQUENCES

The Figure 53 and Figure 54 show the matching result from the simulated tree sequence $S'_{0,9}$.

As previously, the Figure 53-A) represents the superimposition of the whole matched tree sequence with the matched edges in the same color. The Figure 53-B) corresponds to the complete tree of the simulated tree sequence $S'_{0,9}$.

The Figure 54 represents the node classification for this matched simulated sequence. There are more matching errors for this simulated sequence than for the real data sequence. One wrong assignment in a tree can affect next trees although our algorithm corrects the majority of errors (see Figure 56 cases A and B for examples of this kind of errors).

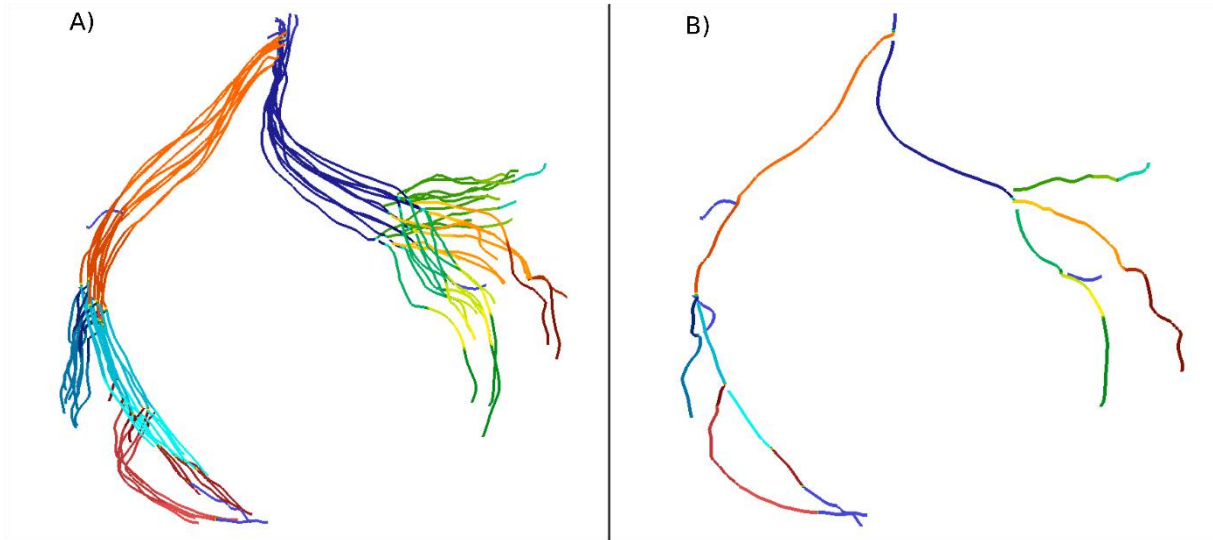


Figure 53: Matching result of the simulated tree sequence $S'_{0,9}$. A) Superimposition of the simulated trees with edge matching represented with same color. B) Global mosaic tree $US'_{0,9}$ of the sequence $S'_{0,9}$ built from the tree T_0 completed with information from $S'_{1,9}$. The same color as A) are used.

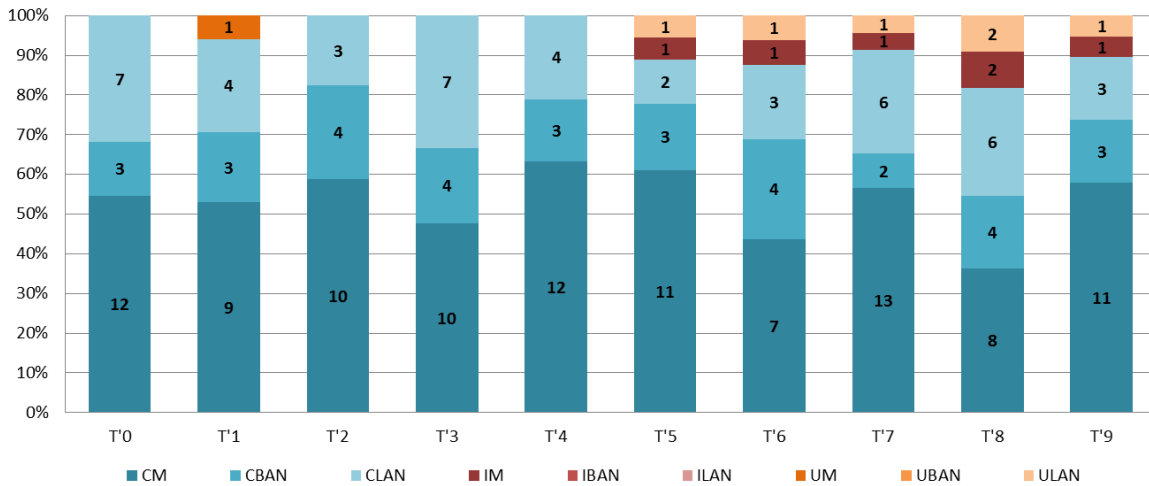


Figure 54: Node classification after the matching of the simulated tree sequence $S'_{0,9}$.

5.2.3. RESULT SYNTHESIS

We have seen the matching results for two sequences (real and simulated data) in detail. The Figure 49 – B) and Figure 55 represent the global matching result for the 42 coronary sequences. We obtain as median value 98.3% correctly matched nodes, 0.7% incorrectly matched nodes and 0.5% missing nodes. 9.5% of the matching tests have no error.

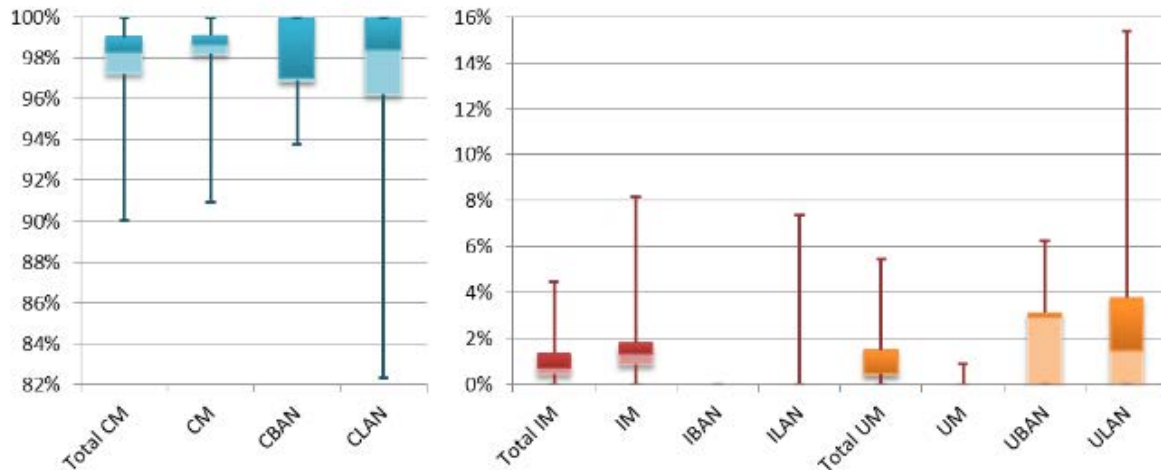


Figure 55: Box plots for the matching result in percentage for the 42 matched sequences. In the left, we have global percent of correctly matched nodes (Total CM) and after the details for real nodes (CM), artificial nodes corresponding to a bifurcation in another tree (CBAN) and artificial nodes corresponding to a leaf in another tree (CLAN). In the right, we have the global percent of incorrectly matched nodes (Total IM) and its associated details (IM, IBAN and ILAN), and after the global percent of unmatched nodes (Total UM) and its detail (UM, UBAN, ULAN).

The majority of matching errors are located in the terminal branches. The Figure 56 shows three major matching errors from the sequence $S''_{0,9} = \{T'_0, T'_1, T'_2, T'_3, T'_4, T_5, T_6, T_7, T_8, T_9\}$ containing 5 simulated trees and 5 original trees. These errors are mainly due to the pair-wise matching algorithm used to match two successive trees proposed in the section 3.1.3 and to some imprecision of artificial node coordinates (choice of similarity measure thresholds).

The proposed mosaic-based matching algorithm takes into account the non-isomorphic nature of the trees in the whole 3D+T sequence and exploits the whole data information. Furthermore, the mosaic-based algorithm is self-correcting during the matching process of the tree sequence. Consequently, the matching is more accurate and complete than in our previous pair-wise matching approach. Moreover, the result for our real data sequence has a lower percentage of incorrectly matched nodes. Indeed, our algorithm checks the matching rightness and so gradually corrects the detected hierarchical conflicts. For all the 42 tests, we have as mean 97.7% of correctly matched nodes over the sequence (median of 98.3%) and 97.3% of correctly inserted and matched artificial nodes.

As explained previously, matching errors are principally located in terminal branches and they can affect the next phases during matching process of the coronary tree sequence. A part of these errors come from the pair-wise matching of the two successive trees T_i and T_{i+1} used to build the union tree $UT_{i,i+1}$. Two leaf nodes can be matched although the branch lengths are different and

consequently an artificial node corresponding to one of these leaf nodes is missed. It can be due to similarity measure thresholds, used to build the association graph, which must be more accurate for some cases. Another part of these matching errors is also related to the threshold value precision in the artificial node insertion case. In the future, these threshold values should be automatically chosen to fit with more databases and be more accurate.

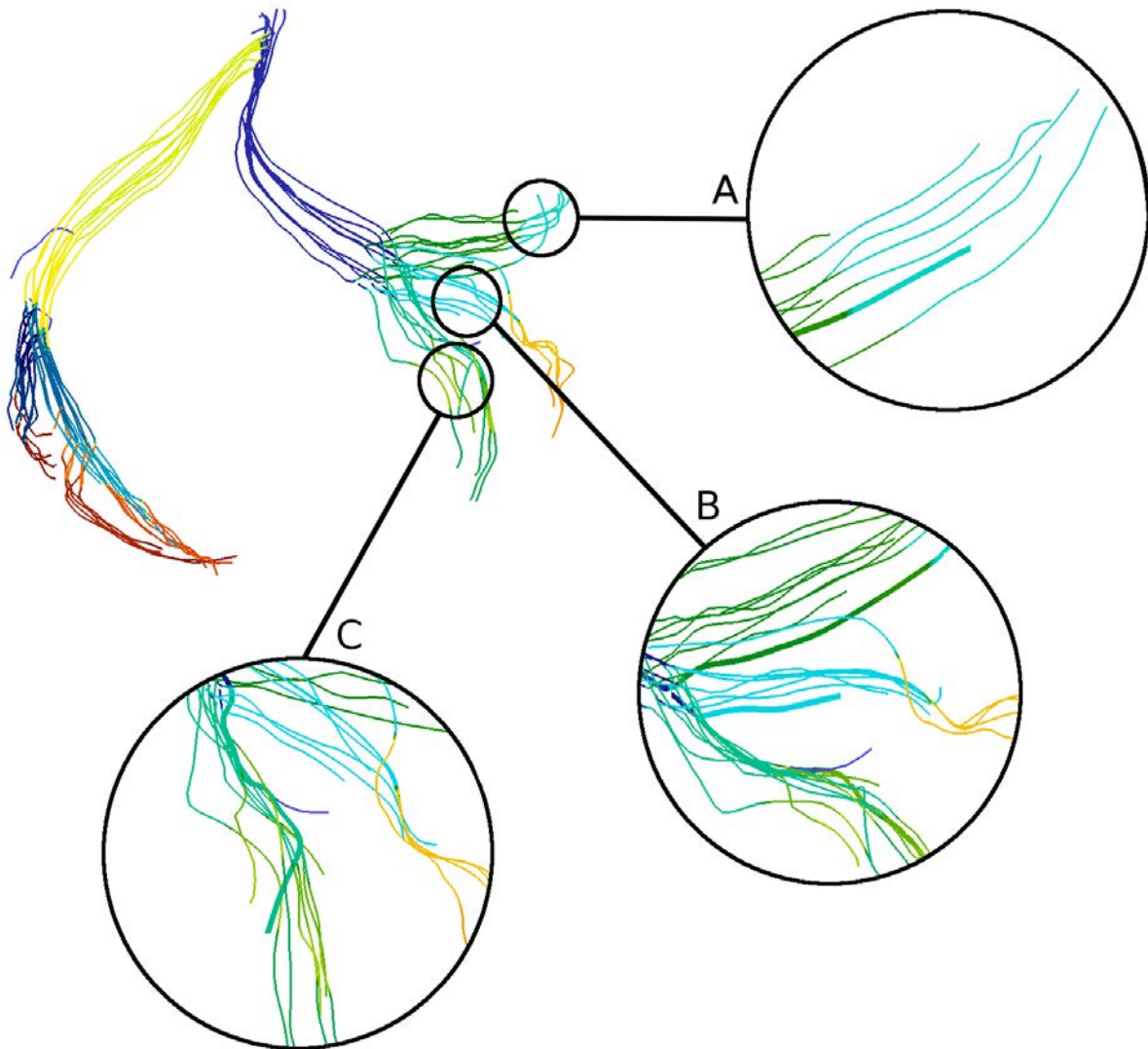


Figure 56: Examples of matching errors. The three zooms show the localization errors in different viewing angles. A) The leaf of the tree in bold is matched with the leaf in other trees. This incorrect matched leaf should be matched with an artificial node in some other trees which have longer terminal branch. B) The error is the same of the previous case A). The leaf of the tree in bold is incorrectly matched and the same artificial node (which corresponds to the current leaf) is missing in all the other trees of the sequence. C) Contrary to the two previous cases, errors are only located on the tree in bold (errors similar to the Figure 51). The leaf node is incorrectly matched and consequently an artificial node corresponding to a bifurcation is missed.

5.3. OPTIMAL VIEWING ANGLE RESULTS

Up to now, we evaluated the proposed coronary tree matching algorithm. In this section, our method used to determine the optimal viewing angle (section 4.4) will be evaluated for different object of interest: a single segment, a multiple segments (from root to the lesion area) and a bifurcation. As a remainder, for each test, different maps are built corresponding to a single criterion and a phase (for example the global optimal view map). Each pixel of the maps represents a coefficient between 0 and 1 (the result of the different criterion or map combination) for a projection and a given gantry orientation (LAO/RAO defining the y-axis and CRA/CAU the x-axis). More the value is near to 1, more the corresponding gantry orientation is optimal for the object of interest.

5.3.1. RESULT FROM SEGMENT OF INTEREST

The first proposed result concerns a segment as object of interest (Figure 57). This segment is not present in each tree of the sequence: it is only matched in five trees ($Q = 5$). The Figure 57-A shows the different steps of the optimal viewing angle determination without threshold. The centerline projection and foreshortening rate are computed 55205 times (11041×5 (see equation (6) which defines the interval of the two angles of the C-arm)) and the different volume projection and external overlap rate are also computed 55205 times. The runtime is very high for this algorithm. The Figure 57-B shows our proposed algorithm for a segment of interest (Algorithm 8). As explained previously (section 4.4), after each map building a threshold used to reduce the execution time but also to remove worse gantry orientation for one phase, is performed. In this case, with a threshold equal to 90%, the centerline projection and foreshortening rate are only calculated 23099 times and the volume projection and external foreshortening rate is computed 15329 times. With these different thresholds, the global optimal view map has only 2494 retained gantry orientations ($nb_{(\theta_j, \varphi_k)}$).

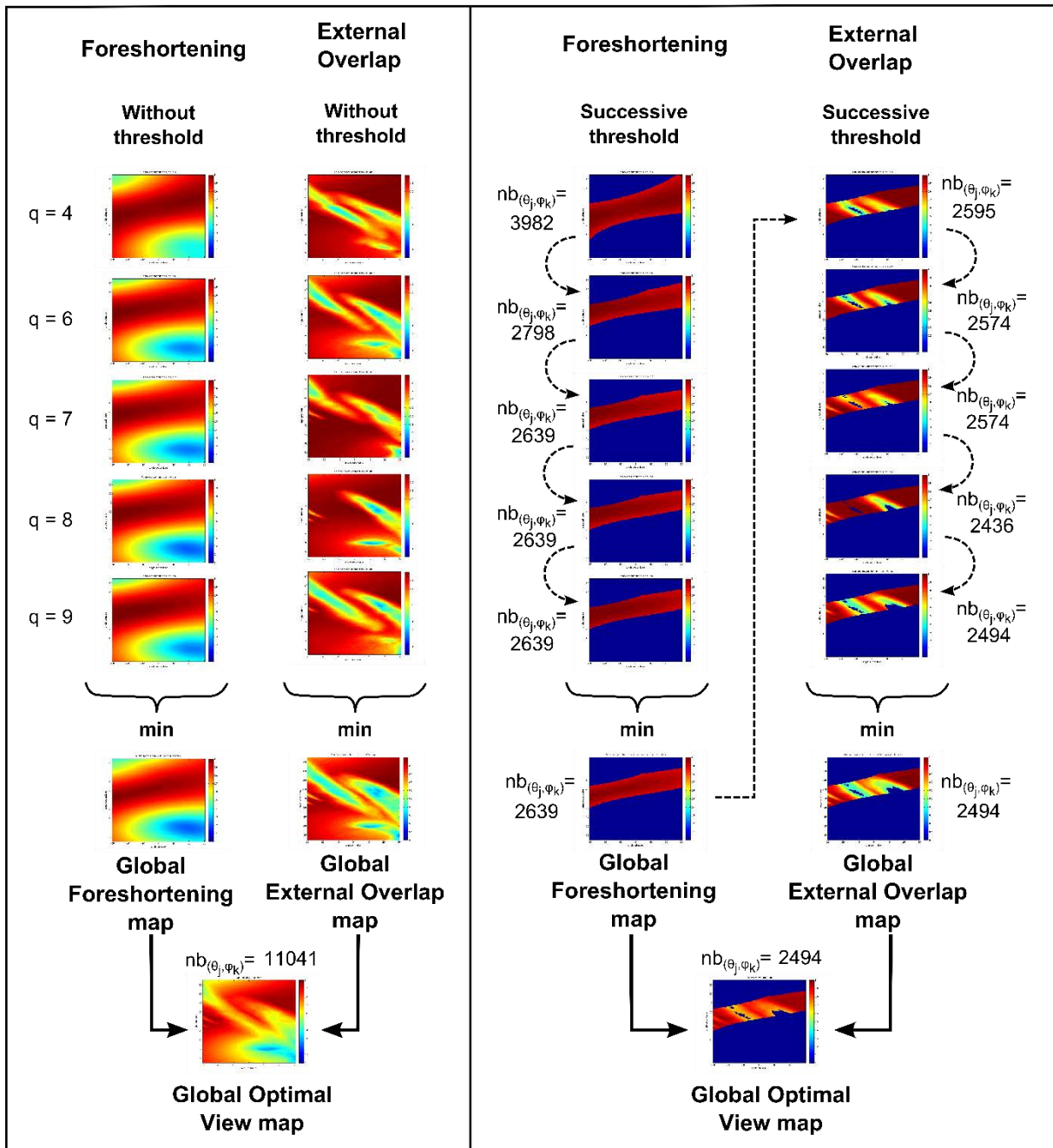


Figure 57: Process of the optimal view map determination for a segment in the whole cardiac cycle. A) The foreshortening and external overlap maps are computed successively without threshold for each matched phase. The global foreshortening and external overlap maps are calculated with the minimum value of the previous temporary maps. The global optimal view map corresponds to the mean of the two last maps. B) Same process as A) but with two successive thresholds. First the foreshortening maps are computed with a threshold of 90% for each phase to perform a high selection of angles until the global foreshortening maps. Second, the external overlap maps are calculated with a threshold of 30% to only remove worse values. The global optimal view map is computed with the two previous global map. $nb_{(\theta_j, \varphi_k)}$ represents the selected gantry orientation number after each threshold.

The Figure 57 proposed the global optimal view map of the selected matched segment with 2494 possible gantry orientations. Thanks to the thresholds and the using of minimum value to obtain the different global maps, all of these cases give the segment of interest under a correct orientation in

the whole cardiac cycle. However, the physician has too many choices. That's why we proposed to retained only the best angles with maximum values. In the Figure 58, we select respectively 5% and 2% of the better gantry orientations. Two different gantry orientations are proposed. The first angle $(\theta_j, \varphi_k) = (-7^\circ, -27^\circ)$ in green in the Figure 58 which corresponds to LAO 7° and CRAN 27°, gives the better value with 98.47% of foreshortening and 95.99% of external overlap. A second angle $(\theta_j, \varphi_k) = (-40^\circ, 25^\circ)$ in pink in the Figure 58 which corresponds to LAO 40° and CRAU 25° and situated in another group in the optimal view map, gives another maximum value with 94.11% of foreshortening and 99.04% of external overlap.

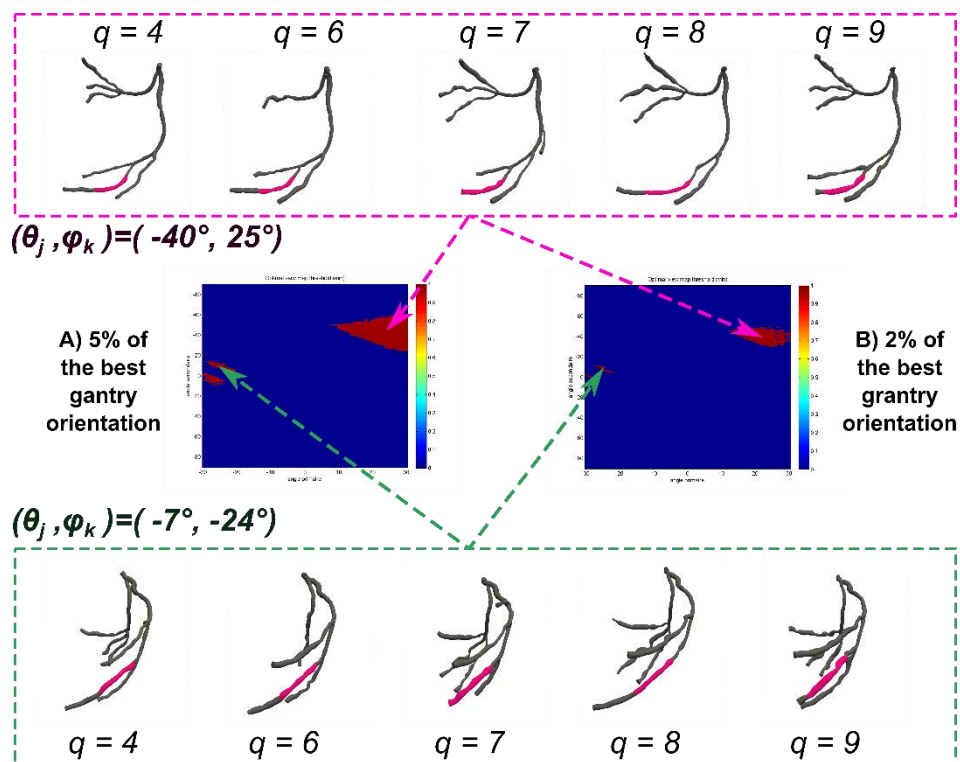


Figure 58: Optimal viewing angles for a segment of interest matched in five cardiac phases. The optimal view map is obtained in the Figure 57. Two optimal gantry orientations are proposed and the left coronary artery with the segment in pink is shown under these two views. A) Optimal view map with the 5% better gantry orientation (551 retained angles). B) Optimal view map with the 2% better gantry orientation (220 retained angles).

5.3.2. RESULTS FOR BIFURCATION OF INTEREST

The second result, which we propose in this section, concerns a bifurcation as object of interest (Figure 59). The bifurcation is matched in all trees of our coronary tree sequence ($Q = 10$). Consequently, the maps for the four criteria (foreshortening, external and internal overlap, bifurcation

opening angle) are computed for each phase. The Figure 59-A shows the optimal view map without threshold. The different threshold values chosen for any bifurcation are different than for a segment. Indeed, with the number of criteria which doubles, if we keep a high threshold value in each phase, the selection associated with criteria may be too important and for result, only few or none optimal views can be proposed to the physicians (see next section 5.3.3). Thus, we chose only to remove worse values for each phase and each criterion. In the Figure 59-B, we propose a threshold at 30%. We didn't choose a higher threshold because of the bifurcation opening rate: the selection becomes too important in each phase. The computational time is also reduced with our method (Algorithm 9). The centerline projection, foreshortening and bifurcation opening angle rates are computed 69202 times against 110410 times without thresholds. The volume projection, external and internal overlap rates are computed 60621 times against aslo 110410 times without thresholds.

With the threshold values and the creation of the different global maps for each criterion with the minimum value of each phase, our proposed algorithm takes into account the non-isomorphic nature of the coronary trees of the sequence. In the Figure 59-A, when $q = 0$ the external overlap map is different than the other phases. In this tree, there is an additional branch which exists only in this phase. If the mean or the median is used to compute the global external overlap map of the cardiac cycle, then the external map of the first tree, which have additional branches, should be under-represented in the global map. In this case, the proposed optimal viewing angle should not be optimal in the whole cardiac cycle: the additional branch is under-appreciated.

As for the segment of interest, the global optimal view map of a selected bifurcation (Figure 59) can have too many potential angles (6064 in this test). To reduce this choice, in the Figure 60, we propose a threshold to keep only the 5% or the 2% better gantry orientations with maximum value in the global optimal view map. The Figure 60 shows also the optimal viewing angle $(\theta_j, \varphi_k) = (37^\circ, -18^\circ)$ corresponding to RAO 37° and CRA 18° (with 97.13% of foreshortening, 44.13% of bifurcation opening, 93.72% of external overlap and 84.81% of internal overlap) and the left coronary tree and the selected bifurcation in pink under this orientation for the different phases of the sequence. The selected bifurcation have no overlap with the other structure and with itself and the children branches of the bifurcation are well separated (high bifurcation angle).

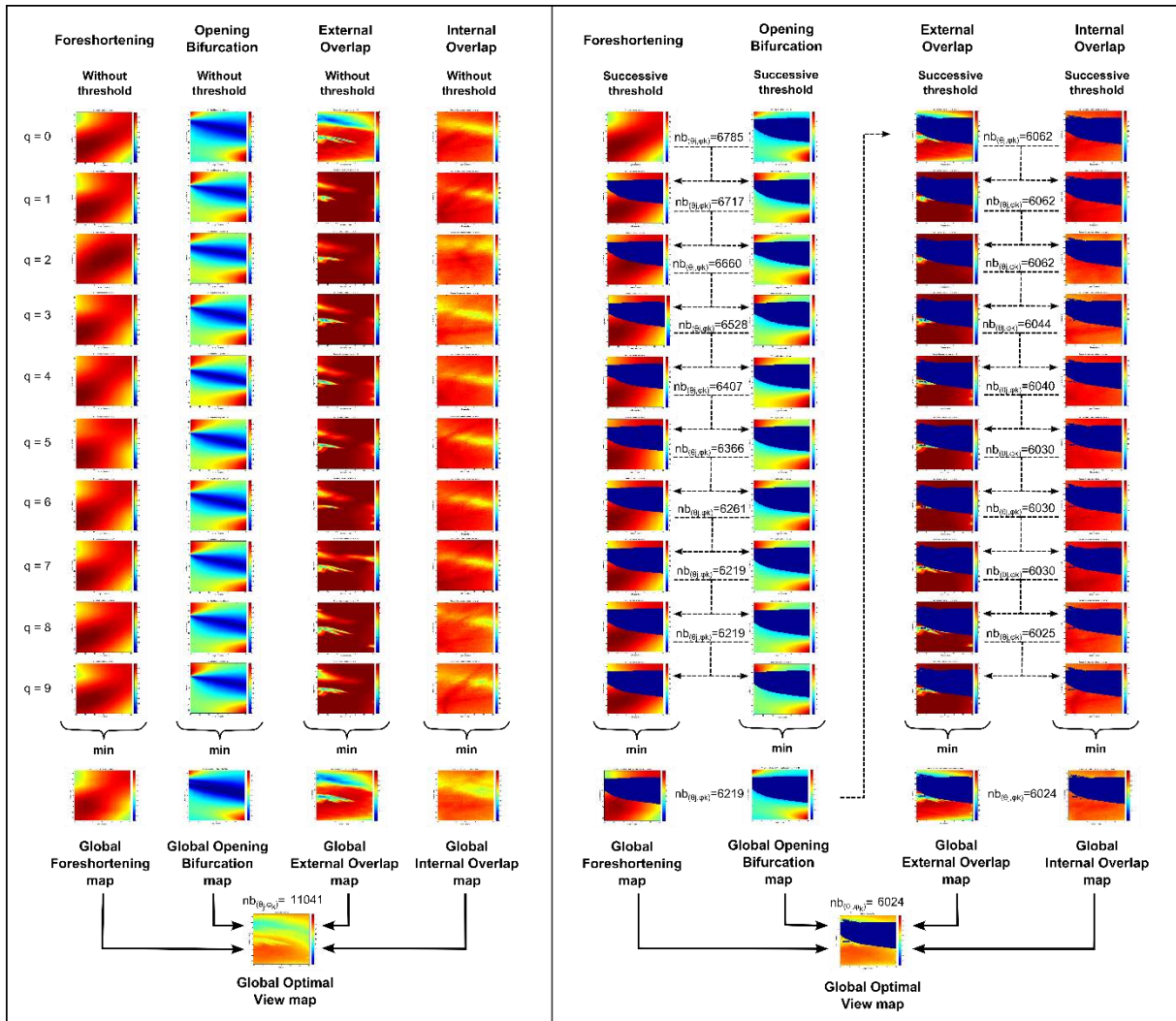


Figure 59: Process to optimal view map determination for a bifurcation in the whole cardiac cycle. The bifurcation is matched in all the ten phases. A) The four criteria are computed without threshold for each matched phase. B) The four criteria are calculated successively with a threshold to remove worse values under 30%. First the foreshortening map and bifurcation opening map are computed successively for each map and the threshold is performed. Next, when the global foreshortening and bifurcation opening maps are obtained with their minimum value on the sequence, the two overlap are computed in the same way. The global optimal view map is obtained with the mean of the four global maps. $nb_{(\theta, \phi_k)}$ represents the selected gantry orientation number after each threshold.

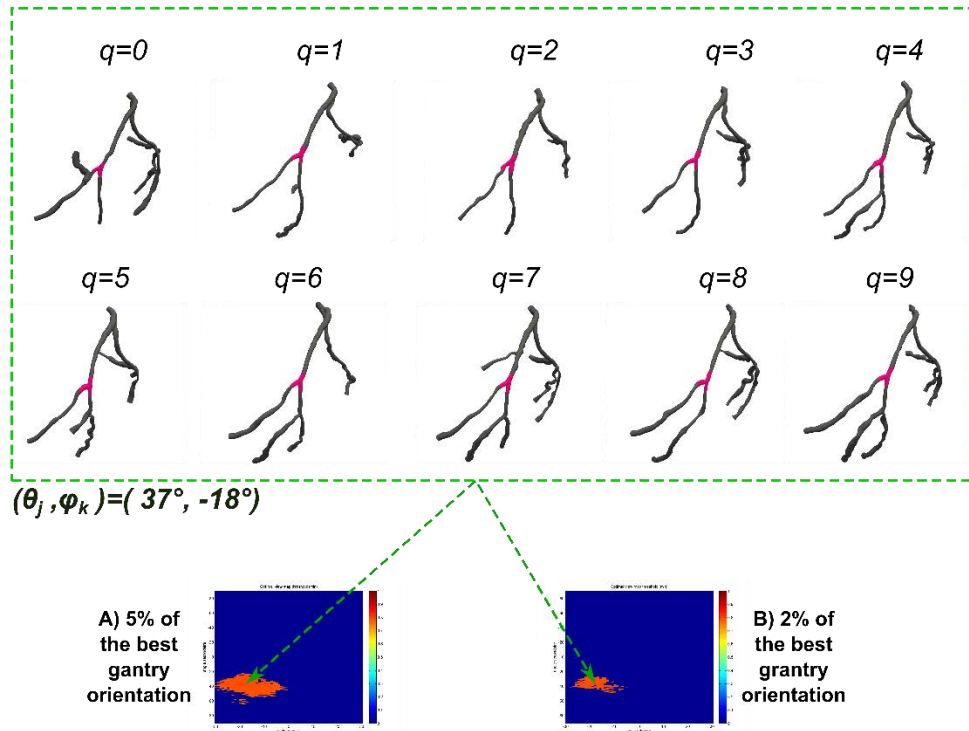


Figure 60: Optimal viewing angles for a bifurcation in the whole cardiac cycle. The optimal view map corresponds to the Figure 59. In top, the left coronary tree with the bifurcation in pink is display under the best angiographic view for the different q phases.

5.3.3. RESULTS FOR MULTIPLE SEGMENTS OF INTEREST

The third proposed result is about multiple segments. The optimal view map is obtained in the same way as the Figure 57 (Algorithm 8). In the GUI, the user selects the segment of interest and all segments from the root to the selected segment are merged into one. As explained previously, this segment allows determining optimal viewing angles for the guidance of the catheter and also the stent deployment.

The first column of the Figure 61 shows the optimal view map for a multiple segment with different thresholds. The last six columns describe optimal view maps for each segment which composes the multiple segment. In this test, six segments composed the multiple segment object. If the optimal view maps of the segments 1, which represents the selected single segment (stenotic segment), and the multiple segment are compared, we can see that the selected segment is under-represented in the optimal view map of the multiple segment. More the multiple segment is longer with tortuous centerline and have more segments, more is difficult to find an optimal viewing angles for both the guidance and the stent deployment. That's why, we choose to combine the different optimal view

maps of the six segments to obtain one single global optimal view map (which can be similar to the one of the multiple segment). The second column of the Figure 61 shows the combination of the global optimal view maps of the six segments with the same weights (i.e. mean of the maps). However, as explained previously in section 4.4, we choose different weights for the six segments according to their distance to the selected segment and their length (see Table 8). As explained in the section 4.4, with this two different criteria, to choose their weight factor we give advantage to the selected segment. Indeed, in the case of the determination of the optimal viewing map for a multiple segment, we want an optimal gantry orientation useful to the catheter guidance, but we want also that this chosen view can be used to deploy the stent. The third column of the Figure 61 shows this different merging of the maps. The threshold used in the optimal view map computation for a single segment is important (Algorithm 8). We cannot choose a higher threshold value as previously in the section 5.3. Indeed, as in the last line in the Figure 61, the likelihood that obtaining no optimal view angles is very high. Between the six threshold optimal view maps, there is no region of interest shared. That's why we choose a threshold similar to the section 5.3.2 which removes worse values but which doesn't perform a high selection. In the Figure 61, we choose a threshold value equal to 40%.

The global optimal view map obtained for the multiple segments is less accurate than the mean or the merging of the six global optimal view maps obtained for the different segments (which compose the multiple segment). As show in the Figure 63, the multiple segment represented in colour is only matched in five phases (T_4, T_6, T_7, T_8 and T_9). The global optimal view map corresponding to the combination of the six segments take into account more information (variation of topology) about the cardiac cycle. Indeed, the selected segment (segment 1 in pink in the Figure 63) is matched only in five phases (as the multiple segment) but its father segment (segment 2 in green in the Figure 63) has a correspondence in seven coronary trees and the others (segment 3 to segment 6, respectively in purple, orange, blue and yellow in the Figure 63) are matched in all trees of the 3D+T sequence. The global optimal view map of each segment is at least computed in more cardiac phases than the multiple segment.

Moreover, as shown in the Figure 62, the external overlap map of the multiple segment (in the left) is also less accurate than the external overlap map of the combination of the six segments (in middle). As explained in the subsection 4.4, with the different segment concatenation, the multiple segment has less overlap with the other structures. The multiple segment is considered as only one entity and not as different branches such as the bifurcation, so the internal overlap is not computed.

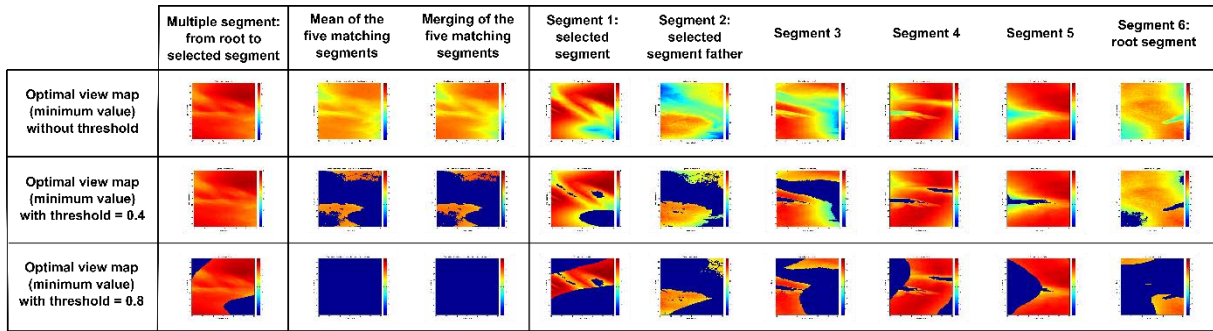


Figure 61: Optimal view map for different segments. In the right, the five last columns represent the optimal view maps for five successive segments in the tree hierarchy. In the left, the first column corresponds to the optimal view maps when the five previous segments are merged into one (multiple segments case). In the middle, the two last columns represent the optimal view map from the linear combination of the optimal view map of the five segments (same weights for the first and weights calculated from the two criteria proposed in section 4.4). The first line represents the optimal view map without threshold and the others with different thresholds.

Table 8: The different weights used to the optimal view map combination.

Segments	Point number	Weighted Length	Weighted Distance	Total weight (mean)
Segment 1 (pink)	27	0.2269	0.2857	0.2563
Segment 2 (green)	13	0.1092	0.2381	0.1737
Segment 3 (purple)	17	0.1495	0.1905	0.1667
Segment 4 (orange)	17	0.1429	0.1429	0.1429
Segment 5 (blue)	38	0.3193	0.0952	0.2073
Segment 6 (yellow)	7	0.0588	0.0475	0.0532

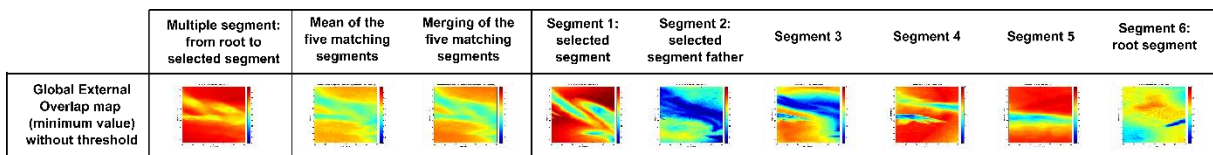
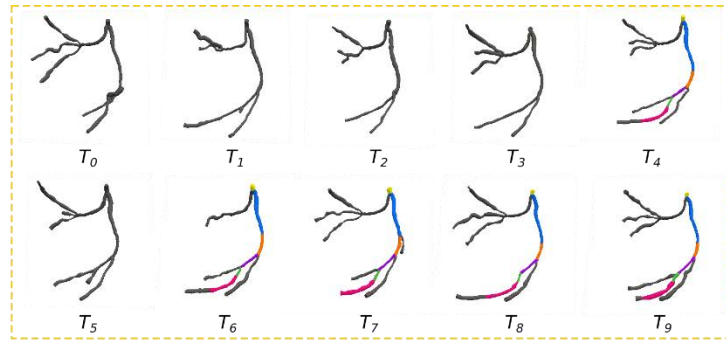
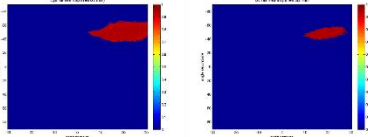


Figure 62: Global External Overlap map for different segments. In right, the last five columns represent the global external overlap maps for the five successive segments in the tree hierarchy. In the left, the first column corresponds to the global external overlap map when the five segments are merged into one single (multiple segment case). In the middle, the two last columns represent the global external overlap map from the linear combination of the global external overlap map of the five segments (same weights for the first and weights calculated from the two criteria proposed in section 4.4).

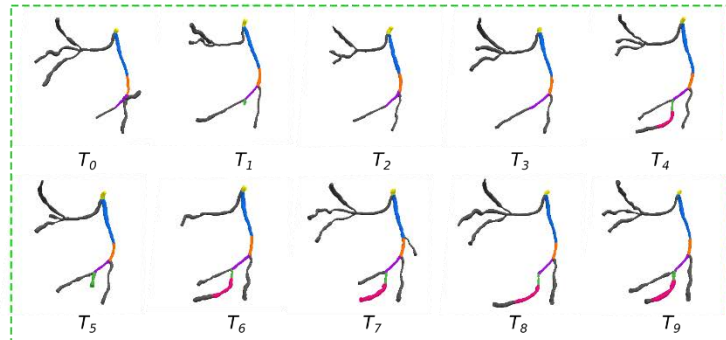
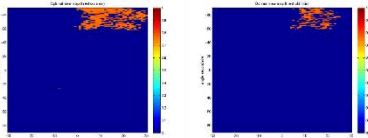
**A) Multiple segment:
from root to selected
segment**

$$(\theta_j, \varphi_k) = (-43^\circ, 16^\circ)$$



**B) Mean of the five
matching segments**

$$(\theta_j, \varphi_k) = (-74^\circ, 17^\circ)$$



**C) Merging of the five
matching segments
(two criteria)**

$$(\theta_j, \varphi_k) = (-60^\circ, 20^\circ)$$

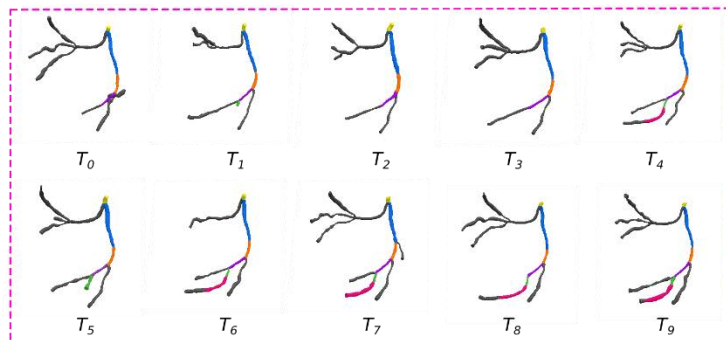
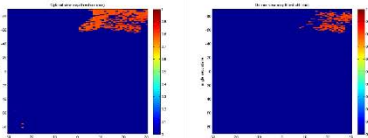


Figure 63: Optimal viewing angles for a multiple segment. The optimal view maps in the left correspond to the 5% better angiographic projection angles and the following maps represent the 2% better values. A) Optimal viewing angle for the multiple segment as object of interest. B) Optimal viewing angle for the combination between the six segments (same weight). C) Optimal viewing angle for the combination between the six segments with different weights computed in relation to the hierarchical level of the segment according to the selected segment (in pink) and the length segments. The different tree sequences in the right show the left coronary tree under the found optimal gantry orientation. The pink segments represent the selected segment (segment 1), the green segment is the father of the selected segment (segment 2), the segment 3 is in purple, the segment 4 in orange, the segment 5 in blue and the root segment (segment 6) in yellow.

The Figure 63 shows also the optimal view maps for the three cases (multiple segment global map, mean of the six global maps and merging of the six global max) with first 5% of the best values and next 2%. In right, the coronary tree sequence is displayed respectively under the best computed gantry orientation. In the case A, only five trees are taken into account to determine the optimal viewing angle. In the two next cases B and C, all phases are exploited. As we can see, the optimal view map is rather similar in our test: the optimal viewing angles in A, B and C are near. The three Table 9, Table 10 and Table 11 show in detail the foreshortening rates, external overlap rates and mean of these two criteria (optimal view map) for the different obtained optimal viewing angles, respectively $(-43^\circ, 16^\circ)$ for the multiple segment (Figure 63-A)), $(-74^\circ, 17^\circ)$ for the mean of the six global optimal

view maps of the single segments Figure 63-B)), and (-60°, 20°) for the merging of the six global optimal view maps of the single segments Figure 63-C)). The last column of each table represents the foreshortening and overlap maps obtained with the three different methods. The other columns show the percentages for the different segments which compose the multiple segment. In the Table 9, the external overlap rate and the optimal value is equal to 0% because of the threshold used after the external overlap map building. The value of the external overlap rates was under 40% for this gantry orientation.

Table 9: Foreshortening and external overlap rates obtained for the optimal viewing angle (-43°, 16°) determined from the optimal view map of the multiple segment (Figure 63-A)).

Type of rates	Segment 1	Segment 2	Segment 3	Segment 4	Segment 5	Segment 6	Multiple segment
Foreshortening rates	94.75%	84.97%	88.96%	95.32%	92.07%	63.61%	99.44%
External Overlap rates	98.75%	40.35%	20.93%	85.64%	90.36%	58.44%	91.24%
Optimal view percentage (mean)	96.75%	62.66%	54.94%	90.48%	91.21%	60.83%	95.34%

Table 10: Foreshortening and external overlap rates obtained for the optimal viewing angle (-74°, 17°) determined from the mean of the global optimal view maps of the six segments (same weight factor) (Figure 63-B)).

Type of rates	Segment 1	Segment 2	Segment 3	Segment 4	Segment 5	Segment 6	Mean of the six
Foreshortening rates	82.55%	79.30%	90.50%	92.05%	87.53%	69.17%	83.53%
External Overlap rates	94.60%	45.83%	60.26%	87.88%	85.91%	56.25%	71.79%
Optimal view percentage (mean)	88.57%	62.57%	75.38%	89.97%	86.77%	62.71%	77.66%

Table 11: Foreshortening and external overlap rates obtained for the optimal viewing angle (-60°, 20°) determined from the merging of the global optimal view maps of the six segments (different weight factors) (Figure 63-C)).

Type of rates	Segment 1	Segment 2	Segment 3	Segment 4	Segment 5	Segment 6	Merging of the six
Foreshortening rates	90.07%	82.66%	90.91%	93.98%	89.58%	64.28%	88.01%
External Overlap rates	97.31%	45.76%	44.12%	87.75%	86.39%	52.44%	73.47%
Optimal view percentage (mean)	93.69%	64.21%	67.52%	90.86%	87.98%	58.36%	80.74%

5.3.4. RESULT SYNTHESIS

In this section, the determination of the optimal viewing angle for different objects of interest from coronary tree sequence corresponding to the whole cardiac cycle are evaluated with four criteria (foreshortening, external and internal overlap, bifurcation opening angle). We propose a first case of optimal view map obtained for a single segment matched in only five cardiac phases among ten phases (since this segment is missed in the other five phases). As result, the chosen gantry orientation gives a correct visualisation with few foreshortening and overlapping of the selected segment. With the thresholded optimal view map, we can propose several optimal views focus only on a single selected segment to the physician.

Next case, we evaluate our algorithm for a bifurcation matching from all coronary trees corresponding to all cardiac phases. The optimal viewing angles propose the selected bifurcation with few foreshortening, internal and external overlapping and also with a great bifurcation opening.

As last case, for a multiple segment from root to the same selected segment, we compare three different methods to determine optimal viewing angles. The first method considers the multiple segment as one single object. The second and the third methods consider respectively the mean and the weighted (depending on the length and hierarchical level of segment) of all optimal view maps of all segments which compose the multiple segment. These three methods give similar and correct result. Now, we cannot determine the best method because only few tests are realized.

With the different objects of interest, several thresholds associated with four criteria are performed to compare the efficiency of our algorithms. For a given single segment, the first threshold is used to perform an important selection among the better foreshortening values. We choose to keep the gantry orientation which have more than 90% as foreshortening rate. This selection allow to reduce the number of potential optimal viewing angles to compute (supress the worst according to the foreshortening criterion). Consequently, a lower number of gantry orientation is evaluated for the next criterion, the external overlap rate (reduction of the execution time). A larger threshold value might be too strict and consequently only few or none view would be proposed to the physicians.

For a given bifurcation or multiple segment, a lower value is used for the different thresholds (30% or 40%) with the same reason. We want to suppress the gantry orientation which have the worst rate. When the different optimal view maps of the different segments are combined (same or different weight factors), the selection of good gantry orientations is important in the combined optimal view map. In the case of a bifurcation, more criteria are used compared to only one single segment

(foreshortening and bifurcation opening rates are computed first, and next, after the threshold, the two overlap rates are computed). The different global maps of each criterion may be very different and the global optimal view map can have, with a lower threshold value (30% or 40%), an important angle selection. These threshold values will be evaluated more precisely through a validation of our method on larger data sequence number.

5.4. CONCLUSION

Ce chapitre propose une évaluation des différents algorithmes décrits au cours des chapitres 3 et 4. Dans un premier temps, nous avons comparé trois algorithmes de mise en correspondance d'arbres deux à deux dont deux issus de la littérature (mais adaptés à nos données). Ces derniers ont servi de base à notre approche d'appariement d'arbre deux à deux. Pour chacun des algorithmes, une évaluation de différentes mesures de similarité a été réalisée afin d'obtenir la meilleure combinaison. Notre algorithme basé sur la construction d'un graphe d'association donne de meilleurs résultats (testé sur 36 appariements) en sélectionnant les couples de nœuds potentiellement appariables grâce à un seuil de distance entre leurs coordonnées puis en confirmant cette similarité grâce à trois critères (coordonnées, tangente et direction) à partir desquels une distance basée sur l'algorithme de Dynamic Time Warping est calculée. Comme expliqué précédemment, le choix des seuils pour les différentes mesures de similarité est important. Si la valeur est trop faible, l'insertion des nœuds artificiels peut être incorrect et les appariements faussés. A l'inverse, si la valeur est trop stricte, la similarité entre branches peut être non trouvée et les nœuds artificiels ne seront alors pas insérés. Cette méthode, reposant sur l'insertion de nœuds artificiels et sur l'étape de correction hiérarchique, propose un nombre de nœuds correctement appariés plus important ; avec pour conséquence un nombre de nœuds non appariés réduit de 15,8% en comparaison avec les deux autres méthodes issues de la littérature. 1,06% des nœuds sont incorrectement appariés et 0.74% des nœuds sont manquants pour les différentes évaluations effectuées.

Cette méthode, donnant des résultats satisfaisants pour la mise en correspondance entre deux arbres, n'était cependant pas suffisamment précise pour appairer une séquence de plusieurs arbres coronaires. Effectivement, elle ne prend en compte que les différences topologiques des arbres directement voisins temporellement (successifs). C'est pourquoi, dans un deuxième temps, nous avons proposé un algorithme de mise en correspondance globale d'arbres basé sur la construction d'un arbre mosaïque et évalué sur 42 séquences de 10 arbres coronaires gauches. Grâce à cette méthode, les

arbres de la séquence ont été enrichis. Ils sont beaucoup plus complets et possèdent un nombre plus important de nœuds artificiels permettant d'identifier les bifurcations manquantes et les branches terminales de longueurs différentes. Nous obtenons ainsi sur les séquences testées 98,3% de nœuds correctement appariés avec 0,7% de nœuds de la séquence entière incorrectement mis en correspondance et 0,5% de nœuds manquants. La mise en correspondance de la séquence entière ne comporte que très peu d'erreurs. Ces quelques erreurs sont localisées principalement en fin de branches (ces erreurs ont moins d'incidence que celle portée sur les bifurcations) et sont majoritairement dues à l'application de l'algorithme de mise en correspondance d'arbre deux à deux utilisé en tant que première étape. Malgré les corrections liées aux conflits hiérarchiques, ces erreurs sont tout de même répercutées lors de la création de l'arbre mosaïque.

Le résultat de cette mise en correspondance globale de la séquence entière a été utilisé pour la détermination d'incidences optimales afin d'obtenir une vue angiographique optimale de la zone de lésion tout le long du cycle cardiaque, peu importe les mouvements des artères coronaires. Quel que soit l'objet d'intérêt (segment, segment multiple ou bifurcation) nos algorithmes proposent au cardiologue des angles de vues où l'objet est bien déployé et sans superposition sur l'ensemble du cycle cardiaque. Les trois méthodes proposées concernant le segment multiple nous donnent des incidences similaires. Grâce aux cartes de vues optimales, nous pouvons proposer plusieurs choix au cardiologue selon son besoin et ses préférences. Effectivement, la meilleure incidence, ayant le moins de superposition, de chevauchement (interne et externe) et une plus grande ouverture d'angle dans le cas des bifurcations, est ainsi proposée avec également les 5% ou 2% des meilleurs angles. Puisque nos différents algorithmes ont été évalués sur peu de données (étude de faisabilité), nous ne pouvons donc actuellement pas mettre en avant une des méthodes développées. Le calcul de l'incidence optimale sur le cycle cardiaque dépend de la mise en correspondance de la séquence dynamique. Si celle-ci est faussée, ces erreurs d'appariements seront répercutées sur la recherche d'angle de vue optimale.

L'utilisation des différents seuils associés à nos mesures dans nos algorithmes de recherche d'incidences optimales permet, dans un premier temps, de réduire le temps de calcul, mais aussi de supprimer les couples d'angles ne correspondant pas à nos critères (trop de chevauchement par exemple). La valeur de ces seuils peut légèrement varier en fonction du nombre d'arbres de la séquence. Une évaluation plus poussée par rapport à ces valeurs sera prochainement réalisée sur plus de données.

CONCLUSIONS ET PERSPECTIVES

SOMMAIRE

6.1.	Bilans et conclusions	154
6.2.	Perspectives.....	157
6.2.1.	Amélioration des méthodes	157
6.2.2.	Validation des méthodes.....	158
6.2.3.	Autres applications.....	159

Au cours de cette thèse, nous nous sommes intéressés aux lésions coronaires en imagerie rotationnelle RX et plus particulièrement au planning de l'angioplastie coronarienne. Notre démarche vise à aider le cardiologue à choisir une incidence 2D dite optimale permettant au mieux de guider et de déployer le stent au niveau de la sténose à partir de reconstructions (angiographie rotationnelle) ou de segmentations (images CT) 3D des artères coronaires de toutes les phases cardiaques. Nous souhaitons proposer au cardiologue une vue bien entendu compatible avec les caractéristiques du C-arm qui prenne surtout en compte le mouvement et les différentes déformations subies par les artères coronaires tout le long du cycle cardiaque. Ce chapitre, concluant le manuscrit, dresse un bilan des travaux réalisés et présente les travaux futurs qui pourraient être envisagés et les nouvelles perspectives de recherche.

6.1. BILANS ET CONCLUSIONS

Pour déterminer l'incidence optimale, il est nécessaire de suivre chaque segment pour l'ensemble des phases du cycle cardiaque. Pour cela, nous avons proposé tout d'abord une nouvelle méthode automatique de mise en correspondance d'une séquence dynamique d'arbres vasculaires. Cette méthode prend en compte les différences topologiques entre les arbres de la séquence liées à la variabilité de leur segmentation (depuis une séquence d'images de CT) ou reconstruction (depuis une séquence d'angiographie rotationnelle) 3D pour une phase cardiaque donnée, ainsi que les différents déplacements des branches au cours du cycle cardiaque. L'algorithme proposé se décompose en plusieurs étapes.

Tout d'abord, après construction des graphes (structure d'arbres) issus des volumes 3D (surfaces des contours vasculaires) et des lignes centrales associées, nous avons créé un algorithme de mise en correspondance d'arbres successifs deux à deux. Cet algorithme est basé sur la construction d'un graphe d'association et de la recherche de clique maximum correspondant aux couples de nœuds appariés. Deux types de mesure de similarité sont utilisés au cours de la construction du graphe d'association. Le premier type, appelé contrainte unaire, est utilisé pour sélectionner les couples d'angle potentiellement appariables grâce à un critère de distance euclidienne entre les coordonnées spatiales des nœuds. Le deuxième type, appelé contrainte binaire, est utilisé pour mesurer la similarité des lignes centrales entre les couples de nœuds préalablement retenus. Nous utilisons, comme contrainte binaire, trois critères (coordonnées, tangente et direction des nœuds) permettant de calculer une distance basée sur le *Dynamic Time Warping* [KePa01, RaKe05] afin de comparer la

similarité entre nœuds et segments des deux arbres. L'avantage de cette méthode est de proposer des critères à la fois géométrique permettant de prendre en compte le mouvement non-rigide des artères coronaires, et hiérarchique permettant de prendre en compte l'absence de structures.

Une étape ultérieure, consistant à ajouter des nœuds artificiels dans les arbres de la séquence, permet également de tenir compte des variations topologiques entre arbres. Ces nœuds artificiels correspondent à une bifurcation existante ou à une branche de longueur inférieure à celle d'un autre arbre voisin. Une vérification hiérarchique est réalisée à chaque insertion de nœuds artificiels pour résoudre les conflits associés. Au cours de l'évaluation de cette méthode sur 36 appariements d'arbres deux à deux, nous avons déterminé la meilleure combinaison de mesures permettant la mise en correspondance la plus correcte. Un des inconvénients majeurs de cette méthode est le choix empirique des différentes valeurs de seuils et de pondération (4 valeurs de seuils et 3 de pondérations) associées aux mesures proposées. Une des perspectives de notre travail serait alors de tester notre méthode sur plus de données coronariennes afin d'obtenir un paramétrage automatique indépendant des données.

Afin d'améliorer encore la qualité de la mise en correspondance de la séquence d'arbres, nous avons proposé une méthode globale d'appariement exploitant toutes les phases cardiaques au lieu de seulement deux à deux successivement. Cette méthode globale se décompose en plusieurs étapes :

- appliquer la mise en correspondance deux à deux,
- calculer l'union des deux arbres successifs préalablement appariés avec un label pour chaque nœud (nœud apparié, nœud seul, nœud artificiel, etc.),
- fusionner l'ensemble des arbres d'union permettant d'obtenir un arbre mosaïque.

Cet algorithme s'autocorrige au fur et à mesure de la fusion d'arbres et permet de prendre en compte toutes les différences topologiques des arbres et de les répercuter sur chaque arbre de la séquence. Au final, nous obtenons un arbre mosaïque représentant un arbre complet de la séquence entière mais également d'enrichir chaque arbre de la séquence qui possède maintenant d'avantage de nœuds (nœuds artificiels).

Cette méthode de mise en correspondance d'une séquence entière a été validée sur 42 séquences (réelles et simulées). La méthode est robuste face à la variabilité topologique pour l'ensemble des phases cardiaques. Elle présente en effet un pourcentage satisfaisant de nœuds correctement appariés ainsi qu'un faible taux d'erreurs ou de nœuds non appariés. Ces erreurs sont situées dans la majorité des cas dans les branches terminales et sont liées à des erreurs d'appariement lors de la première étape (mise en correspondance d'arbre deux à deux). Une amélioration de cet

algorithme est également envisageable pour obtenir le moins d'erreurs possibles d'appariement notamment, par l'utilisation de nouvelles mesures de similarité par exemple. Cet algorithme, pour une séquence de 10 arbres coronaires, est rapide d'exécution (moins d'une minute) mais peut être optimisé (avec l'utilisation des possibilités de traitement multitâche du GPU – *Graphics processing unit* – par exemple).

Nous avons développé cet algorithme de mise en correspondance globale d'une séquence dynamique d'arbres coronaires afin de suivre au cours du cycle cardiaque la zone d'intérêt. A partir de cet appariement de toute la séquence, nous avons ainsi proposé également une nouvelle méthode de recherche d'incidence optimale tout le long du cycle cardiaque dans le cas de traitement de lésions coronaires. Notre méthode, prenant en compte à la fois les mouvements cardiaque et les différences de topologie entre les arbres de la séquence, propose ainsi une vue angiographique optimale pour toutes les phases du cycle cardiaque. L'utilisateur choisi par le biais de l'interface graphique de l'application développée, un objet d'intérêt qui peut être soit un segment, un segment multiple (chemin de la racine au segment d'intérêt) ou une bifurcation. Grâce à ce choix, notre méthode propose au cardiologue soit une vue optimale pour le guidage du cathéter depuis la racine jusqu'à la zone de lésion ou de manière plus globale pour le déploiement final du stent. Une étude de faisabilité de l'approche proposée a été réalisée. Comme résultats préliminaires, les vues angiographiques obtenues présentes bien les différents objets d'intérêts avec peu de raccourcissement, de chevauchement (interne et externe) et un angle d'ouverture de bifurcation important (dans le cas des bifurcations). Pour pouvoir valider et montrer la robustesse de la méthode, une évaluation sur d'avantage de données réelles et simulées est envisagée. Cela va nous permettre de déterminer si les seuils proposés actuellement pour l'optimisation et la suppression des plus mauvais angles de vue pour une phase donnée dépendent de différents paramètres, comme le nombre d'arbres de la séquence (c'est-à-dire le nombre de phases cardiaques lors de l'acquisition) par exemple.

Dans le cas de la détermination de l'angle de vue optimale pour un segment multiple, trois différentes méthodes ont été proposées. Ces trois méthodes proposent des résultats similaires en pourcentage de raccourcissement et de chevauchement externe. L'approche utilisant la pondération des différents segments simples (composant le segment multiple), privilégie le segment sténosé par rapport à l'autre méthode utilisant uniquement la moyenne entre les cartes. Le taux de raccourcissement et de chevauchement externe est effectivement plus important lors de la pondération des cartes. Cette combinaison de segments simples exploite plus d'information (c'est-à-dire plus de phases du cycle cardiaque) que la méthode utilisant le segment multiple en tant qu'objet

unique. Elle semblerait plus précise. De plus, la séparation du segment multiple devrait apporter une précision plus importante de l'angle de vue optimale du chemin. En effet, le chevauchement du segment multiple avec lui-même n'est pour l'instant pas considéré (chevauchement interne). Cependant, à cause du faible nombre d'évaluations, nous ne pouvons pas déterminer pour l'instant si une méthode est meilleure qu'une autre dans le cadre d'un segment multiple mais aussi d'une bifurcation. Dans le futur, nous souhaiterons également déterminer l'influence d'une mauvaise mise en correspondance sur l'incidence optimale calculée. En effet, si un objet est mal apparié sur au moins une phase du cycle cardiaque, les cartes obtenues peuvent être incorrectes. Par conséquent, l'incidence choisie peut ne pas être optimale sur l'ensemble du cycle cardiaque.

6.2. PERSPECTIVES

Cette thèse propose donc différentes méthodes automatiques allant de la mise en correspondance de deux arbres coronaires successifs, à l'appariement globale de séquence complète 3D+T puis à la détermination d'incidence optimale pour un segment unique, un segment multiple ou une bifurcation tout le long du cycle cardiaque. Ces différentes méthodes sont intégrées à une application disposant d'une interface graphique décrite dans l'Annexe A, facilitant ainsi leurs utilisations. Les résultats obtenus pour ces différentes méthodes sont prometteurs. Diverses améliorations et évolutions sont ici envisagées.

6.2.1. AMELIORATION DES METHODES

Tout d'abord, pour proposer dans un temps raisonnable la mise en correspondance puis la détermination d'un angle de vue optimale à partir d'une séquence d'arbres coronaires 3D, l'exécution du programme peut être optimisée grâce à la parallélisation de sa mise en œuvre par un traitement multitâche GPU par exemple. La mise en correspondance de la séquence entière est réalisée rapidement mais le temps d'exécution pour la détermination de l'angle de vue optimale est beaucoup plus important à cause du nombre de projections possibles et du nombre de phases.

Ensuite, les valeurs des paramètres utilisés pour la mise en correspondance peuvent varier en fonction des données. L'utilisation d'une méthode calculant ces paramètres de manière automatique

permettrait d'améliorer l'appariement et de réduire les erreurs dans l'algorithme de mise en correspondance de la séquence complète.

En outre, la méthode de détermination d'incidence optimale peut également être améliorée. Nous considérons pour l'instant la zone sténosée sur un segment entier et non une portion. Dans le futur, via l'interface graphique, nous envisageons de permettre à l'utilisateur de choisir plus précisément la zone de lésion pour avoir un angle de vue encore plus adapté. De même pour les bifurcations, l'utilisateur doit pouvoir choisir la longueur de la bifurcation sur les différentes branches. Ces différentes possibilités sont réalisables sans trop de difficulté. En effet, grâce à la mise en correspondance à travers l'algorithme du Dynamic Time Warping, nous connaissons la mise en correspondance point par point afin de suivre une partie de segment sur l'ensemble du cycle cardiaque.

Une amélioration peut également être apportée dans le cas de la recherche d'incidence optimale pour un segment multiple. De nouveaux critères peuvent être rajoutés pour obtenir un résultat encore plus robuste et plus précis pour la méthode où le segment multiple est considéré comme objet unique (temps de calcul plus court par rapport à celui de la combinaison des différents segments composant le segment multiple). Par exemple, nous pouvons considérer un critère permettant de supprimer les angles de vue où le segment projeté se superpose avec lui-même (similaire au critère de chevauchement interne pour les bifurcations). De plus, nous pouvons privilégier plutôt le critère de raccourcissement par rapport à celui du chevauchement externe puisque le segment multiple a moins de chance de se superposer avec les autres structures.

Une autre perspective portant sur l'amélioration des travaux concerne l'application développée. Celle-ci encore en cours de développement devra être adaptée et testée à plus grande échelle ainsi qu'être optimisée pour être envisagée en routine clinique. Avant cela, nous devons montrer les différents résultats et les différents choix au cardiologue, allant de l'angle de vue optimale à la carte optimale retenant les 5% ou 2% meilleures incidences .

6.2.2. VALIDATION DES METHODES

Les différentes méthodes ont été évaluées à partir d'une séquence réelle de 10 volumes et de séquences simulées contenant également de 10 volumes issues de la modification de la séquence réelle. La méthode d'appariement d'arbres coronaires deux à deux a été testée sur 36 séquences et celle de mise en correspondance globale d'une séquence entière a été évaluée sur 42 séquences. Pour

pouvoir valider au mieux ces deux méthodes et rendre le choix des paramètres plus adapté et ces méthodes plus robustes, nous prévoyons de les évaluer sur une autre base de données issue de scanners dynamiques de 10 ou 20 volumes par cycle cardiaque. Ces scanners seront dans un premier temps segmentés phase par phase afin d'obtenir une séquence 3D+T d'arbres coronaires avec les informations de lignes centrales. Grâce à ces données supplémentaires, une amélioration sera portée sur la définition des bifurcations à partir des lignes centrales afin d'avoir une bifurcation plus réaliste et d'obtenir une incidence optimale pour une bifurcation plus robuste. Nous pourrions également utiliser, dans les algorithmes de mise en correspondance, des mesures de similarité portées sur des informations de volume. Actuellement nous ne pouvons pas les exploiter. Effectivement, le diamètre des artères coronaires sur les différents arbres est incohérent sur nos données disponibles. A plus long terme, nous souhaiterions obtenir une reconstruction 3D+T d'arbres artériels coronaires issus d'angiographie rotationnelle pour se passer du CT. En effet, la majorité des angioplasties ne sont planifiées qu'à partir d'angiographie rotationnelle et sans disposer de CT.

La deuxième partie de la thèse (recherche d'incidence optimale) propose une étude de faisabilité. Différents tests ont été réalisés à partir de la séquence de données réelles : une incidence optimale a été obtenue pour la majorité des segments et des bifurcations composant cette séquence. Cependant, d'un point de vue clinique, nous ne pouvons pas valider complètement la méthode au vue de la quantité de données disponibles. Dans le futur, nous aimerions comparer les incidences optimales obtenues avec celle définie par un expert et valider ainsi notre méthode sur d'avantage de cas réels.

6.2.3. AUTRES APPLICATIONS

A partir des méthodes développées, d'autres applications et évolutions sont envisageables.

Tout d'abord, toujours dans le domaine des lésions artérielles coronaires, il est possible de suivre point par point le déplacement des branches coronaires grâce à l'exploitation plus poussée de l'algorithme Dynamic Time Warping (utilisation du chemin d'alignement entre les chemins – *path*) [KePa01, KeRa05, RaKe05, SaCh78]. Avec un nombre important de données, une étude portant sur le mouvement des artères coronaires dans le temps peut être envisagée.

De plus, nos méthodes de mise en correspondance et de recherche d'incidence optimale ne prennent pas en compte l'insertion du cathéter dans la structure et donc la déformation des artères liée à cet outil. Il serait alors envisageable de proposer une vue angiographique tenant compte de cette

forte déformation durant l'angioplastie. En effet, le passage de ces outils (guide, cathéter, etc.) apporte rigidité aux artères (et donc une tortuosité plus faible de celle-ci) et l'angle de vue optimale peut alors être modifié.

Nos deux algorithmes de mise en correspondance peuvent être utilisés dans d'autres applications (telles que celles étudiées au sein du LTSI). Nous pourrions les tester dans le cas de la mise en correspondance d'arbres vasculaires, différents des coronaires, comme par exemple des veines coronaires assez proches en terme de topologie des artères coronaires, et pourquoi pas les vaisseaux du foie qui ont par contre une hiérarchie beaucoup plus importante que les coronaires (nombre de nœuds de bifurcations beaucoup plus grand). Cela permettrait de montrer l'adaptabilité de notre algorithme et sa robustesse.

ANNEXE A

INTERFACE GRAPHIQUE

Comme expliqué au cours des précédents chapitres, une application disposant d'une interface graphique (GUI) a été développée afin de réaliser la mise en correspondance d'une séquence d'arbres coronaires 3D (à partir des contours vasculaires et de leurs squelettes) et de déterminer ensuite l'angle de vue optimale pour un objet d'intérêt. Cette application, encore en cours de développement, est réalisée en langage C++ avec la bibliothèque Qt⁷ couplé à VTK⁸ (*Visualisation Toolkit*).

L'interface graphique, comme montré dans la Figure 64, est séparée en plusieurs parties. La première partie située à gauche concerne la visualisation des arbres coronaires. Seuls deux onglets sont pour l'instant disponibles. Le premier appelé « *Setup Window* » permet de visualiser les données de la séquence 3D+T préalablement chargées (surfaces des contours vasculaires et squelettes). Grâce à un *slider* (curseur de déplacement), les arbres de la séquence sont affichés indépendamment un par un. L'utilisateur peut interagir avec les arbres (sélectionner des branches de l'arbre ou changer l'angle de visualisation par exemple). Toujours sur le même onglet, une modélisation simplifiée d'un C-arm et de la table est affichée (réalisation sous le logiciel Blender⁹) permettant de positionner un volume et de le visualiser selon une incidence donnée. Le deuxième onglet « *Matching Window* » permet de visualiser l'arbre mosaïque (section 3.2) calculé après la mise en correspondance de tous les arbres de la séquence. Une superposition de tous les volumes est également disponible.

⁷ <http://www.qt.io/>

⁸ <http://www.vtk.org/>

⁹ <https://www.blender.org/>

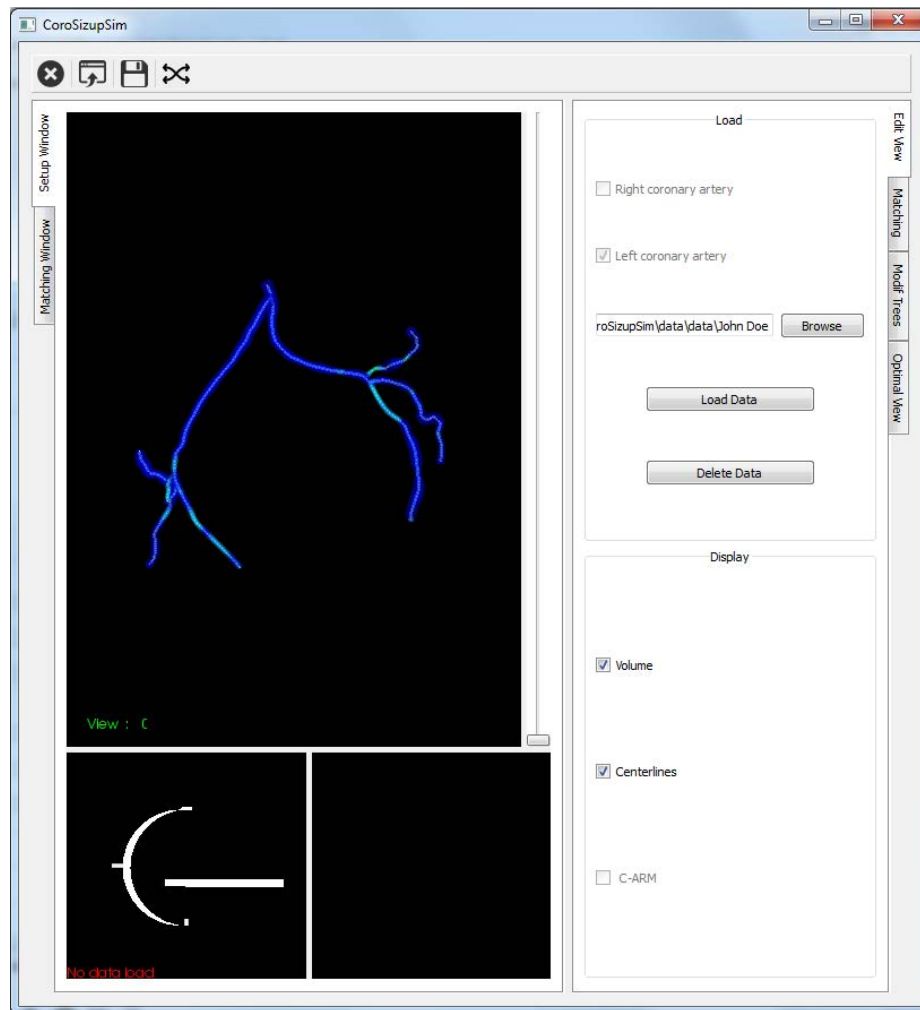


Figure 64: Interface graphique de l'application développée permettant de charger les données, de visualiser l'arbre en 3D sous différentes incidences et d'exécuter les différents algorithmes programmés au cours de cette thèse.

La seconde partie de l'interface correspond au panneau permettant de gérer tous les paramètres des différents algorithmes proposés. Cette partie est également séparée grâce à des onglets (Figure 65).

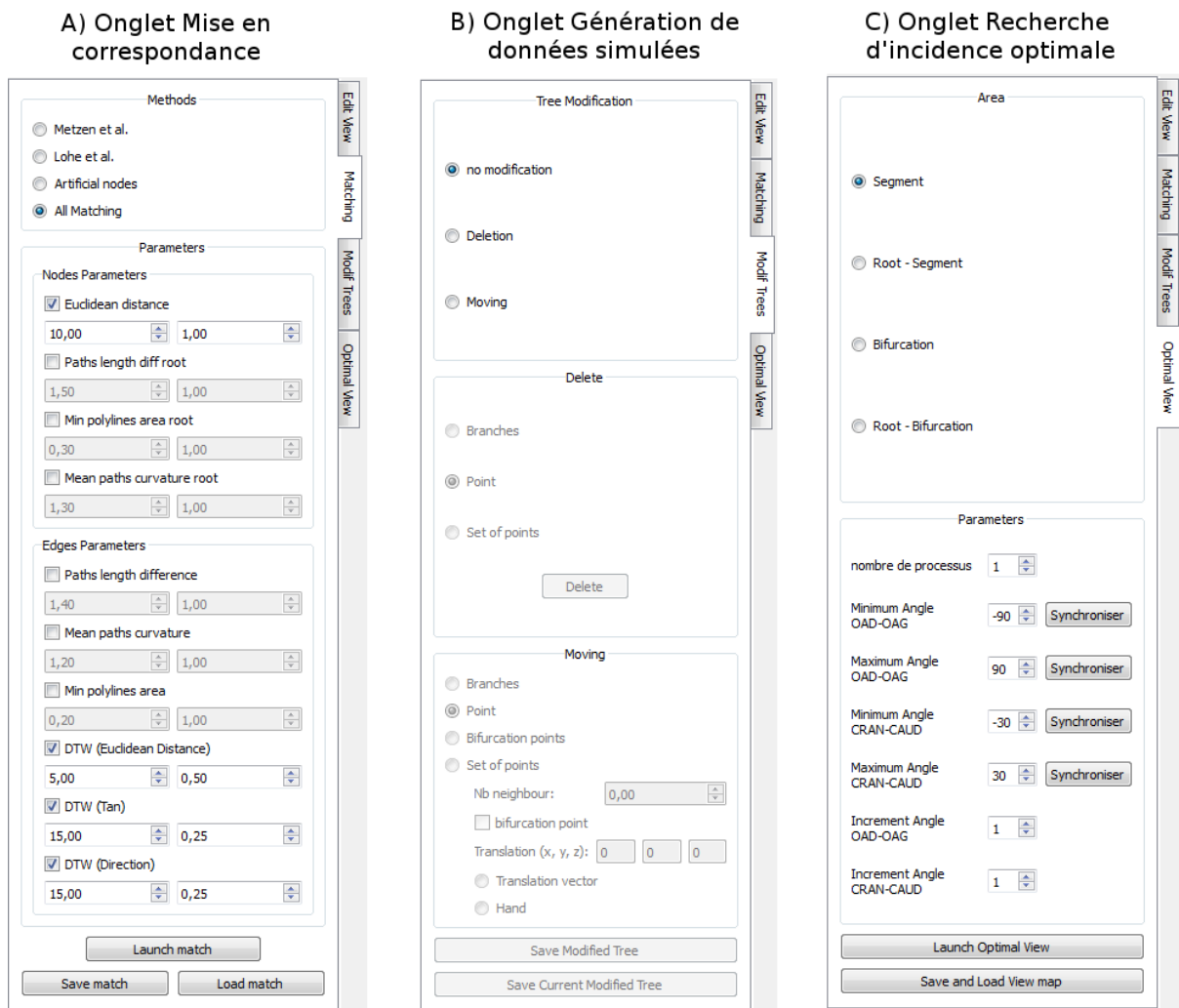


Figure 65: Capture d'écran de trois onglets de l'interface permettant d'exécuter les algorithmes développés pour la mise en correspondance (A) et la recherche d'incidence optimale (C). L'onglet (B) permet de modifier interactivement les données réelles pour générer des données simulées.

Le premier onglet « *Edit View* » permet de charger la séquence d'arbres coronaires et de gérer l'affichage de ses arbres soit les meshes, soit les squelettes (Figure 64). L'utilisateur choisi s'il veut afficher les arbres coronaires droits et/ou gauches de la séquence ainsi que le chemin de son dossier dans l'ordinateur. La deuxième partie de l'onglet décrit « *Edit View* » permet de gérer l'affichage des données venant d'être chargées. L'utilisateur peut choisir d'afficher ou non les lignes centrales ou le volume de l'arbre coronaire associé. Cependant, le programme ne crée ni les volumes des arbres coronaires, ni leurs lignes centrales (section 1.2.4). Ces données sont considérées comme disponibles. La segmentation ou reconstruction 3D des arbres coronaires sont stockées dans des fichiers, de même pour l'extraction des lignes centrales. Dans notre cas, pour une série donnée, l'utilisateur dispose de plusieurs dossiers comme nous le montre la Figure 66-A). Ces dossiers représentent chacun une phase d'un cycle cardiaque. Chaque dossier contient les fichiers DICOM, les informations de volumes de

l'arbre coronaires droit et gauche ou encore leurs lignes centrales. La Figure 66-B) nous montre les différents fichiers. La segmentation des images Scanner et la création de ces différents fichiers ont été réalisés par Jérôme Velut dans le cadre du projet EuHeart [VLPG10]. Le fichier *.mhd* représente le fichier DICOM du patient et le fichier *.vtp* représente quant à lui les caractéristiques des arbres :

- la ligne centrale de l'arbre coronaire gauche (*lca_centerline.vtp*),
- la ligne centrale de l'arbre coronaire droit (*rca_centerline.vtp*),
- la modélisation 3D des vaisseaux pour l'arbre coronaire gauche (*lca_lumen.vtp*),
- la modélisation 3D des vaisseaux pour l'arbre coronaire droit (*rca_lumen.vtp*).

Nom	Modifié le	Type	A)	
00	22/03/2016 14:29	Dossier de fichiers		
10	22/03/2016 14:29	Dossier de fichiers		
20	22/03/2016 14:29	Dossier de fichiers		
30	22/03/2016 14:29	Dossier de fichiers		
40	22/03/2016 14:29	Dossier de fichiers		
50	22/03/2016 14:29	Dossier de fichiers		
60	22/03/2016 14:29	Dossier de fichiers		
70	22/03/2016 14:29	Dossier de fichiers		
80	22/03/2016 14:29	Dossier de fichiers		
90	22/03/2016 14:29	Dossier de fichiers		

Nom	Modifié le	Type	Taille	B)
00.raw	26/06/2012 17:16	Fichier RAW	45 312 Ko	
00_dicom.mhd	26/06/2012 17:16	Fichier MHD	1 Ko	
lca_centerline.vtp	19/05/2011 11:09	Fichier VTP	30 Ko	
lca_lumen.vtp	19/05/2011 11:10	Fichier VTP	856 Ko	
rca_centerline.vtp	19/05/2011 10:54	Fichier VTP	22 Ko	
rca_lumen.vtp	19/05/2011 10:54	Fichier VTP	607 Ko	

Figure 66: Exemple de fichier contenant les données réelles. A) les dossiers correspondent aux différentes phases du cycle cardiaque. B) les fichiers contiennent le scanner coronarien CT ainsi que le volume (surfaces des contours vasculaires) et les lignes centrales des arbres droits et gauches.

Le deuxième onglet « *Matching* » concerne la mise en correspondance entre arbres, étapes nécessaire pour pouvoir calculer un angle de vue optimale. Dans cet onglet (Figure 65-A)), l'utilisateur peut choisir plusieurs méthodes de mise en correspondance.

- « *Metzen et al.* » [MKSZ09] correspond à la méthode d'appariement entre deux arbres successifs basée sur la construction d'un graphe d'association et la recherche de la clique maximum et adapté à nos données (section 3.1.2).
- « *Lohe et al.* » [LKZP07] correspond également à une méthode de mise en correspondance entre deux arbres successifs basée cette fois ci sur la construction d'un arbre de recherche et également adapté à nos données (section 3.1.1).
- « *Artificial nodes* » correspond à l'algorithme proposé en section 3.1.3 basée sur la construction d'un graphe d'association et d'extraction de clique maximum, mais également sur l'insertion de nœud artificiel avec une correction de conflit.

- « *All matching* » propose de mettre en correspondance tous les arbres de la séquence grâce à la construction d'un arbre mosaïque (section 3.2). En sélectionnant ce type de méthode, le résultat peut être visualisé dans l'onglet « *Matching window* » décrit précédemment.

Par la suite, l'utilisateur peut choisir différents paramètres pour la méthode de mise en correspondance sélectionnée. Les différents critères de similarités proposés et testés dans le chapitre 3 et chapitre 5 peuvent être combinés (par exemple les critères issus des coordonnées, de la tangente et de la direction des nœuds à partir desquels une distance euclidienne est calculée basée sur le Dynamic Time Warping), leur valeur de sélectivité (seuil) choisie et leur pondération également. L'interface propose par défaut pour chaque algorithme les meilleures combinaisons proposées dans le chapitre 5. Pendant le calcul de la mise en correspondance, le programme enregistre simultanément plusieurs fichiers permettant d'enregistrer à la fois les arbres de la séquence modifiés avec l'insertion des nœuds artificiels et également avec l'information d'appariement des différents segments (label identique pour chaque arête appariée). Cet enregistrement pourra par exemple être utile pour pouvoir charger directement la séquence appariée dans l'interface.

Une fois la mise en correspondance accomplie, l'utilisateur peut utiliser l'onglet « *Optimal view* » pour déterminer l'incidence optimale pour un objet sélectionné. La Figure 65-C) nous montre les différents paramètres possibles. Comme expliqué dans le chapitre 4, l'utilisateur peut sélectionner via l'interface plusieurs types d'objet : un segment, un segment multiple (de la racine à la région d'intérêt), une bifurcation. Sur l'onglet de visualisation d'arbre (onglet « *Setup window* »), l'utilisateur choisit un des arbres de la séquence et avec le clic droit de la souris sélectionne le segment d'intérêt grâce au point de contrôle des lignes centrale affichée (zone de lésion). Si l'objet est une bifurcation, l'utilisateur devra sélectionner le segment correspondant à la branche parente. Pour l'instant, l'utilisateur ne peut pas déterminer la longueur de sa bifurcation ou du segment sélectionné. Mais prochainement, ce paramètre sera disponible dans l'interface. Grâce à la mise en correspondance, le programme détermine automatiquement tous les segments, les segments multiples ou les bifurcations similaires dans les autres arbres de la séquence. L'objet peut ainsi être sélectionné depuis n'importe quel arbre coronaire.

Une fois l'objet sélectionné, l'utilisateur peut lancer les calculs concernant la détermination de l'angle optimal. Dans le même onglet, plusieurs paramètres sont à déterminer :

- le nombre de threads pour une possible implémentation en parallèle (disponible dans la bibliothèque Boost¹⁰),
- le maximum et le minimum de l'angle OAD/OAG,
- le maximum et le minimum de l'angle CRAN/CAU, et
- le pas d'incrémentation des angles.

La recherche de d'incidence optimale sera alors calculée dans l'intervalle sélectionné par cette interface graphique pour les deux angles. Une fois l'angle de vue optimale obtenu, le programme enregistre une image des différents arbres coronaires sous cette incidence, ainsi que des fichiers avec les différentes valeurs des cartes calculées pour chaque critère et phase du cycle cardiaque (voir section 5.3 pour un exemple de carte). Par la suite, nous souhaitons intégrer à l'interface les différents résultats obtenus pour ce dernier algorithme afin de permettre à l'utilisateur de visualiser en même temps l'arbre coronaires 3D sous différentes phases du cycle cardiaque, l'angle optimal proposé par notre algorithme ainsi que la carte globale selon trois méthodes d'obtention de l'angle de vue optimale.

L'interface graphique, en plus d'exécuter nos différents algorithmes, permet également de modifier des données réelles afin de générer des données synthétiques. L'onglet « *Modif Trees* » (Figure 65-B)) permet ainsi, dans un premier temps, de supprimer des branches entières ou des portions de branches à partir de la visualisation des lignes centrales dans l'interface. Chaque arbre de la séquence peut être ainsi modifié. Dans un deuxième temps, les points des lignes centrales ou les points de bifurcation peuvent être déplacés influencés à moindre mesure par cette translation. Plusieurs solutions de déplacement sont disponibles. L'utilisateur peut tout d'abord choisir de déplacer une branche entière ou de déplacer uniquement un point de la ligne centrale. Ensuite, comme dit précédemment, l'utilisateur peut déplacer un point de bifurcation. Le voisinage de chaque branche liée à cette bifurcation sera également déplacé. De même si l'utilisateur choisi de translater un ensemble de points de la ligne centrale, le nombre de points impactés par cette modification doit être indiqué. L'utilisateur peut effectuer la translation soit avec le clic gauche de la souris directement sur le point de l'arbre considéré ou alors définir la valeur de la translation directement depuis l'interface. Pour finir, l'arbre modifié peut être ensuite enregistré sous le même format *.vtp* que nos données réelles. L'interface ne permet pas de modifier le volume des arbres coronaires mais uniquement les lignes centrales. La déformation des meshes pourra être envisagée par la suite. Les données synthétiques ne sont alors utilisées que pour tester les différents algorithmes de mise en correspondance.

¹⁰ <http://www.boost.org/>

BIBLIOGRAPHIE

- [AMCG99] AUSIELLO, GIORGIO ; MARCHETTI-SPACCAMELA, ALBERTO ; CRESCENZI, PIERLUIGI ; GAMBOSI, GIORGIO ; PROTASI, MARCO ; KANN, VIGGO: *Complexity and Approximation* : Springer Berlin Heidelberg, 1999 — ISBN 978-3-642-63581-6
- [AMFE07] AGUILAR, WENDY ; MARTINEZ-PEREZ, M. ELENA ; FRAUEL, YANN ; ESCOLANO, FRANCISCO ; LOZANO, MIGUEL ANGEL ; ESPINOSA-ROMERO, ARTURO: Graph-based methods for retinal mosaicing and vascular characterization. In: *Graph-Based Representations in Pattern Recognition* : Springer, 2007, pp. 25–36
- [BaBH11] BALASUNDARAM, BALABHASKAR ; BUTENKO, SERGIY ; HICKS, ILLYA V.: Clique Relaxations in Social Network Analysis: The Maximum k -Plex Problem. In: *Operations Research* vol. 59 (2011), Nr. 1, pp. 133–142
- [BBPP99] BOMZE, IMMANUEL M. ; BUDINICH, MARCO ; PARDALOS, PANOS M. ; PELILLO, MARCELLO: The Maximum Clique Problem. In: DU, D.-Z. ; PARDALOS, P. M. (Hrsg.): *Handbook of Combinatorial Optimization*. Boston, MA : Springer US, 1999 — ISBN 978-1-4419-4813-7, pp. 1–74
- [BeHa13] BENLIC, UNA ; HAO, JIN-KAO: Breakout Local Search for maximum clique problems. In: *Computers & Operations Research* vol. 40 (2013), Nr. 1, pp. 192–206
- [BeMc92] BESL, PAUL J. ; MCKAY, NEIL D.: Method for registration of 3-D shapes. In: SCHENKER, P. S. (Hrsg.): , 1992, pp. 586–606
- [BeMV15] BENSEGHIR, THOMAS ; MALANDAIN, GRÉGOIRE ; VAILLANT, RÉGIS: A tree-topology preserving pairing for 3D/2D registration. In: *International Journal of Computer Assisted Radiology and Surgery, Medical Imaging 2006: Physiology, Function, and Structure from Medical Images*. vol. 10 (2015), Nr. 6, pp. 913–923
- [Beng02] BENGOTXEA, ENDIKA: *Inexact graph matching using estimation of distribution algorithms*. Paris, Ecole Nationale Supérieure des Télécommunications, 2002
- [Bens15] BENSEGHIR, THOMAS: *Topology Preserving Vascular Registration: Application to Percutaneous Coronary Intervention*, Université de Nice-Sophia Antipolis, 2015
- [Berg73] BERGE, CLAUDE: *Graphes et hypergraphes* : Dunod, 1973 — ISBN 978-2-04-009755-4
- [BiGS10] BI, JINGYING ; GRASS, MICHAEL ; SCHÄFER, DIRK: Optimization of acquisition trajectories for 3D rotational coronary venography. In: *International Journal of Computer Assisted Radiology and Surgery* vol. 5 (2010), Nr. 1, pp. 19–28
- [Bill03] BILLE, PHILIP: *Tree edit distance, alignment distance and inclusion* (IT University Technical Report Series Nr. TR-2003-23). The IT University of Copenhagen : Citeseer, 2003
- [Bill05] BILLE, PHILIP: A survey on tree edit distance and related problems. In: *Theoretical Computer Science* vol. 337 (2005), Nr. 1–3, pp. 217–239

- [BJGB09] BOUSSE, A. ; JIAN ZHOU ; GUANYU YANG ; BELLANGER, J.-J. ; TOUMOULIN, C.: Motion Compensated Tomography Reconstruction of Coronary Arteries in Rotational Angiography. In: *IEEE Transactions on Biomedical Engineering* vol. 56 (2009), Nr. 4, pp. 1254–1257
- [BLWH06] BÜLOW, THOMAS ; LORENZ, CRISTIAN ; WIEMKER, RAFAEL ; HONKO, JANETT: Point based methods for automatic bronchial tree matching and labeling. In: MANDUCA, A. ; AMINI, A. A. (Hrsg.): . vol. 6143. San Diego, CA, 2006, pp. 614300–614300–10
- [BMSG14] BAKA, N. ; METZ, C.T. ; SCHULTZ, C.J. ; VAN GEUNS, R.-J. ; NIESSEN, W.J. ; VAN WALSUM, T.: Oriented Gaussian Mixture Models for Nonrigid 2D/3D Coronary Artery Registration. In: *IEEE Transactions on Medical Imaging* vol. 33 (2014), Nr. 5, pp. 1023–1034
- [BMVA06] BLONDEL, C. ; MALANDAIN, G. ; VAILLANT, R. ; AYACHE, N.: Reconstruction of coronary arteries from a single rotational X-ray projection sequence. In: *IEEE Transactions on Medical Imaging* vol. 25 (2006), Nr. 5, pp. 653–663
- [BoBP06] BOGINSKI, VLADIMIR ; BUTENKO, SERGIY ; PARDALOS, PANOS M.: Mining market data: A network approach. In: *Computers & Operations Research* vol. 33 (2006), Nr. 11, pp. 3171–3184
- [Boll13] BOLLOBAS, BELA: *Modern Graph Theory* : Springer Science & Business Media, 2013 — ISBN 978-1-4612-0619-4
- [BoMu76] BONDY, JOHN ADRIAN ; MURTY, UPPALURI SIVA RAMACHANDRA: *Graph theory with applications*. vol. 290 : London : Macmillan, 1976
- [BuAl83] BUNKE, H ; ALLERMANN, G: Inexact graph matching for structural pattern recognition. In: *Pattern Recognition Letters* vol. 1 (1983), Nr. 4, pp. 245–253
- [BuBü93] BUNKE, H ; BÜHLER, U: Applications of approximate string matching to 2D shape recognition. In: *Pattern Recognition* vol. 26 (1993), Nr. 12, pp. 1797–1812
- [BuCs95] BUNKE, H. ; CSIRIK, J.: Parametric string edit distance and its application to pattern recognition. In: *IEEE Transactions on Systems, Man, and Cybernetics* vol. 25 (1995), Nr. 1, pp. 202–206
- [BuMe97] BUNKE, H. ; MESSMER, B. T.: Recent Advances in Graph Matching. In: *International Journal of Pattern Recognition and Artificial Intelligence* vol. 11 (1997), Nr. 01, pp. 169–203
- [Bunk00] BUNKE, H.: Recent developments in graph matching. In: *15th International Conference on Pattern Recognition, 2000. Proceedings*. vol. 2. Barcelona, 2000, pp. 117–124
- [Bunk97] BUNKE, H.: On a relation between graph edit distance and maximum common subgraph. In: *Pattern Recognition Letters* vol. 18 (1997), Nr. 8, pp. 689–694
- [Busy06] BUSYGIN, STANISLAV: A new trust region technique for the maximum weight clique problem. In: *Discrete Applied Mathematics, International Symposium on Combinatorial Optimization CO'02*. vol. 154 (2006), Nr. 15, pp. 2080–2096
- [Bute03] BUTENKO, SERGIY: *Maximum Independent Set and Related Problems, with Applications*. Gainesville, FL, USA, University of Florida, 2003
- [CAMS05] CHARNOZ, ARNAUD ; AGNUS, VINCENT ; MALANDAIN, GRÉGOIRE ; SOLER, LUC ; TAJINE, MOHAMED: Tree Matching Applied to Vascular System. In: BRUN, L. ; VENTO, M. (Hrsg.): *Graph-Based*

Representations in Pattern Recognition, Lecture Notes in Computer Science. vol. 3434 : Springer, 2005 — ISBN 978-3-540-25270-2, pp. 183–192

- [CaPa90] CARRAGHAN, RANDY ; PARDALOS, PANOS M.: An exact algorithm for the maximum clique problem. In: *Operations Research Letters* vol. 9 (1990), Nr. 6, pp. 375–382
- [Cast10] CASTRO, MIGUEL: *Navigation endovasculaire augmentée : mise en correspondance d'images pré- et per-opératoires.*, Université de Rennes 1, 2010
- [Cast12] CASTA, CHRISTOPHER: *Estimation 3D conjointe forme/structure/mouvement dans des séquences dynamiques d'images. Application à l'obtention de modèles cardiaques patients spécifiques anatomiques et fonctionnels.*, Université de Lyon 1 - INSA, 2012
- [CCMM06] CHOE, TAE EUN ; COHEN, ISAAC ; MUNWAI, LEE ; MEDIONI, GÉRARD: Optimal Global Mosaic Generation from Retinal Images. In: , 2006 — ISBN 978-0-7695-2521-1, pp. 681–684
- [CFSV04] CONTE, D. ; FOGGIA, P. ; SANSONE, C. ; VENTO, M.: Thirty years of graph matching in pattern recognition. In: *International Journal of Pattern Recognition and Artificial Intelligence* vol. 18 (2004), Nr. 03, pp. 265–298
- [ÇGGF16] ÇIMEN, SERKAN ; GOOYA, ALI ; GRASS, MICHAEL ; FRANGI, ALEJANDRO F.: Reconstruction of Coronary Arteries from X-ray Angiography: A Review. In: *Medical Image Analysis* vol. 32 (2016), pp. 46–68
- [Char07] CHARNOZ, ARNAUD: *Recalage d'organes intra-patient à partir de l'étude de leur réseau vasculaire: application au foie*, Université Strasbourg 1 - Louis Pasteur, 2007
- [ChCa00] CHEN, S.J. ; CARROLL, J.D.: 3-D reconstruction of coronary arterial tree to optimize angiographic visualization. In: *IEEE Transactions on Medical Imaging* vol. 19 (2000), Nr. 4, pp. 318–336
- [ChCa98] CHEN, S.-Y. JAMES ; CARROLL, JOHN D.: Computer assisted coronary intervention by use of on-line 3D reconstruction and optimal view strategy. In: *Medical Image Computing and Computer-Assisted Intervention — MICCAI'98*. vol. 1496, 1998 — ISBN 978-3-540-65136-9, pp. 377–385
- [ChHa01] CHANDRA, BARUN ; HALLDÓRSSON, MAGNÚS M: Approximation Algorithms for Dispersion Problems. In: *Journal of Algorithms* vol. 38 (2001), Nr. 2, pp. 438–465
- [ChHW87] CHAMS, M. ; HERTZ, A. ; DE WERRA, D.: Some experiments with simulated annealing for coloring graphs. In: *European Journal of Operational Research* vol. 32 (1987), Nr. 2, pp. 260–266
- [ChRa03] CHUI, HAILI ; RANGARAJAN, ANAND: A new point matching algorithm for non-rigid registration. In: *Computer Vision and Image Understanding* vol. 89 (2003), Nr. 2-3, pp. 114–141
- [CLZW14] CHEN, XIAODONG ; LI, JUNWEI ; ZHANG, ZHAOXIA ; WANG, YI ; JIA, ZHONGWEI ; PU, KUI ; YU, DAOYIN: A new algorithm for calculating optimal viewing angles in coronary angiography based on 4-D cardiac computed tomography. In: *Journal of X-Ray Science and Technology* (2014), Nr. 2, pp. 137–145
- [CSRT02] CAN, A. ; STEWART, C.V. ; ROYSAM, B. ; TANENBAUM, H.L.: A feature-based technique for joint, linear estimation of high-order image-to-mosaic transformations: mosaicing the curved

- human retina. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 24 (2002), Nr. 3, pp. 412–419
- [CTFC98] CHEN, S.W. ; TUNG, S.T. ; FANG, C.Y. ; CHERNG, SHEN ; JAIN, ANIL K.: Extended Attributed String Matching for Shape Recognition. In: *Computer Vision and Image Understanding* vol. 70 (1998), Nr. 1, pp. 36–50
- [DuRG94] DUMAY, A.C.M. ; REIBER, J.H.C. ; GERBRANDS, J.J.: Determination of optimal angiographic viewing angles: basic principles and evaluation study. In: *IEEE Transactions on Medical Imaging* vol. 13 (1994), Nr. 1, pp. 13–24
- [Fan12] FAN, WENFEI: Graph pattern matching revised for social network analysis. In: : ACM Press, 2012 — ISBN 978-1-4503-0791-8, pp. 8
- [FeNT15] FEUILLÂTRE, HÉLÈNE ; NUNES, JEAN-CLAUDE ; TOUMOULIN, CHRISTINE: An improved graph matching algorithm for the spatio-temporal matching of a coronary artery 3D tree sequence. In: *IRBM* vol. 36 (2015), Nr. 6, pp. 329–334
- [FeNT16a] FEUILLÂTRE, HÉLÈNE ; NUNES, JEAN-CLAUDE ; TOUMOULIN, CHRISTINE: Mosaic-based spatio-temporal matching of non-isomorphic coronary tree. In: *submit to PMB* (2016)
- [FeNT16b] FEUILLÂTRE, HÉLÈNE ; NUNES, JEAN-CLAUDE ; TOUMOULIN, CHRISTINE: Pair-wise tree matching algorithm applied to non-isomorphic coronary trees. In: *submit to EURASIP Journal on Image and Video Processing* (2016)
- [FiBo81] FISCHLER, MARTIN A. ; BOLLES, ROBERT C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: *Communications of the ACM* vol. 24 (1981), Nr. 6, pp. 381–395
- [FNLT15] FEUILLÂTRE, HÉLÈNE ; NUNES, JEAN-CLAUDE ; LI, SI ; TOUMOULIN, CHRISTINE: Coronary Tree Matching Based on Association Graphs with Artificial Nodes. In: *6th European Conference of the International Federation for Medical and Biological Engineering (MBEC)*. vol. 45. Cham : Springer International Publishing, 2015 — ISBN 978-3-319-11127-8, pp. 244–247
- [FrHW89] FRIDEN, C. ; HERTZ, A. ; DE WERRA, D.: STABULUS: A technique for finding stable sets in large graphs with tabu search. In: *Computing* vol. 42 (1989), Nr. 1, pp. 35–44
- [FSFR13] FISCHER, ANDREAS ; SUEN, CHING Y. ; FRINKEN, VOLKMAR ; RIESEN, KASPAR ; BUNKE, HORST: A Fast Matching Algorithm for Graph-Based Handwriting Recognition. In: *Graph-Based Representations in Pattern Recognition*. vol. 7877 : Springer Berlin Heidelberg, 2013 — ISBN 978-3-642-38220-8, pp. 194–203
- [FSNB13] FEUILLÂTRE, HÉLÈNE ; SANOKHO, CUNKA BASSIROU ; NUNES, JEAN-CLAUDE ; BEDOSSA, MARC ; TOUMOULIN, CHRISTINE: Automatic determination of optimal view for the visualization of coronary lesions by rotational X-ray angiography. In: *IRBM* vol. 34 (2013), Nr. 4-5, pp. 291–295
- [GAGM09] GARCIA, JOEL A. ; AGOSTONI, PIERFRANCESCO ; GREEN, NATHAN E. ; MADDUX, JAMES T. ; CHEN, S.-Y. JAMES ; MESSENGER, JOHN C. ; CASSERLY, IVAN P. ; HANSGEN, ADAM ; U. A.: Rotational vs. standard coronary angiography: An image content analysis. In: *Catheterization and Cardiovascular Interventions* vol. 73 (2009), Nr. 6, pp. 753–761

- [GCHM05] GREEN, NATHAN E. ; CHEN, S.-Y. JAMES ; HANSGEN, ADAM R. ; MESSENGER, JOHN C. ; GROVES, BERTRON M. ; CARROLL, JOHN D.: Angiographic views used for percutaneous coronary interventions: A three-dimensional analysis of physician-determined vs. computer-generated views. In: *Catheterization and Cardiovascular Interventions* vol. 64 (2005), Nr. 4, pp. 451–459
- [GeSS93] GENDREAU, MICHEL ; SORIANO, PATRICK ; SALVAIL, LOUIS: Solving the maximum clique problem using a tabu search approach. In: *Annals of Operations Research* vol. 41 (1993), Nr. 4, pp. 385–403
- [GJBG10] GUANYU YANG ; JIAN ZHOU ; BOULMIER, D. ; GARCIA, M.-P. ; LIMIN LUO ; TOUMOULIN, C.: Characterization of 3-D Coronary Tree Motion From MSCT Angiography. In: *IEEE Transactions on Information Technology in Biomedicine* vol. 14 (2010), Nr. 1, pp. 101–106
- [GMCK09] GARCIA, JOEL A. ; MOVASSAGHI, BABAK ; CASSERLY, IVAN P. ; KLEIN, ANDREW J. ; JAMES CHEN, S.-Y. ; MESSENGER, JOHN C. ; HANSGEN, ADAM ; WINK, ONNO ; U. A.: Determination of optimal viewing regions for X-ray coronary angiography based on a quantitative analysis of 3D reconstructed models. In: *The International Journal of Cardiovascular Imaging* vol. 25 (2009), Nr. 5, pp. 455–462
- [Grah06] GRAHAM, MICHAEL W.: *Robust Graph-Theoretic Methods for Matching and Labeling Anatomical Trees*, The Pennsylvania State University, 2006
- [GrHi06a] GRAHAM, M.W. ; HIGGINS, W.E.: Optimal graph-theoretic approach to 3D anatomical tree matching. In: *3rd IEEE International Symposium on Biomedical Imaging: Nano to Macro*. Arlington, VA, 2006 — ISBN 0-7803-9576-X, pp. 109–112
- [GrHi06b] GRAHAM, MICHAEL W. ; HIGGINS, WILLIAM E.: Globally optimal model-based matching of anatomical trees. In: *Medical Imaging 2006: Image Processing*. vol. 6144, 2006, pp. 614415–614415–15
- [GrLC04] GROSSO, A. ; LOCATELLI, M. ; CROCE, F. DELLA: Combining Swaps and Node Weights in an Adaptive Greedy Approach for the Maximum Clique Problem. In: *Journal of Heuristics* vol. 10 (2004), Nr. 2, pp. 135–152
- [GRSM99] GIBSON, C. MICHAEL ; RYAN, KATHRYN ; SPARANO, ANTHONY ; MOYNIHAN, JOHN L. ; RIZZO, MICHAEL ; KELLEY, MICHAEL ; MARBLE, SUSAN J. ; LAHAM, ROGER ; U. A.: Angiographic methods to assess human coronary angiogenesis. In: *American heart journal* vol. 137 (1999), Nr. 1, pp. 169–179
- [Gu05] GU, JIA: *Mise en correspondance spatio-temporelle par chaînes attribuées et splines plaques minces pour la reconstruction 3-D des coronaires en imagerie R-X*, Université de Rennes 1, 2005
- [GuDa08] GUTURU, P. ; DANTU, R.: An Impatient Evolutionary Algorithm With Probabilistic Tabu Search for Unified Solution of Some NP-Hard Problems in Graph and Set Theory via Clique Finding. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* vol. 38 (2008), Nr. 3, pp. 645–666
- [GVBL13] GARCIA, M-P ; VELUT, J. ; BOULMIER, D. ; LECLERCQ, C. ; GARREAU, M. ; HAIGRON, P. ; TOUMOULIN, C.: Coronary Vein Extraction in MSCT Volumes Using Minimum Cost Path and Geometrical Moments. In: *IEEE Journal of Biomedical and Health Informatics* vol. 17 (2013), Nr. 2, pp. 336–345

- [GVLW06] GOLLAPUDI, RAGHAVA R. ; VALENCIA, RAFAEL ; LEE, STEVE S. ; WONG, GARRETT B. ; TEIRSTEIN, PAUL S. ; PRICE, MATTHEW J.: Utility of three-dimensional reconstruction of coronary angiography to guide percutaneous coronary intervention. In: *Catheterization and Cardiovascular Interventions* vol. 69 (2006), Nr. 4, pp. 479–482
- [GXTL10] GAO, XINBO ; XIAO, BING ; TAO, DACHENG ; LI, XUELONG: A survey of graph edit distance. In: *Pattern Analysis and Applications* vol. 13 (2010), Nr. 1, pp. 113–129
- [HADA99] HUANG, J. ; ABENDSCHEIN, D. ; DAVILA-ROMAN, V. G. ; AMINI, A. A.: Spatio-temporal tracking of myocardial deformations with a 4-D B-spline model from tagged MRI. In: *IEEE Transactions on Medical Imaging* vol. 18 (1999), Nr. 10, pp. 957–972
- [Hans05] HANSSON, GÖRAN K.: Inflammation, Atherosclerosis, and Coronary Artery Disease. In: *New England Journal of Medicine* vol. 352 (2005), Nr. 16, pp. 1685–1695
- [Heck78] HECKEL, PAUL: A technique for isolating differences between files. In: *Communications of the ACM* vol. 21 (1978), Nr. 4, pp. 264–268
- [HEMP99] HARIS, K. ; EFSTRATIADIS, S.N. ; MAGLAVERAS, N. ; PAPPAS, C. ; GOURASSAS, J. ; LOURIDAS, G.: Model-based morphological segmentation and labeling of coronary angiograms. In: *IEEE Transactions on Medical Imaging* vol. 18 (1999), Nr. 10, pp. 1003–1015
- [HrMc90] HRECHAK, ANDREW K. ; MCHUGH, JAMES A.: Automated fingerprint recognition using structural matching. In: *Pattern Recognition* vol. 23 (1990), Nr. 8, pp. 893–904
- [HSDG08] HANSIS, E. ; SCHAFER, D. ; DOSSEL, O. ; GRASS, M.: Evaluation of Iterative Sparse Object Reconstruction From Few Projections for 3-D Rotational Coronary Angiography. In: *IEEE Transactions on Medical Imaging* vol. 27 (2008), Nr. 11, pp. 1548–1555
- [JaSa01] JAGOTA, ARUN ; SANCHIS, LAURA A.: Adaptive, Restart, Randomized Greedy Heuristics for Maximum Clique. In: *Journal of Heuristics* vol. 7 (2001), Nr. 6, pp. 565–585
- [Karp72] KARP, RICHARD M.: *Reducibility among combinatorial problems* : Springer, 1972
- [KePa01] KEOGH, EAMONN J. ; PAZZANI, MICHAEL J.: Derivative Dynamic Time Warping. In: *Proceedings of the 2001 SIAM International Conference on Data Mining*. Philadelphia, PA : Society for Industrial and Applied Mathematics, 2001 — ISBN 978-0-89871-495-1, pp. 1–11
- [KeRa05] KEOGH, EAMONN ; RATANAMAHATANA, CHOTIRAT ANN: Exact indexing of dynamic time warping. In: *Knowledge and Information Systems* vol. 7 (2005), Nr. 3, pp. 358–386
- [KKN06] KAFTAN, JENS N. ; KIRALY, ATILLA P. ; NAIDICH, DAVID P. ; NOVAK, CAROL L.: A novel multipurpose tree and path matching algorithm with application to airway trees. In: *Medical Imaging 2006: Physiology, Function, and Structure from Medical Images*. vol. 6143. San Diego, CA, 2006, pp. 61430N
- [KMJK08] KITSLAAR, PIETER H. ; MARQUERING, HENK A. ; JUKEMA, WOUTER J. ; KONING, GERHARD ; NIEBER, MAARTEN ; VOSSEPOEL, ALBERT M. ; BAX, JEROEN J. ; REIBER, JOHAN H. C.: Automated determination of optimal angiographic viewing angles for coronary artery bifurcations from CTA data. In: *Medical Imaging 2008: Visualization, Image-guided Procedures, and Modeling*, 2008, pp. 69181J

- [KoBa84] KOCHANEK, DORIS H. U. ; BARTELS, RICHARD H.: Interpolating splines with local tension, continuity, and bias control. In: *11th annual conference on Computer graphics and interactive techniques (SIGGRAPH '84)*, 1984 — ISBN 978-0-89791-138-2, pp. 33–41
- [Kuml04] KUMLANDER, DENISS: A new exact algorithm for the maximum-weight clique problem based on a heuristic vertex-coloring and a backtrack search. In: *Proc. 5th Int'l Conf. on Modelling, Computation and Optimization in Information Systems and Management Sciences*, 2004, pp. 202–208
- [KuND02] KUON, EBERHARD ; NIEDERST, PIERRE N. ; DAHM, JOHANNES B.: Usefulness of rotational spin for coronary angiography in patients with advanced renal insufficiency. In: *The American journal of cardiology* vol. 90 (2002), Nr. 4, pp. 369–373
- [LABF09] LESAGE, DAVID ; ANGELINI, ELSA D. ; BLOCH, ISABELLE ; FUNKA-LEA, GARETH: A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes. In: *Medical Image Analysis* vol. 13 (2009), Nr. 6, pp. 819–845
- [Libb02] LIBBY, PETER: Inflammation in atherosclerosis. In: *Nature* vol. 420 (2002), Nr. 6917, pp. 868–874
- [LiQu10] LI, CHU-MIN ; QUAN, ZHE: An Efficient Branch-and-Bound Algorithm Based on MaxSAT for the Maximum Clique Problem. In: *Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*, 2010, pp. 128–133
- [LiRi13] LIVI, LORENZO ; RIZZI, ANTONELLO: The graph matching problem. In: *Pattern Analysis and Applications* vol. 16 (2013), Nr. 3, pp. 253–283
- [LKZP07] LOHE, TOBIAS ; KRÖGER, TIM ; ZIDOWITZ, STEPHAN ; PEITGEN, HEINZ-OTTO ; JIANG, XIAOYI: Hierarchical matching of anatomical trees for medical image registration. In: *Medical Biometrics* : Springer, 2007, pp. 224–231
- [LoBe04] LOHOU, CHRISTOPHE ; BERTRAND, GILLES: A 3D 12-subiteration thinning algorithm based on P-simple points. In: *Discrete Applied Mathematics* vol. 139 (2004), Nr. 1-3, pp. 171–195
- [Loho01] LOHOU, CHRISTOPHE: *Contribution à l'analyse topologique des images : étude d'algorithmes de squelettisation pour images 2D et 3D selon une approche topologie digitale ou topologie discrète* : Marne-la-Vallée, 2001
- [LuSz97] LUCZAK, T. ; SZPANKOWSKI, W.: A suboptimal lossy data compression based on approximate pattern matching. In: *IEEE Transactions on Information Theory* vol. 43 (1997), Nr. 5, pp. 1439–1451
- [MaAY10] MALOD-DOGNIN, NOËL ; ANDONOV, RUMEN ; YANEV, NICOLA: Maximum Cliques in Protein Structure Comparison. In: *Experimental Algorithms: 9th International Symposium (SEA 2010)*. vol. 6049, 2010 — ISBN 978-3-642-13192-9, pp. 106–117
- [MaBP14] MASLOV, EVGENY ; BATSYN, MIKHAIL ; PARDALOS, PANOS M.: Speeding up branch and bound algorithms for solving the maximum clique problem. In: *Journal of Global Optimization* vol. 59 (2014), Nr. 1, pp. 1–21
- [Maca02] MACAMBIRA, ELDER MAGALHÃES: An Application of Tabu Search Heuristic for the Maximum Edge-Weighted Subgraph Problem. In: *Annals of Operations Research* vol. 117 (2002), Nr. 1/4, pp. 175–190

- [MaGD10] MARTÍ, RAFAEL ; GALLEGRO, MICAEL ; DUARTE, ABRAHAM: A branch and bound algorithm for the maximum diversity problem. In: *European Journal of Operational Research* vol. 200 (2010), Nr. 1, pp. 36–44
- [Mann99] MANNINO, CARLO: An Augmentation Algorithm for the Maximum Weighted Stable Set Problem. In: *Computational Optimization and Applications* vol. 14 (1999), Nr. 3, pp. 367–381
- [MeBu98] MESSMER, B.T. ; BUNKE, H.: A new algorithm for error-tolerant subgraph isomorphism detection. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 20 (1998), Nr. 5, pp. 493–504
- [MGDP13] MARTÍ, RAFAEL ; GALLEGRO, MICAEL ; DUARTE, ABRAHAM ; PARDO, EDUARDO G.: Heuristics and metaheuristics for the maximum diversity problem. In: *Journal of Heuristics* vol. 19 (2013), Nr. 4, pp. 591–615
- [MKSZ07] METZEN, JAN HENDRIK ; KRÖGER, TIM ; SCHENK, ANDREA ; ZIDOWITZ, STEPHAN ; PEITGEN, HEINZ-OTTO ; JIANG, XIAOYI: Matching of tree structures for registration of medical images. In: *Graph-Based Representations in Pattern Recognition* : Springer, 2007, pp. 13–24
- [MKSZ09] METZEN, JAN HENDRIK ; KRÖGER, TIM ; SCHENK, ANDREA ; ZIDOWITZ, STEPHAN ; PEITGEN, HEINZ-OTTO ; JIANG, XIAOYI: Matching of anatomical tree structures for registration of medical images. In: *Image and Vision Computing* vol. 27 (2009), Nr. 7, pp. 923–933
- [MoSt65] MOTZKIN, THEODORE S. ; STRAUS, ERNST G.: Maxima for graphs and a new proof of a theorem of Turán. In: *Canadian Journal of Mathematics* vol. 17 (1965), Nr. 4, pp. 533–540
- [MTLP12] MARKELJ, P. ; TOMAŽEVIČ, D. ; LIKAR, B. ; PERNUŠ, F.: A review of 3D/2D registration methods for image-guided interventions. In: *Medical Image Analysis* vol. 16 (2012), Nr. 3, pp. 642–661
- [Müll07] MÜLLER, MEINARD: Dynamic Time Warping. In: *Information Retrieval for Music and Motion* : Springer Berlin Heidelberg, 2007 — ISBN 978-3-540-74047-6, pp. 69–84
- [MWMG04] MADDUX, JAMES T. ; WINK, ONNO ; MESSENGER, JOHN C. ; GROVES, BERTRON M. ; LIAO, ROBERT ; STRZELCZYK, JODI ; CHEN, SHIUH-YUNG ; CARROLL, JOHN D.: Randomized study of the safety and clinical utility of rotational angiography versus standard angiography in the diagnosis of coronary artery disease. In: *Catheterization and Cardiovascular Interventions* vol. 62 (2004), Nr. 2, pp. 167–174
- [Myer94] MYERS, EUGENE W.: Algorithmic Advances for Searching Biosequence Databases. In: *Computational Methods in Genome Research*. Boston, MA : Springer US, 1994 — ISBN 978-1-4613-6042-1, pp. 121–135
- [MyXu10] MYRONENKO, ANDRIY ; XUBO SONG: Point Set Registration: Coherent Point Drift. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 32 (2010), Nr. 12, pp. 2262–2275
- [NeWu70] NEEDLEMAN, SAUL B. ; WUNSCH, CHRISTIAN D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. In: *Journal of Molecular Biology* vol. 48 (1970), Nr. 3, pp. 443–453

- [Öste02] ÖSTERGÅRD, PATRIC R.J.: A fast algorithm for the maximum clique problem. In: *Discrete Applied Mathematics* vol. 120 (2002), Nr. 1-3, pp. 197–207
- [Öste99] ÖSTERGÅRD, PATRIC R.J.: A New Algorithm for the Maximum-Weight Clique Problem. In: *Electronic Notes in Discrete Mathematics* vol. 3 (1999), pp. 153–156
- [Ouki15] OUKILI, AHMED: *Reconstruction statique 3D à partir d'un faible nombre de projections. Application : Coronarographie RX rotationnelle*, Université de Rennes 1, 2015
- [Oyar15] OYARZUN LAURA, CRISTINA: *Graph-matching and FEM-based Registration of Computed Tomographies for Outcome Validation of Liver Interventions*, Technische Universität Darmstadt, 2015
- [OyDr11] OYARZUN LAURA, CRISTINA ; DRECHSLER, KLAUS: Computer assisted matching of anatomical vessel trees. In: *Computers & Graphics* vol. 35 (2011), Nr. 2, pp. 299–311
- [PaLP96] PARK, KYUNGCHUL ; LEE, KYUNGSIK ; PARK, SUNGSOO: An extended formulation approach to the edge-weighted maximal clique problem. In: *European Journal of Operational Research* vol. 95 (1996), Nr. 3, pp. 671–682
- [Palu07] PALUBECKIS, GINTARAS: Iterated tabu search for the maximum diversity problem. In: *Applied Mathematics and Computation* vol. 189 (2007), Nr. 1, pp. 371–383
- [PaXu94] PARDALOS, PANOS M. ; XUE, JUE: The maximum clique problem. In: *Journal of Global Optimization* vol. 4 (1994), Nr. 3, pp. 301–328
- [PaYB12] PATTILLO, JEFFREY ; YOUSSEF, NATALY ; BUTENKO, SERGIY: Clique Relaxation Models in Social Network Analysis. In: *Handbook of Optimization in Complex Networks*. vol. 58 : Springer New York, 2012 — ISBN 978-1-4614-0856-7, pp. 143–162
- [Peli02] PELILLO, MARCELLO: Matching free trees, maximal cliques, and monotone game dynamics. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 24 (2002), Nr. 11, pp. 1535–1541
- [Peli99] PELILLO, MARCELLO: Replicator Equations, Maximal Cliques, and Graph Isomorphism. In: *Neural Computation* vol. 11 (1999), Nr. 8, pp. 1933–1955
- [PeSZ99] PELILLO, MARCELLO ; SIDDIQI, KALEEM ; ZUCKER, STEVEN W.: Matching hierarchical structures using association graphs. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 21 (1999), Nr. 11, pp. 1105–1120
- [Peti03] PETITJEAN, CAROLINE: *Recalage non rigide d'images par approches variationnelles statistiques. Application à l'analyse et à la modélisation de la fonction myocardique en IRM*, Université René Descartes - Paris V, 2003
- [Pier00] PIER, JEAN-PAUL: *Development of Mathematics 1950-2000* : Springer Science & Business Media, 2000 — ISBN 978-3-7643-6280-5
- [Pinh13] PINHEIRO, MIGUEL AMAVEL: Graph and point cloud registration for tree-like structures: survey and evaluation. In: *International Student Conference on Electrical Engineering*, 2013

- [PISr00] PLAMONDON, R. ; SRIHARI, S.N.: Online and off-line handwriting recognition: a comprehensive survey. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 22 (2000), Nr. 1, pp. 63–84
- [Pros12] PROSSER, PATRICK: Exact Algorithms for Maximum Clique: A Computational Study. In: *Algorithms* vol. 5 (2012), Nr. 4, pp. 545–587
- [PuHo06] PULLAN, WAYNE ; HOOS, HOLGER H.: Dynamic Local Search for the Maximum Clique Problem. In: *Journal of Artificial Intelligence Research* vol. 25 (2006), Nr. 1, pp. 159–185
- [Pull08] PULLAN, WAYNE: Approximating the maximum vertex/edge weighted clique using local search. In: *Journal of Heuristics* vol. 14 (2008), Nr. 2, pp. 117–134
- [PuMB11] PULLAN, WAYNE ; MASCIA, FRANCO ; BRUNATO, MAURO: Cooperating local search for the maximum clique problem. In: *Journal of Heuristics* vol. 17 (2011), Nr. 2, pp. 181–199
- [RaJu93] RABINER, LAWRENCE ; JUANG, BIING-HWANG: *Fundamentals of Speech Recognition* : Prentice Hall, 1993 — ISBN 0-13-015157-2
- [RaKe05] RATANAMAHATANA, CHOTIRAT ANN ; KEOGH, EAMONN: Three Myths about Dynamic Time Warping Data Mining. In: *Proceedings of the 2005 SIAM International Conference on Data Mining*. Philadelphia, PA : Society for Industrial and Applied Mathematics, 2005 — ISBN 978-0-89871-593-4, pp. 506–510
- [RiSC12] RIVEST-HENAULT, D. ; SUNDAR, H. ; CHERIET, M.: Nonrigid 2D/3D Registration of Coronary Artery Models With Live Fluoroscopy for Guidance of Cardiac Interventions. In: *IEEE Transactions on Medical Imaging* vol. 31 (2012), Nr. 8, pp. 1557–1572
- [RLBP10] ROHKOHL, C. ; LAURITSCH, G. ; BILLER, L. ; PRÜMMER, M. ; BOESE, J. ; HORNEGGER, J.: Interventional 4D motion estimation and reconstruction of cardiac vasculature without motion periodicity assumption. In: *Medical Image Analysis* vol. 14 (2010), Nr. 5, pp. 687–694
- [RMNA04] RAMAN, SUBHA V. ; MORFORD, RONALD ; NEFF, MATTHEW ; ATTAR, TALAL T. ; KUKIELKA, GILBERT ; MAGORIEN, RAYMOND D. ; BUSH, CHARLES A.: Rotational X-ray coronary angiography. In: *Catheterization and Cardiovascular Interventions* vol. 63 (2004), Nr. 2, pp. 201–207
- [SaCh07] SALVADOR, STAN ; CHAN, PHILIP: Toward accurate dynamic time warping in linear time and space. In: *Intelligent Data Analysis* vol. 11 (2007), Nr. 5, pp. 561–580
- [SaCh78] SAKOE, H. ; CHIBA, S.: Dynamic programming algorithm optimization for spoken word recognition. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* vol. 26 (1978), Nr. 1, pp. 43–49
- [SAHN98] SATO, Y. ; ARAKI, T. ; HANAYAMA, M. ; NAITO, H. ; TAMURA, S.: A viewpoint determination system for stenosis diagnosis and quantification in coronary angiographic image acquisition. In: *IEEE Transactions on Medical Imaging* vol. 17 (1998), Nr. 1, pp. 121–137
- [SAOM07] SILVA, GEIZA C. ; DE ANDRADE, MARCOS R. Q. ; OCHI, LUIZ S. ; MARTINS, SIMONE L. ; PLASTINO, ALEXANDRE: New heuristics for the maximum diversity problem. In: *Journal of Heuristics* vol. 13 (2007), Nr. 4, pp. 315–336
- [Scha07] SCHAEFFER, SATU ELISA: Graph clustering. In: *Computer Science Review* vol. 1 (2007), Nr. 1, pp. 27–64

- [Schä14] SCHÄLING, BORIS: *The Boost C++ libraries: [introduces 72 libraries with more than 430 examples]*. 2nd English ed. Aufl. Laguna Hills, Calif : XML Press, 2014 — ISBN 978-1-937434-36-6
- [ScNG09] SCHOONENBERG, GERT ; NEUBAUER, ANNE ; GRASS, MICHAEL: Three-Dimensional Coronary Visualization, Part 2: 3D Reconstruction. In: *Cardiology Clinics* vol. 27 (2009), Nr. 3, pp. 453–465
- [SeKK04] SEBASTIAN, T.B. ; KLEIN, P.N. ; KIMIA, B.B.: Recognition of shapes by editing their shock graphs. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 26 (2004), Nr. 5, pp. 550–571
- [Selk77] SELKOW, STANLEY M.: The tree-to-tree editing problem. In: *Information Processing Letters* vol. 6 (1977), Nr. 6, pp. 184–186
- [ShZh90] SHASHA, DENNIS ; ZHANG, KAIZHONG: Fast algorithms for the unit cost editing distance between trees. In: *Journal of Algorithms* vol. 11 (1990), Nr. 4, pp. 581–621
- [SiGu06] SINGH, ALOK ; GUPTA, ASHOK KUMAR: A hybrid heuristic for the maximum clique problem. In: *Journal of Heuristics* vol. 12 (2006), Nr. 1-2, pp. 5–22
- [SiLL01] SIEK, JEREMY G. ; LEE, LIE-QUAN ; LUMSDAINE, ANDREW: *Boost Graph Library, The: User Guide and Reference Manual*. 1st. Aufl. : Addison-Wesley Professional., 2001 — ISBN 978-0-201-72914-6
- [SLBK04] SCHENKER, ADAM ; LAST, MARK ; BUNKE, HORST ; KANDEL, ABRAHAM: Classification of web documents using graph matching. In: *International Journal of Pattern Recognition and Artificial Intelligence* vol. 18 (2004), Nr. 03, pp. 475–496
- [SMKM05] SIANOS, GEORGIOS ; MOREL, MARIE-ANGÈLE ; KAPPETEIN, ARIE PIETER ; MORICE, MARIE-CLAUDE ; COLOMBO, ANTONIO ; DAWKINS, KEITH ; VAN DEN BRAND, MARCEL ; VAN DYCK, NIC ; U. A.: The SYNTAX Score: an angiographic tool grading the complexity of coronary artery disease. In: *EuroIntervention* vol. 1 (2005), Nr. 2, pp. 219–227
- [SMRH13] SAN SEGUNDO, PABLO ; MATIA, FERNANDO ; RODRIGUEZ-LOSADA, DIEGO ; HERNANDO, MIGUEL: An improved bit parallel exact maximum clique algorithm. In: *Optimization Letters* vol. 7 (2013), Nr. 3, pp. 467–479
- [SoFe06] SOLNON, CHRISTINE ; FENET, SERGE: A study of ACO capabilities for solving the maximum clique problem. In: *Journal of Heuristics* vol. 12 (2006), Nr. 3, pp. 155–180
- [SPSK15] SERRADELL, EDUARD ; PINHEIRO, MIGUEL AMAVEL ; SZNITMAN, RAPHAEL ; KYBIC, JAN ; MORENO-NOGUER, FRANCESC ; FUA, PASCAL: Non-Rigid Graph Registration Using Active Testing Search. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 37 (2015), Nr. 3, pp. 625–638
- [SRLG11] SERRADELL, EDUARD ; ROMERO, ADRIANA ; LETA, RUBEN ; GATTA, CARLO ; MORENO-NOGUER, FRANCESC: Simultaneous correspondence and non-rigid 3D reconstruction of the coronary tree from single X-ray images. In: *2011 International Conference on Computer Vision, 2011* — ISBN 978-1-4577-1102-2, pp. 850–857
- [SSGD03] SUNDAR, H. ; SILVER, D. ; GAGVANI, N. ; DICKINSON, S.: Skeleton based shape matching and retrieval. In: *Shape Modeling International, 2003* — ISBN 978-0-7695-1909-8, pp. 130–139

- [SSKL13] SHREINER, D. ; SELLERS, G. ; KESSENICH, J. M. ; LICEA-KANE, B. ; KHONOS OpenGL ARB WORKING GROUP (Hrsg.): *OpenGL programming guide: the official guide to learning OpenGL, version 4.3*. Eighth edition. Aufl. Upper Saddle River, NJ : Addison-Wesley, 2013 — ISBN 978-0-321-77303-6
- [SSRB05] SHECHTER, G. ; SHECHTER, B. ; RESAR, J.R. ; BEYAR, R.: Prospective motion correction of X-ray images for coronary interventions. In: *IEEE Transactions on Medical Imaging* vol. 24 (2005), Nr. 4, pp. 441–450
- [Step94] STEPHEN, GRAHAM A.: *String Searching Algorithms* : World Scientific, 1994 — ISBN 978-981-4501-86-6
- [SuYa98] SUGANTHAN, P. N. ; YAN, H.: Recognition of handprinted Chinese characters by constrained graph matching. In: *Image and Vision Computing* vol. 16 (1998), Nr. 3, pp. 191–201
- [TaCh06] TANG, W. H. ; CHUNG, ALBERT C.S.: Cerebral Vascular Tree Matching of 3D-RA Data Based on Tree Edit Distance. In: *Third International Workshop on Medical Imaging and Augmented Reality, Lecture Notes in Computer Science*. vol. 4091, 2006 — ISBN 978-3-540-37220-2, pp. 116–123
- [Tai79] TAI, KUO-CHUNG: The Tree-to-Tree Correction Problem. In: *Journal of the ACM* vol. 26 (1979), Nr. 3, pp. 422–433
- [Tava12] TAVARD, FRANÇOIS: *Recalage et fusion d'informations multimodales pour l'optimisation de la thérapie de resynchronisation cardiaque*, Université de Rennes 1, 2012
- [THKW11] TU, SHENGXIAN ; HAO, PEIYUAN ; KONING, GERHARD ; WEI, XIANGLONG ; SONG, XUDONG ; CHEN, AIHUA ; REIBER, JOHAN H.C.: In vivo assessment of optimal viewing angles from X-ray coronary angiography. In: *EuroIntervention* vol. 7 (2011), Nr. 1, pp. 112–120
- [TJHO12] TU, SHENGXIAN ; JING, JING ; HOLM, NIELS R. ; ONSA, KEVIN ; ZHANG, TAO ; ADRIAENSSENS, TOM ; DUBOIS, CHRISTOPHE ; DESMET, WALTER ; U. A.: In vivo assessment of bifurcation optimal viewing angles and bifurcation angles by three-dimensional (3D) quantitative coronary angiography. In: *The International Journal of Cardiovascular Imaging* vol. 28 (2012), Nr. 7, pp. 1617–1625
- [TLGD05] TURGEON, GUY-ANNE ; LEHMANN, GLEN ; GUIRAUDON, GERARD ; DRANGOVA, MARIA ; HOLDSWORTH, DAVID ; PETERS, TERRY: 2D-3D registration of coronary angiograms for cardiac procedure planning and guidance. In: *Medical Physics* vol. 32 (2005), Nr. 12, pp. 3737
- [TMPH05] TSCHIRREN, J. ; MCLENNAN, G. ; PALAGYI, K. ; HOFFMAN, E.A. ; SONKA, M.: Matching and anatomical labeling of human airway tree. In: *IEEE Transactions on Medical Imaging* vol. 24 (2005), Nr. 12, pp. 1540–1547
- [ToSe03] TOMITA, ETSUJI ; SEKI, TOMOKAZU: An Efficient Branch-and-Bound Algorithm for Finding a Maximum Clique. In: *Discrete Mathematics and Theoretical Computer Science*. vol. 2731 : Springer Berlin Heidelberg, 2003 — ISBN 978-3-540-40505-4, pp. 278–289
- [Tu12] TU, SHENGXIAN: *Three-dimensional quantitative coronary angiography and the registration with intravascular ultrasound and optical coherence tomography*, Leiden University, 2012
- [Ullm76] ULLMANN, J. R.: An Algorithm for Subgraph Isomorphism. In: *Journal of the ACM* vol. 23 (1976), Nr. 1, pp. 31–42

- [Vali02] VALIENTE, GABRIEL: *Algorithms on Trees and Graphs* : Springer, 2002 — ISBN 978-3-540-43550-1
- [VLPG10] VELUT, J. ; LENTZ, P. A. ; PHILIPOT, C. ; GARCIA, M. P. ; TOUMOULIN, C.: A qualitative and quantitative study of coronary artery MRA. In: *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, 2010, pp. 5724–5727
- [WaFi74] WAGNER, ROBERT A. ; FISCHER, MICHAEL J.: The String-to-String Correction Problem. In: *Journal of the ACM* vol. 21 (1974), Nr. 1, pp. 168–173
- [WaHi07] WARREN, JEFFREY P. ; HICKS, ILLYA V.: *Combinatorial branch-and-bound for the maximum weight independent set problem*, 2007
- [WHGL14] WANG, YANG ; HAO, JIN-KAO ; GLOVER, FRED ; LÜ, ZHIPENG: A tabu search based memetic algorithm for the maximum diversity problem. In: *Engineering Applications of Artificial Intelligence* vol. 27 (2014), pp. 103–114
- [WKCC02] WINK, O. ; KEMKERS, R. ; CHEN, S. J. ; CARROLL, J. D.: Coronary Intervention Planning Using Hybrid 3D Reconstruction. In: *Medical Image Computing and Computer-Assisted Intervention — MICCAI 2002*. vol. 2488 : Springer Berlin Heidelberg, 2002 — ISBN 978-3-540-44224-0, pp. 604–611
- [WKCC03] WINK, ONNO ; KEMKERS, RICHARD ; CHEN, SHIUH-YUNG J ; CARROLL, JOHN D: Intra-procedural coronary intervention planning using hybrid 3-dimensional reconstruction techniques1. In: *Academic Radiology* vol. 10 (2003), Nr. 12, pp. 1433–1441
- [Wu10] WU, YINGHUI: *Extending graph homomorphism and simulation for real life graph matching*, University of Edinburgh, 2010
- [WuHa12] WU, QINGHUA ; HAO, JIN-KAO: Coloring large graphs based on independent set extraction. In: *Computers & Operations Research* vol. 39 (2012), Nr. 2, pp. 283–290
- [WuHa15a] WU, QINGHUA ; HAO, JIN-KAO: Solving the winner determination problem via a weighted maximum clique heuristic. In: *Expert Systems with Applications* vol. 42 (2015), Nr. 1, pp. 355–365
- [WuHa15b] WU, QINGHUA ; HAO, JIN-KAO: A review on algorithms for maximum clique problems. In: *European Journal of Operational Research* vol. 242 (2015), Nr. 3, pp. 693–709
- [WuHG12] WU, QINGHUA ; HAO, JIN-KAO ; GLOVER, FRED: Multi-neighborhood tabu search for the maximum weight clique problem. In: *Annals of Operations Research* vol. 196 (2012), Nr. 1, pp. 611–634
- [WYCA14] WANG, XUEHU ; YANG, JIAN ; CHEN, YANG ; AI, DANNI ; HU, YINING ; WANG, YONGTIAN: Optimal Viewing Angle Determination for Multiple Vessel Segments in Coronary Angiographic Image. In: *IEEE Transactions on Nuclear Science* vol. 61 (2014), Nr. 3, pp. 1290–1303
- [Xie12] XIE, LIZHE: *Reconstruction 3D des artères coronaires en imagerie rotationnelle RX*, Université de Rennes 1, 2012
- [YaSz07] YANG, QINGWU ; SZE, SING-HOI: Path Matching and Graph Matching in Biological Networks. In: *Journal of Computational Biology* vol. 14 (2007), Nr. 1, pp. 56–67

- [YBTS06] YANG, GUANYU ; BOUSSE, ALEXANDRE ; TOUMOULIN, CHRISTINE ; SHU, HUAZHONG: A multiscale tracking algorithm for the coronary extraction in MSCT angiography. In: *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE* : IEEE, 2006, pp. 3066–3069
- [ZaBV09] ZASLAVSKIY, M. ; BACH, F. ; VERT, J.-P.: Global alignment of protein-protein interaction networks by graph matching methods. In: *Bioinformatics* vol. 25 (2009), Nr. 12, pp. i259–1267
- [ZhQi11] ZHENG, SUN ; QI, YAN: Motion estimation of 3D coronary vessel skeletons from X-ray angiographic sequences. In: *Computerized Medical Imaging and Graphics* vol. 35 (2011), Nr. 5, pp. 353–364
- [ZMDG76] ZIR, LEONARD M. ; MILLER, STEPHEN W. ; DINSMORE, ROBERT E. ; GILBERT, J. P. ; HARTHORNE, J. W.: Interobserver variability in coronary angiography. In: *Circulation* vol. 53 (1976), Nr. 4, pp. 627–632

LISTE DES PUBLICATIONS

ARTICLES DE JOURNAUX

Feuillâtre H., Nunes J.-C., Toumoulin C., An improved graph matching algorithm for the spatio-temporal matching of a coronary artery 3D tree sequence, *IRBM*, 2015, 36(6), pp. 329-334.

Feuillâtre H., Sanokho B. Nunes J.-C., Bedossa M., Toumoulin C., Automatic determination of optimal view for the visualization of coronary lesions by rotational X-ray angiography, *IRBM*, 2013, 34(4), pp. 291-295.

ARTICLES DE JOURNAUX SOUMIS

Feuillâtre H., Nunes J.-C., Toumoulin C. Pair-wise tree matching algorithm applied to non-isomorphic coronary trees, soumis avril 2016, *EURASIP Journal on Image and Video Processing*.

Feuillâtre H., Nunes J.-C., Toumoulin C. Mosaic-based spatio-temporal matching of non-isomorphic coronary tree, soumis avril 2016, *Physics in Medicine and Biology (PMB)*.

Feuillâtre H., Nunes J.-C., Toumoulin C. Optimal view determination for coronary intervention planning from rotational X-ray angiography, en cours de rédaction.

CONFÉRENCES

Feuillâtre H., Li S., Nunes J.-C., Toumoulin C., Coronary Tree Matching Based on Association Graphs with Artificial Nodes, 6th European Conference of the International Federation for Medical and Biological Engineering (MBEC), Croatia, 2014, 45, pp. 244-247.

Feuillâtre H., Sanokho B. Nunes J.-C., Bedossa M., Toumoulin C., Détermination automatique de l'angle de vue optimale pour la visualisation des lésions coronaires en angiographie rotationnelle R-X, *RITS*, Bordeaux, 2013.

Résumé

Les travaux de cette thèse s'inscrivent dans le cadre du planning de traitements minimalement invasifs des lésions des artères coronaires. Le cardiologue réalise un examen coronarographique, puis dans la continuité, une angioplastie transluminale. L'angiographie rotationnelle à rayons X permet de visualiser sous différentes incidences 2D la lumière des artères coronaires sur plusieurs cycles cardiaques et aussi d'obtenir une reconstruction 3D+T des arbres coronaires. A partir de cette séquence, notre objectif est de déterminer automatiquement une incidence optimale 2D du segment sténosé compatible avec les angles du C-arm afin d'aider le cardiologue lors de l'intervention.

Différentes étapes sont considérées pour calculer la position angulaire optimale du C-arm. Afin de suivre la zone de lésion durant le cycle cardiaque, une première méthode est proposée pour mettre en correspondance tous les arbres de la séquence 3D+T. Tout d'abord, un appariement deux à deux des arbres successifs est réalisé afin de construire un arbre d'union. Ces derniers sont ensuite fusionnés afin d'obtenir un arbre mosaïque représentant l'arbre le plus complet de la séquence. L'utilisation de mesures de similarités géométriques et hiérarchiques ainsi que l'insertion de nœuds artificiels permet de prendre en compte les différents mouvements non-rigides des artères coronaires subits au cours du cycle cardiaque et les variations topologiques dû à leurs extractions. Cet appariement nous permet de proposer une deuxième méthode afin d'obtenir une vue angiographique 2D optimale de la zone de lésion tout le long du cycle cardiaque. Cette incidence est proposée spécifiquement pour trois types de région d'intérêt (segment unique, segment multiple ou bifurcation) et est calculée à partir de quatre critères (raccourcissement, chevauchement interne et externe ou angle d'ouverture de bifurcation). Une vue 2D déployée du segment projeté avec le moins de superposition avec les structures vasculaires avoisinantes est obtenue. Nous donnons également la possibilité au cardiologue d'avoir une incidence optimale privilégiant soit le déploiement du stent ou soit le guidage d'outils de la racine de l'arbre à la zone sténosée. Nos différents algorithmes ont été évalués sur une séquence réelle de 10 phases segmentées à partir d'un CT et de 41 séquences simulées.

Abstract

The thesis work deals with the planning of minimally invasive surgery of coronary artery lesions. The physician performs a coronarography following by a percutaneous transluminal angioplasty. The X-ray rotational angiography permits to visualize the lumen artery under different projection angles in several cardiac cycles. From these 2D projections, a 3D+T reconstruction of coronary arteries can be obtained. Our goal is to determine automatically from this 3D+T sequence, the optimal angiographic viewing angle of the stenotic segment.

Several steps are proposed to compute the optimal angular position of the C-arm. Firstly, a mosaic-based tree matching algorithm of the 3D+T sequence is proposed to follow the stenotic lesion in the whole cardiac cycle. A pair-wise inexact tree matching is performed to build a tree union between successive trees. Next, these union trees are merged to obtain the mosaic tree which represents the most complete tree of the sequence. To take into account the non-rigid movement of coronary arteries during the cardiac cycle and their topology variations due to the 3D reconstruction or segmentation, similarity measures based on hierarchical and geometrical features are used. Artificial nodes are also inserted. With this global tree sequence matching, we propose secondly a new method to determine the optimal viewing angle of the stenotic lesion throughout the cardiac cycle. This 2D angiographic view which is proposed for three regions of interest (single segment, multiple segment or bifurcation) is computed from four criteria: the foreshortening, the external and internal overlap and the bifurcation opening angle rates. The optimal view shows the segment in its most extended and unobstructed dimension. This 2D view can be optimal either for the deployment of the stent or for the catheter guidance (from the root to the lesion). Our different algorithms are evaluated on real sequence (CT segmentation) and 41 simulated sequences.