Volumetric tracking of 3D deformable shapes

Benjamin Allain

▶ **To cite this version:**

Benjamin Allain. Volumetric tracking of 3D deformable shapes. Computational Geometry [cs.CG]. Université Grenoble Alpes, 2017. English. NNT : 2017GREAM017 . tel-01681320v2

HAL Id: tel-01681320

https://theses.hal.science/tel-01681320v2

Submitted on 12 Jan 2018

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.

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Informatique**

Arrêté ministériel : 7 Août 2006

Présentée par

Benjamin Allain

Thèse dirigée par **Edmond Boyer**
et codirigée par **Jean-Sébastien Franco**

préparée au sein **Inria Grenoble, Laboratoire Jean Kuntzmann**
et de **Ecole Doctorale MSTII**

Suivi volumétrique de formes 3D non rigides

Volumetric Tracking of 3D Deformable Shapes

Thèse soutenue publiquement le **31 mars 2017**,
devant le jury composé de :

Monsieur Adrien Bartoli
Professeur à l'Université d'Auvergne, Président
Madame Lourdes Agapito
Professeur à University College London, Rapporteur
Monsieur Michael M. Bronstein
Professeur associé à Università della Svizzera Italiana, Rapporteur
Monsieur Edmond Boyer
Directeur de Recherche à Inria, Directeur de thèse
Monsieur Jean-Sébastien Franco
Maître de Conférences à Grenoble INP, Co-Directeur de thèse

Abstract

In this thesis we propose algorithms for tracking 3D deformable shapes in motion from multiview video. Although series of reconstructed 3D shapes can be obtained by applying a static reconstruction algorithm to each temporal frame independently, such series do not represent motion. Instead, we want to provide a temporally coherent representation of the sequence of shapes resulting from temporal evolutions of a shape. Precisely, we want to represent the observed shape sequence as a 3D surface mesh whose vertices move in time but whose topology is constant.

In contrast with most existing approaches, we propose to represent the motion of inner shape volumes, with the aim of better accounting for the volumetric nature of the observed object. We provide a fully volumetric approach to the fundamental problems of deformable shape tracking, which are the association between corresponding shape elements and the deformation model. In particular, we extend to a volumetric shape representation the EM-ICP tracking framework and the association-by-detection strategy.

Furthermore, in order to better constrain the shape tracking problem, we propose a model for the temporal evolution of deformation. Our deformation model defines a shape space parametrized by variables that capture local deformation properties of the shape and whose values are automatically learned during the tracking process.

We validate our tracking algorithms on several multiview video sequences with ground truth (silhouette and marker-based tracking). Our results are better or comparable to state of the art approaches.

Finally, we show that volumetric tracking and the shape representation we choose can be leveraged for producing shape animations which combine captured and simulatated motion.

Résumé

Dans cette thèse nous proposons des algorithmes pour le suivi 3D du mouvement des objects déformables à partir de plusieurs caméras vidéo. Bien qu'une suite de reconstructions tridimensionnelles peut être obtenue par des méthodes de reconstruction statique, celle-ci ne représente pas le mouvement. Nous voulons produire une représentation temporellement cohérente de la suite de formes prises par l'object. Précisément, nous souhaitons représenter l'objet par une surface maillée 3D dont les sommets se déplacent au cours du temps mais dont la topologie reste identique.

Contrairement à beaucoup d'approches existantes, nous proposons de représenter le mouvement du volume intérieur des formes, dans le but de mieux représenter la nature volumétrique des objets. Nous traitons de manière volumétrique les problèmes fondamentaux du suivi déformable que sont l'association d'éléments semblables entre deux formes et la modélisation de la déformation. En particulier, nous adaptons au formes volumétriques les modèles d'association EM-ICP non-rigide ansi que l'association par détection par apprentissage automatique.

D'autre part, nous abordons la question de la modélisation de l'évolution temporelle de la déformation au cours d'une séquence dans le but de mieux contraindre le problème du suivi temporel. Pour cela, nous modélisons un espace de forme construit autour de propriétés de déformations locales que nous apprenons automatiqument lors du suivi.

Nous validons nos algorithmes de suivi sur des séquences vidéo multi-caméras avec vérité terrain (silhouettes et suivi par marqueurs). Nos résultats se révèlent meilleurs ou équivalents à ceux obtenus avec les méthodes de l'état de l'art.

Enfin, nous démontrons que le suivi volumétrique et la représentation que nous avons choisie permettent de produire des animations 3D qui combinent l'acquisition et la simulation de mouvement.

Acknowledgements

I thank my thesis commitee, especially the reviewers, for their insghtful comments about my work. I thank my PhD supervisors Jean-Sébastien Franco and Edmond Boyer for their scientific support, their availability and their involvement in my work. I thank my collaborators Tony Tung, Chun-Hao Huang, Slobodan Ilic, Li Wang, Franck Hetroy-Wheeler and Mickaël Heudre for the work we accomplished together. I thank the members of the Morpheo group, with whom I had insightful discussions as well as pleasant recreational times. I thank my family, especially my parents, for the support they provided.

Contents

Chapter 1

Introduction

1.1 Introduction

Many scenes of interest are dynamic scenes, often involving human actions or interactions. Capturing dynamic scenes has been done since the invention of video cameras, for a wide variety of applications including visual telecommunication, intangible cultural heritage preservation, scientific experimentation, and more generally for recording any visual event. Most applications concern visualization, analysis and edition of acquired dynamic scenes. With monocular video, which is the main acquisition technology, the projective nature of the image formation process removes much of the scene information, including 3D shape geometry and motion. On the contrary, 4D capture aims at reconstructing a much more informative representation of the scene that includes 3D shape geometry, appearance and 3D motion. This richer representation allows for new applications such as free-viewpoint video, precise analysis of 3D shape and motion, and direct edition of the 3D scenes instead of editing projected images.

Three-dimensional reconstruction of a dynamic scene may be performed as a temporal sequence of independent 3D scene reconstructions. However, this approach completely ignores the fact that this scene sequence represents the temporal evolution of physical objects, which is a highly constrained phenomenon. We can go far beyond this naive approach by considering a dynamic scene as the temporal evolution of shapes. By recovering the motion and deformation of the observed 3D shapes, we build a much more informative representation of the scene. Beside the application of motion analysis, motion information enables the exploitation of temporal redundancy, which is useful for improving the quality of the reconstruction (in terms of shape and appearance) which might have been altered by several factors (noise, occlusions, lighting, reconstruction artifact). Motion information is also crucial for compressing dynamic scene representations with motion compensation, which is often necessary for real-time applications.

In this thesis, we propose new algorithms for the shape tracking problem. In particular, we propose the use of volumetric shape representations. This chapter presents the problem of shape tracking, its key components and the problematics which motivate our work.

1.2 Problem

Reconstructing a static 3D shape from images taken from multiple viewpoints is a well-known problem for which efficient solutions already exist. When we consider a dynamic scene viewed by multiple synchronized video cameras, we can apply such a solution independently to each temporal frame of the synchronized videos, and obtain in result a sequence of 3D shapes representations.

When this shape sequence is visualized in time, the human brain perceives an illusion of motion, in the same way as when visualizing a sequence of video frames. However, in both cases, no motion information is present in the representation. At each frame, we know where is the information (ex: space occupancy or pixel color), but we do not know how it moves between frames. In particular, given a physical shape point at time t, we do not know where this physical point lies at any other time instant. Such a 3D shape sequence representation lacks *temporal coherence*.

In contrast, we want to build a temporally coherent shape sequence representation, where each shape of the sequence is expressed as a deformed version of a common shape reference, also called shape *template*. For example, if the common shape reference is represented as a surface mesh, we can define a temporally coherent mesh sequence by providing at each time instant a 3D location to each vertex of the shape reference. This is illustrated in Figure 1.1 where the undeformed template shape and its deformed versions have been colorized with the same vertex-based coloring in order to display the temporal coherence. Such a visualization is not possible for the input sequence, which lacks temporal coherence.

We pose the shape tracking problem as the production of a temporally coherent shape sequence from a temporally inconsistent shape sequence.

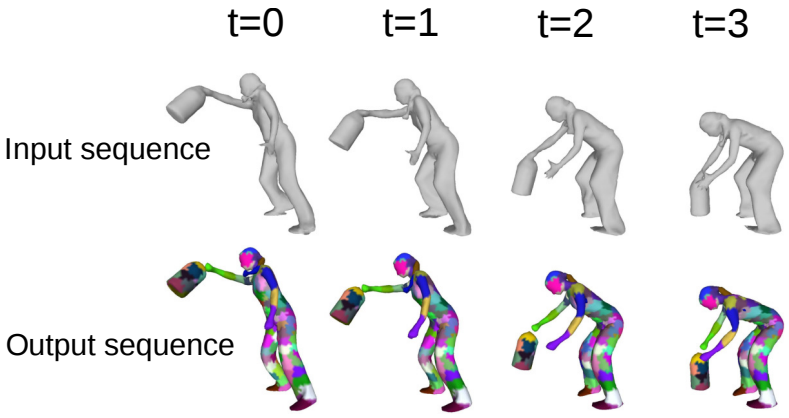


Figure 1.1 – Input and output of the shape tracking problem. Surface colors illustrate the temporal coherence of the output sequence.

1.3 Strategy Overview

Most deformable shape tracking approaches can be decomposed into four key components.

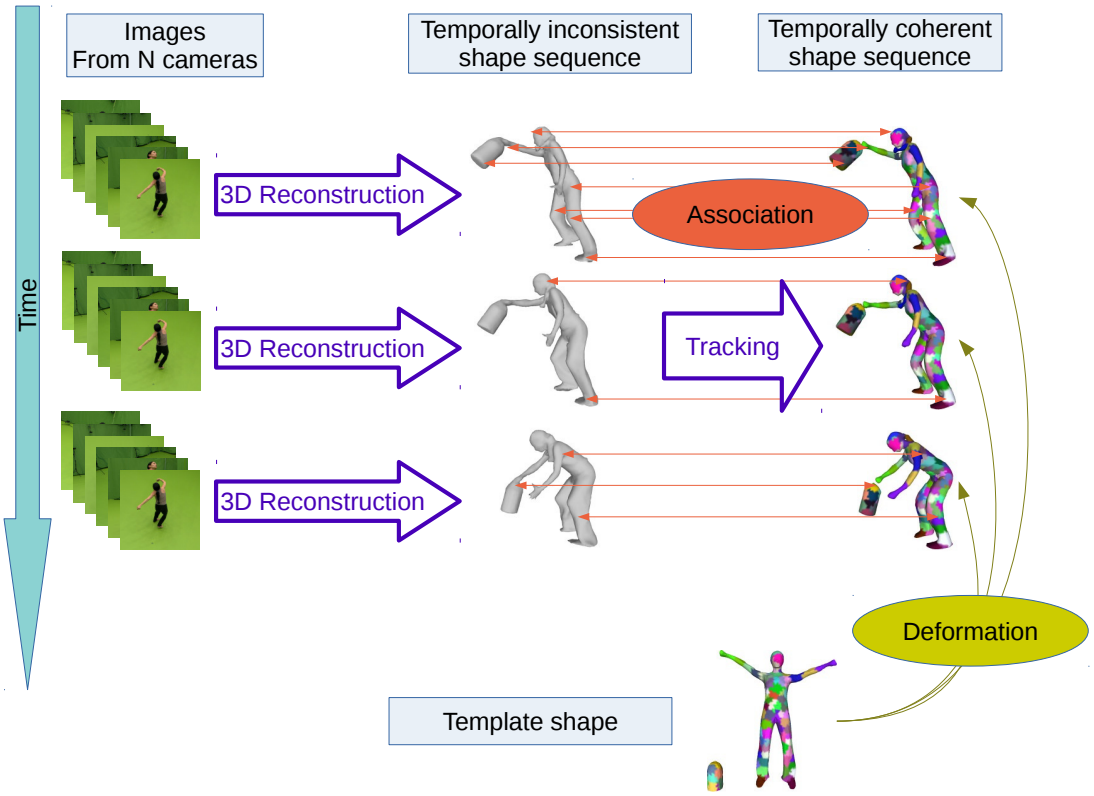


Figure 1.2 – Tracking strategy overview.

Shape reconstruction For obtaining a shape sequence from multiple videos, one generally applies a static reconstruction algorithm at each time frame independently. This process produces a temporally inconsistent shape sequence, which is an intermediate result of the entire pipeline. Several reconstruction strategies exist in the literature, and choosing one has an impact on other elements of our tracking strategy. First, the output shape representation has an impact on the design of our algorithm. Second, the static reconstruction algorithms produce geometric artefacts which perturb tracking algorithms.

Shape representation For manipulating and deforming shapes, algorithms require to represent shapes in memory. In our case, we need a shape representation for the output of the static reconstructions stage, and a shape representation for the temporally coherent shape sequence we wish to obtain.

We must choose a shape representation which is consistent with the true shape being represented. The chosen shape representations must also be convenient for expressing a deformation model as well as an association model, two concepts that we explain in the following paragraphs.

Association model In order to align a template shape with another shape, we need to align their corresponding elements. This requires to find correspondences between the two shapes. This association problem is highly ambiguous because it is high-

dimensional, and also because the visual information alone is usually insufficient for matching corresponding shape elements. For example, finding temporal correspondences on a textureless flat area of a shape is impossible without any additional prior about deformation.

Deformation model In order to represent deformation and motion, we define a deformation model, which has two roles. The first role is to provide a deformation representation. This is provided by means of a parametrization of shape deformation. The second role is to encode priors about the deformation in order to regularize the ambiguous association problem.

In this thesis, we propose to use new volumetric shape representations for shape tracking, and we propose new deformation and association models that are adapted to volumetric shape representations.

1.4 Strategy

In this section, we discuss in more detail the concepts we have introduced, and we explain our strategy with respect to each of them.

1.4.1 Shape Representation

The 3D shapes we want to represent are closed surfaces or volumes enclosed by a closed surface. There exists no representation that can exactly represent any shape with limited memory, but shapes with sufficient regularity can be reasonably approximated with discrete representations such as meshes. In practice, the fidelity of the shape approximation increases with the number of parameters of the representation, and so does the computational complexity of algorithms. Hence one needs to compromise between the computational complexity of tracking algorithms and the level of detail of the shape approximation.

We want to represent shapes whose topology is preserved under motion. This property can be enforced within the deformation model, but it requires to use a shape representation that represents shape topology. This criterion is in favor of mesh-based representations and it eliminates unstructured representations such as point clouds and polygon soups.

In this thesis, we are interested in representing volumetric shapes with volumetric representation in order to express the deformation model and the association model on volumetric elements. Hence we need a shape representation which decomposes the inner volume of shape in elementary cells. Many of the existing volumetric decompositions do not suit our requirements. Eulerian voxel grids are inconvenient for preserving topology between temporal frames, and gives irregular cells when they are intersected with the shape volume. The Delaunay tetrahedralization produces elongated tetrahedra between existing vertices, so it does not produce regular cells. We decide to use the centroidal Voronoi Tessellation [1], which partitions a shape volume in volumetric cells

with high regularity of shape, size and connectivity. Centroidal Voronoi tessellations can be obtained from watertight surface meshes [1] or from point clouds [2].

1.4.2 Shape Reconstruction

The first stage of shape tracking usually consists in capturing a sequence of shapes by performing series of static shape reconstructions. For example, this can be performed by multi-view stereo approaches [3]. Although static shape reconstruction has been widely studied, it remains a difficult problem for which no ideal method exists. The problem is difficult because it is high-dimensional and because the input data is noisy, incomplete and ambiguous. As a result, static shape reconstruction algorithms produce approximate reconstructions. Since shape tracking algorithms take reconstructed shape sequences as input, it is necessary to understand how they can be produced, how they are represented, and what artefacts or noise can be expected. We review the main reconstruction approaches in the next chapter.

1.4.3 Association Model

Shape alignment is driven by correspondences between shapes. Yet finding reliable correspondences from visual data is challenging. In images, the appearance information is subject to noise, occlusions including self-occlusions, it depends on the illumination, it is sometimes viewpoint-dependent and it is uninformative in the case of textureless surfaces. These issues makes the use of appearance difficult and unreliable.

Interestingly, shape tracking can be formulated as a purely geometric problem and yet provide state of the art results [4]. With the geometric tracking strategy, the appearance information is only used for static shape reconstruction, which is considered as a preprocessing stage. Hence, a geometric tracking algorithm takes a geometric shape sequence as its only input, which makes the approach agnostic to the static shape reconstruction pipeline. This versatility makes geometric tracking algorithms theoretically applicable to any shape sequence expressed in a compatible representation.

Finding shape correspondences from geometric information is also an ambiguous problem. A famous and simple approach, the Iterative Closest Point (ICP) [5], establishes correspondences by associating each point of one shape to the closest point on the other shape, and refines the association iteratively during the alignment process. In spite of its simplicity, this algorithm is sufficient for simple registration problems. It has been extended to non-rigid alignment, and has been provided robustness to noise and outliers in the EM-ICP framework thanks to probabilistic association models [6, 7, 4]. These surface tracking algorithms establish correspondences between surface points. However, the presence of strong noise in the reconstructed surface can lead to serious errors in the correspondence estimation. In this work, we extend these association models to volumetric shape representations, which yields stronger robustness to geometric and topological noise.

ICP-type algorithms tend to propagate and accumulate alignment errors during sequence processing. This is a consequence of the iterative nature of the ICP association model and the presence of local minima in the associated objective function: the

basin of convergence of the global minimum is usually small, sometimes smaller than inter-frame motion, which makes the algorithm likely to converge to suboptimal local minima.

In order to avoid local minima, other association models have been proposed for estimating correspondences directly without deforming any shape. In particular, successful results have been obtained by predicting surface correspondences with machine learning approaches [8]. Inspired from these works, we propose a direct association model for volumetric shapes represented by centroidal Voronoi tessellations.

1.4.4 Deformation Model

Deforming a shape representation requires to define a parametrization of deformation. The simplest deformation parametrization for a mesh is the location of each of its vertices. Such a parametrization is very permissive. In particular, it does not guarantee any local cohesion of motion and allows for physically unfeasible shapes. Using such a permissive parametrization for shape tracking requires the use of a deformation regularization which encodes deformation priors about the deforming object.

Deformation priors can also be directly embedded in the deformation parametrization. For example, the deformation of a human-shaped mesh can be parametrized by a kinematic skeleton [9]. While skeleton-based parametrizations drastically reduces the search space compared to free mesh deformation, they also oversimplify the actual deformation. Moreover, they are only applicable to articulated shapes.

In between the extreme cases we have just mentioned, there are deformation models which are general enough for deforming any surface or volume while enforcing priors of local cohesion and local rigidity in the deformation parametrization and regularization [10, 11, 12]. We build upon this group of works to propose new deformation models with two objectives in mind. The first goal is to propose a deformation model for volumetric shape representations, with the aim of providing a stronger regularization than surface-based regularization when tracking shape whose nature is volumetric. The second goal is to propose a deformation model which learns deformation properties of the shape being tracked and uses them at the same time for providing a regularization which is more adapted than a generic deformation regularization.

1.5 Problematics

Although non-rigid shape tracking methods have significantly progressed during the last decade, state-of-the-art algorithms still have shortcomings, including critical failures cases. How can we improve the quality of shape tracking algorithms? In this thesis, we explore several directions with the purpose of improving geometric non-rigid shape tracking. This directions involve each components of model-based shape tracking, which are a deformation model, a temporal evolution model, and an association model. The research directions concern three themes: the temporal evolution of shapes, the use of volumetric shape representations, and the combination of acquired and simulated dynamics. We list the questions related to this themes below, for which this thesis gives some answers.

Temporal evolution When observing the motion of a shape, the correlation between neighborhing time samples is high, because of the continuity of motion as a function of time. Should we use that temporal correlation for estimating poses? In other words, should we pose the non-rigid shape tracking problem by considering set of frames simultaneously with their interdependence, or should we pose it as a set of independent pose estimation problems? What is at stake in this decision? How can we exploit the temporal redundancy? Can we find associations independently for each time sample?

Volumetric shape representation Although non-rigid tracked objects often have a volumetric physical nature, most works on non-rigid shape tracking represent shapes with surfaces, the use of volumetric representation being exceptional. We believe that this imbalance has more to do with the dominance of surface representation in the computer graphics and vision communities than with purely scientific reasons. What do volumetric representations can offer to shape tracking? What are the existing volumetric representations for shapes? Which ones best suits shape tracking? How can volumetric representations be used for deformation models, and for association models? In what situation are volumetric representations advantageous with respect to surface representations?

Combining acquired and simulated motion 4D motion capture allows to play back an acquired shape-motion sequence. Even if the viewpoint is free of constraints, the acquired motion is unchanged. Could we modify the acquired motion in order to generate a full range of new animations? In particular, could we modify the motion in such a way that it is compatible with a virtual environment? This would take advantage of realism of the acquired motion sequence and the flexibility of purely simulated animation. The problem is: how to make a plausible motion which resembles the acquired motion, but is however different? Can volumetric shape tracking provide a framework for this purpose?

1.6 Outline

In Chapter 3, we propose a method that simultaneously tracks a shape sequence and learns a dedicated deformation model by analysing the deformation occurring inside a temporal window. This tracking method provides robustness and precision improvements, but it still exhibits the defaults of surface-based tracking methods when applied to objects that are volumetric.

In Chapter 4, we propose to revisit shape tracking with a volumetric shape representation and compare it to surface-based tracking. The proposed framework extends the framework of chapter 3 to the volumetric case by representing shapes with centroidal Voronoi tessellations (CVT), which decompose volumes in cells with high regularity. This volumetric decomposition is used for defining a deformation model as well as an association model.

Previous chapters use association models based on the EM-ICP family, which rely strongly on the assumption that the pose estimation of the preceding timeframe is rea-

sonably correct. This dependency between neighborhing frames makes the EM-ICP methods sensitive to large inter-frame motions and it make them propagate tracking errors during sequence processing. On the other hand, detection-based association models, which estimate correspondences independently of neighboring timeframes, offers a strong resilience to tracking failures. We propose in Chapter 5 to use detection-based associations to volumetric shape tracking case, by replacing the EM-ICP association model with a volumetric detection-based association model, based on a volumetric local feature.

In previous chapters, volumetric shape tracking is motivated by the prospect of better tracking quality. In Chapter 6, we propose an application of volumetric shape tracking to computer graphics. We show that volumetric shape tracking provides a framework for a new kind of capture-based shape animation, where acquired dynamics are modified under the influence of a virtual environment. The CVT cell-based decomposition is advantageous for modifying the acquired motion with rigid-body mechanics and procedural effects.

Chapter 2

Related Work

Tracking deformable shapes from visual data is a relatively ancient research subject. Pioneering scientific works in motion capture dates back to the XIXth century [13], only a few decades after the invention of photography. However, most works on dense 3D deformable motion capture from video cameras have emerged since the late 1990's, with the technological advances of digital video cameras. Because its main historical applications are computer graphics and medical imaging, the topic of dense markerless motion capture is tightly related to these fields.

As motivated in the introduction of this thesis, our work addresses the problem of *geometric* tracking of 3D deformable motion. In this chapter, we review previous works related to this problem.

Most works on deformable shape tracking can be viewed as the combination of a deformation model and an association model. The association model defines correspondences between shape elements which need to be aligned, and the deformation model describes how the shape can be deformed. Our survey of these components is oriented by our objective, tracking fast and large non-rigid deformation, and by our need for robustness to geometric and topological noise in the input shapes. Our analysis of existing techniques also focus on the possibility to transpose them to volumetric tracking.

Before reviewing deformation and association models, we review existing shape acquisition systems and associated shape reconstruction methods, which constitute the preprocessing step of geometric shape tracking.

2.1 Static Shape Reconstruction

Tracking non-rigid 3D shape in motion with precision requires an acquisition system capable of acquiring 3D shapes at sufficient temporal frame-rate. We present the prevailing technologies with their strengths and disadvantages.

2.1.1 Wide Baseline Multi-Camera Setup

When capturing a 3D scene with the aim of reconstructing it, it is common to use a system of multiple cameras distributed around the volume of interest. Observing a scene from several viewpoints provides a better coverage, it reduces the risk of occlusion and

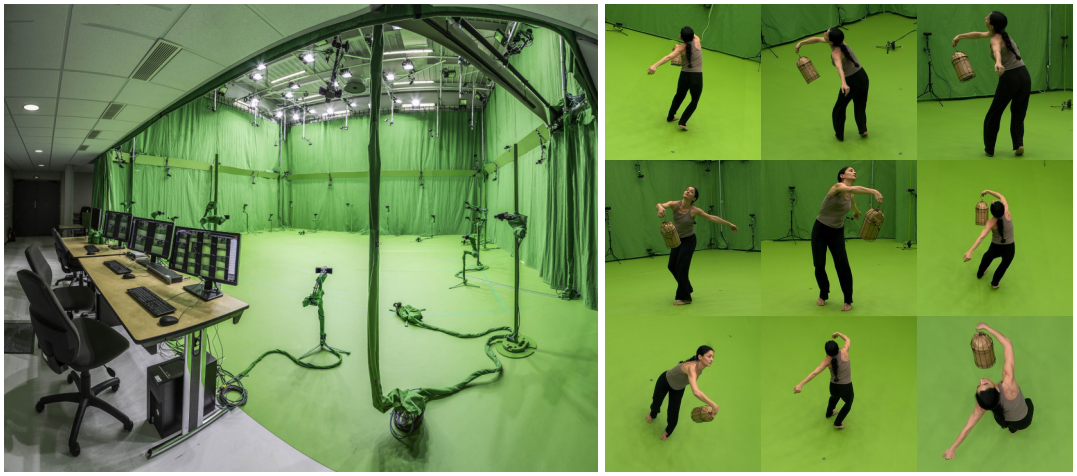


Figure 2.1 – *Kinovis* platform, at Inria Grenoble. Left: acquisition room. Right: simultaneous images from a subset of cameras.

it allows for a precise triangulation of 3D points. Reconstructing a shape from multiple viewpoints is a problem for which many solutions have been proposed in the static case. Consequently, a temporally inconsistent sequence of reconstructed shapes can be obtained by applying a static reconstruction method to each temporal frame of the multiview video sequence.

Multiple-camera systems have several advantages. They allow for a large acquisition volume (see Fig. 2.1), which is often necessary for acquiring sports, dance or multiple people interactions. Color cameras are passive sensors so they do not interfere with most of other sensors. This gives the opportunity to build a hybrid acquisition systems by combining them with active sensors technologies such as structured light [14], time-of-flight, X-ray sensors [15] or marker-based motion capture [16]. When they surround the scene, Multiple-camera systems gather information about almost all sides of the observed object, which gives the possibility to fully reconstruct the shape. Finally, thanks to the high resolution of today's video cameras, they provide a high quality of reconstruction.

However, the quality of the 3D reconstruction is obtained at the cost of a highly controlled environment: known background (fixed or monochromatic), uniform and powerful lighting conditions, fixed and synchronized cameras, and they require a calibration process. Moreover, they generate very high volumes of data (40GB/s of raw images with the Inria Kinovis system[1]), which require a high computational power to be processed. They also turn out to be expensive and tedious to install. These limitations are critical for some applications, especially those whose acquisition context is already constrained, such as movie shooting setups, sports events and medical environments. Therefore several works aim at relaxing the requirements of multiple-camera systems. For instance, background segmentation, 3D reconstruction and tracking from multiple uncalibrated and hand-held cameras in complex outdoor conditions have already been performed [17], which opens the way for new applications of 4D modeling.

[1]There are 68 cameras working at 50fps with a 2048x2048 image resolution.

Characteristics of reconstructed shapes. Multi-view stereovision, the problem of reconstructing a 3D shape from several calibrated views, has been heavily studied. A first group of methods consists in first extracting silhouettes of the foreground object in images, then reconstruct a 3D shape from the silhouettes only. Each silhouette image from a camera defines a 3D generalized cone, which contains the 3D object. By intersecting the cones from all cameras, we obtain the visual hull [18, 19], which is a coarse over-estimation of the observed shape. This method provides a rough shape approximation when using only a few cameras, but the accuracy of the reconstruction is much better with tens of cameras (see Fig. 2.2). Although increasing the number of cameras improves the reconstruction accuracy of convex and hyperbolic areas, the visual hull will never contain parabolic concavities because of its mathematical properties. For being categorized as "outside of the visual hull", a 3D point needs to be on the way of a direct light ray going from a background point to a camera. This is impossible for a point lying inside a parabolic concavity.

Figure 2.2 – Visual hull reconstruction from 68 views. Same temporal frame as the images displayed in Fig. 2.1

While being an imperfect shape approximation, the visual hull relies only on silhouette information, which can be obtained by background segmentation for which simple, fast and robust algorithms exist in the case of monochromatic backgrounds. Moreover, there exist real time algorithms for computing visual hulls approximately using a voxel-grid representation [20] or exactly as watertight surface meshes [21, 22]. Since it can be obtained with robust and fast algorithms, the visual hull in an advantageous shape approximation on which shape tracking methods can rely. They can also be used as an initial solution for other detail-preserving static reconstruction algorithms.

An other class of reconstruction methods rely on photo-consistency and look for correspondences between image points with similar appearance between different cam-

eras. These methods are able to reconstruct shape detail including concavities, but they are very sensitive to illumination conditions, surface material, image noise and camera calibration. Moreover, they require the shape to have a non-uniform texture. Most of these algorithms produce either an exact surface representation such as a mesh or an implicit surface, or a cloud of oriented points from which we can estimate a surface with the Poisson reconstruction algorithm [23]. They are usually computationally expensive, at least an order of magnitude slower than visual hull reconstruction methods. For more information on multi-view stereo reconstruction algorithms, we refer the reader to the survey of Seitz *et al*. [3] and the more recent tutorial of Furukawa *et al*. [24].

Visual hull and multi-view stereo reconstruction techniques have common shortcoming: they produce topological artefacts. For example, if two parts of the shape are in contact, the reconstruction will fuse them. The consequence is that the topology of the reconstructed shapes is not reliable. For the shape tracking problem, it means that we should not rely on the topology of reconstructed shapes for estimating temporal correspondences.

2.1.2 Depth Sensors

Depth sensors are imaging sensors that provide an image where a pixel does not encode a color, but the distance between the sensor and the observed surface. From this depth image (depth map), we can directly reconstruct a partial surface of the observed scene. Depth sensors can be distinguished into two categories: passive sensors and active sensors. Passive sensors are usually a setup of two or three cameras with a narrow baseline. Reconstruction is often performed by photometric stereovision techniques. The main active technologies are time of flights sensors and structured light. Time of flight sensors emit a light ray toward the scene and measure the time spent by the light to go from the sensor, being reflected and touching the sensor back, which is linearly related to the distance between the sensor and the object. Structured light sensors use a light projector that projects a known texture (patterns) and a camera that detects the pattern. The parallax induces small location differences between the projected and sensed patterns, which are measured for estimating a depth map.

Depth sensors are usually portable devices, can be can be low cost, and they provide surface detail. However, they only provide a partial scene representation: only ones side of the scene is reconstructed, the object topology is not preserved because of occlusions and observation noise. Passive sensors have a limited depth precision and robustness to noise, while active sensors have a restrained depth range and they suffer from strong limitations regarding the combination of several sensors because of interference.

Active depth sensors are of growing interest, because the recent technological advances have permitted the production of such devices at low cost, and they are being integrated to mobile devices (project Tango, Intel realsense). They might become as common as regular cameras on smartphones, opening the way for consumer applications.

2.1.3 Marker-based motion capture

Marker-based motion capture systems do not acquire a shape, but only the motion of a sparse set of point of the shape. There exist several methods for estimating the motion of the original shape from the motion of these points and a kinematic skeleton, but most shape deformation detail is lost and the initial shape must be obtained by other means. However, when used jointly with a shape acquisition system, a marker tracking system can provide a sparse tracking ground truth which we will use for validating our surface tracking methods.

The points being tracked are materialized by active or passive physical markers that usually modify the subject's appearance. Some systems even requires to wear a special tight suit. Thanks to their high acquisition frequency (that reaches several hundreds of fps) and their high accuracy, they can track very fast motion. Marker-based motion capture benefits from a high technological maturity and has been strongly adopted in the industry of movie and video game production. However, it often requires heavy manual labour for correcting marker tracking failures. Moreover, markers are rather invasive: they perturb the subject's behavior and appearance (visible markers, tight clothes or special suit). Consequently, marker-based motion capture system are restricted to certain type of shape and motions, so we only use them for validating surface tracking algorithms.

Although the experiments presented in this thesis mainly concern multiple camera systems, most of the concepts and ideas that we develop are not limited to any type of shape acquisition system. However, since our approach uses volumetric models, the reconstructed shapes must clearly define the interior of shapes. In case of systems of depth sensors, this requires additional processings or is simply impossible.

2.2 Deformation Models

For tracking deformable shapes, we need a deformation model, which consists in the combination of a shape representation, a deformation parametrization and optionally a deformation regularization.

The set of all possible continous deformations of a shape defined by a manifold is an infinite-dimensional space. For representing and estimating deformations, we need to restrict deformation to smaller spaces for three reasons. First, just as the shape representation, the deformation representation has to be finite-dimensional in order be stored in computer memory. Second, shape registration is an ill-posed problem. Third, input data is noisy and incomplete.

When tracking deformable shapes, the reduction of the deformation space is obtained by means of the deformation parametrization and a deformation regularization. Reducing the space of deformations amounts to enforce a prior about which deformations are feasible. This prior is meant to compensate for the data noise and the ill-posedness of the problem. Therefore the deformation model should be chosen so that it reflects the actual shape motion, and the strength of the prior should be adapted to the degree of ill-posedness of the problem and to the level of noise.

Shape deformation models are heavily used in computer vision and computer graphics. Depending on the application, the requirements are different. While computer animation synthesis requires high perceived realism of the deformation, computer vision applications require deformation models that are robust to noisy data-driven constraints. In this section, we review the existing deformation models that relate the problem of tracking deformable shapes from multiple cameras.

2.2.1 Shape Representations

Before defining how to deform a shape, we need to represent this 3D shape. We categorize existing shape representations into manifold representations and non-manifold representations.

Manifold Representations The most popular 3D shape representation is probably the polyhedron, also called surface mesh or polygon mesh. It is easily manipulated by its graph of vertices, edges and facets. Any orientable two-manifold can be approximated by a polyhedron, given a sufficient number of vertices. Polyhedra are continuous surfaces, but they are not differentiable. This can be an issue for representing smooth surfaces in some applications. If surface smoothness is required, one may consider higher-order representations such as NURBS, or a smooth implicit surface, which is defined as the zero of a real-valued smooth function defined on the volume. While smoothness may be critical for shape rendering applications, it rarely affects shape tracking.

A shape can also be defined by an occupancy function defined on a volume discretization such as a regular grid of voxels. One needs to ensure that the discretized domain covers the shape, which may require a high memory. For differentiability and merging purposes, the occupancy function might be replaced by a smooth function, such as the truncated signed distance function (TSDF) [25] which has proven successful for fusing range sensor data [26, 27]. In this case, the surface is implicitly defined by an interpolation of the discrete function.

In our work, we want to express functions that are defined on the shape interior volume, such as volumetric cohesion constraints and volumetric correspondences, so we are interested in volumetric shape discretizations that include inner volumetric elements. Several volumetric discretizations have been proposed and used for building deformation models. Regular cubic voxel grids have been used for deformation models, *e.g.* [11], but they produce cells with irregular shape and size near the surface of the shape. They also suffer from an alignment bias and an inefficient memory cost. Zhou *et al.* [28] propose to build a volumetric graph from a regular voxel grid and the surface mesh. The resulting graph is not manifold, and it does not define a shape tessellation. Other approaches use a tetrahedralization of a shape such as the constrained Delaunay tetrahedralization [29], which creates tetrahedra by inserting inner edges between surface vertices. This approach produces irregular tetrahedra, all of them being in contact with the surface. Furthermore, the number of tetrahedra is not controlled, and their shapes depend on the surface meshing, so it is often needed to remesh the surface. One can also discretize a volumetric shape with a Voronoi tessellation, which decomposes a shape into an arbitrary number of inner volumetric cells. The cells form a partition of

the initial shape, and their connectivity respects the topology of the initial shape. The Voronoi tessellation issued from a random set of point contains irregular cells, but the point set can be optimized in order to obtain a centroidal Voronoi tessellation (CVT) [1], which is formed by cells of quasi-regular shape, size and connectivity. Hence, CVTs are the best candidate for discretizing volumetric shapes in our application.

Non-manifold representations There are also non-manifold representations of shapes. For example, a set of unconnected primitives may define a shape. Point clouds and polygon soups are fuzzy representations of shapes. These representations are outputted by noisy sensors or reconstruction methods that are able to detect shape elements independently, without recovering a complete shape. Such representations do not contain neighborhood information on shape elements.

A depth map, or range image, may be viewed as a shape representation. It is a partial representation which only represents the parts of the surface that are visible from the sensor viewpoint. A depth map can be trivially transformed into a point cloud. Although samples (pixels) have a neighborhood relationship inherited from the sensor geometry (the pixel lattice), this topology does not relate to the shape topology, because of image discontinuities produced by the projection.

When performing shape tracking, it is often preferable to use a manifold representation, because the knowledge of the neighborhood relationship helps to enforce shape consistency. Hence, in this work, we represent surfaces with watertight meshes, which are provided by a static 3D reconstruction step, and we represent their inner volume with a CVT computed from the surface mesh.

2.2.2 Deformation Parametrizations and Regularizations

Once a shape representation is chosen, one needs to define how to deform it. The deformation can be expressed by a set of parameters (such as scalars, vectors, transformations). Some parametrizations require constraints between their parameters. Choosing an appropriate deformation parametrization is crucial, because most of them already encode a prior about the deformation.

Deformation priors can also be encoded as additional constraints on the deformation parameters. These constraints may be either hard or soft constraints. Soft constraints can often be interpreted as a Bayesian prior probability density on the deformation parameters.

It has to be noted that the level of detail of the deformation representation does not need to match the level of detail of the shape representation. For example, a highly detailed shape may undergo a low-frequency deformation. Decorrelating these two levels of detail allows for a better adjustment of the level of detail, which removes unnecessary degrees of freedom. This prevents the deformation from overfitting noisy correspondences, and it leads to a lower computational complexity.

We group deformation models in three categories: kinematic structures, locally shape-preserving structures, and shape spaces. Deformation models of theses categories can also be combined into hybrid deformation models.

Kinematic structures When the motion is known to be articulated, a popular approach consists in parametrizing the motion of a surface with a kinematic skeleton. This is particularly adapted to human body motion, which is mainly determined by the pose of the anatomic skeleton. The deformed surface is usually defined with linear blend skinning [9] (LBS), a popular mesh animation method issued from the computer animation community. The deformation parameters of a kinematic skeleton are the joint angles (*i.e.* the angles between neighbor bones). This method requires to know the articulated structure of the shape, which can then be automatically rigged to the surface [30]. The articulated structure and its rigging can also be inferred from an already tracked sequence [29]. A kinematic skeleton is a motion parametrization that encodes a strong prior about the deformation. This prior provides robustness to the approaches that use a kinematic skeleton and LBS for tracking mesh sequences [31, 32]. A strong limitation of this deformation model is its inability to include deformation detail of the surface such as local skin and flesh motion. It is especially unsuitable to the tracking of human beings wearing large clothes and other non-articulated shapes.

For animating shapes with more general deformations, one can use cage-based shape deformation. As a kinematic skeleton, a deformation cage is a kinematic structure described with less parameters than the initial shape mesh being deformed. Contrary to a regular kinematic skeleton, which is a tree structure lying inside a shape, a deformation cage is a volumetric mesh which surrounds a shape and acts as an external skeleton. The deformation is defined for any point lying inside the cage: its location is interpolated from the cage vertices coordinates using generalized barycentric coordinates such as mean value coordinates [33], harmonic coordinates [34] and Green coordinates [35]. Deformation cages offer more degrees of freedom than kinematic skeletons while remaining more constrained than vertex-wise displacement fields. Originally devised for shape animation, cage-based deformation models have been applied to shape tracking [36, 37]. They have proven better than skeleton for deforming rodents [34, 38] and human dressed with loose garments [39], whose deformations are insufficiently described by skeletons. However, deformation cages are underconstrained. Although this is not a limitation for shape animation, it is an issue for pose estimation because it makes the inverse kinematics unstable. In particular, cage vertices which lie near highly deformed regions are ill-conditioned and tend to be ejected far from the mesh, which leads to incorrect deformation estimation [39]. Therefore, pose estimation methods need to regularize the cage deformation. Savoye *et al.* [36] try to locally preserve the local shape geometry by preserving Laplacian coordinates of the mesh. Duveau *et al.* [37] constrain the cage deformation parameters to a manifold learned from a database. Each of this methods comes its own issues. Thiery *et al.* [39] propose a regularization which stabilizes the cage vertices, but their work only consider a noise-free temporally coherent mesh sequences as input. Thus their framework does not perform shape tracking. Although deformation cages are *volumetric* deformation models in the sense that they define a deformation of their inner volume, they are only *weakly* volumetric since they do not model the interior deformation independently of the surface mesh deformation. The volumetric deformation simply follows the cage deformation through interpolation, leaving no freedom for more local internal volume deformation. The numerical instability of deformation cages make them inconvenient for robust shape tracking. Although

they can express a wider variety of motion than kinematic skeletons, their deformation expressivity is still limited.

Local shape preservation In order to address more general shape motion, a group of approaches use higher-dimensional deformation parametrizations (such as a displacement fields), with priors that express local shape preservation. They assume that locally, the motion can be approximated by a transformation that lies in a restrained set. This set can be the set of isometric surface transformations, elastic transformations, affine transformations or rigid transformations. Isometric surface deformation is particularly suitable for modeling non-rigid inextensible surfaces such as paper sheets and inextensible textiles [40]. The isometric deformation constraint can be enhanced with surface smoothness priors such as thin-plate spline [41]. However, when applied to a closed surface, the isometric surface constraint does not prevent the inner shape volume from collapsing, which can be a problem if the real shape is supposed to be incompressible. Vicente *et al.* [42] address this issue by applying the the isometric deformation constraints between the vertices of a volumetric mesh. For modeling extensible surfaces, several methods have emerged. Some methods assume the surface is made of a homogeneous material and use a physical local linear elasticity model involving the Young's modulus (which models the material stiffness) and the Poisson ratio (which relates the deformation between orthogonal directions) [43]. However, many objects, such as the human body, deform in a way which is not well described by the local linear elasticity model.

For tracking a more general class of objects, several approaches enforce local rigidity constraints on the deformation. A popular regularization of mesh deformation in computer graphics and vision consists in preserving the values of the Laplacian operator of the 3D coordinates of surface vertices [44, 10]. Since the Laplacian encodes the local shape geometry, local shape structures tend to be preserved. Because the Laplacian of vertices coordinates is not invariant to rotations, this approach is inadequate for large deformations. To overcome this limitation, Sorkine *et al.* [45] propose to estimate local rotations on the surface, and compensate for rotations in the term that compares Laplacian coordinates. This *as-rigid-as-possible* deformation regularization has been used for surface tracking [46]. However, if the deformed surface reaches a configuration where estimating the local rotations is very ambiguous (which can happen when using noisy data-driven constraints), then the Laplacian energy computation is erroneous, which leads to an optimization failure [47]. Laplacian mesh regularization has been extended to volumetric meshes for deformation retargeting [28] and shape tracking [48]. With tetrahedral meshes, the risk of tetrahedron inversion arises. It can be handled by introducing a non-inversion term [49] which, however, complicates the optimization.

Several approaches enforcing a locally rigid behavior are based on local patches defined over the mesh structure [11, 50, 12]. Each patch motion is parametrized by a rigid transform, and the structure is maintained by elastic constraints between neighborhing patches transformations. Vertex coordinates are then interpolated from the transformations of neighborhing patches (with LBS, for example). In another work [51], the rigid patches are inferred from a sequence of observations rather than arbitrarily chosen. In this case, the deformation is regularized by the maximum number of patches, which

is a hyperparameter of the deformation model. By parametrizing the motion by vertex groups rather than individual vertices, patch-based approaches decorrelate the deformation complexity from the shape complexity. This is particularly useful for preserving template shape detail despite noisy associations. We build our work on patch-based deformation models as these approaches are relatively robust while being able to address a wide range of shape deformation. While those approaches use surface patches, we extend patch-based deformation models to volumetric shapes.

Shape spaces The deformation models we propose also relate to shape space methods, which regularize shape deformation by restraining the deformation to a particular shape space. Such a shape space may be known *a priori* [52], or learned from examples [53, 37, 54, 55]. Shape spaces can provide a strong regularization, which is useful when fitting highly incomplete and noisy data. They are popular for shape animation from sparse data such as marker-based motion capture or partial scans [53]. They can also be used for shape and pose estimation in images [54]. The shape space model can be linear (ex: PCA [53, 54]), non-linear (ex: GP-LVM [37]), or simply described by a set of shape samples [55]. Shape spaces are usually limited to a particular shape category (ex: undressed human body), which limit their applicability. Moreover, as all example-based methods, they are limited by the representativeness expressed in the database. The deformation model we propose in Chapter 3 can be seen as a shape space approach which is not based on a shape space known a priori, but on the poses of the sequence being tracked. Hence it does not suffer from the two limitations we have just mentioned.

Hybrid deformation models The different kinds of deformation models we described are not necessarily incompatible. Indeed, many works combine several of them. Kinematic skeleton can be combined with Laplacian-based regularization [56, 57] or patch-based methods [58] in order to benefit from the robustness of skeletal regularization while preserving surface deformation detail. Shape spaces are often built on top of other deformation models in order to build a stronger deformation prior. There are examples with kinematic skeletons [53, 59], deformation cages [37] and locally rigid patch-based structures [55]. Our approach can be seen as a hybrid deformation model which combines a locally rigid patch-based structure with shape-space constraints.

2.2.3 Temporal Deformation Models

Most deformable shape tracking approaches do not use a dynamic model for motion. The exploitation of temporal contiguity of acquisitions is often limited to using the previous frame's pose as an initial solution for the current frame. Some approaches enforce temporal continuity by penalizing the deformation amplitude between consecutive frames. For example, Vlasic *et al.* [31] enforces temporal smoothness of skeleton poses by penalizing the quadratic deviation of each joint angle between adjacent temporal frames. This scheme globally improves the robustness of the tracking, but when a pose is incorrectly estimated, the failure tends to propagate to the following frames. Vlasic *et al.* [31] also enforce temporal smoothness of the surface motion by applying a bilateral filter on vertex coordinates. This filtering pass reduces the texture sliding

artefacts caused by tracking failures, which are mostly visible when the object is static. However, this post-processing pass does not constitute a dynamic motion model that improves the tracking robustness.

Other approaches assume there is some repetition in the poses in a sequence, and exploit it for tracking purposes. Huang *et al*. [55] select key poses from the sequence and use them as initialization for tracking other frames. Budd *et al*. [60] measure shape similarity between all pairs of temporal frames and build a minimum spanning tree over the set of frames. Pose estimates are then propagated along the tree structure rather than in the sequential order. These approaches build a pose space rather than a dynamic model.

Dynamic models have been applied to motion analysis and synthesis. The hidden Markov model (HMM) [61], successful in speech modeling, models dynamics with a finite state machine with probabilistic transitions and considers that state values are observed indirectly through a noisy probabilistic process. Linear Dynamical systems (LDS) are systems of linear differential equations. The HMM and LDSs are well-known and easy to train, but they cannot express complex motions such as human motion [62]. There are non-linear dynamical systems (NLDS), whose motion expressivity is better, but they have many parameters and they require large training datasets. The Gaussian process latent variable models (GP-LVM) [63] have been adapted to time series to make Gaussian process dynamical models (GPDM) [62]. While this approach requires little training data, it is time-consuming and it is not adapted to long sequences with broad motion variety that includes branching (in the latent space) such has dance. Furthermore, this work only considers the motion of a kinematic skeleton. In order to model more complex motion, not limited to a particular gesture or activity, some approaches have proposed hierarchical or hybrid approaches that combine a simple motion model (such as LDS) with a finite state model. Each state models a simple motion, while the finite state machine handles transitions between states [64, 65]. These approaches are more expressive, but they are often complex to train and their parameters require tuning, in particular the number of states. While most dynamic systems applications to human motion consider the motion of a kinematic skeletons, Tung *et al*. [65] characterizes local dynamics of general surfaces. Thanks to a hybrid system using a set of LDS for each state, it models local curvature evolutions. However, this work only provides a tool for analysis, since it considers only input sequences that are already tracked. Therefore it does not constitute a deformation model that regularizes shape deformation.

Several works address the problem of modeling the dynamic of soft tissues such as loose clothes and human flesh. Such physical motion is highly dependent on the previous deformations so it cannot be explained by static shape-pose models such as SCAPE [53].

The Dyna [59] model extends the SCAPE skeleton-based shape-pose model [53] with a dynamic model for soft tissue motion of human bodies. Their dynamic model aims at predicting only the component of non-rigid deformation which is not explained by the shape and static pose. This component is predicted by an autoregressive model (AR) from the velocities and accelerations of each limbs and the root node of the skeleton, as well as the two previous estimate of this component. This AR model is trained on a database of dynamic motions for which ground truth (surface tracking) is available.

While this model is able to produce realistic flesh jiggling motion, it does not model the pose dynamics. It is also limited to undressed human bodies.

Our deformation model addresses temporal evolution of poses by minimizing a pose-distance between the pose estimated at time t and the pose space defined by the poses of the sequence within a surrounding temporal window. Our approach can be seen as a blending of shape-space approaches and dynamic models.

In our work, we propose two deformation models. In Chapter 3, we propose a surface deformation model that adapts to the variability of the poses inside a temporal window thanks to a local model of deformation variability. In Chapter 4, we propose a volumetric deformation model which enforces local volumetric cohesion of shape interiors. This model is combined with the temporal deformation model of chapter 3.

2.3 Association Models

Inferring motion from visual data can be done with several cues. In the case of geometric tracking, the motion is inferred from the geometry of a sequence of shapes. For most tracking methods, it requires to establish correspondences between local parts of two shapes. These correspondences are established either between a template shape model and a reconstructed shape, or between two shapes reconstructed at different time instants. The correspondences are used for driving the deformation of one shape into the other one, thanks to a deformation model (Section 2.2). In this section, we review the existing approaches for estimating shape correspondences in the case of deformable shape tracking.

Proximity and temporal continuity If the motion is continuous and if the temporal shape acquisition rate is sufficient, then the displacement of a material point between two frames is small. Many association models rely on this fact for estimating shape correspondences. The simplest association model consists in associating each point of shape A to the closest point from shape B. This association model is used by the famous ICP algorithm [5], which registers a shape with another one by iteratively performing a correspondence estimation step and a pose estimation step until convergence. The ICP association model suffers from several limitations. When the frame rate is insufficient for the speed of motion and deformation, the correspondences are incorrect and they can lead the registration algorithm to a sub-optimal pose estimate which can be highly inaccurate. The same issue arises when establishing correspondences between a shape template model and the first shape of a sequence. A second limitation is the lack of robustness to outliers and missing data. Many variants of this association model have been proposed in order to overcome its limitations.

Global association models Many association models estimate correspondences from elements of a shape A to elements of a shape B. We define a *local* association model as a model which defines the correspondences independently for each point of shape A. This is the case of the ICP association model. On the contrary, there are *global* association

models, which enforce global consistency among all correspondences. Examples of global consistence constraints are the surjectivity and the injectivity of the association, as well as the preservation of the topology or the metric.

Local association methods have the advantage of being often simple and fast. The Independence between correspondences allows implementations that take advantage of parallel hardware architectures such as multi-core CPUs and GPUs. On the contrary, global association methods tend to be slow, memory-consuming, more complex (as they often involve optimization) and difficult to implement in parallel. However, they tend to provide more consistent solutions than local association models.

Global methods usually require a "local" criterion for matching points. They act as a regularization. Local matching may be used as an initial solution of the global matching problem.

Many global correspondence estimation approaches rely on topology or metric preservation. They try to find correspondences that preserve the neighborhood relationship (or geodesic distance) between shape elements. Topological noise requires to implement a robust matching scheme [66, 67], or to rely on additional cues [67].

Pose-dependent and pose-independent association models We can distinguish two kinds of correspondence methods. Pose-independent methods estimate correspondences directly given two shapes. Pose-dependent methods estimate correspondences given a pose estimate. They are usually used to solve the alignment and correspondence problem jointly. Pose-dependent methods are used in variants of the *iterative closest point* (ICP) algorithm [5] which iteratively alternates between a correspondence estimation step and a pose estimation step. Each step depends on the results of the preceding one. Pose-dependent methods are well-suited to temporal tracking, since the pose estimated at the previous frame is usually quite close to the current true pose. However, if the algorithm provides a wrong estimate for the previous pose, the correspondence estimation for the current pose is likely to fail, and it will make the current pose estimation fail too. Thus, tracking errors may propagate through the sequence. Pose-independent methods do not propagate errors temporally, but they cannot benefit from the previously inferred pose.

Sparse and dense correspondences Considering sparse correspondences instead of dense ones reduces the computational complexity of the association problem, and it may also speed up the deformation estimation. Sparse correspondences can be used for estimating a coarse shape deformation. Dense correspondences can be inferred from sparse ones later on with interpolation techniques such as a inference in a MRF [67]. Sparsity may be obtained by several methods. One can detect feature points and keep only the most likely matches [68]. Another approach consists in dividing the shapes into regions. The regions can be defined randomly with similar sizes [12]. Regions may also correspond to shape protrusions, which can be matched thanks to spectral embeddings of surface [68], point clouds [69] or volumetric meshes [70].

Soft correspondences Correspondences are often estimated between shape elements (ex: vertices) which form a discrete representation of the shape. Some methods repre-

sent correspondences by hard assignments such as functions. Yet, choosing a unique image between several plausible hypotheses reduces the chances of choosing the true one. By representing correspondences with a soft assignment, we can conserve several correspondence hypothesizes simultaneously. This gives the opportunity to disambiguate the correspondences using the deformation model.

Shape registration with soft assignment as been popularized by the *softassign* method [71], which casts the problem of point set matching as a linear assignment problem (linear programming), solved with a convex relaxation. Like the ICP algorithm, the registration algorithm alternates between steps that estimate the transformation and the point assignment. The soft assignment coefficients are integrated as linear weights in the cost functional, so the solution does not need to be deterministic. Rigid registration with soft assignments was formalized later [6] as a probabilistic framework where each point of shape B is a noisy observation of a point of shape A and the true association is defined by a hidden latent variable. Inference is performed with the Expectation-Maximization (EM) algorithm [72], which leads to an algorithm named EM-ICP [6]. The observation noise is modeled by an isotropic Gaussian distribution whose mean is the transformed point of shape A and whose variance is a global parameter of the model. The variance of the observation noise is a parameter of the probabilistic model which has the same role as the temperature in simulated annealing, excepted that its value is automatically handled by the EM algorithm. The EM-ICP algorithm was later generalized to articulated shapes [7] and non-rigid surfaces [4]. Most non-rigid deformation models make the maximization step of the EM algorithm intractable (though there are exceptions [51]). Therefore, the ECM (Expectation-Conditional-Maximization) algorithm can be used instead, where the maximization step is split into several maximization steps which consider different subsets of parameters. The maximization step can also be replaced by an increasing step without losing the guarantee of convergence (Generalized EM algorithm).

Actually, non-rigid shape registration with an EM-ICP-like algorithm was performed earlier in a quite different context. Same year as the publication of the ICP algorithm [5], Hinton *et al*. [73] published an algorithm for hand-printed character recognition that fits to an image a Gaussian mixture model (GMM) augmented with elastic constraints between the Gaussian distributions. However, the dimensionality of the problem is different than in the case of surface registration: they fit a one-dimensional manifold in a two-dimensional Euclidean space, while surface registration is about fitting a two-dimensional manifold in a three-dimensional Euclidean space.

Another soft correspondence model is the *sums of Gaussians* appearance similarity model [74]. This work performs pose estimation by maximizing the similarity of appearance between an input image and a human body model projected to the image plane, considering a very coarse region-based appearance model. By considering a very coarse appearance model (which is a set of monochromatic isotropic Gaussian regions) for representing each image as well as the 3D human body model, the appearance similarity is expressed as the overlap of regions of similar color. The regions being Gaussians, the similarity expression is analytic, which allows for an efficient maximization with a gradient ascent in real-time. With this region overlap formulation, the approach do not establish explicit associations between regions. However, such an appearance and vis-

ibility model seems too coarse for driving a deformation model with higher degrees of freedom than a kinematic skeleton. Furthermore, this appearance model has only been applied to subject wearing clothes with large monochromatic regions.

Features Pose-dependent correspondence estimation methods usually rely on distance to perform correspondence (closest point). They are often combined with simple feature compatibility tests such as surface orientation compatibility. Distance is also used for optimal transport (global) models. However, pose-independent association models can not rely solely on distance and absolute orientation, so they must consider other cues. The success of feature points and descriptors such as SIFT [75] for solving image correspondence problems has inspired the field of shape correspondence and shape retrieval [76]. Generalizations to non-Euclidean surfaces have been proposed in order to define feature points and descriptors of functions defined on non-Euclidean surfaces [77]. While such a function can be based on the appearance of the shape (like in the flat image descriptors), pure geometric features and descriptors can be obtained by considering a geometric function such as a surface curvature (note: the Gaussian curvature is preserved by isometric deformations). Some of the successful surface descriptors involve curvature [77] or diffusion properties [78, 79]. Features and descriptors can also be learned with a machine learning algorithm such as convolutional neural networks trained on a relevant task [80]. Although shape features tend to work well with noise-free computer-generated shapes, they are rarely robust to the type of noise present in shapes reconstructed from visual data [76].

Robustness to outliers Shape reconstructed from multiple views often contain outlying data, such as phantom shape components. These artefacts must be discarded from the association, otherwise the pose estimation is be perturbed. Fitting a model to data that includes outliers can be addressed with the theory of robust estimation with the M-estimators [81]. Pose estimation often amounts to minimize a sum of registration residuals over the pose parameters. The least squares estimator, which corresponds to a probabilistic model with a Gaussian noise model for residuals, is highly perturbed by the large residuals of outliers. Using an M-estimator with a robust penalty function (which reduces the influence of large residuals) makes the pose estimation more robust to outliers [82]. Popular robust penalty functions are Huber's and Tukey's penalty functions, whose behaviors are still quasi-quadratic for inliers. They have proven successful for fitting deformable models to depth images [83, 84, 27]. Solving for the M-estimator can be done with several techniques [85]. Most of them amount to reformulate the robust problem as a weighted least squares problem, which is then solved by the Gauss-Newton or the Levenberg-Marquardt algorithm. The weights are either updated at each iteration of the solver (for the square root method and the iteratively reweighted least squares method [81]), or they are additional variables that are constrained by additional residuals (for the lifting method [86]). The lifting method seems to provide better convergence properties than other methods for surface fitting [84] as well as for bundle adjustment [85].

Outliers can also be handled by introducing an outlier class (also called garbage class) in the association model. Instead of being associated to a vertex of the model, an

observation can be associated to the garbage class, which groups all outliers. This approach suits particularly well the soft association methods. In the EM-ICP approaches, the garbage class can be represented with a uniform probability distribution over the observation space with a small likelihood [4]. For a given observed vertex, the association with the garbage class becomes dominant when all other possible associations are unlikely.

Robust penalty functions of M-estimators have a scale parameter which defines the threshold between inliers and outliers. This parameter can be difficult to estimate, since low values will treat all association as outliers (and fail the pose estimation), while high values will fail to detect outliers. In the EM-ICP framework with a garbage class, the observation noise level is automatically estimated, and the only garbage class parameter that we need to choose represents a probabilistic prior on the rate of outliers. This softer way of detecting outliers requires less fine tuning than the scale parameter of robust penalty functions of the M-estimators.

Correspondences learning From a database with ground truth correspondences, one can train an algorithm to predict shape correspondences. This approach has proven successful for real-time human pose estimation from a range image [87, 8]. The supervised learning algorithms considered are regression forests and more recently convolutional neural networks (CNN) [88]. These algorithms are usually quite fast, which is an advantage for real-time applications. Moreover, they provide pose-independent association models, which provide robustness to the tracking process. However they require a database with ground truth correspondences. Such a database may be difficult to obtain, especially when using algorithms that require very large amounts of samples (*e.g.* CNNs). This problem can be partially solved by synthesizing a database. Another issue is the representativity of the database. As every machine-learning technique, the trained algorithm is likely to fail on cases that are too far from the examples present in the database. Finally, the learning-based methods usually predict correspondences to a template shape model, which limits the tracking algorithm to a class of shapes (ex: human bodies). However, very recent works [88, 89] propose template-free learning-based algorithms for establishing correspondences between depth or RGBD images. Practically, the training phase of machine learning approaches often requires large computational resources, especially for CNNs (which require large training sets).

In order to benefits from the advantage of correspondence learning in the context of volumetric shape tracking, we want to learn and predict volumetric template correspondences. To our knowledge, no existing work addresses this problem in the volumetric case, so we propose a solution in Chapter 5 based on a new volumetric feature and regression forests.

In our work, we propose two association models. The first one (Chapter 4) is an extension of the EM-ICP probabilistic association model to volumetric correspondences and relates to distance field alignment. The second one (Chapter 5) also establishes volumetric correspondences, but in a pose-independent manner, thanks to the combination of supervised learning and a new volumetric feature.

Chapter 3

An Adaptive Temporal Deformation Model

3.1 Introduction

When a deformable object is in motion, it generates a sequence of poses. The pose space spanned by these poses is conditioned by the intrinsic deformation properties of the object, whose nature are usually mechanical (eg: material properties). In the case of general object tracking, these properties are unknown. Nevertheless, tracking requires to regularize deformation. One successful approach consists in penalizing the deviation from a reference pose with the help of a local constraint that acts uniformly over the shape. This deformation model is applied for each time sample independently. These approaches use only the geometry of a single pose instance, thus ignoring the information embedded in the temporal evolution. Is it possible to infer intrinsic deformation properties of an object from a captured sequence of its motion? If this is possible, it would allow to automatically build a deformation model which is specialized for this particular object. Such a deformation model would have several applications. First, it would help understanding the object dynamics. Second, when tracking a sequence, this specialized deformation model would provide a better deformation regularization than a general deformation model, thus it may improve tracking quality. Finally, if used for sequence editing, this deformation model could help generating new sequences where the deformation is consistent with the object deformation space.

This idea leads to several questions. Which deformation properties should be considered, and how to represent them? The deformation model should have enough complexity for characterizing the deformation space while being simple enough for allowing robust inference. How to infer deformation properties while tracking a sequence? This is not obvious, since estimating the deformation properties requires the motion information, and vice-versa. Should we estimate the deformation properties over the full sequence, or over smaller temporal windows? Finally, does the resulting tracking approach improves quality?

In this chapter we propose a Bayesian framework where an input shape sequence is conditioned by a set of intrinsic deformation parameters. These deformation parameters consist in a mean pose over the considered temporal window, and a set of coefficients

that characterize the spatial distribution of the level of rigidity over the shape. We use the Expectation-Maximization algorithm for simultaneously inferring the intrinsic deformation parameters and the deformation at each time sample. The simultaneity of these two tasks is a key novelty of our approach. We perform a qualitative evaluation by visualizing the properties inferred. We also perform a quantitative evaluation of the tracking results on a dataset with ground truth. Our method compares favorably with state of the art surface tracking approaches.

The remainder of the chapter is organized as follows. The next section discusses related work. The framework is detailed in sections 3.4, 3.5 and 3.6, then Section 3.7 describes the inference. Section 3.8 presents various applications and experimental results. Section 3.9 concludes with a discussion on our contributions.

3.2 Related Work

The analysis of deformable surfaces captured by multi-video systems has gained lot of interest during the last decade due to the rapid progression of computer and image sensing technologies. We focus here on works that relate to dynamic properties of shapes.

Kinematic structures. Many popular tracking methods propose to rigidly constrain a model using an articulated structure, for instance a skeleton or a cage, which must be scaled and rigged to a 3D template, and optimally positioned through a sequence of models representing the observed subjects [90, 31, 91, 32]. The template is usually deformed using a skinning technique, according to the optimized structure across the sequence [30]. Such kinematic structures provide intrinsic information on the associated shapes through their parameter evolutions (e.g. their averages can define a mean pose). These approaches require a priori knowledge on the observed shapes, such as the topology and the rigid parts, and cannot be applied to arbitrary object shapes. Moreover global template deformation across time is subject to loss of local details such as cloth wrinkles and folds.

Locally rigid structures. The literature also contains several methods that relax the constraint on the shape structure using looser rigidity priors. A body of works consider deformations that preserve local intrinsic surface properties, e.g. isometric deformations [92, 93, 94, 95]. Such properties relate to local rigidities, for instance in [96, 97] local surface distortions are constrained, however they are usually known priors. While efficient to register or match surfaces, intrinsic surface properties are not necessarily sufficient to track complex shapes such as human bodies. In that case, several approaches introduce local deformation models to drive surface evolutions. For instance, in [12], the observed surface is treated as a piece-wise body with locally rigid motions. We consider a similar model to represent surface deformations which is used to learn local rigidities as well as mean poses along with the tracking. Interestingly, recent approaches also in this category were proposed to characterize local surface deformations. In [51], the authors propose a probabilistic framework for rigid tracking and segmentation of

Figure 3.1 – Example of patch template used.

dynamic surfaces where the rigid kinematic structure is learned along time sequences. Our framework does not assume such structure but learns instead local rigidities and mean poses. In [65], the authors model complex local deformation dynamics using linear dynamical systems by observing local curvature variations, using a shape index, and perform rigidity-based surface patch classification. The latter approach assumes surface alignment is given, in contrast to our proposed generative model that simultaneously performs surface tracking and local rigidity estimation.

Shape Spaces. Following the work of Kendall [98], a number of works consider shape spaces that characterize the configurations of a given set of points, the vertices of a mesh for instance. This has been used in medical imaging to estimate mean shapes through Procrustes analysis, e.g. [99]. In this case, the shape of the object is the geometrical information that remains when the pose (i.e., similarities) is filtered out. Thus Procrustes distances can be used to measure shape similarities and to estimate shape averages with Fréchet means. We follow here a different strategy where a shape space represents the poses of a single shape and where we estimate a mean pose instead of a mean shape. This relates to other works in this category that also consider shape spaces to model shape poses with mesh representations. They can either be learned, e.g. [53, 54] or defined a priori, e.g. [52] and are used to constrain mesh deformations when creating realistic animations [53, 52] or estimating shape and poses from images[54]. While sharing similarities in the deformation model we consider, our objective is not only to recover meaningful shape poses but also to measure pose similarities and intrinsic shape properties. Unlike [53, 54], we do not need a pose or shape database and the associated hypothesis of its representativeness. Moreover, our methodology specifically addresses robust temporal window integration.

3.3 Method Overview

We assume given a temporal sequence of 3D reconstructions, incoherent meshes or point clouds, obtained using a multi-view reconstruction approach, e.g. [100, 48, 101]. As in all the chapters of this thesis, we also assume that a template mesh model of the scene is available, e.g. a particular instance within the reconstructed sequence un-

der consideration. The problem of local surface rigidity and mean pose analysis is then tackled through the simultaneous tracking and intrinsic parameter estimation of the template model. We embed intrinsic motion parameters (e.g. rigidities) in the model, which control the motion behavior of the object surface. This implies that the estimation algorithm is necessarily performed over a sub-sequence of frames, as opposed to most existing surface tracking methods which in effect implement tracking through iterated single-frame pose estimation. We first define a mean pose over a set of poses (§3.4) and introduce our parametrization of the template deformation illustrated with Fig. 3.1 and its associated pose distance. Second, we define the local deformation model which characterizes local rigidity. Third, we present a Bayesian generative model which includes a probabilistic formulation of the deformation model and describes how the deformation template surface generates noisy measurements (§3.6). We then show how to perform estimation over the sequence through Expectation-Maximization (§3.7).

3.4 Mean Pose Model

We define a pose as a deformation $\mathbf{T}$ of the template shape mesh. Assuming we have a pose distance d measuring the amount of non-rigid deformation between two poses $d(\mathbf{T}^i, \mathbf{T}^j)$, we can regularize shape deformation by minimizing the distance between a pose estimate $\mathbf{T}$ and a reference pose. Several tracking approaches use the undeformed template pose $\mathbf{Id}$ as a reference (e.g. [12]), which is undeniably a valid pose. However, large deformation from the template pose are likely to be discarded by this regularization, even if they are correct. Other approaches (e.g. [31]) use the pose estimate from the previous frame as a reference pose, which is closer to the current true pose. However, the previous pose estimate is unreliable as it might include tracking failures, which leads to error accumulation during sequence processing. Here we propose to define a mean pose over a temporal window around the current frame and use it as a reference pose. This provides a reference pose which is relatively close to the true current pose, but with a low risk of estimation error thanks to the averaging. Hence it combines the advantages of both strategies.

Mean pose definition To retrieve the mean pose of a given sequence, we provide a definition suitable for the analysis of complex temporal mesh sequences. Following Fréchet's definition of a mean [102], we introduce the *mean pose* $\bar{\mathbf{T}}$ of a given set of poses $\{\mathbf{T}^t\}_{t\in\mathcal{T}}$ over the time sequence $\mathcal{T}$ as the pose minimizing the Fréchet's variance, which is sum of squared distances to all poses in the set:

$$\bar{\mathbf{T}} = \arg\min_{\mathbf{T}} \sum_{t\in\mathcal{T}} d^2(\mathbf{T}, \mathbf{T}^t), \tag{3.1}$$

where d is a distance function that measures the similarity of two poses. This distance should evaluate the non-rigidity of the transformation between two poses of a shape and hence should be independent of any global pose. In order to define such a pose distance, we need to precisely define a pose by parametrizing the non-rigid deformation of the template shape.

Pose space parameterization To express non-rigid deformability of shapes, while de-correlating the resolution of deformation parameters from mesh resolution, we opt for a patch-based parameterization of the surface similar to [12]. The reference mesh is partitioned in an overlapping set of patches, pre-computed by geodesic clustering of vertices. Each patch P_k is associated to a rigid transformation $\mathbf{T}_k^t \in SE(3)$ at every time t. Each position $\mathbf{x}_{k,v}$ of a mesh vertex v as predicted by the transform of P_k can then be computed from its template position $\mathbf{x}_v^0$ as follows:

$$\mathbf{x}_{k,v} = \mathbf{T}_k(\mathbf{x}_v^0). \tag{3.2}$$

We thus define a *pose* of the shape space as the set of patch transforms $\mathbf{T} = \{\mathbf{T}_k\}_{k\in\mathcal{K}}$ that express a given mesh deformation. Note here that a pose in the shape space does not necessarily correspond to a proper geometric realization of the reference mesh, since large discontinuities may appear between neighborhing patches. In practice, we preserve the surface consistency by interpolating each vertex location between the locations predicted by neighborhing patch transformations. The interpolation is described in the experimental section §3.8.

Pose distance definition Defining a pose distance that measures local non-rigid deformation and is independent from global rigid transformation is not easy. We propose a distance that measures whether the relative poses of neighboring patches are preserved during motion between the two shape poses $\mathbf{T}^i$ and $\mathbf{T}^j$.

Given two neighboring patches P_k and P_l, their relative pose is described by the transformation $\mathbf{T}_{k-l} = \mathbf{T}_l^{-1} \circ \mathbf{T}_k$. Comparing $\mathbf{T}_{k-l}^i$ and $\mathbf{T}_{k-l}^j$ tells how non-rigid the deformation between pose $\mathbf{T}^i$ and $\mathbf{T}^j$ is for the patch pair (P_k, P_l). Given a distance d_{kl} on $SE(3)$ for comparing $\mathbf{T}_{k-l}^i$ and $\mathbf{T}_{k-l}^j$, we can sum the distance value $d_{kl}(\mathbf{T}_{k-l}^i, \mathbf{T}_{k-l}^j)$ for all neighboring patch pairs in order to define a distance between $\mathbf{T}^i$ and $\mathbf{T}^j$ that considers the full shape:

$$d^2(\mathbf{T}^i, \mathbf{T}^j) = \sum_{(P_k,P_l)\in\mathcal{N}} d_{kl}^2(\mathbf{T}_{k-l}^i, \mathbf{T}_{k-l}^j). \tag{3.3}$$

Defining a distance d_{kl} on the non-Euclidean group of rigid transformation $SE(3)$ that is meaningful with respect to our problem is not trivial. For example, the Frobenius norm (taken on the matrix representation with a rotation matrix and a translation vector) is unsatisfactory as it does not focus on how the rigid transformations operate on the domain of interest, i.e. the template shape. Therefore, instead of defining a distance intrinsic to $SE(3)$, we follow the work of Pottman *et al.* [103] and we integrate the Euclidean distance between transformed points over the domain of interest. Like Botsch *et al.* [11], our integration domain is the union of the patches of the pair considered. For simplicity, we consider a discrete version of the integral based on vertex sampling of the template mesh, which we assume to be uniform [12]. Hence, the elementary distance d_{kl} is defined by

$$d_{kl}^2(\mathbf{T}^i, \mathbf{T}^j) = \sum_{v\in P_k\cup P_l} \|\mathbf{T}_{k-l}^i(\mathbf{x}_v^0) - \mathbf{T}_{k-l}^j(\mathbf{x}_v^0)\|^2. \tag{3.4}$$

The computation of d_{kl} is visually explained in Fig. 3.2. It can be verified that d defines a distance as it inherits this property from the L^2 norm used between vertices[1]. The independence from global rigid transformations is provided by the use of the relative transformation $\mathbf{T}_{k-l}$.

Note than when one pose is the undeformed template pose (e.g. $\mathbf{T}^j = \mathbf{Id}$), the squared distance is equivalent to the deformation energies of [11, 12][2]. Hence our model is more general since it allows for comparison between any pair of poses.

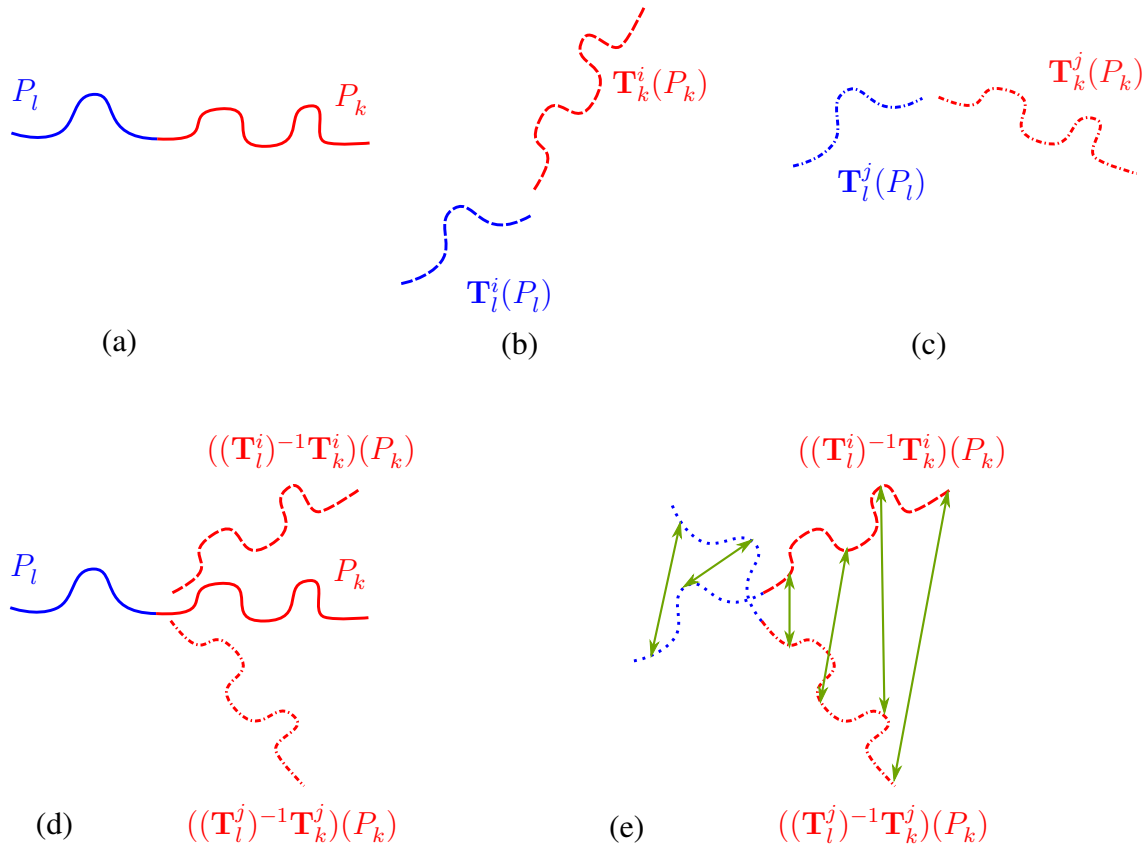


Figure 3.2 – Illustration of the pose distance. Poses $\mathbf{T}^i$ (b) and $\mathbf{T}^j$ (c) are rigidly backmapped to the template space (a) by aligning their patches l with the patch l of the template (d). The backmapped versions of patch k are compared by measuring the Euclidean distance between several corresponding points of each version (e). The comparison also takes place on the patch l as extended according to the template pose.

[1]The axiom of separation requires additional assumptions, which are met in practical cases. An example of sufficient condition is the presence of at least three non-aligned vertices in each union of neighbor patches. This condition can be relaxed by assuming a lower bound of the connectivity degree of each patch.

[2]The only difference is the weighting of each term of the sum, specific to each paper.

3.5 Local Rigidity Model

The pose distance (section §3.4) measures the non-rigidity uniformly over the shape. This corresponds to a uniform prior about the level of (non-)rigidity over the shape. Here we introduce parameters that allow for non-uniform non-rigidity measurement. To keep this information in its simplest form, for each pair of patches $(k,l) \in \mathcal{N}$, we introduce *binary rigidity variable* $\mathrm{c}_{kl} \in \{0,1\}$ which will condition the patch pair to accordingly be rigid or flexible. This variable is an intrinsic parameter attached to the original deformable model and is thus time-independent. We note the full set of rigidity variables $\mathrm{C} = \{\mathrm{c}_{kl}\}_{(k,l)\in\mathcal{N}}$. Note that the binary behavior of the rigidity variables will be relaxed by the EM inference in the probabilistic model (see §3.7), thus providing a continuous range of behavior between flexibility and rigidity.

We embed the influence of rigidity variables in the pose distance defined earlier (3.3), by computing two versions of the elementary distances d_{kl}, biased by rigidity variables C:

$$d^2(\mathbf{T}^i, \mathbf{T}^j, \mathrm{C}) = \sum_{(P_k,P_l)\in\mathcal{N}} d^2_{kl}(\mathbf{T}^i_{k-l}, \mathbf{T}^j_{k-l}, \mathrm{c}_{kl}), \tag{3.5}$$

$$\text{where } d^2_{kl}(\mathbf{T}^i, \mathbf{T}^j, \mathrm{c}_{kl}) = \sum_{v\in P_k\cup P_l} \beta_{kl}(v, \mathrm{c}_{kl}) \, \|\mathbf{T}^i_{k-l}(\mathbf{x}^0_v) - \mathbf{T}^j_{k-l}(\mathbf{x}^0_v)\|^2, \tag{3.6}$$

with $\beta_{kl}(v, \mathrm{c}_{kl})$ a uniform function over all vertices of the patch pair if $\mathrm{c}_{kl} = 1$, which encourages common rigid behavior of the two patches, and a non-uniform function encouraging more elasticity when $\mathrm{c}_{kl} = 0$:

$$\beta_{kl}(v, 0) \propto \exp\left(-\frac{b_{kl}(v)}{\eta \bar{D}}\right), \tag{3.7}$$

where $b_{kl}(v)$ is the distance between the vertex v and the border between P_k and P_l on the template, $\bar{D}$ is the average patch diameter and η is a global coefficient controlling the flexibility. The $\beta_{kl}(\cdot, 0)$ has larger values on the border between the patches, which allows more flexibility while enforcing continuity between the patches. The coefficients $\beta_{kl}(v, 0)$ are normalized such that $\sum_{P_k\cup P_l} \beta_{kl}(v, 0) = \sum_{P_k\cup P_l} \beta_{kl}(v, 1)$ in order to make both modes as competitive.

3.6 Probabilistic Bayesian Generative Model

The expression (3.1) is useful to characterize the mean over a set of poses *already known*. Our goal however is to estimate this mean in the context where such poses are indirectly observed through a set of noisy and sparse 3D point clouds of the surface. Thus we cast the problem as the joint estimation of mean pose and fitting of the model to each set of observations. For our purposes, we assume the set of poses $\{\mathbf{T}^t\}_{t\in\mathcal{T}}$ are defined for a set $\mathcal{T}$ corresponding to observations in a temporal sequence. The observed point clouds are noted $\mathbf{Y} = \{\mathbf{Y}^t\}_{t\in\mathcal{T}}$, where $\mathbf{Y}^t = \{\mathbf{y}^t_o\}_{o\in\mathcal{O}_t}$ is the set of point coordinates $\mathbf{y}^t_o$ for an observation o among the set of observations $\mathcal{O}_t$ at time t. Note that this set $\mathcal{O}_t$

is different than $\mathcal{V}$ in general as it is obtained from a 3D reconstruction or depth camera, without any direct correspondence to the deformable shape surface model earlier defined.

To express the noisy predictions of observations, we follow the principle of EM-ICP [6] by introducing a set of assignment variables k_o^t indicating, for each observation o, which patch this observation is assigned to. We express the generative model through the following joint probability distribution, which corresponds to the Bayesian network illustrated in Fig. 3.3:

$$p(\bar{\mathbf{T}}, \mathbf{T}, \mathbf{Y}, \mathrm{C}, \mathrm{K}, \sigma) = p(\bar{\mathbf{T}})\; p(\mathrm{C}) \prod_{t \in \mathcal{T}} \left(p(\mathbf{T}^t \mid \bar{\mathbf{T}}, \mathrm{C}) \prod_{o \in \mathcal{O}_t} p(\mathbf{y}_o^t \mid \mathrm{k}_o^t,\; \mathbf{T}^t, \sigma^t) \right), \tag{3.8}$$

with $\sigma = \{\sigma^t\}_{t \in \mathcal{T}}$ the set of noise parameters of the observation prediction model, and $\mathrm{K} = \{\mathrm{k}_o^t\}$ the set of all patch selection variables.

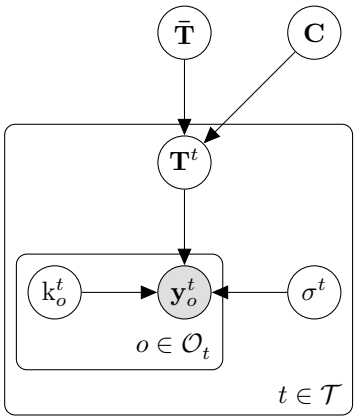


Figure 3.3 – Graphical representation of the Bayesian network. Variables drawn inside a rectangular block are replicated for each index value of the block as in [104].

Observation prediction model. Each observation's point measurement is predicted from the closest vertex v within patch $P_{k=\mathrm{k}_o^t}$. Because the prediction is noisy, this prediction is perturbed by Gaussian noise of variance ${\sigma^t}^2$:

$$p(\mathbf{y}_o^t \mid \mathrm{k}_o^t\; \mathbf{T}^t \sigma^t) \;=\; \mathcal{N}(\mathbf{y}_o^t \mid \mathbf{T}_{\mathrm{k}_o^t}^t(\mathbf{x}_v^0), \sigma^t). \tag{3.9}$$

Pose constraining model. We constrain the fitted poses to be close to the mean pose, using the distance defined earlier (3.3). We embed the influence of rigidity variables in this term, by computing two versions of the distance, biased by rigidity variables C:

$$p(\mathbf{T}^t \mid \bar{\mathbf{T}}, \mathrm{C}) \;\propto\; \exp\left(- \sum_{(k,l) \in \mathcal{N}} d_{kl}^2(\bar{\mathbf{T}}, \mathbf{T}^t, \mathrm{c}_{kl}) \right). \tag{3.10}$$

Mean model prior. In the probabilistic framework, the mean pose is a random variable which is no longer defined by (3.1). In the absence of any prior, the mean pose is unconstrained and could theoretically have completely loose patches unrelated to each other. To avoid this and give the mean pose a plausible deformation, we consider the following a prior which expresses that the intrinsic mean pose should not significantly deviate from the original reference pose (represented by the identity transform $\mathbf{Id}$):

$$p(\bar{\mathbf{T}}) \propto \exp(-d^2(\bar{\mathbf{T}}, \mathbf{Id})) = \exp\left(-\sum_{(P_k,P_l)\in\mathcal{N}} \sum_{v\in P_k\cup P_l} \|\bar{\mathbf{T}}_k(\mathbf{x}_v^0) - \bar{\mathbf{T}}_l(\mathbf{x}_v^0)\|^2\right). \tag{3.11}$$

Coupling parameter prior. We assume the rigidity variables to be independent and identically distributed (IID), each of them following a Bernoulli distribution of parameter $p_{\text{rigid}} \in [0, 1]$:

$$p(\mathrm{c}_{kl}) = \begin{cases} p_{\text{rigid}}, & \text{if } \mathrm{c}_{kl} = 1 \\ 1 - p_{\text{rigid}}, & \text{if } \mathrm{c}_{kl} = 0 \end{cases} \tag{3.12}$$

$$\text{and } p(\mathrm{C}) = \prod_{(k,l)\in\mathcal{N}} p(\mathrm{c}_{kl}). \tag{3.13}$$

The IID assumption means there is no dependence between the intrinsic non-rigid behaviors of different parts of the shape.

3.7 Expectation-Maximization Inference

We apply Expectation-Maximization [72] to compute Maximum A Posteriori (MAP) estimates of the tracking and average shape parameters given noisy 3D measurements, using the joint probability described in (3.8) as described in [104]. The assignment variables K and rigidity coupling variables C are treated as latent variables, which we group by the name $Z = \{\mathrm{K}, \mathrm{C}\}$. For the purpose of clarity let us also rename all parameters to estimate as $\Theta = \{\bar{\mathbf{T}}, \mathbf{T}, \sigma\}$. Expectation-Maximization solves the MAP problem

$$\Theta = \arg\max_{\Theta} p(\mathbf{Y}|\Theta)p(\Theta) \tag{3.14}$$

by iteratively maximizing the following auxiliary function Q given the knowledge of the previous parameter estimate Θ^m:

$$\Theta^{m+1} = \arg\max_{\Theta} Q(\Theta|\Theta^m) \tag{3.15}$$

$$= \arg\max_{\Theta} \sum_{Z} p(Z|\mathbf{Y}, \Theta^m) \ln p(\mathbf{Y}, Z|\Theta) + \ln p(\Theta). \tag{3.16}$$

The **E-Step** consists in computing the posterior distribution $p(Z|\mathbf{Y},\Theta^m)$ of latent variables given observations and the previous estimate. Computing this posterior distribution by enumerating each possible value of Z would be impossible because of the combinatorial number of Z values. However, we show that factorizing $p(Z|\mathbf{Y},\Theta^m)$ in a product of factors involving a single latent variable reduces the E-step to the computation of a single table for each latent variable. It can be noted given the form of (3.8) that according to the D-separation criterion [104], the set of latent patch selection variables and set of rigidity variable are independent given Θ, and the latent patch selection variables are mutually independents given Θ, thus following the factorization of the joint probability distribution:

$$p(Z|\mathbf{Y},\Theta) = p(\mathrm{C}|\Theta)\prod_{t\in\mathcal{T}}\prod_{o\in\mathcal{O}_t} p(\mathrm{k}_o^t|\mathbf{Y},\Theta). \tag{3.17}$$

We can express $p(\mathrm{k}_o^t|\mathbf{Y},\Theta)$ with the help of coefficients $\tilde{\gamma}_{ot}(\mathrm{k}_o^t,\Theta)$ and their normalized versions $\gamma_{ot}(\mathrm{k}_o^t,\Theta)$ as follows:

$$p(\mathrm{k}_o^t|\mathbf{Y},\Theta) = \gamma_{ot}(\mathrm{k}_o^t,\Theta) \tag{3.18}$$

$$\text{where}\quad \gamma_{ot}(\mathrm{k}_o^t,\Theta) = \frac{\tilde{\gamma}_{ot}(\mathrm{k}_o^t,\Theta)}{\sum_{\mathrm{k}_o^t}\tilde{\gamma}_{ot}(\mathrm{k}_o^t,\Theta)} \tag{3.19}$$

$$\text{where}\quad \tilde{\gamma}_{ot}(\mathrm{k}_o^t,\Theta) = p(\mathbf{y}_o^t,\mathrm{k}_o^t|\Theta) \tag{3.20}$$

$$= p(\mathrm{k}_o^t)\mathcal{N}(\mathbf{y}_o^t \mid \mathbf{T}^t_{\mathrm{k}_o^t}(v),\sigma^t), \tag{3.21}$$

where v is the closest vertex on patch k with respect to $\mathbf{y}_o^t$.

Although the Θ set does not D-separate the individual $\mathrm{c}_{kl} \in \mathrm{C}$ variables, the posterior $p(\mathrm{C}|\Theta)$ can be factorized by relying on the MRF-type factorization of the joint distribution $p(\mathrm{C},\Theta)$ as a product of potentials:

$$p(\mathrm{C},\Theta) = p(\mathrm{C})p(\bar{\mathbf{T}})\prod_{t\in\mathcal{T}} p(\mathbf{T}^t,|\bar{\mathbf{T}},\Theta) \tag{3.22}$$

$$= \prod_{(k,l)\in\mathcal{E}} \tilde{\lambda}_{kl}(\mathrm{c}_{kl},\Theta) \tag{3.23}$$

$$\text{where } \tilde{\lambda}_{kl}(\mathrm{c}_{kl},\Theta) = p(\mathrm{c}_{kl})\exp(-d^2_{kl}(\bar{\mathbf{T}}_{k-l},\mathbf{Id}))\prod_{t\in\mathcal{T}}\exp(-d^2_{kl}(\bar{\mathbf{T}}_{k-l},\mathbf{T}^t_{k-l},\mathrm{c}_{kl})). \tag{3.24}$$

Since each potential $\tilde{\lambda}_{kl}$ involves only one rigidity variable c_{kl}, it follows that

$$p(\mathrm{C}|\Theta) = \prod_{(k,l)\in\mathcal{E}} p(\mathrm{c}_{kl}|\Theta) \tag{3.25}$$

$$\text{and } p(\mathrm{c}_{kl}|\Theta) = \lambda_{kl}(\mathrm{c}_{kl},\Theta) \tag{3.26}$$

$$\text{where } \lambda_{kl}(\mathrm{c}_{kl},\Theta) = \frac{\tilde{\lambda}_{kl}(\mathrm{c}_{kl},\Theta)}{\sum_{\mathrm{c}_{kl}}\tilde{\lambda}_{kl}(\mathrm{c}_{kl},\Theta)}, \tag{3.27}$$

which leads to the factorization

$$p(Z|\mathbf{Y},\Theta) = \prod_{t\in\mathcal{T}}\prod_{o\in\mathcal{O}_t}\gamma_{ot}(\mathrm{k}_o^t,\Theta)\prod_{(k,l)\in\mathcal{E}}\lambda_{kl}(\mathrm{c}_{kl},\Theta). \tag{3.28}$$

At each iteration, the E-step consists in computing the individual coefficients $\gamma_{ot}(\mathrm{k}_o^t,\Theta^m)$ and $\lambda_{kl}(\mathrm{c}_{kl},\Theta^m)$ given the previous estimate Θ^m, for each possible value of their corresponding latent variable, respectively k_o^t or c_{kl}, according to equations (3.19), (3.21), (3.24) and (3.27). Equations (3.24) and (3.27) correspond to a reevaluation of probabilities of rigid coupling between patches, based on the previous m-th estimates of temporal and mean poses. Equations (3.19) and (3.21) correspond to the probability assignment table of time t's observation o to each patch in the model. This corresponds to the soft matching term commonly found in EM-ICP methods [6].

The **M-Step** maximizes expression (3.16). The joint distribution $p(\mathbf{Y},Z,\Theta)$ can be shown to factorize similarly to (3.28) but with the unnormalized coefficients as functions of the parameters Θ:

$$p(\mathbf{Y},Z,\Theta) = \prod_{t\in\mathcal{T}}\prod_{o\in\mathcal{O}_t}\tilde{\gamma}_{ot}(\mathrm{k}_o^t,\Theta)\prod_{(k,l)\in\mathcal{E}}\tilde{\lambda}_{kl}(\mathrm{c}_{kl},\Theta). \tag{3.29}$$

We show in appendix A that given (3.28) and (3.29), the objective function simplifies to

$$Q(\Theta|\Theta^m) = \sum_{t\in\mathcal{T}}\sum_{o\in\mathcal{O}_t}\sum_{\mathrm{k}_o^t}\gamma_{ot}(\mathrm{k}_o^t,\Theta^m)\ln\tilde{\gamma}_{ot}(\mathrm{k}_o^t,\Theta) \tag{3.30}$$

$$+\sum_{(k,l)\in\mathcal{E}}\sum_{\mathrm{c}_{kl}}\lambda_{kl}(\mathrm{c}_{kl},\Theta^m)\ln\tilde{\lambda}_{kl}(\mathrm{c}_{kl},\Theta). \tag{3.31}$$

We follow an Expectation-Conditional-Maximization algorithm, and optimize the auxiliary function successively with respect to each subset of a partition of the parameters Θ. This leads to the following update equations:

$$\mathbf{T}^{t,m+1} = \arg\min_{\mathbf{T}^t}\sum_{(k,l)\in\mathcal{N}}\sum_{\mathrm{c}_{kl}}\lambda_{kl}(\mathrm{c}_{kl},\Theta^m)\,d_{kl}^2(\bar{\mathbf{T}}^m,\mathbf{T}^t,\mathrm{c}_{kl}) \tag{3.32}$$

$$+\frac{1}{{\sigma^{t,m}}^2}\sum_{o\in\mathcal{O}_t}\sum_{\mathrm{k}_o^t}\gamma_{ot}(\mathrm{k}_o^t,\Theta^m)\|\mathbf{y}_o^t-\mathbf{T}_{\mathrm{k}_o^t}^t(\mathbf{x}_v^t)\|^2,$$

$${\sigma^{t,m+1}}^2 = \frac{1}{3}\frac{\sum_{o\in\mathcal{O}_t}\sum_{\mathrm{k}_o^t}\gamma_{ot}(\mathrm{k}_o^t,\Theta^m)\,\|\mathbf{y}_o^t-\mathbf{T}_{\mathrm{k}_o^t}^{t,m+1}(\mathbf{x}_v^t)\|^2}{\sum_{o\in\mathcal{O}_t}\sum_{\mathrm{k}_o^t}\gamma_{ot}(\mathrm{k}_o^t,\Theta^m)}, \tag{3.33}$$

$$\bar{\mathbf{T}}^{m+1} = \arg\min_{\bar{\mathbf{T}}}\; d^2(\bar{\mathbf{T}},\mathbf{Id}) + \sum_{t\in\mathcal{T}}\sum_{(k,l)\in\mathcal{N}}\sum_{\mathrm{c}_{kl}}\lambda_{kl}(\mathrm{c}_{kl},\Theta^m)d_{kl}^2(\bar{\mathbf{T}},\mathbf{T}^{t,m+1},\mathrm{c}_{kl}). \tag{3.34}$$

Expression (3.32) corresponds to simultaneous updates of all patch transformations for a given time t, weighed by E-step probabilities. (3.33) updates the per-time frame noise parameter with an E-step weighed contribution of each observation. (3.34) computes

the mean pose, accounting for all poses in the time sequence. Note that, for ease of resolution, we decouple the estimation of $\mathbf{T}^{t,m+1}$ and $\bar{\mathbf{T}}^{m+1}$, which is why (3.34) uses the result $\mathbf{T}^{t,m+1}$. Moreover, instead of fully minimizing the non-linear problems (3.32) and (3.34) at each M-step, which would be time-consuming, we follow a generalized EM algorithm [72] and only perform a decreasing step by applying only a few iterations of the Gauss-Newton algorithm to each system.

We parametrize each rigid transform as a translation and a rotation. For each rotation, the rotation group is "linearized" around the previous estimate by parametrizing the tangential space of $SO(3)$ in the matrix space $M_{3\text{x}3}(R)$ at the previous estimate. By directly parametrizing the tangential space of $SO(3)$, we keep our system as small as possible, since we use only 3 parameters for each rotation (which is the dimension of the $SO(3)$ manifold) and we do not need any additional constraints for enforcing the rigidity of the transformations. Computational details can be found in [12].

3.8 Experiments

We evaluate the proposed generative model using 3D sequences reconstructed from real human performances captured by multiple view videos. We propose two datasets, GOALKEEPER and DANCER, which provide two different actions and clothing situations with high resolution inputs acquired from 48 cameras. These were processed by extracting visual hull reconstructions, and two neutral topology frames were selected to provide the template model after smoothing and simplifying the obtained mesh down to $5k$ vertices. Additionally, we also validate using two public datasets made available by the community. The FREE [105] dataset consists of a photocoherent mesh sequence of a dancer with approximately $135k$ vertices per frame, exhibiting particularly fast and difficult dancing motion. The MARKER dataset [32] provides another type of challenging situation with a two-person sequence of reconstructions, with martial art motions. It also provides markers on one of the persons which we will use for quantitative evaluation. For both these public sequences, we use the templates provided downsampled to $5k$ vertices. We provide a supplemental video[3] with the processed results for these datasets.

In all visualizations, we render mesh poses by computing vertex position $\mathbf{x}_v^t$ at time t as a linear blend of positions $\mathbf{x}_k^t$ of expression (3.2), weighed by a set of Gaussian weights $\alpha_k(v)$ materializing the region of influence of patch P_k on the mesh. These weights are maximal at the center of mass of P_k and their sum over all non-zero patch influences are normalized to 1 for a given vertex v:

$$\mathbf{x}_v^t = \sum_k \alpha_k(v)\, \mathbf{x}_k^t \quad . \tag{3.35}$$

3.8.1 Tracking Evaluation

We first evaluate the tracking performance of the algorithm. Full sequences may be processed but because of the motion of subjects in the sequence, all poses of the sequence

[3] http://hal.inria.fr/hal-01016981

Figure 3.4 – Tracking excerpts from the DANCER dataset. Colors code patches.

cannot be initialized with a single static pose, as this would surely be susceptible to local minima. We thus process the four datasets using a sliding window strategy for $\mathcal{T}$, where processing starts with a single pose, then additional poses are introduced in the time window after the previous window converges. We provide tracking results with sliding window size 20 which corresponds to approximately one second of video. We show the resulting poses estimated by our algorithm on the four datasets in Fig. 3.4, Fig. 3.8a and Fig. 3.8b. Runtime is approximately 15 seconds per time step on a workstation[4] and can be further improved.

We also provide a comparison with state of the art methods Liu *et al.* [32] and with a purely patch-based strategy [4], on the MARKER dataset. We reproduce [4] results by neutralizing mean updates and rigid coupling updates from our method, which corresponds to removing these terms from the energy and closely mimics [4]. Note that [32] is a kinematic tracking strategy, where both subjects are rigged to a kinematic skeleton providing a strong, fixed and dataset specific rigidity prior. On the other hand, [4] only use patch rigidity and inter-patch elasticity priors, that are weaker than [32] and our method. The MARKER dataset provides sparse marker positions, at which we estimate geometric positional error with respect to the surface. To this purpose we match the closest vertex on the template model provided, and follow it with the different methods, computing geometric errors in position with respect to the corresponding marker's position in these frames. The average errors are shown in Table 3.1. We also provide a temporal error graph for our method and [4] in Fig. 3.5.

Table 3.1 shows that our method achieves comparable tracking performance to state of the art surface tracking techniques. The slightly higher error with respect to [32] is not unexpected given that they use a stronger kinematic skeleton prior. Regarding [4], the graph and table show a small advantage in error for our method along the sequence, as well as a smaller variance of the error, showing the better constraining provided by our framework. The graph also shows significantly higher error values with [4] than with our method around frames 60, 250, 325, 390 and 460. These error peaks are imputable

[4]The workstation is equipped with a dual Intel Xeon E5-2665 processor running at 2.40 GHz

Table 3.1 – Mean error and standard deviation over the sequence of the MARKER dataset.

method	mean error (mm)	standard deviation (mm)
no coupling, no mean pose [4]	55.11	48.02
our method	43.22	29.58
Liu *et al.* [32]	29.61	25.50

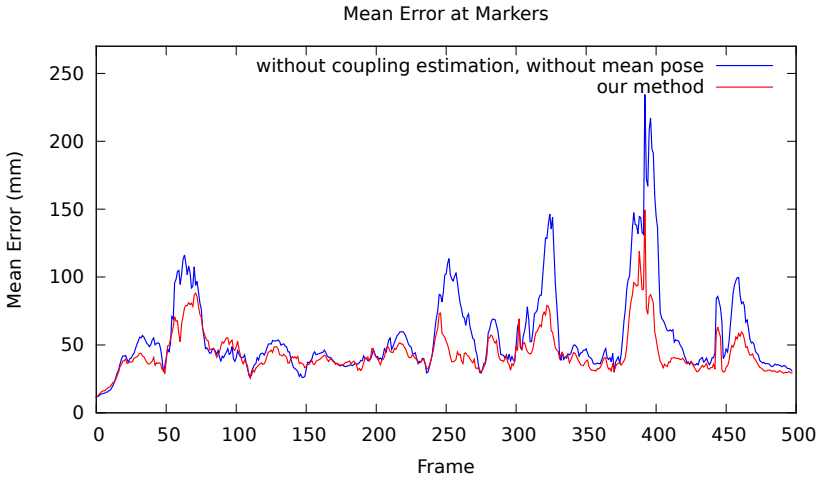
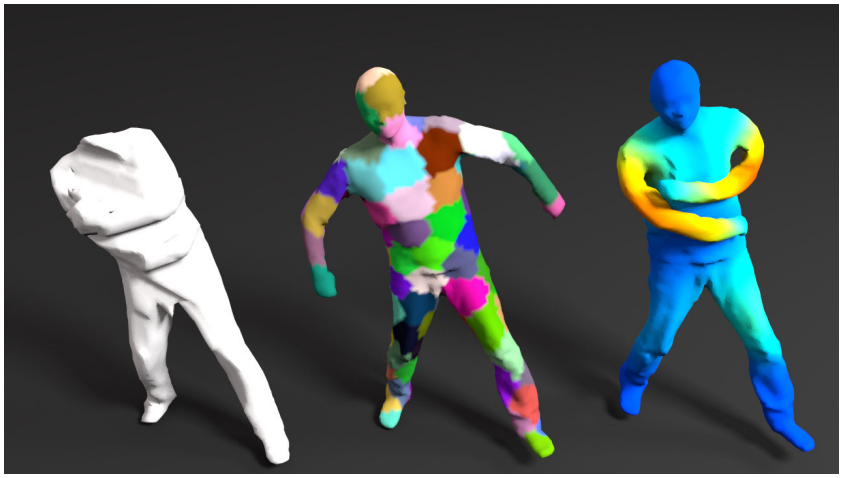


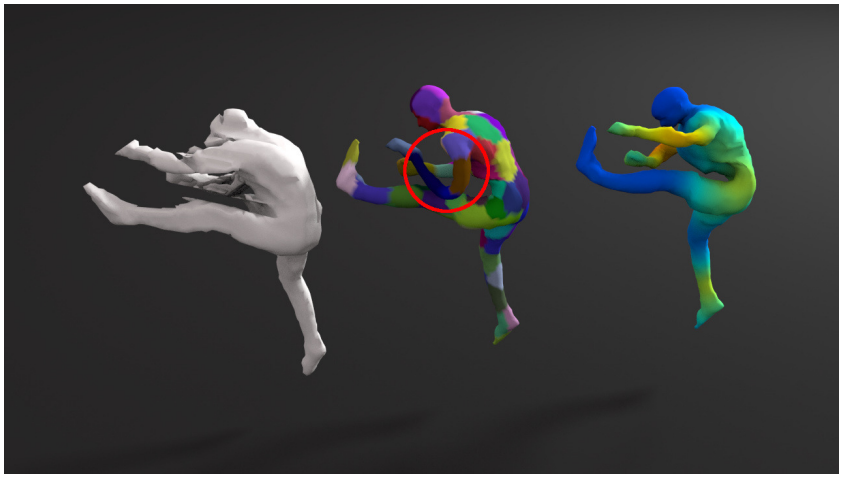
Figure 3.5 – Mean error for temporal evolution over MARKER dataset.

to difficult segments of the input sequence where [4] loses track of limbs (see Fig. 3.6a and Fig. 3.6b) while our method does not. The high error values around frame 390 are due to ambiguous input meshes where the head of the second character (not seen in Fig. 3.6b) is out of the field of view. Around this frame, our method still outperforms [4] which misaligns an arm (see Fig. 3.6b). These results substantiate stronger robustness for our method over [4].

Limitations. Regarding limitations, the model may fall into local minima when the noise level of inputs is too high similarly to all patch-based methods but this was not a strong limitation on the datasets. As the model favours rigidity and isometric surface deformations, the surface sometimes overfolds in non-rigid sections (as sometimes seen in the supplemental video and in Fig. 3.7), which we will address in Chapter 4. Since our framework does not directly optimize the blended vertex positions (3.35), the resulting tracking precision is sub-optimal. Optimizing the blended version complicates the optimization, but it is necessary for achieving high-precision tracking [89]. However our work is more focused on improving tracking robustness than achieving such an accuracy.



(a) Frame 325



(b) Frame 390

Figure 3.6 – Input mesh (left), tracked mesh with [4] (middle) and with our method (right). The absence of the head on the top-left input mesh is imputable to the visual hull computation method, which assumes full visibility for each camera.

3.8.2 Mean Pose and Rigidity Estimation

We visualize the rigidity coupling probabilities over the surface with heat-colored probabilities, by diffusing this probability over vertices of influence of patch pairs to obtain a smooth rendering, as shown in Fig. 3.8. Fig. 3.8a shows tracking results with color coded rigidity coupling probabilities with sliding window size 20. The method accurately reports instantaneous rigidity deviation, such as when the subject folds his elbows or shoulders. Blue regions correspond to regions of the mesh that have no non-rigid distortion with respect to the estimated mean pose. Fig. 3.8b shows estimates of mean poses for full sequences, colored with the estimated rigidity coupling probabilities over full sequence (no sliding window). It can be noted that the method accurately reports where the most common deviations occur.

The supplemental video shows mean pose sequences for several sliding window

Figure 3.7 – Overfolding effect on the DANCER dataset.

sizes. We observe a temporal smoothing of the initial deformation: fast deformation is filtered out. This effect is stronger with wide windows. We interpret this phenomenon as follows: when the temporal window slides along the sequence, it produces a mean pose sequence analogous to the convolution of the estimated pose sequence with a gate function, with the same size as the window size. This process can be seen as a low-pass filtering of the sequence poses.

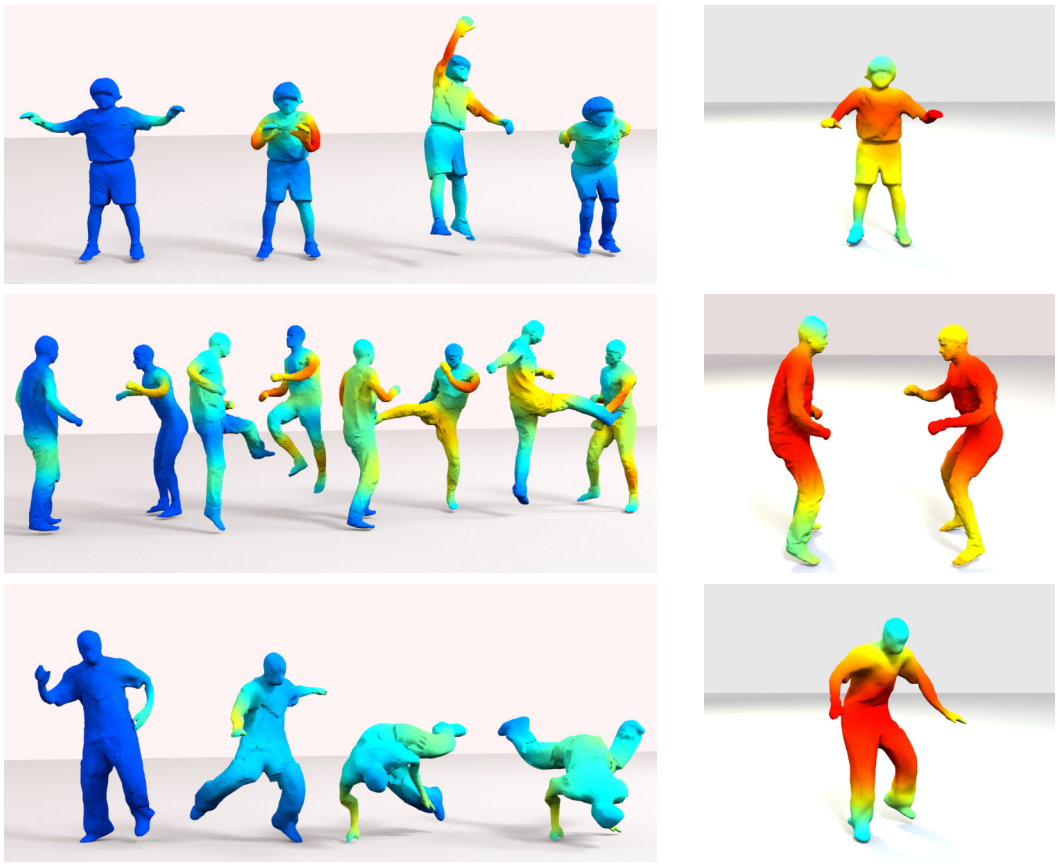
We also observe that the mean pose is not affected by global rigid motion of the shape (noticeable with the DANCER dataset). This is an expected consequence of using a pose distance that is invariant under global rigid transforms in (3.1).

While some template-based tracking approaches require a template mesh which has both a good topology and a neutral pose (which can be difficult to find in a given sequence), our framework removes the pose neutrality condition from the template mesh, since the estimated mean pose provides a neutral pose.

Limitations. The level of detail of the local rigidity analysis is the size of a patch. Finer detail of the rigidity model may be obtained when using a finer patch resolution. However, the pose estimation of smaller patches is more subject to noise, which leads to capturing noise in the variability analysis instead of true deformation variability.

3.9 Conclusions

In this chapter, we have proposed two parameters for representing intrinsic deformation properties of a deformable object: a mean pose, and a set of local rigidity probabilities. We have built a deformation model conditioned by these parameters, and we've proposed a Bayesian framework which allows for the simultaneous estimation of these intrinsic parameters and template-based tracking of a sequence with the generalized EM algorithm. Our experiments show improved or comparable tracking accuracy than state



(a) Tracking Excerpts.

(b) Mean poses computed on full sequences.

Figure 3.8 – Tracking excerpts from GOALKEEPER, MARKER and FREE datasets. Best viewed in color. Please watch supplemental video for more visualizations.

of the art tracking methods. Moreover, the inferred intrinsic parameters seem plausible when visualized over the shape.

The contribution of this chapter is not only to provide a new state of the art shape tracking algorithm, but also to open a novel research direction for improving deformable shape tracking: the combination of deformation analysis and shape tracking in a unified model which allows to perform these two tasks simultaneously and in collaboration with each other. Central to this unified framework is the embedding of intrinsic deformation parameters in the deformation model. It allows the deformation model to be refined from a general-purpose surface deformation model to a deformation model specialized to the observed dynamics. Hence, the overall tracking approach is still applicable to general shapes, and it benefits from the advantage of specialized deformation models: a better conditioning of deformation. Thus this approach creates a bridge between general and specialized deformation models.

The proposed framework has several limitations. The parameters we chose for representing intrinsic deformation properties are relatively simple, so they capture only partially the true intrinsic deformation properties. They are nevertheless sufficient for

inferring relevant and useful deformation properties, thereby validating the general approach. We hope that future works of the community will propose models with a greater expressivity. Moreover, our approach assumes that the intrinsic deformation properties can be expressed as local surface deformation constraints. This assumption is questionable when the nature of the object is volumetric, since local volume consistency constraints cannot be expressed with our surface-based model. Sometimes, this problem leads to artifacts in the tracking results such as the overfolding of surfaces and the non-preservation of local volumes. Using a deformation model which is able to express volumetric constraints would also be a way to obtain a deformation model that suits the object. This idea is developed in the next chapter, where we propose to revisit deformable shape tracking with volumetric shape representations.

Chapter 4

Exploring a Volumetric Shape Representation

4.1 Introduction

In the previous chapter, we showed that it is possible to specialize a general deformation model to a particular object thanks to the learning of intrinsic deformation properties, and that it is beneficial to the tracking accuracy. We have used a deformation model that represents shapes with a surface and penalizes local surface deformation, which constitutes a prior of elastic shell deformation. Such deformation models are relevant for cloth, in particular for clothed human motions as the observed surface is cloth. However, these surface-based models miss the volumetric nature of deformation when the object is intrinsically volumetric, for example the human body motion which conditions clothes deformation. In particular, we observed artefacts such as surface overfolding and other local volume changes when tracking human body motion. In this chapter, we propose an algorithm which uses a volumetric representation for the shape tracking problem, with the prospects of improving the tracking of volumetric deformable objects.

Ever since the initial promises of free viewpoint video [106], many models of shape capture have been explored. In any case, however, surface-based models, such as meshes, have been largely dominant to represent and track shapes. This is due to several factors, primarily to the fact that visual observations generally lie on the shape surface, but also to the popularity of surface-based representations in the vision and graphics communities and the availability of efficient tools to manipulate them. Yet it has been observed that certain forms of volume-preserving deformations may be beneficial to model shape deformations in graphics applications such as [107, 28, 11], or to enforce volumetric constraints, nevertheless based on surface tessellations, in dynamic shape capture [48].

While the idea has led to interesting preliminary results, a full volumetric treatment of dynamic shape capture is still to be investigated and its benefits evaluated. Among the expected advantages of this approach are its ability to express volume conservation as well as its ability to enforce local volumetric deformation constraints. In this chapter, we address this problem with a twofold contribution: we first propose a dedicated volumetric deformation model based on *Centroidal Voronoi Tessellations* (CVT) [1] as

well as a volumetric observation model, which extend the framework of Chapter 3, and second we propose an evaluation of the method based on a hybrid multi-camera and marker-based capture dataset [32].

4.1.1 Previous Work

A large set of techniques exist to capture moving shapes as a time independent sequence of meshes representing the object's surface [108, 109]. For this process, many volumetric parameterizations have also been devised, based on regular or hierarchical Eulerian grid discretizations [110, 111], although they are mainly dedicated to single time occupancy representation. Some approaches have taken these representations a step further, by examining short term motion characteristics of the shape using regular volume grids [112, 113, 114], yet they do not retrieve long term motion information of the sequence, nor do they embed specific motion models in the volume.

Various methods attempt leveraging time consistency to retrieve temporally coherent shape models, in the vast majority of cases manipulating a surface model. While in some cases this process is purely data-driven, by aligning surfaces across frames using sparse matching and stereo refinement [115], in most cases a deformation prior is used to drive the method toward the solution within a plausible state space. In its weakest form and without further assumptions, pure spatio-temporal continuity of the observed surface can be used [116]. At the other end of the spectrum a full kinematic rigging of a template model can be assumed, where the surface is expressed from kinematic parameters using e.g. the linear blend skinning deformation framework [117] popularized for 3D animation in the computer graphics community. These parameters can then be estimated for best fitting the model reprojections to image and silhouette data [31, 90, 91, 32]. For tracking more general subjects and situations, more generic surface deformation frameworks have been explored to bypass the rigging stage and allow for more general non-rigid motion components. Central to these methods is the idea of enforcing a cohesive behavior of the surface, such as locally rigid behavior [118], Laplacian deformation [46, 48, 50], inextensibility [40], or elasticity between piecewise-rigid surface patches [12, 50].

Among the existing surface capture methods, only a handful use volumetric representations. Some methods have proposed reparameterizing temporally aligned sequences using a volumetric cage embedding [36, 39] inspired from the animation community [119, 34]. However, no volume deformation model strong enough to solve the full tracking problem has yet emerged from these works. Among the methods successfully using volume preserving constraints, most use a Delaunay tetrahedrization of reconstructed template surface points [46, 48, 50] to enforce as-rigid-as-possible or Laplacian deformation constraints common to 3D animation techniques [107, 10]. It can be noted that the proposed decomposition is not fully volumetric as it only involves tessellating surfaces without introducing inner vertices. In contrast, we propose a fully volumetric treatment of the problem, using an intrinsically volumetric tessellation, deformation model and data terms for rewarding volume alignment.

4.1.2 Approach Overview

We formulate the tracking problem as the MAP estimation of multiple poses of a given geometric template model, non-rigidly adjusted to a set of temporally inconsistent shape measurements. In multi-view camera systems, these measurements typically take the form of time independent 3D mesh reconstructions obtained from a visual hull or multi-view stereo method, which is what we assume here. To efficiently make use of volumetric information, we need to express volume conservation and overlapping constraints from the template to the observed shape volumes. For representational and computational efficiency, we thus need a proper discretization of the interior of the shape. While uniformly located in the volume, regular grids are inherently anisotropic and biased toward the axis of the template basis. Furthermore, their intersection with the object surface yields boundary voxels of irregular shape (Fig. 4.1(a)). On the other hand, the Constrained Delaunay tetrahedrization of the boundary vertices, previously used in [48, 50], yields a set of highly non-uniform tetrahedra spanning the whole interior of the volume, whose cardinality is not controlled but imposed by the surface discretization (Fig. 4.1(b)). Taking instead the Voronoi diagram of a uniform set of samples of the interior volume decorrelates the cardinality of the decomposition from the geometry, but still yields cells of irregular shape (Fig. 4.1(c)). Rejection sampling may statistically impose additional regularity, but this would only come with asymptotic guaranties attainable at high computational cost. We therefore propose to use CVT (Fig. 4.1(d)), informally a Voronoi tessellation where the samples are iteratively repositioned to coincide with the center of mass of their cell, which achieves the desired properties [1]: isotropy, rotational invariance, uniform cells of compact and regular form factor, regular intersection of boundary cells and surface, independent cardinality and practical computation.

After introducing how to define and compute CVTs in the context of our approach (§4.2), we show how this discretization can be used to define adequate volumetric deformation (§4.3) and observation (§4.4) models in the form of Bayesian prior and likelihoods. The MAP estimation proposed on this basis in §4.5 is evaluated in §4.6.

4.2 Centroidal Voronoi Tessellation (CVT)

Definitions. To tessellate the shape, we manipulate Voronoi diagrams that are restricted, or clipped, to its inner volume. More formally, let $\mathcal{S}$ be a set of 3D point samples of a volumetric domain Ω, either the template to be fitted or the observed shape for our purposes. The *Clipped Voronoi diagram* of $\mathcal{S}$ in Ω is defined as the intersection of the Voronoi diagram of $\mathcal{S}$ in $\mathbb{R}^3$ with the domain Ω. Thus the Voronoi cell Ω_s of a sample s is the set of points from Ω that are closer to s than to any other sample:

$$\Omega_s = \{\mathbf{x} \in \Omega \mid \forall s' \in \mathcal{S} \setminus \{s\} \quad \|\mathbf{x} - \mathbf{x}_s\| < \|\mathbf{x} - \mathbf{x}_{s'}\|\}\,, \tag{4.1}$$

where cells Ω_s are mutually exclusive and define a partition of Ω:

$$\bigcup_{s \in \mathcal{S}} \overline{\Omega_s} = \overline{\Omega}\,, \tag{4.2}$$

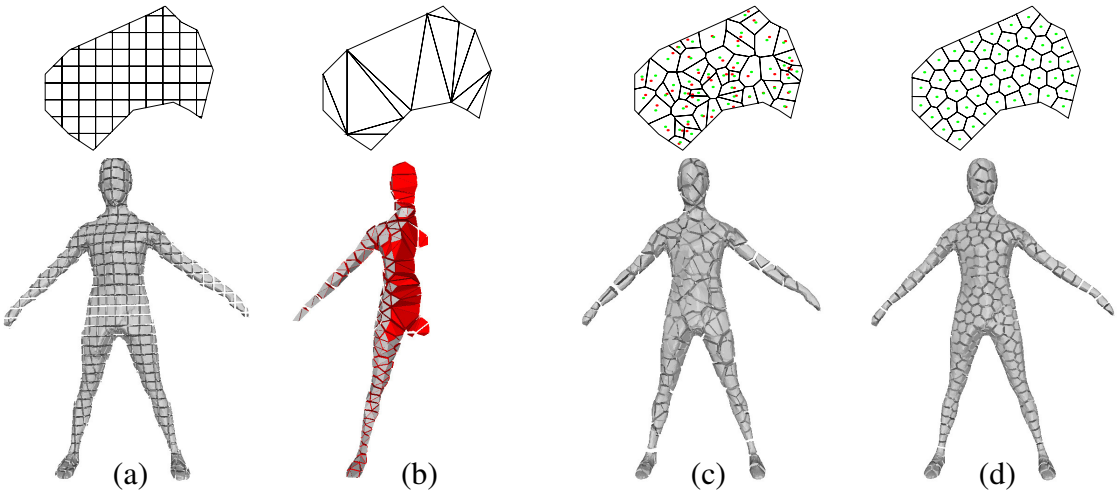


Figure 4.1 – Possible volumetric decompositions of the template and observed shapes. (top) 2D schematic view. (bottom) 3D decomposition example. (a) Voxels on a regular grid. (b) A sliced Constrained Delaunay tetrahedrization showing the elongated inner tetrahedra generated. (c) Voronoi cells with random centroids shown in red, center of mass of each cell in green. (d) Centroidal Voronoi tessellation cells, where the center of mass and cell centroid coincide.

where $\overline{\Omega_s}$ and $\overline{\Omega}$ denote topological set closures. If the border $\partial\Omega$ of Ω is a polyhedral surface, then each cell also has a polyhedral border.

A *Centroidal Voronoi tessellation* is a clipped Voronoi tessellation of Ω for which each sample s is the center of mass of its (clipped) Voronoi cell Ω_s. CVT cells are of regular size and shapes, and also define a regular connectivity of the samples set, two samples being connected if and only if their respective CVT cells share a face. This connectivity thus encodes the shape volume and topology, a property we will use in the following sections.

Computing a CVT. It has been proven [1] that local minima of the energy

$$E(\mathcal{S}) = \sum_{s \in \mathcal{S}} \int_{\mathbf{x} \in \Omega_s} \|\mathbf{x} - \mathbf{x}_s^t\|^2 dV \tag{4.3}$$

define CVTs on Ω. Thus a CVT can be obtained by iteratively estimating the sample locations that minimize (4.3), with a quasi-Newton method such as the L-BFGS algorithm [120], for a sample population of desired size and uniform initial position.

4.3 Deformation Model

4.3.1 Principle

Once a regular, anisotropic volumetric decomposition of the template shape is obtained, we can use it as a fundamental building block to build a volumetric deformation model

of the shape, which will constrain the estimation. Botsch *et al.* [11] show that a non-linear elastic deformation energy can be devised using small volumetric deformations, typically voxels. While such a reasoning could be advantageously transposed to the CVT discretization proposed, eliminating the grid orientation bias, doing so comes at a high computational cost. It has been shown [121, 12] that the complexity of the deformation model is best decorrelated from the geometry itself, for example by using rigid surface patches in lieu of the original surface vertices [12]. We have shown in Chapter 3 a way to improve the quality and temporal stability using a similar surface decomposition, by inferring a mean pose and sequence rigidity behaviour.

We extend the latter ideas to the case of a complete volumetric treatment of the deformation problem. In so doing, we cluster together groups of CVT cells in rigid volume patches P_k using a k-medoids algorithm. Note that such patches can lie either against the surface or completely inside the template shape's volume, which is of particular interest to express non-rigid deformation of the model while preserving the local volume and averting over-compression or dilation. We associate to each patch a rigid transform $\mathbf{T}_k^t \in SE(3)$ at every time t. Each position $\mathbf{x}_{k,q}$ of a mesh vertex or inner sample is indiscriminately labeled as a point q. Its position can be written as a transformed version of its template position $\mathbf{x}_q^0$ as follows, once the rigid transform of its patch is applied:

$$\mathbf{x}_{k,q} = \mathbf{T}_k(\mathbf{x}_q^0). \tag{4.4}$$

This makes it possible to define a *pose* of the shape as the the set of patch transforms $\mathbf{T} = \{\mathbf{T}_k\}_{k\in\mathcal{K}}$, which expresses a given volumetric shape deformation.

4.3.2 Formulation

To prevent patch poses of the shape from being arbitrary, we follow the approach presented in Chapter 3 which constrains the shape to be close to a sequence rest pose $\bar{\mathbf{T}}$ and to follow constant rigidity characteristics C over the sequence. These rigid characteristics are defined for neighboring patch pairs in the volume (P_k, P_l), as a binary valued property c_{kl}, whose value in $\{0, 1\}$ reflects whether the relative motion between patches P_k and P_l is respectively articulated or rigid. To define the rest pose $\bar{\mathbf{T}}$ we rely on the following measure [11, 122] of the relative deformation energy between two arbitrary poses $\mathbf{T}^i$ and $\mathbf{T}^j$ of the template shape, given a rigidity configuration C:

$$\begin{aligned}
\mathcal{E}(\mathbf{T}^i, \mathbf{T}^j|\mathrm{C}) &= \sum_{(P_k,P_l)\in\mathcal{N}} \mathcal{E}_{kl}(\mathbf{T}^i, \mathbf{T}^j|\mathrm{c}_{kl}), \quad \text{with} \\
\mathcal{E}_{kl}(\mathbf{T}^i,\mathbf{T}^j|\mathrm{c}_{kl}) &= \sum_{q\in P_k\cup P_l} \beta_{kl}(q,\mathrm{c}_{kl})\|\mathbf{T}^i_{k-l}(\mathbf{x}_q^0) - \mathbf{T}^j_{k-l}(\mathbf{x}_q^0)\|^2,
\end{aligned} \tag{4.5}$$

where $\mathbf{T}^i_{k-l} = \mathbf{T}_l^{i\,-1} \circ \mathbf{T}^i_k$ is the relative transformation between patches P_k and P_l for pose i, and $\mathcal{N}$ is the set of neighboring patch pairs on the surface. The energy measures the rigid deviation from pose i to j of every neighboring patch pair, as the sum over each of the samples s of the pair, of the discrepancy in relative positions of the vertex as displaced by P_k on one hand, and P_l on the other. If the two patches are rigidly linked ($\mathrm{c}_{kl} = 1$), then the discrepancy of all samples of the pair should be equally

penalized, therefore $\beta_{kl}(s,1)$ is chosen to be constant over all samples s of the pair. On the other hand, if the patch pair is articulated ($\mathrm{c}_{kl} = 0$), only samples that lie near the boundary between the two patch volumes should be penalized for deviating relative positions: those samples materialize the locus of the inter-patch articulation, whereas samples that aren't close the inter-patch boundary can move more freely. We express this using $\beta_{kl}(q,0) \propto \exp(-\frac{b_{kl}(s)}{\eta \bar{D}})$ where $b_{kl}(s)$ is the distance between the sample s and the boundary between P_k and P_l on the template, $\bar{D}$ is the average patch diameter and η is a global coefficient controlling the flexibility.

4.3.3 Resulting Pose Likelihoods

The relative pose energy described in (4.5) makes it possible to express the expected behavior of the estimated models as a prior and likelihood over the poses:

$$p(\bar{\mathbf{T}}) \propto \exp(-\mathcal{E}(\bar{\mathbf{T}}, \mathbf{Id})), \tag{4.6}$$

$$p(\mathbf{T}^t \mid \bar{\mathbf{T}}, \mathrm{C}) \propto \exp\left(-\mathcal{E}(\mathbf{T}^t, \bar{\mathbf{T}}|\mathrm{C})\right). \tag{4.7}$$

$p(\bar{\mathbf{T}})$ is the prior over the rest pose, which should minimize the relative displacement energy to the default template pose (transformed by identity $\mathbf{Id}$). This terms ensures minimal cohesion of the volume patches of the rest pose model, as it enforces mutual patch elasticity.

$p(\mathbf{T}^t|\bar{\mathbf{T}}, \mathrm{C})$ is the likelihood of a given tracked pose at time t, which should minimize the relative displacement energy with respect to the sequence rest pose $\bar{\mathbf{T}}$ given a current rigidity state C. This ensures the inter-patch cohesion of pose $\mathbf{T}^t$ as well as a general proximity to the rest pose, which stabilizes the resulting pose estimates. In turn the rest pose will be simultaneously estimated as the pose which minimizes the relative deformation energy to all poses in the sequence.

For deeper understanding of this model, we refer the reader to the sections of Chapter 3 that detail the mean pose model §3.4, the local rigidity model §3.5 and their integration to the probabilistic model §3.6.

4.4 Observation Model

4.4.1 Probabilistic Shape Fitting

The observed shape Ω^t at time t is described by the point cloud $\mathbf{Y}^t = \{\mathbf{y}_o^t\}_{o \in \mathcal{O}_t}$. To describe how a deformed template can explain the observed shape, we propose a generative data term following EM-ICP, expressing how a given deformed template point predicts the position of an observed shape point o. A set of cluster association variables k_o^t is therefore instantiated for every observed point in time, indicating which cluster generates this point. For simplicity, each observation o is associated to its cluster k_o^t via the best candidate q of cluster k_o^t. The best candidate is chosen as the closest compatible sample in the cluster during iterative resolution. We consider that each cluster P_k generates observations perturbed by a Gaussian noise with isotropic variance σ^2:

$$p(\mathbf{y}_o^t \mid \mathrm{k}_o^t, \mathbf{T}_k^t, \sigma) = \mathcal{N}(\mathbf{y}_o^t \mid \mathbf{T}_{\mathrm{k}}^t(\mathbf{x}_q^0), \sigma). \tag{4.8}$$

Note that o indiscriminately refers to surface or volume sample points of the observed shape, as the principles we describe here apply to both, with the restriction that observed surface points only associate to surface template points, and volume samples are associated to volume samples of the template. As often proposed in ICP methods, we additionally filter associations using a compatibility test, described in the following sections. The compatibility test is specific to the nature (surface or volume) of the observed point and is detailed in the next paragraphs. If there is no compatible candidate in the cluster, then we set conditional probability density (4.8) to zero. We deal with outliers by introducing an outlier class among values of k, which generate observations with a uniform probability density over the scene.

4.4.2 Compatibility Tests

Compatibility tests are useful for pruning the association graph for obvious mismatches that would perturb and otherwise slow down convergence. We use two compatibility tests respectively designed for surface fitting and volumetric fitting.

Surface Observations. While surface points may be matched based on position only, obvious orientation incompatibilities can be filtered by detecting large discrepancies between the normal of the deformed template candidate point v, and the normal of surface observation vertex o:

$$\vec{\mathbf{n}}_o^t \,.\, \mathbf{R}_k^t(\vec{\mathbf{n}}_v^0) \geq \cos(\theta_{\max}), \tag{4.9}$$

where $\vec{\mathbf{n}}_o^t$ is the surface normal of observation o, $\vec{\mathbf{n}}_v^0$ is the surface normal of the template at vertex v, $\mathbf{R}_k^t$ is the rotation component of $\mathbf{T}_k^t$, and $\theta_{\max}$ is an arbitrary threshold.

Volume Observations. We introduce a compatibility test specific to volumetric fitting, by assuming that the distance of inner surface points to the shape's surface remains approximately constant under deformation. Let us define the distance between an inner shape point x and the shape's surface by:

$$d(x, \partial\Omega) = \min_{p \in \partial\Omega} d(x, p). \tag{4.10}$$

In our observation model, this hypothesis can be leveraged by the following compatibility test: a volumetric observation o can be associated to a template point s only if

$$d(\mathbf{x}_s^0, \partial\Omega^0) = d(\mathbf{y}_o^t, \partial\Omega^t). \tag{4.11}$$

To account for small deviations to this assumption, which might occur under e.g. slight compression or dilation of the perceived shape, we relax the equality constraint up to a precision ϵ, where ϵ accounts for the distance-to-surface inconsistency caused by the discrete sampling of the template. Using the triangular inequality, it can be shown that this error is bounded by the maximum cell radius over the set of the template's CVT cells. This leads to the following compatibility test:

$$d(\mathbf{y}_o^t, \partial\Omega^t) - \epsilon \leq d(\mathbf{x}_s^0, \partial\Omega^0) \leq d(\mathbf{y}_o^t, \partial\Omega^t) + \epsilon \tag{4.12}$$

For the particular case of silhouette-base observed shapes, it can be noted that reconstruction algorithms based on the visual hull inherently provides inflated estimates of the true shape. This phenomenon results in an overestimation of the distance to the surface when computed on the reconstructed shape. Hence, we only impose a volumetric inclusion constraint instead of complete depth correspondence, i.e. we only keep the right inequality from expression (4.12) in this case:

$$d(\mathbf{x}_s^0, \partial\Omega^0) \leq d(\mathbf{y}_o^t, \partial\Omega^t) + \epsilon. \tag{4.13}$$

Contrary to the surface compatibility test, this test does not depend on pose parameters $\mathbf{T}$, consequently it is robust to convergence failures of inference algorithms.

4.5 Inference

The model proposed with (4.6), (4.7) and (4.8), defines a joint likelihood over the rest pose, the rigidity configuration, all temporal poses, the observed points and their selection variables, and prediction noise σ:

$$p(\mathrm{C})p(\bar{\mathbf{T}}) \prod_{t\in\mathcal{T}} \left(p(\mathbf{T}^t|\bar{\mathbf{T}}, \mathrm{C}) \prod_{o\in\mathcal{O}_t} p(\mathbf{y}_o^t \mid \mathrm{k}_o^t, \mathbf{T}^t, \sigma^t) \right), \tag{4.14}$$

We have shown in Chapter 3 that this likelihood can be maximized using an Expectation Maximization algorithm [72, 104], yielding maximum a posteriori estimates of the pose parameters $\bar{\mathbf{T}}$, $\mathbf{T}$ and prediction noise σ. This results in an algorithm iterating between two steps.

Intuitively, The **E-Step** computes all observation cluster assignment probabilities over K, based on the distance to the predicted template positions under the currently estimated poses. Compatibility rules are applied at this stage. Probabilities over inter-cluster rigid links C are also estimated based on the current deformation energy of the poses. The **M-Step** updates the rest pose $\bar{\mathbf{T}}$, all poses $\mathbf{T}$, and prediction noise σ, using the assignment and rigid link probabilities to weigh individual observation contributions to each cluster transform estimate.

4.6 Experiments

4.6.1 Datasets

We validate our framework using four synchronized and calibrated multiple-camera datasets, labeled GOALKEEPER-13, DANCER [122], MARKER [32], and the newly proposed BALLET, whose content reflect a wide variety of shape tracking situations. DANCER is a long sequence (1362 frames, 2048×2048 resolution, 48 viewpoints) showing slow and medium speed dance moves, and thus offers good opportunity to verify the tracking stability. GOALKEEPER-13 (2048×2048, 150 frames, 48 viewpoints) illustrates a specific soccer goalie plunging move, of particular interest when the goalie

is on the floor, where the reconstruction data is of poor quality due to grazing camera angle and challenges the tracking performance. Both previous sequences otherwise have very high quality and detailed reconstructions. BALLET is a more challenging full HD (1920×1080) sequence we have acquired with fewer cameras (9 viewpoints and 500 frames) and thus coarser reconstructions, consisting in a number of ballet moves with various levels of difficulty, including fast moves, spinning and crossing legs. MARKER (1296×972, 500 frames, 12 viewpoints) is a sequence with two actors performing karate moves, illustrating the robustness to several subjects, and which was captured simultaneously with a set of sparse markers offering a reference and comparison basis with [32]. The reconstructions are of coarser quality due to relatively noisy inputs and occasional jumps where actor heads get clipped.

4.6.2 Experimental Protocol

We first select a template among the better frames with correct reconstruction topology, then compute a CVT using §4.2 with 5'000 samples per person (10'000 for the two-person MARKER sequence) and 250 clusters per person (500 for MARKER), as illustrated in Fig. 4.7. Each shape reconstruction is obtained from a silhouette-based reconstruction algorithm [109] and CVTs are also extracted (1 minute/frame). We then apply the algorithm described using a sliding window strategy over a 10 frame window, where the rest position is computed for each time slice to locally stabilize the estimation. The sequences are initially solved for frames in the vicinity of the template pose, then the solution is classically propagated to future windows as initialization. Convergence has been achieved for all frames, typically in a few hundred iterations, with a convergence time per frame of the order of a minute to a few minutes. The provided supplemental video[1] illustrates the results obtained with these sequences.

4.6.3 Quantitative Analysis

Volume Stability. We verify here the assertion that the volumetric parameterization of the tracking produces poses with stable volumes. As we use silhouette based reconstructions, it is not relevant to compare the estimated volumes with the observed shape volumes. Instead, we compute the standard deviation to this volume, in Table 4.1 and provide a comparison of these results with best runs of two state of the art methods [4, 122]. This comparison supports the initial intuition of volumetric stability in the sequence, as the standard deviation of the estimated volumes is significantly lower for our method.

Silhouette reprojection error. We evaluate the silhouette reprojection error as the symmetric pixel difference between the reprojected and the silhouette projection of the reconstructions used as observed input shapes. We then express this value as a percentage of the area of the silhouette region in each view. Table 4.2 shows statistics of this error percentage among all images for each dataset, and Figures 4.2, 4.3 and 4.4

[1] Video available at https://hal.inria.fr/hal-01141207

tracking method	std. dev. (L)	
	MARKER	BALLET
Cagniart *et al.* 2010 [4]	3.85	1.22
Allain *et al.* 2014 [122]	4.32	1.20
our method	**2.24**	**0.95**

Table 4.1 – Variation of the estimated volume over the sequence for MARKER and BALLET datasets.

tracking method	mean	std. dev.	median	max
BALLET				
Cagniart *et al.* [4]	5.74	1.88	5.48	15.20
Allain *et al.* [122]	5.81	1.70	5.61	13.77
our method, without volume fitting	4.62	1.94	**4.28**	17.20
our method	**4.56**	**1.21**	4.43	**11.00**
GOALKEEPER-13				
Cagniart *et al.* [4]	10.51	5.06	9.62	38.74
Allain *et al.* [122]	12.67	5.38	11.82	35.46
our method	**7.19**	**2.70**	**6.74**	**20.52**
MARKER				
Cagniart *et al.* [4]	10.31	3.09	9.90	35.10
Allain *et al.* [122]	9.37	2.95	**8.74**	26.09
our method	**8.95**	**2.09**	**8.75**	**20.77**

Table 4.2 – Mean and statistics of silhouette reprojection error over BALLET and GOALKEEPER-13 and MARKER datasets, expressed in percentage of silhouette area.

show the temporal evolution of the error percentage averaged over all cameras. Table 4.2 shows favorable comparisons to state of the art methods [4, 122]. In particular, the mean error and maximum error achieved by our method over the sequences is significantly lower, and exhibits lower variance. The strongest improvement is achieved on the GOALKEEPER-13 dataset, especially in the interval between frames 30 and 110 (see Fig. 4.3). The input visual hulls on this interval are particularly ambiguous because of a noisy topology (see Fig. 4.8(a) bottom image), which favors our volumetric model with respect to surface-based models. Additionally we test the influence of the volumetric data term by comparing the results with a run where it is disabled (surface data-term only), all other parameters being equal, on the MARKER dataset. Interestingly, the method still achieves better mean error than state of the art, but with less stable behavior.

Marker reference error. We use the MARKER sequence provided by Liu *et al.* [32] to sparsely compare the output quality of our method against state of the art methods. This comparison is illustrated in Table 4.3 and plotted against time in the sequence in Fig. 4.5. Again we illustrate turning off the surface data term, in which case the

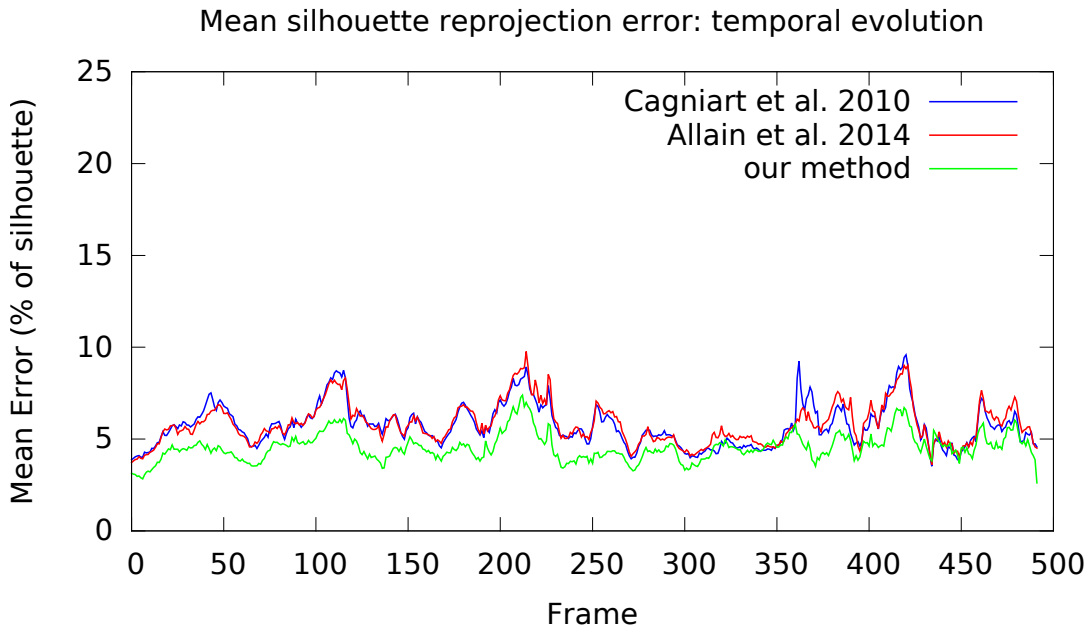


Figure 4.2 – Mean silhouette reprojection error: temporal evolution over BALLET dataset.

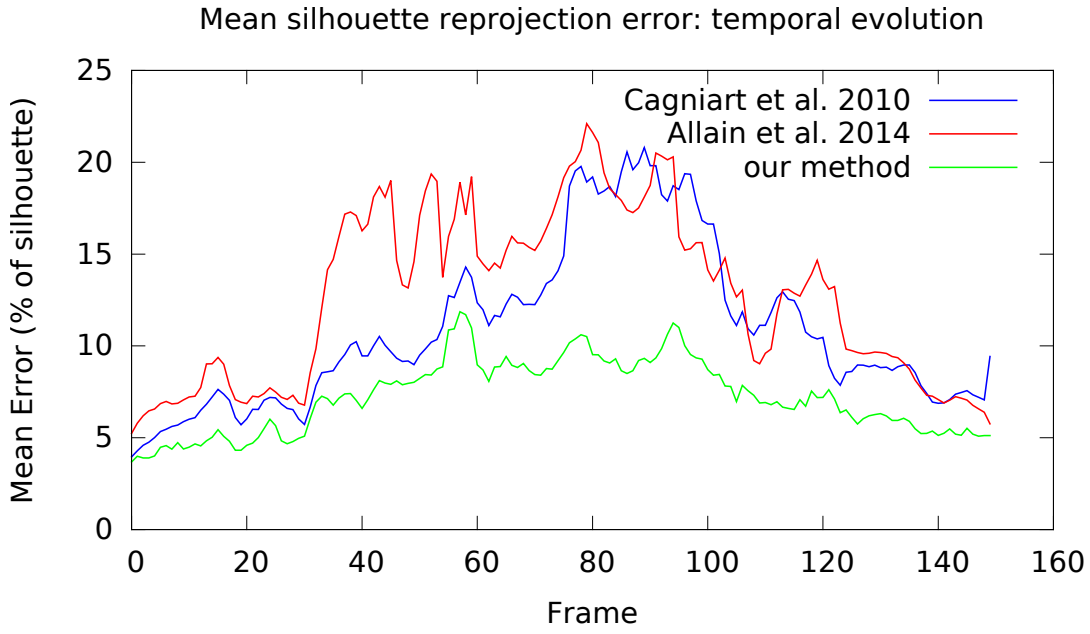


Figure 4.3 – Mean silhouette reprojection error: temporal evolution over GOALKEEPER-13 dataset.

estimation is slightly worse. The method proposed performs consistently better than comparable surface-based state of the art. Note that Liu *et al*. fully rig a skeleton to the template, which provides slightly better mean results than ours thanks to the stronger

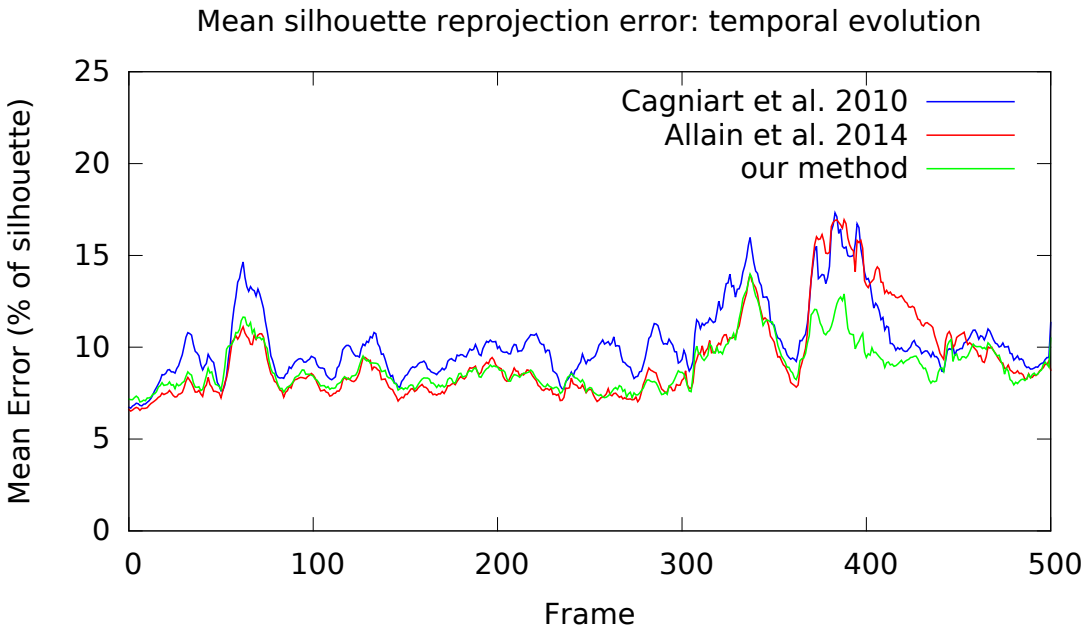


Figure 4.4 – Mean silhouette reprojection error: temporal evolution over MARKER dataset.

tracking method	mean	std. dev.
Cagniart *et al.* [4]	55.11	48.02
Allain *et al.* [122]	43.22	29.58
our method, without surface fitting	42.60	29.32
our method	**38.41**	**26.70**
Liu *et al.* [32]	**29.61**	**25.50**

Table 4.3 – Mean and standard deviation of the marker registration error, expressed in millimeters, MARKER dataset. Note that Liu *et al.* assume a rigged skeleton is associated to the template, a stronger and more restrictive assumption.

assumption. On the other hand, our method is generic and can be applied to arbitrary objects.

4.6.4 Qualitative Assessment

To illustrate the benefits of the approach, in particular where the improvements can be seen, we provide excerpts of the datasets (see supplemental video for further examples). Fig. 4.8 shows the improvements of the method over surface-based methods [4, 122] in poses of strong contortion, such as an elbow or knee bending gesture. Because of their use of the elastic energy on the surface, these methods tend to dilute error compensation over a smooth and extended location of the folds, yielding curvy elbow and knee shapes in the tracking. A usual side effect here is the local decrease of the shape volume in the vicinity of the fold. In contrast, our method being volumetrically constrained, it

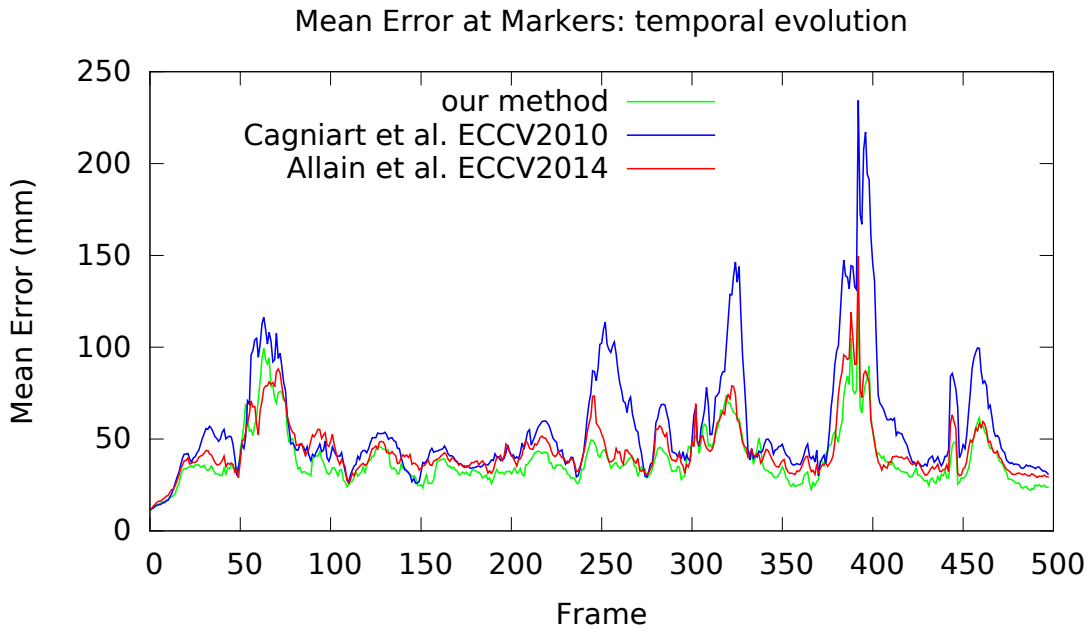


Figure 4.5 – Mean error for temporal evolution over MARKER dataset.

penalizes such local volume changes and prefers to focus the bending energy on fewer volumetric patch articulations, yielding more consistent and accurate pose estimates. This effect is particularly visible in Fig. 4.6, which shows that the undesirable over-folding effect of surface-based tracking (of Chapter 3) is corrected with our volumetric tracking approach. The GOALKEEPER-13 dataset illustrates the increased robustness in the presence of highly perturbed reconstructions thanks to the volumetric constraints, where other methods yield random results. The reconstructed visual hull input is very ambiguous on the frame shown in Fig. 4.8 because of the presence of concavities and the strong topology mismatch creates errors for surface-based methods.

4.7 Conclusions

In this chapter, we have proposed a novel volumetric tracking framework. Based on CVT decomposition and its properties, this framework extends the surface tracking framework of Chapter 3 to the volume tracking problem. In particular, we have proposed a volumetric deformation model which preserves the shape volume locally by enforcing local rigidity constraints, and a robust volume fitting term which aligns inner distance fields thanks to a discrete probabilistic assignment scheme between CVT cells. Numerical analysis shows significant improvement over state of the art tracking methods, both in terms of tracking error over the surface and silhouette reprojection. The framework is also shown to conform to initial intuition in being more stable in terms of the errors and volume measures of the fitted template shapes.

Let's now draw the conclusions of this chapter about the use of volumetric representation in deformable shape tracking. Among surveyed volumetric representation of

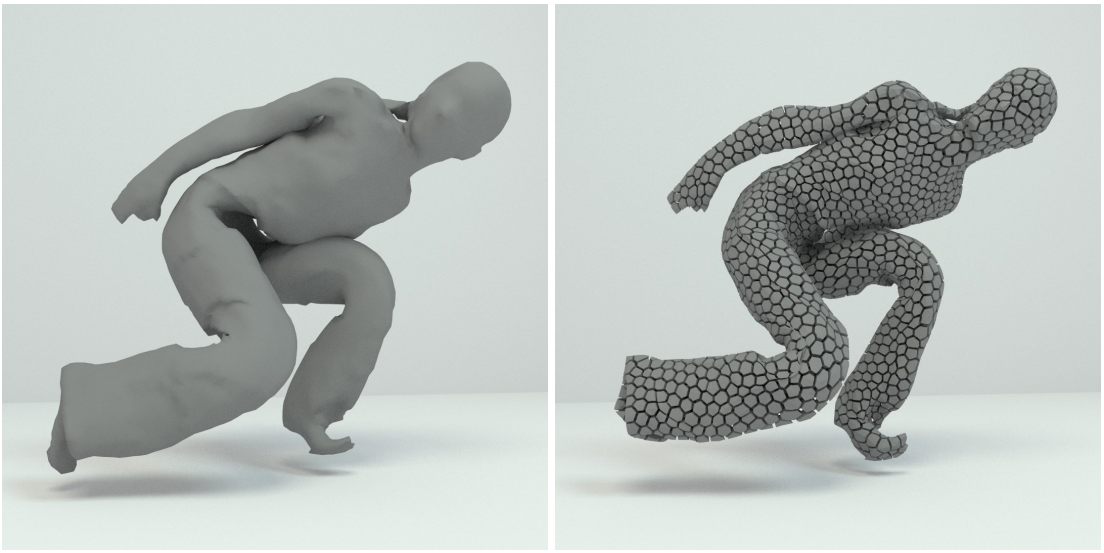


Figure 4.6 – Tracking excerpts from on the BALLET dataset, tracked with the surface-based method of Chapter 3 (left) and with our volumetric method (right). Note the absence of overfolding of the volumetric approach in the pelvis-abdomen region.

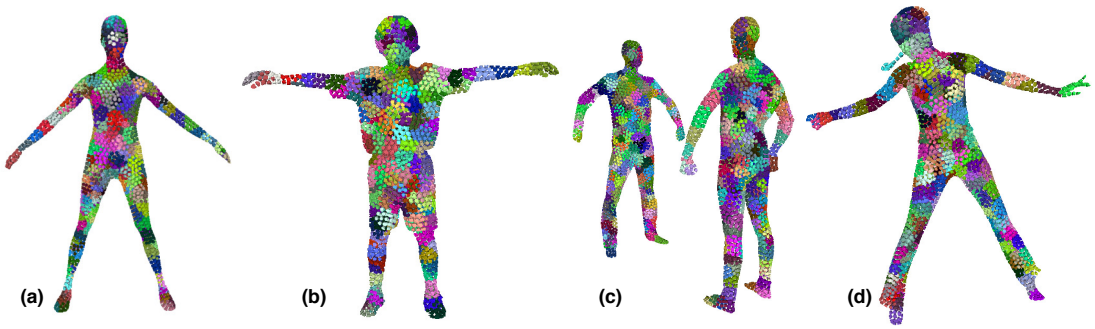


Figure 4.7 – Patched volumetric decompositions of the template for sequences (a) BALLET, (b) GOALKEEPER-13, (c) MARKER and (d) DANCER. A color has been assigned to each patch for a visualization purpose.

shapes (voxel grids, Delaunay tetrahedralization, Voronoi tessellations and centroidal Voronoi tessellations), the CVT turned out to be the best choice for our requirements regarding volumetric shape tracking, as it provides an exact shape decomposition with regular cell elements, it provides a regular connectivity and it is an efficient volume representation.

Contrary to surface representations, volumetric representations provide a support for expressing functions whose domain is the shape volume. This can be leveraged for shape tracking through a deformation model as well as an association model. With a volumetric representation, a deformation model can be equipped with a local volume rigidity constraint which enforces local volume preservation and which is a stronger constraint than local surface rigidity. This provides robustness to the shape tracking algorithm, and it is not subject to typical artefacts of surface-based tracking approaches. Currently, the volumetric clustering proposed for volumes yields uniform sizes over the

entire template shape, which can be a limitation for parts that are thinner than the cluster size, such as arms. We could address this in future work with adaptive cluster densities, ensuring the volumetric prior is equally efficient regardless of thickness.

Regarding the association models, volumetric representations allows to establish correspondences between volumetric elements. This is particularly useful when surface matching is ambiguous because of strong surface noise in the observations. In this case, volumetric matching provides robustness, since matching inner points can disambiguate the matching of surface elements, thanks to the distance field criterion. The proposed association model could be further improved with the help of a more discriminative cell descriptor than the distance field, as it would make the EM-ICP algorithm less influenced by the previous pose estimate. In Chapter 5, we take this idea a step further and shift away from ICP by estimating volumetric associations independently from any pose estimate.

The assumption of volume-constrained deformation makes difficult the fine-tracking of large clothes whose motion is not constrained by local volume preservation. Nevertheless, the most significant numerical improvement over surface-based tracking is achieved on a sequence which exhibits human motion with large clothes. This shows that volumetric tracking can provide robustness even at the border of its fundamental assumption.

To conclude, tracking shape with volumetric representations is advantageous with respect to surface-based tracking in the following situations: when the deformation is mostly or partially explained by local rigidity and volume preservation, or when surface observations are noisy and ambiguous. In those situations, it provides robustness as well as local deformation improvements. More generally, our work paves the way for the use of other volumetric priors in shape tracking. Eventually, volumetric tracking is directly useful for applications which require inner shape motion as input, such as the capture-based animation framework presented in Chapter 6.

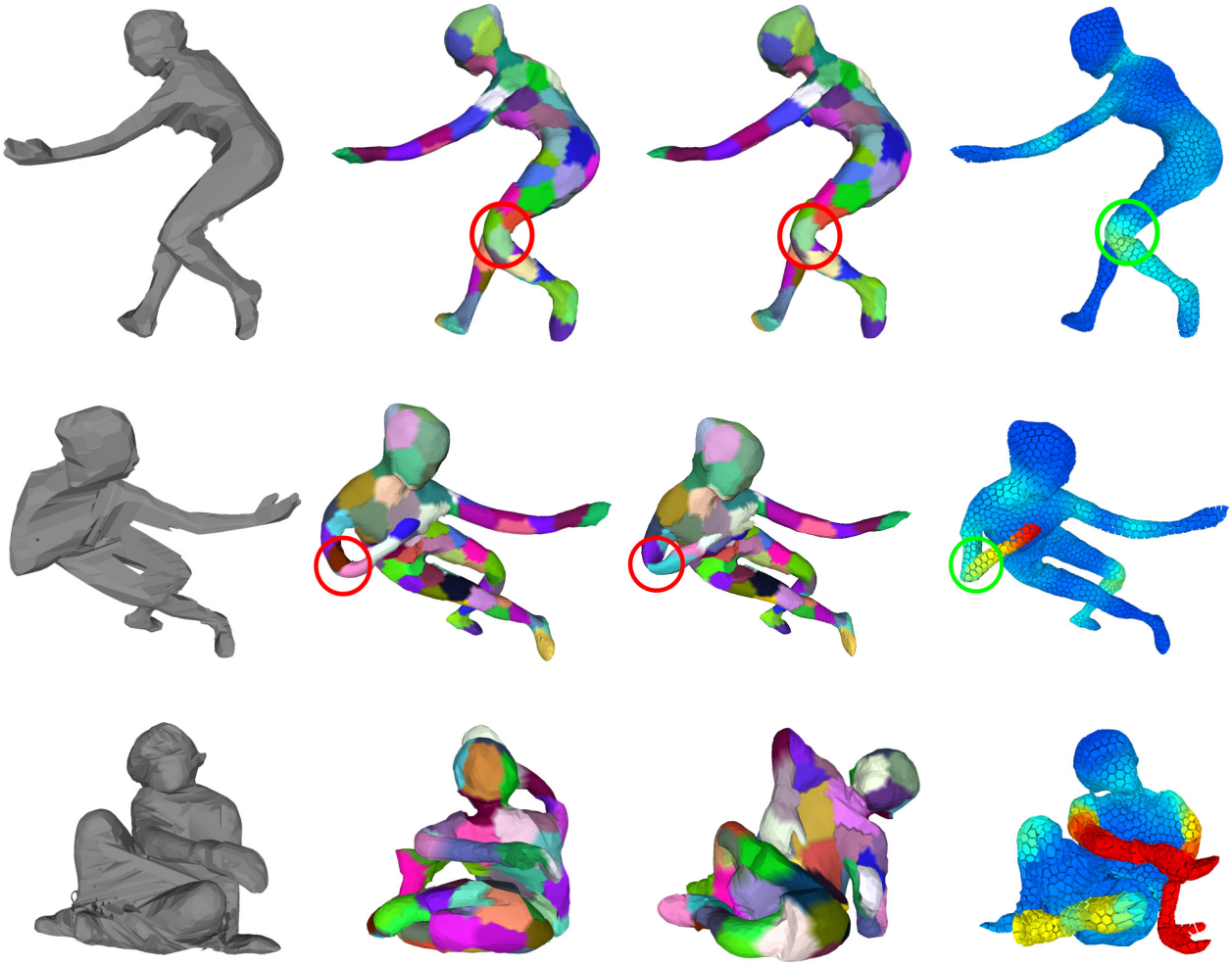


Figure 4.8 – Frames of the BALLET (top and middle) and GOALKEEPER-13 datasets (bottom). (a) Visual hull input. (b) Tracking result of Cagniart *et al.* [4]. (c) Allain *et al.* [122]. (d) Our method. Note the improved angular shapes on the dancer's knee (top) and elbow (middle), and the improved robustness (bottom).

Chapter 5

Finding Volumetric Correspondences with Regression Forests

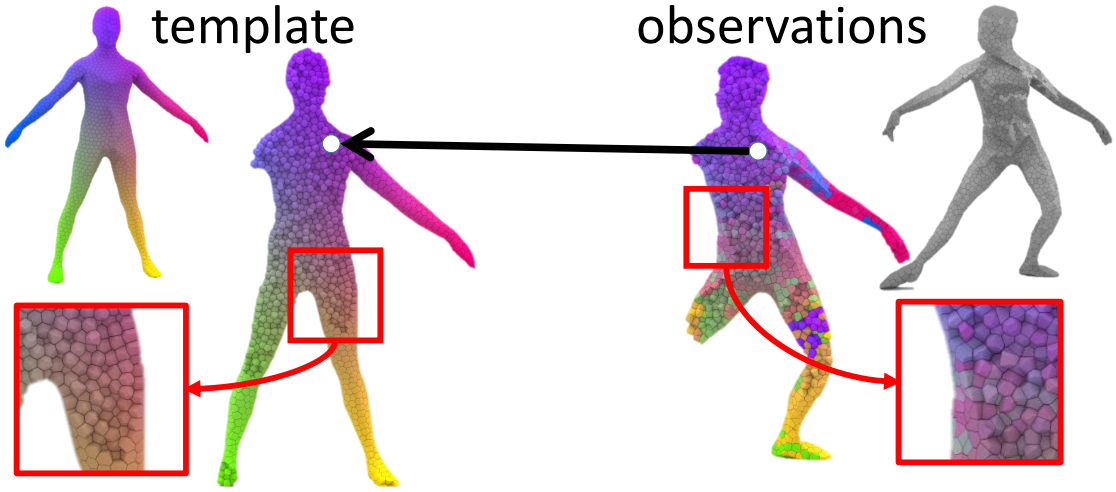


Figure 5.1 – We represent 3D shapes using centroidal Voronoi tessellations. The volumetric cells of the observations are matched to cells of the template.

5.1 Introduction

In the previous chapter, we discovered the benefits of volumetric deformation models and volumetric associations when combined in an ICP-type tracking algorithm [5]. Although the resulting algorithm is generic and efficient, it suffers from the shortcomings of ICP-type algorithms. Those iterative registration algorithms alternate between two steps: the first step consists in establishing correspondences between geometric elements of the deformed template model and those of the reconstructed shape, based on Euclidean proximity; the second step consists in aligning these corresponding geometric elements by deforming the template under the constraints of a deformation model. This iterative algorithm is sensitive to initialization, so a standard sequence tracking strategy consists in initializing the algorithm with the pose estimate from the previous frame, which is assumed to be sufficiently close to the solution. Although this strategy works

in simple cases, it is sensitive to fast motion and it tends to propagate and accumulate errors during sequence processing. In other words, leveraging temporal redundancy comes with the risk of transferring erroneous information and compromising long-term resilience.

Alternative association models have been proposed in order to overcome the shortcomings of the ICP. In particular, template correspondences can be estimated directly with discriminative machine learning algorithms [123, 8]. These association models allow for tracking-by-detection algorithms, where template correspondences are estimated only once, independently for each temporal frame, and independently of any pose estimate. This provides resilience to tracking failures and it improves convergence rates. Although surface-based features are used in many cases to describe local shapes and construct the associations, volumetric features have proven to be a promising direction for 3D shape description with surface-based templates [124].

The emergence of 3D tracking-by-detection approaches raises questions with respect to the main issues of this thesis: volumetric representations for shape tracking, and the use of temporal redundancy. Can we estimate volumetric associations by detection? What are the benefits over surface-based detection? What features are relevant for volumetric correspondences? Moreover, we ask ourselves whether machine learning-based detection are going to overcome ICP-based tracking methods. More generally, should we exploit temporal redundancy and thus introduce strong dependences between the tracking of temporal frames, or should we pursue a fully independent frame processing? What are the costs and benefits of each strategy?

In order to answer these questions, we propose a fully volumetric tracking-by-detection framework which combines the volumetric deformation model presented in chapter 4 and a new volumetric detection-based association model. Our volumetric framework is build around the CVT as a unified volumetric decomposition which is used for shape representation, feature computation, association domain and deformation model. Specifically, the observed and template shapes are all tessellated as a set of uniform and anisotropic cells (see Fig. 5.1), which bring benefits at all stages and yield a volumetric representation of regular cell shape and connectivity with controllable cell complexity.

While benefiting from local volume preservation properties inherent to this representation and the associated deformation model, we leverage the configurations of cells to build volumetric distance fields which we use to construct our volumetric feature space. On this basis, we propose a full framework to register a template shape to an observed shape, as two generic CVT cell sets. Because features are expressed in the volume, the proposed method is well suited to obtain fully volumetric detections, in turn helping the volumetric template tracking to be more robust. Thanks to its significantly low memory footprint, we use the representation to propose a multi-template learning framework, where large training sets can be assembled from multiple tracked action sequences for several human subjects. Specifically, every different subject's template is mapped to a generic, subject-agnostic template where the actual learning takes place, to benefit all subsequent tracked subjects. This framework consequently yields better or comparable detection and tracking performance than current state of the art 3D temporal tracking or tracking by detection methods.

5.2 Related Work

3D tracking by detection. The tracking by detection strategy applied to human skeletal poses estimation (Kinect) [87] has shown robustness to tracking failure and reasonable convergence efficiency in real-world applications. It was first transposed to the problem of 3D shape tracking through the work of Taylor *et al*. [8] and presented similar targeted benefits, with the initial intention to substitute ICP-based optimization. The method achieves this goal by learning the mapping from input 3D points from depth sensors, to the human template surface domain, termed the Vitruvian manifold. This yields discriminative associations that replace the step of proximity search in ICP-based tracking methods. Variants of this work have explored changing the entropy function used to train random forests from the body-part classification entropy to the variance on surface embeddings for better data separation [125], or replacing surface-based features with 3D volume features computed on a voxel grid in a local coordinate frame [124]. Both increase the precision by finishing convergence with an ICP-based loop after the discriminative association stage. All these methods are nevertheless based on surface points, thus relying on heterogeneous shape representations, deformation models, target primitives and feature spaces. Our proposal builds a unified framework for all these purposes and takes advantage of volumetric tracking strategies as described below. Also, we introduce a multi-template strategy, where a template is assigned to each subject and mapped to a generic template, allowing to learn from all subject motions sequences for the benefit of any subsequent subject tracking task.

3D volumetric tracking. While many visual tracking techniques employ skeletons [91, 31] or surface-based representations [122, 126], volume-based representations have also been proposed to address various issues. On one hand, topology changes or online surface reconstructions are better handled if surfaces are implicitly represented in volumes as *e.g*. truncated signed distance field (TSDF) [127, 26, 27], with high memory cost due to regular grids storing empty space information. On the other hand, volumetric techniques have also been devised for robustness in long term tracking, as a way to alleviate the so-called *candy-wrapper artifacts*, namely, collapsing surfaces in animations. Without explicitly tessellating surface interiors, Zhou *et al*. [28] introduce internal nodes to construct a volumetric graph and preserve the volumes by enforcing Laplacian constraints among them. Instead, Budd *et al*. [56] and De Aguiar *et al*. [48] perform a constrained tetrahedralization on surfaces to create interior edges. In Chapter 4, we generate internal points by CVT decomposition and thereby propose a generative tracking strategy that yields high quality performance. These techniques are nevertheless based on ICP-variants, whereas we aim at detecting associations discriminatively.

3D features. In many cases, surface-based features are used for recognition or shape retrieval, such as heat kernel signatures (HKS) [78] and wave kernel signatures (WKS) [79]. Both exploit the Laplacian-Beltrami operator, the extension of the Laplacian operator to surface embeddings. These features are nonetheless known for their lack of resilience to artifacts present in noisy surface acquisitions, especially significant topology changes. MeshHoG [77] and SHOT [128] attach a local coordinate frame at each point to achieve

invariant representations and reach better performance for noisy surfaces. More detailed reviews can be found in [76] and [129] for triangular surfaces and point clouds, respectively. In the context of discriminative 3D tracking, pixel difference features have been used to build random forests with depth [87, 8] or RGBD images [89], and features have been extracted from CNNs trained on body region classification tasks [88] from depth images. One common trait of the aforementioned features is that the computation involves only surface points. Huang *et al*. [124] show that features can be built based on local coordinate frames in a regular-grid volume. However, these features are only computed on surface vertices and do not address the need for fully volumetric correspondences as proposed in our work.

5.3 Preliminaries and Method Overview

Given a volumetric domain Ω defined by a shape in $\mathbb{R}^3$, CVT is a clipped Voronoi tessellation of Ω which holds the property that the seed location of each cell coincides with its center of mass. Cells are of regular size and shapes as in Fig. 5.1. A surface is expressed as the border of Ω, *i.e*. $\partial\Omega$.

Let $\mathcal{S}$ denote the set of all cell centroids. Both the template shape Ω_M and the observed data Ω_Y are expressed by their CVT samplings, $\mathcal{S}_M$ and $\mathcal{S}_Y$ with locations $\mathbf{M} \subset \Omega_M$ and $\mathbf{Y} \subset \Omega_Y$ using the method in [130]. We adopt the volumetric deformation framework of Chapter 4 that groups cells into K clusters, each having a rigid transformation $\mathbf{T}_k \in SE(3)$. The collection of all transformations, $\mathbf{T} = \{\mathbf{T}_k\}_{k=1}^{K}$, encodes the *pose* of the shape. As a result, the problem amounts to estimating the best $\hat{\mathbf{T}}$ such that the deformed template cells $\mathbf{M}(\hat{\mathbf{T}})$ resembles $\mathbf{Y}$ as much as possible. Matching cells $i \in \mathcal{S}_Y$ with cells $s \in \mathcal{S}_M$ is therefore an indispensable task. To this end, each point in $\mathbf{Y}$ is first mapped to the template domain Ω_M, where the closest point in $\mathbf{M}$ is sought as the correspondence (as represented by the green line in Fig. 5.3). This mapping $\mathbf{r} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is accounted for by a regression forest which is learned off-line with many pre-tracked CVTs (§ 5.4.1). Given the detected associations, the best pose $\hat{\mathbf{T}}$ is estimated using an EM-ICP algorithm (§ 5.4.2).

5.4 Learning and Tracking

5.4.1 Learning

We explain in this section how to learn the mapping $\mathbf{r} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ from the observation domain to the template domain with a regression forest [131], which is a set of T binary decision trees. An input cell is first described as a feature vector $\mathbf{f}$ in § 5.4.1. Taking $\mathbf{f}$ as input, during training each tree learns the split functions that best separate data recursively at branch nodes, while during testing the cell is routed through each tree, reaching T leaves that store statistics (a mode in $\mathbb{R}^3$ in our case) as predictions (§ 5.4.1). We first discuss the scenario with one single template and then generalize to multiple ones in § 5.4.1.

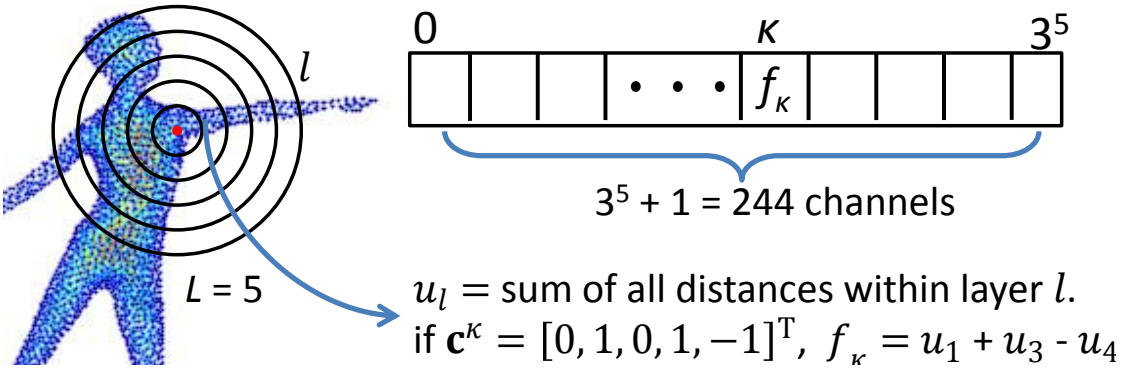


Figure 5.2 – Right: the distance field defined by a CVT sampling $\mathcal{S}$, where each cell stores the distance $d(s, \partial\Omega)$. Blue to red colors means from close to far. Red dot: cell center s to be described. Left: illustration of our feature $\mathbf{f}$. $L = 5$ in this toy example. See text for more explanations.

Feature

The feature $\mathbf{f}$ we use for building trees is designed with several principles in mind. In order to be discriminative for shape matching, our feature should be able to characterize the local neighborhood of any point of the *volumetric* shape. This rules out the descriptors that rely on surface normals such as SHOT [128]. For time and memory efficiency of forest training and prediction, we want our feature vector coefficients to be computable separately. This requirement is not met by the descriptors that rely on unit length normalization. In order to be able to match any deformed pose with the template, we would like our feature to be pose-invariant. Therefore, we build it on the Euclidean distance from cell centroids s to the surface $\partial\Omega$: $d(s, \partial\Omega) = \min_{p\in\partial\Omega} d(s, p)$ because it naturally encodes the relative location with respect to the surface and it is invariant to rotations, translations and quasi-invariant to changes of poses. While some detection approaches obtain the invariance to rigid transforms by augmenting the training set with many rotated or translated versions of each example [87, 8, 88], embedding this invariance in the feature bypasses this training set augmentation and thus improves the memory and time requirements of the training step. Finally, our feature needs to be robust to the topological noise present in the input data.

Given a distance field defined by a CVT sampling $\mathcal{S}$, our feature is similar in spirit to Haar feature in the Viola-Jones face detector [132], except that the rectangular neighborhood is replaced with a sphere. As visualized in Fig. 5.2, we open an L-layer spherical support region in the Euclidean space around each cell. An L-dimensional vector $\mathbf{u}$ is defined accordingly, where each element u_l is the sum of the distances of all cells falling within layer l. The feature value is the linear combination of all u_l, with coefficients c_l chosen from a set $\mathcal{C} = \{-1, 0, 1\}$. Formally, suppose $\mathbf{c}$ are L-dimensional vectors whose elements are the bootstrap samples of $\mathcal{C}$. Let $\mathbf{c}^\kappa$ denote one particular instance of $\mathbf{c}$, *i.e.*, $\mathbf{c}^\kappa \in \mathcal{C}^L$. The feature value is then expressed as an inner product: $\mathbf{u}^\top \mathbf{c}^\kappa$, corresponding to one feature attribute κ. We consider all possible $\mathbf{c}^\kappa$ and also take the distance d itself into account. $\mathbf{f}$ is hence a vector of $(3^L + 1)$ dimensions, where 3^L is the cardinality of $\mathcal{C}^L$ and each element f_κ is defined as:

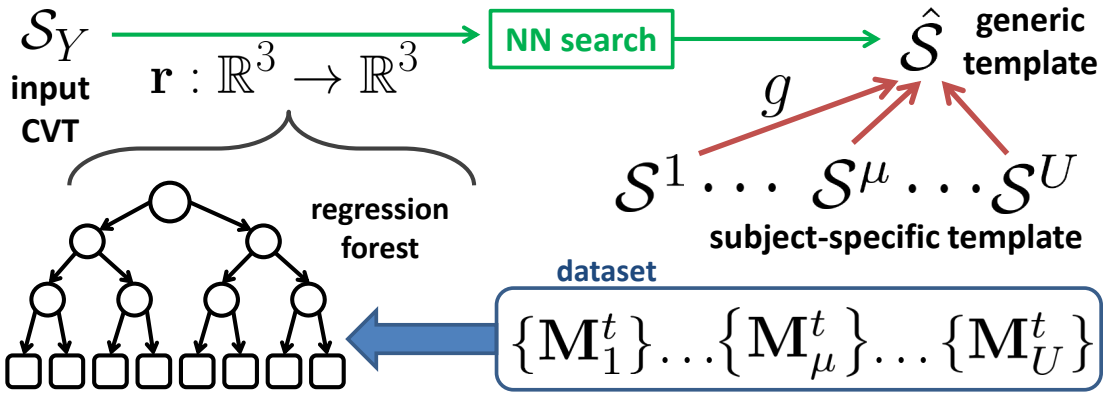


Figure 5.3 – The schematic flowchart of the multi-template learning framework. Red arrows: mappings g^μ that associate the indices from each subject-specific template $\mathcal{S}^\mu$ to the common template $\hat{\mathcal{S}}$. $\mathbf{M}_\mu^t$ are the temporal evolutions of each template. Blue: training; green: prediction.

$$f_\kappa \triangleq \begin{cases} \mathbf{u}^\top \mathbf{c}^\kappa = \sum_l c_l^\kappa u_l, & \kappa < 3^L,\ c_l^\kappa \in \{-1, 0, 1\} \\ d(s, \partial\Omega), & \kappa = 3^L \end{cases}. \tag{5.1}$$

Since each dimension f_κ is computation-wise independent, $\mathbf{f}$ is suitable for decision forests, which select feature channels κ randomly to split the data during training. Being derived from $d(s, \partial\Omega)$, $\mathbf{f}$ inherits the invariance to rigid-body motions. In addition, we normalize distances by their standard deviation in one surface, achieving scale invariance to a certain extent. However, $\mathbf{f}$ is not invariant to pose changes as the contained cells in each layer vary with poses. Although considering geodesic spherical supports instead of Euclidean ones would overcome this issue and yield quasi-invariance to pose changes, the resulting feature would be highly sensitive to topological noise. Thus, we keep the Euclidean supports and let forests take care of the variations caused by pose changes in learning.

Training and prediction

The aim of forests is to map an observed cell to the template domain Ω_M, typically chosen to be in the rest pose. Given a set of CVTs corresponding to the template Ω_M deformed in various poses, we associate each cell $s \in \mathcal{S}_M$ to its locations at the rest pose, denoted as $\mathbf{x}_s^0 \in \mathbf{M}^0$, forming a pool of sample-label pairs $\{(s, \mathbf{x}_s^0)\}$ as the dataset. Suppose $\mathcal{D}_N$ is the set of samples arriving at a certain branch node. The training process is to partition $\mathcal{D}_N$ recursively into two subsets $\mathcal{D}_L$ and $\mathcal{D}_R$ by simple thresholding on a chosen feature channel. Our splitting candidate $\phi = (\kappa, \tau)$ is therefore the pair of thresholds τ and feature attribute indices κ in Eq. 5.1. In branch nodes, many candidates ϕ are randomly generated and the one that maximizes the information gain G, $\phi^* = \arg\max_\phi G(\phi)$, is stored for the later prediction use.

We use the typical definition of information gain:

$$G(\phi) = H(\mathcal{D}_N) - \sum_{i \in \{L,R\}} \frac{|\mathcal{D}_i(\phi)|}{|\mathcal{D}_N|} H(\mathcal{D}_i(\phi)), \tag{5.2}$$

where H is the entropy, measured as the variance in Euclidean space, *i.e.* $H = \sigma^2$. We do not apply the more sophisticated measure [125] because (1) our continuous labels $\mathbf{x}_s^0$ lie in a volumetric domain Ω and (2) templates are usually chosen in canonical T or A poses. The Euclidean approximation holds more naturally here than in [124, 125], where the regression is performed along the surface manifold. The tree recursively splits samples and grows until one of the following stopping criteria is met: (1) it reaches the maximum depth, or (2) the number of samples $|\mathcal{D}_N|$ is too small. A mean-shift clustering is performed in a leaf node to represent the distributions of $\mathbf{x}_s^0$ as a set of confidence-weighted modes $\mathcal{L} = \{(\mathbf{m}, \omega)\}$. $\mathbf{m} \in \mathbb{R}^3$ is the mode location and ω is a scalar weight.

In the prediction phase, a cell $i \in \mathcal{S}_Y$ traverses down the trees and lands on T leaves containing different collections of modes: $\{\mathcal{L}_1 \cdots \mathcal{L}_T\}$. The final regression output $\mathbf{r}_i$ is the cluster centroid with largest weight obtained by performing mean-shift on them. Each observed cell then gets a closest cell p in the reference $\mathcal{S}_M$: $p = \arg\min_{s \in \mathcal{S}_M} \|\mathbf{r}_i - \mathbf{x}_s^0\|_2$. The correspondence pair (i, p) serves as input to the volumetric deformation framework described in § 5.4.2.

Multi-template learning

The above training scenario requires deformed CVTs of consistent topology such that one can easily assign each cell sample s a continuous label which is its rest-pose position $\mathbf{x}_s^0$. It hence applies only to one template. However, the amount of training data for one single template is often limited because a fully volumetric shape and pose modeling framework is still an open challenge. To avoid over-fitting, the rule of thumb is to incorporate as much variation as possible into training. This motivates us to devise an alternative strategy that learns across different CVT topologies.

Given U distinct CVT templates: $\{\mathcal{S}^\mu\}_{\mu=1}^U$ [1], whose temporal evolutions are recovered with the method described in Chapter 4, resulting in a collection of different templates deformed in various poses: $\{\{\mathbf{M}_1^t\} \cdots \{\mathbf{M}_U^t\}\}$ as our dataset. To include all of them into training, we take one generic template $\hat{\mathcal{S}}$ as the reference. Intuitively, if there exists a mapping g that brings each cell $s \in \mathcal{S}^\mu$ to a new cell $g(s) = \hat{s} \in \hat{\mathcal{S}}$, one only needs to change the template-specific labels $\mathbf{x}_s^0$ to the corresponding $\mathbf{x}_{\hat{s}}^0$, which are common to all templates, and the training process in § 5.4.1 can again be applied. In other words, we align topologies by matching every template $\mathcal{S}^\mu$ to $\hat{\mathcal{S}}$. Fig. 5.3 depicts this multi-template learning scheme.

Although various approaches for matching surface vertices exist, only a handful of works discuss matching voxels/cells. Taking *skinning weights* as an example, we demonstrate in the following how to adapt a surface descriptor to CVTs. Note that the goal of this chapter is not to propose a robust local 3D descriptor. With proper modifications, other descriptors can be used as well for shape matching.

[1] The template suffix M is dropped to keep notations uncluttered.

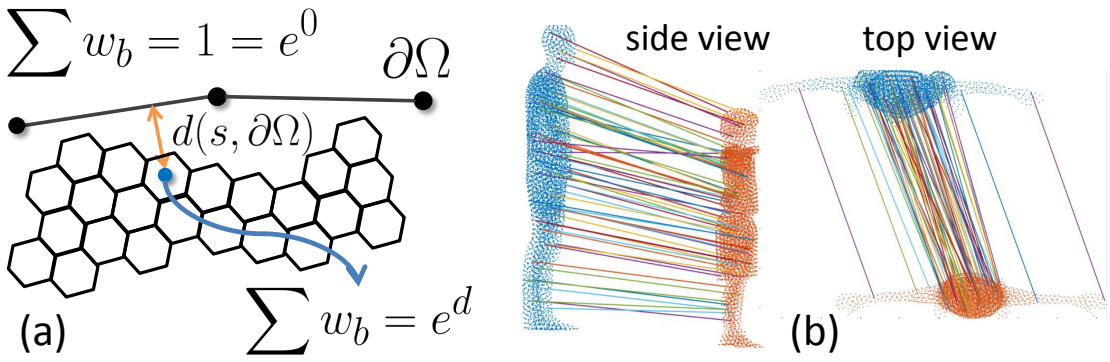


Figure 5.4 – (a): illustration of our strategy adapting skinning weights to CVT cells. Distances $d(s, \partial\Omega)$ are reflected in normalizations. (b): result of matching two templates.

Generalized skinning weights. Skinning weights are originally used for skeleton-based animations, aiming to blend the transformations of body parts (bones). Usually coming as a side product of the skeleton-rigging process [30], it is a vector $\mathbf{w}$ of B-dimensions, each corresponding to a human bone b and B is the number of bones. The non-negative weight w_b indicates the dependency on that part and is normalized to sum up to one, *i.e.* $\sum_b w_b = 1$. As such, a skinning weight vector $\mathbf{w}$ is actually a probability mass function of body parts, offering rich information about vertex locations. To extend it from surface vertices to CVT cells, we first relax the unity-summation constraint as $\mathbf{w}$ is not used to average transformations of bones but only as a descriptor here. The intuition behind the adaptation is that, a CVT cell should have bone dependencies similar to the closest surface point. Therefore, for a cell whose distance to the surface is d, its skinning weight is simply the one of its closest surface point[2], scaled by e^d. Note that this does not violate the unity-summation constraint for surface vertices as their distance d is still zero. We illustrate this concept in Fig. 5.4(a). The mapping g is then determined by searching for the nearest neighbor in the skinning weight space: $g(s) = \arg\min_{\hat{s}\in\hat{\mathcal{S}}} \|\mathbf{w}_{\hat{s}} - \mathbf{w}_s\|_2$.

In practice, we use `Pinocchio` [30] to computes skinning weights, extend them from surface vertices to CVT cells, and match all cells to those of the common template $\hat{\mathcal{S}}$. The resulting skeletons are not used in our method. Fig. 5.4(b) visualizes one example of matching results. Our approach yields reasonable matches, regardless of the difference in body sizes. Due to the descriptiveness of skinning weights, symmetric limbs are not confused. Note that this computation is performed only between user-specific templates $\mathcal{S}^\mu$ and the generic one $\hat{\mathcal{S}}$ off-line once. Input data $\mathcal{S}_Y$ cannot be matched this way, because rigging a skeleton for shapes in arbitrary poses remains a challenging task.

[2]When the shortest distance does not exactly correspond to a vertex but to a point in the middle of a triangle, we use barycentric coordinates as the coefficients to linearly combine the skinning weights of the three vertices.

5.4.2 Tracking

We elaborate in this section on how to apply our regression forest to track a sequence of temporally inconsistent observations. The current state-of-the-art 3D shape tracking methods usually employ non-rigid ICP algorithms [122]. Instead of performing an extensive search over all possible associations, we directly use the correspondence pair (i, p) detected by the forest as initializations. This results in a faster pose estimation. We adopt the CVT-based deformation framework proposed in Chapter 4. However, the approach we describe can easily be adapted to other ICP variants.

Bayesian tracking

Bayesian tracking such as [122] consists in maximizing the *a posteriori* probability $P(\mathbf{T}|\mathbf{Y})$ of the pose parameters $\mathbf{T}$ given the observations $\mathbf{Y}$. It can be further simplified as $P(\mathbf{T}|\mathbf{Y}) \propto P(\mathbf{T}, \mathbf{Y}) = P(\mathbf{T}) \cdot P(\mathbf{Y}|\mathbf{T})$, where the deformation prior $P(\mathbf{T})$ discourages the implausible poses and the likelihood term $P(\mathbf{Y}|\mathbf{T})$ expresses the compatibility between the observations and the pose estimate $\mathbf{T}$. Since maximizing a probability $P(\cdot)$ is equivalent to minimizing $-\log P$, it leads us to the following problem:

$$\hat{\mathbf{T}} = \arg\min_{\mathbf{T}} \left(-\log P(\mathbf{T}) - \log P(\mathbf{Y}|\mathbf{T}) \right). \tag{5.3}$$

In EM-ICP algorithms [4], the conditional likelihood $P(\mathbf{Y}|\mathbf{T})$ is expressed by introducing a set of latent selection variables $\{k_i\}_i$ that explicitly associate the cell k_i of the deformed template model to the observed cell i.

The prior on the latent association variables is usually uniform, which means that an observed point can be associated to any template point with the same probability. This leads to a long exhaustive search among all possible associations and produces a high number of residuals, slowing down the EM-ICP algorithms. Moreover, it is the source of wrong associations that guides the optimization to suboptimal local minimum.

EM-ICP with forest predictions

With a small number of possible associations provided by forests, our algorithm averts the need for an exhaustive search, and therefore highly decreases the running time of each optimization iteration. Moreover, it removes a lot of wrong association hypotheses. We integrate the predictions from the forest as a prior on the selection variable k. The selection variable k_i (for the observed cell i) follows a probability distribution where only the cell predicted by the forest has a non-zero probability.

Usually the forest outputs only one prediction per cell, which is at the mode with higher weight resulting from the mean-shift algorithm. However, because of the symmetry, the good match is often not the mode with highest weight. Thus, it makes sense to consider several modes instead of one in the prediction phase. The robust scheme described in the next section will usually select the good one.

Template / #Vertex / #Cell	Sequence	Frames
Ballet / 6844 / 5000	*Seq1* [133]	500
	Seq2	936
Goalkeeper / 5009 / 5000	*SideJump* [133]	150
	UpJump [122]	239
Thomas / 5000 / 4998	*Seq1*	1500
	Seq2	1400

Table 5.1 – Sequences used in our experiments. For each subjects, the training set is the random 250 tracked CVTs sampled from first sequences and testing on the unseen second sequence. Unreferenced sequences are the ones proposed in this chapter.

Robustness

The detection forest sometimes outputs wrong correspondences, either due the symmetry of human bodies (left-right confusion) or other detection errors (*e.g.* hand-foot confusion). Therefore, the ICP algorithm needs to be robust to wrong correspondences. We achieve this goal by using a noisy observation model [4], where the noise variance is estimated by an EM algorithm.

5.5 Experimental Results

We validate our approach with numerous multi-view sequences, whose profiles are summarized in Table 5.1. For each frame, a coarse visual hull is reconstructed by a shape-from-silhouette method [21], followed by [130] to draw CVT samplings (raw CVTs). Given a CVT template, we then perform the EM-ICP based method of Chapter 4 on the raw CVTs to recover temporal coherent volumetric deformations (tracked CVTs). We evaluate our method in two aspects: detection accuracy (§ 5.5.1) and tracking results (§ 5.5.2). Unless otherwise specified, we follow the experimental protocol below.

Experimental protocol. We first explain the settings common to two experiments. For each subject, up to 250 tracked CVTs are randomly chosen from the first sequence as the training dataset, while the second sequences are completely left out for testing. We open $L = 8$ sphere layers for the feature computation. Each tree is grown with 30% bootstrap samples randomly chosen from the dataset and trees are grown up to depth 20.

Two experiments, however, differ in the input data for testing. To evaluate the quality of estimated associations, we feed the tracked CVTs into forests due to the availability of ground truth indices (§ 5.5.1), whereas raw CVTs are used as the input for tracking experiments in § 5.5.2. Some distinct experimental settings of the two are exposed in Table 5.2.

[3]More precisely, forests in § 5.5.1 are all single-template based except for the one in "multi-template learning" paragraph.

Sect.	Forest	T	Testing data
§ 5.5.1	template-specific[3]	20	1. tCVTs of seq1 (Tr) 2. unseen tCVTs of seq2 (Te)
§ 5.5.2	multi-template	50	unseen rCVTs of seq2

Table 5.2 – Different experimental settings in two sections. tCVTs stand for tracked CVTs while rCVTs represent raw CVTs.

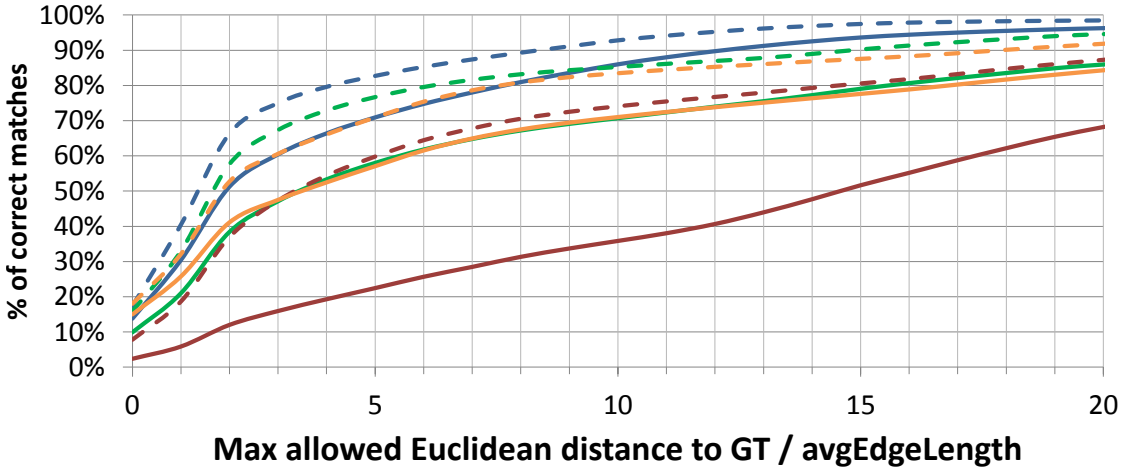
5.5.1 Matching

The contributions of CVT on improving the correspondences detection are evaluated with two experiments. First, we follow the learning framework in [124] but replace their voxel-based features with ours in § 5.4.1, denoted as *CVTfeature*. Next, we further change the regression domain from surfaces to volumes, as described in § 5.4.1 (*fullCVT*). We test on the tracked CVTs and report the results on all frames of training sequences (Tr) and testing ones (Te). The drop between them is a natural phenomenon for every machine learning algorithm and indicates the ability to generalize. If the Euclidean distances between the predicted cell index and the ground truth are smaller than a certain threshold, it is considered as correct.

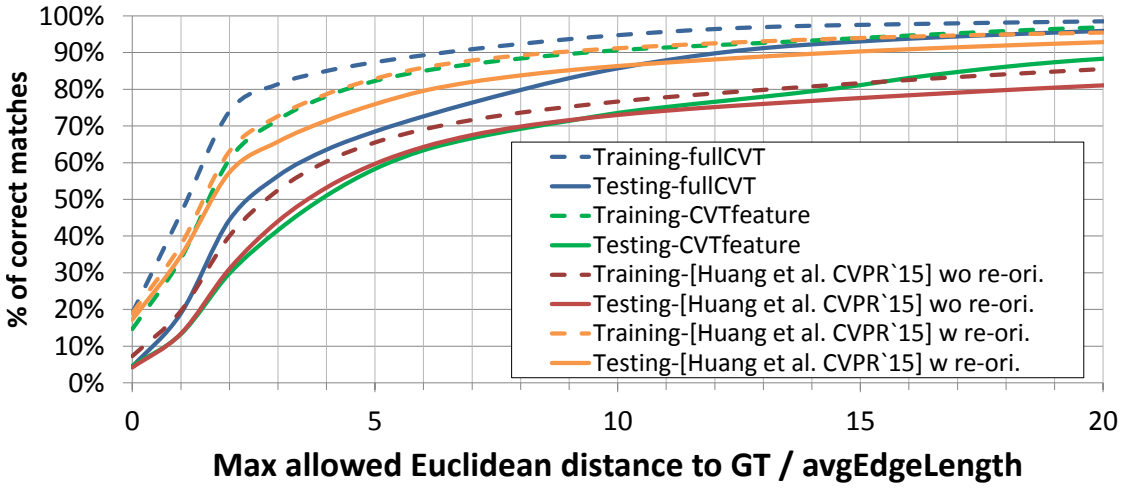
Single-template learning. To align the experimental setting, here the regression forests are subject-specific and consist of only $T = 20$ trees. Fig. 5.5 shows the percentage of correct matches in varying thresholds for *Thomas* and *Ballet*. Since *CVTfeature* and [124] are regressing to surfaces whereas *fullCVT* regresses to volumes, we normalize the x-axis by the average edge length of templates to yield fair comparisons. While the results of *CVTfeature* are comparable to [124] (green vs. red or orange), *fullCVT* attains the improved accuracies (blue vs. red or green), demonstrating the advantages of our fully volumetric framework. Some visual results of the *fullCVT* approach on raw CVT input are shown in Fig. 5.7.

Discussion. It is worth a closer analysis to compare our approach against [124]. Compared to volumes of regular grids, CVT is certainly a more memory-efficient way to describe 3D shapes. In practice, [124] describes each mesh with 150^3 voxels, while we need only 5k cells[4]. Consequently, [124] is not able to include a sufficient amount of training shapes, leading to a major drawback that forests are limited to one single subject and learn merely pose variations. To further decrease the needed number of training meshes, [124] exploits skeletal poses to cancel the global orientation. This in turn makes every mesh in the training dataset face the same direction. During tracking the input data has to be re-oriented likewise using the estimated skeletal poses from the last frame. Our approach, on the other hand, considers distance fields of CVTs which is naturally invariant to rotations and hence does not require re-orientations. We anyway compare to [124] in both settings. Orange curves in Fig. 5.5 shows the results with

[4]Further note that [124] stores a 3D vector in each voxel, whereas we store a scalar in each CVT cell. So the ratio is 3×150^3 to 5k.



(a) *Thomas*



(b) *Ballet*

Figure 5.5 – Cumulative matching accuracy of different approaches. The x-axis is normalized with respect to the average edge length of the templates. The number of trees T is 20 in this experiment. Dashed and solid lines are accuracies on training (Tr) and testing (Te) sequences respectively.

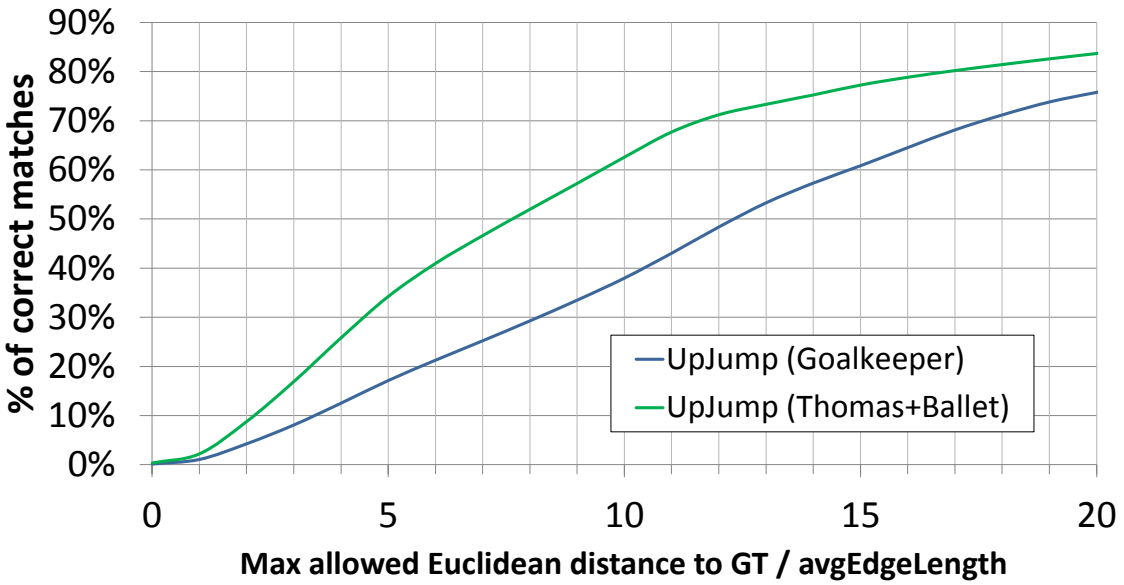


Figure 5.6 – Cumulative matching accuracy of single and multi-template strategy on *Goalkeeper*.

re-orientation, which is better than the proposed strategy in *Ballet*. Nonetheless, without re-orienting data, the accuracy drops substantially during testing (compare red to orange). The efficiency on memory and the invariance of our features are two determining reasons why the presented method is better than [124] and needs just one forest for different subjects in the following experiment.

Multi-template learning. We use the sequences of *Goalkeeper* to verify the advantages of the multi-template learning strategy in § 5.4.1. It is a particularly difficult dataset because motions in the testing sequence *UpJump* have little overlap with those in the training *SideJump*. We report in Fig. 5.6 the correctness of correspondences in *fullCVT* setting. Both curves represent the accuracy on testing *UpJump* sequence. The blue curve corresponds to a forest only trained with *Goalkeeper* tracked CVTs, whereas the green curve corresponds to a forest trained with tracked CVTs of *Ballet* and *Thomas*. For both forests, *UpJump* sequence is unseen during training. Compared with the forest of the blue curve, the one of the green curve is trained with twice the amount of meshes from different subjects, and yet it leads to better prediction accuracy on unseen testing poses. This suggests that including more variation of motions indeed results in better generalization to unseen data. It also confirms the necessity and efficacy of our multi-template strategy. We anyway point out that due to the lack of adequate amount of training data, these encouraging preliminary results need to be confirmed on datasets consisting of more subjects and sequences.

5.5.2 Tracking

We perform several experiments to evaluate our whole tracking-by-detection algorithm and compare with previous approaches using two quantitative metrics. We also show its

Figure 5.7 – Qualitative matching results on the raw CVTs. Templates are displayed at the upper left corner. Best viewed in PDF.

resilience to large pose changes and its generalization capacities on an unknown subject.

Unlike § 5.5.1, here we apply the multi-template strategy in § 5.4.1 to train one universal regression forest, with *Goalkeeper* chosen as the common template $\hat{\mathcal{S}}$. Training $T = 50$ trees up to depth 20 where each one is grown with around 200 CVTs (approximately one million samples) takes about 15 hours on a 24-core Intel Xeon CPU machine. For each subject, we track the testing sequence, which is not part of the training set. Tracking inputs are raw CVTs which have no temporal coherence. Correspondences are predicted by the forest and fed into the volumetric deformation framework described in § 5.4.2. The number of clusters K is 250 for *Ballet* and *Goalkeeper* and 150 for *Thomas*. Some visual results are shown in Fig. 5.8 and in the supplemental video[5]. With the help of regression forests, our approach is able to discover volumetric associations even in challenging poses found in *Thomas* and deform the templates successfully.

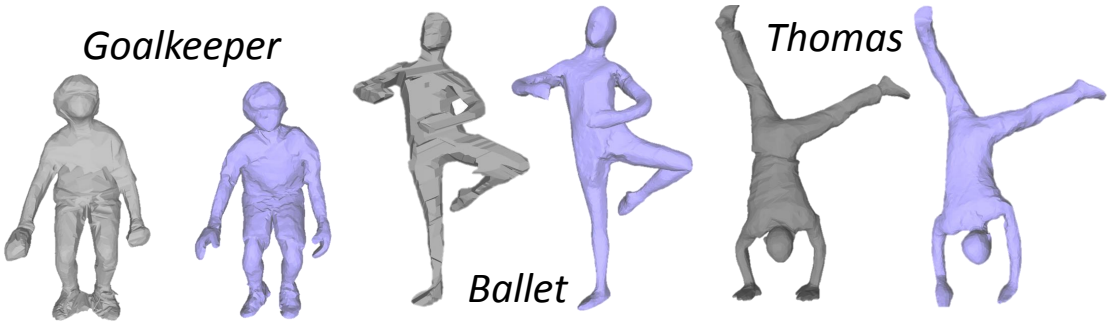


Figure 5.8 – Qualitative tracking results. Gray: input observed visual hulls; purple: deformed templates.

Quantitative evaluation and comparison. We evaluate the tracking results with two complementary metrics: silhouette overlap error, which measures the consistency between the shape and observed silhouettes, and marker location error (using marker-based motion capture data), which sparsely evaluates the surface pose. Numerical results of the silhouette overlap error (Table 5.4) and marker location error (Table 5.3)

[5] https://hal.inria.fr/hal-01300191

show similar or improved results with respect to volumetric ICP-based tracking (from Chapter 4) and surface-based tracking by detection [124].

method	mean (mm)	stddev. (mm)
Proposed	26.37	16.67
Huang *et al.* [124]	124.02	200.16
Allain *et al.* [133]	27.82	18.39

Table 5.3 – Statistics of surface registration error at marker locations, on the *Ballet/Seq2* sequence.

method	mean	stddev.	median	max
Goalkeeper/UpJump				
Proposed	15221	6843	14754	57748
Huang *et al.* [124]	19838	14260	15607	109428
Allain *et al.* [133]	14773	6378	14355	43359
Ballet/Seq2				
Proposed	2620	1041	2557	8967
Huang *et al.* [124]	5427	2809	4863	39559
Allain *et al.* [133]	2606	1008	2571	7642
Thomas/Seq2				
Proposed	9991	7089	7968	78242
Huang *et al.* [124]	28731	23421	22991	354293
Allain *et al.* [133]	10199	7379	8022	81649

Table 5.4 – Silhouette pixel error on sequences *Goalkeeper/UpJump*, *Ballet/Seq2* and *Thomas/Seq2*. Image size is 2048×2048 for *Goalkeeper* and *Thomas* datasets and 1920×1080 for the *Ballet* dataset.

Tracking at low frame rate. One of the expected benefits of our framework over purely ICP-based methods is improved resilience with large pose changes. We test this assertion by tracking the *Thomas* sequence at low frame rate (5fps). Figure 5.9 shows how our method recovers from tracking failures while the volumetric EM-ICP of Chapter 4 does not. This improvement is confirmed by the median silhouette overlap pixel error, which we found to be twice lower with our method (10054 pixels compared to 19998 pixels).

Testing with a new subject. We tested the generalization capacities of our framework with a subject (*Dancer* dataset [122]) which is not in the training data. For this purpose, one can either select an existing template from the training sequences, or build a template model by matching one of the samples from the test sequence to the common reference model using skinning weights, as we do in multi-template training. We use the latter, which is more subject specific and can be expected to yield better results. Most

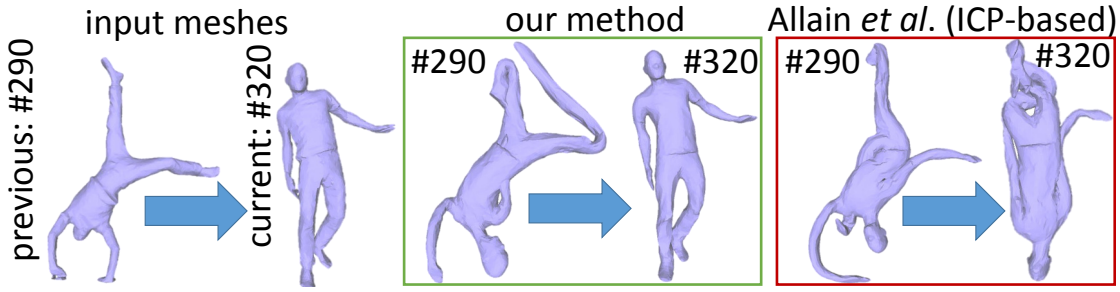


Figure 5.9 – Tracking results of *Thomas* dataset at low frame rate.

poses are correctly tracked in our experiment (see Fig. 5.10). Not unexpectedly for this type of approach, some failures occur on more complex poses unseen in training data and would probably be improved with a larger training set.

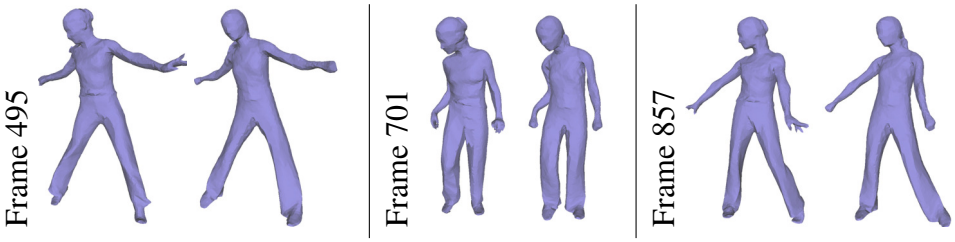


Figure 5.10 – Tracking results with a new subject, *Dancer* dataset. Input mesh (left) and tracked mesh (right).

5.6 Conclusions

In this chapter, we have introduced a fully volumetric tracking-by-detection framework. To our knowledge, this is the first attempt in the computer vision domain. Our framework uses the CVT as a unified volumetric representation which brings advantages at all stages: feature computation, association prediction domain, forest training and deformation model. We have introduced a new volumetric feature, specially designed for our problem. It is invariant to rigid transformations, and it has little sensitivity to topological noise. The CVT is an efficient volumetric shape representation for regression forest learning, as it is memory efficient, especially compared to voxel grids. This allows for larger training datasets. We further devise a multi-template learning strategy to enrich the training variation. This leads to one single forest for different subjects and yields cross-subject learning of discriminative associations.

Our experiments shows two main results. First, our volumetric detection algorithm yields a higher accuracy than the state of the art surface-based detection method considered for the comparison. This is a strong argument for the use of volumetric representations for detection-based shape association problems. Second, when compared to a volumetric ICP, our tracking-by-detection algorithm yields similar results on most testing sequences and shows a significant robustness improvement when there is large

inter-frame motion. While this result was already observed for surface-based tracking [124], our work extends it to the volumetric case.

With respect to the question of whether information should be shared between temporal frames for solving the tracking problem, tracking-by-detection is interesting as it demonstrates that isolating the association estimation process of each time sample provides robustness to tracking failures. This may seem contradictory with the conclusion of chapter 3. However the information transfer happened in different components: the deformation model.

The work presented in this chapter has several limitations. In our framework, the ambiguity between symmetric body parts is a difficulty which is solved at the tracking step. A potential future improvement would be disambiguate this at the feature level, thus providing even more independence between the temporal frames. More generally, our methodology could be easily transposed to other features emphasizing other discriminative characteristics. Furthermore, our experiments were conducted on a datasets of limited size. It would be interesting to confirm our results with larger datasets involving a wider variety of subjects and motions. Moreover, tracking-by-detection requires a labelled database with shape and poses resembling the motion to track and they require a training phase. When compared to ICP-type algorithms, their applicability and convenience are more limited.

To conclude this chapter, volumetric tracking-by-detection combines the advantages of volumetric deformation models and tracking-by-detection approaches in a unified framework, yielding a strong robustness to fast and large motions. It also suffers from disadvantages of each strategy: the assumption of volume-constrained deformation and the need for an appropriate labelled training data set.

Chapter 6

Shape Animation Combining Acquired and Simulated Dynamics

6.1 Introduction

In the previous chapters, we have shown that volumetric representations turn out to be advantageous for tracking deformable objects. Since surface tracking is a byproduct of volume tracking, the latter fullfills the requirements of any application that require surface motion as input such as free-viewpoint video. However, are they applications of shape tracking for which volumetric motion is a requirement ? We believe the availability of volumetric motion opens the way for new applications that would be impossible or complex to realize with surface motion only. We provide here an example which is the production of capture-based volumetric animation.

Creation of animated content has become of major interest for many applications, notably in the entertainment industry, where the ability to produce animated virtual characters is central to video games and special effects. Plausibility of the animations is a significant concern for such productions, as they are critical to the immersion and perception of the audience. Because of the inherent difficulty and necessary time required to produce such plausible animations from scratch, motion capture technologies are now extensively used to obtain kinematic motion data as a basis to produce the animations,

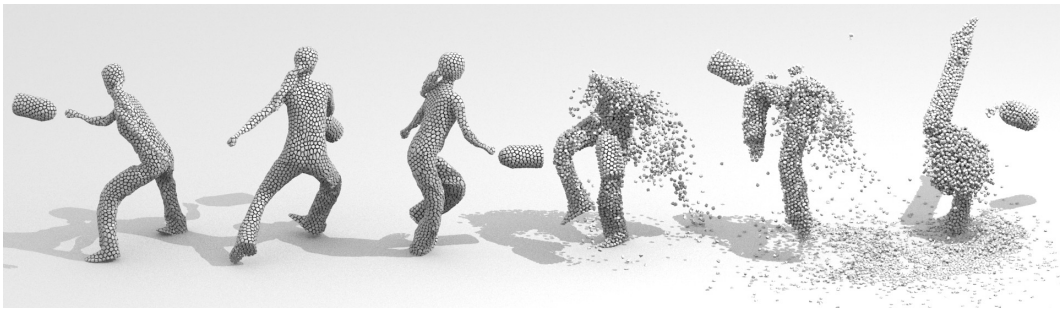


Figure 6.1 – An animation that combines video-based shape motion (left) and physical simulation (right). Our method allows to apply mechanical effects on captured dynamic shapes and generates therefore plausible animations with real dynamics.

and are now standard in the industry.

However, motion capture is usually only the first stage in a complicated process, before the final animation can be obtained. The task requires large amounts of manual labor to rig the kinematic data to a surface model, correct and customize the animation, and produce the specifics of the desired effect. This is why, in recent times, video-based 3D performance capture technologies are gaining more and more attention, as they can be used to directly produce 3D surface animations with more automation, and to circumvent many intermediate stages in this process. They also make it possible to automatically acquire complex scenes, shapes and interactions between characters that may not be possible with the standard sparse-marker capture technologies. Still, the problem of customizing the surface animations produced by such technologies to yield a modified animation or a particular effect has currently no general and widespread solution, as it is a lower level representation to begin with.

In this work, we propose a novel system towards this goal, which produces animations from a stream of 3D shapes acquired with a video-based capture system. The system provides a framework to push the automation of animation generation to a new level, dealing with the capture, shape tracking, and animation generation from end-to-end with a unified representation and solution. In particular, we entirely circumvent the need for kinematic rigging and present results in this report obtained without any manual surface correction.

Although the framework opens many effect possibilities, for the purpose of the demonstration here, we focus our effort on combining the real raw surface data captured with physics and procedural animation, in particular using a physics-based engine. To this aim, we propose to use regular Voronoi tessellations to decompose acquired shapes into volumetric cells, as a dense volume representation upon which physical constraints are easily combined with the captured motion constraints. Hence, shape motions can be perturbed with various effects in the animation, through forces or procedural decisions applied on volumetric cells. Motion constraints are obtained from captured multi-view sequences of live actions. We do not consider skeletal or surfacic motion models for that purpose but directly track volumetric cells instead. This ensures high flexibility in both the class of shapes and the class of physical constraints that can be accounted for. We have evaluated our method with various actor performances and effects. We provide qualitative results for the generated 3D content. They demonstrate that convincing and, to the best of our knowledge, unprecedented animations can be obtained using video-based captured shape models.

In summary, this work considers video-based animation and takes the field a step further by allowing for physics-based or procedural animation effects. The core innovation that permits the combination of real and simulated dynamics lies in the volumetric shape representation we propose. The associated tessellated volumetric cells can be both tracked and physically perturbed hence enabling new computer animations.

6.2 Related work

This work deals with the combination of simulated and captured shape motion data. As mentioned earlier, this has already been explored with marker based mocap data as kinematic constraints. Following the work of [134], a number of researchers have investigated such combination. They propose methods where mocap data can be used either as reference motion [134, 135, 136], or to constrain the physics-based optimization associated to the simulation with human-like motion styles [137, 138, 139, 140]. Although sharing conceptual similarities with these methods, our work differs substantially. Since video-based animations already provide natural animations, our primary objective is not to constrain a physical model with captured kinematic constraints but rather to enhance captured animations with user-specified animation constraints based on physics or procedural effects. Consequently, our simulations are not based on biomechanical models but on dynamic simulations of mechanical effects. Nevertheless, our research draws inspiration from these works.

With the aim to create new animations using recorded video-based animations, some works consider the concatenation of elementary segments of animations, *e.g.* [141], the local deformation of a given animation, *e.g.* [142], or the transfer of a deformation between captured surfaces, *e.g.* [52]. While we also aim at generating new animations, we tackle a different issue in this research with the perturbation of recorded animations according to simulated effects.

Our method builds on results obtained in video-based animations with multi-camera setups, to obtain the input data of our system. Classically, multi-view silhouettes can be used to build free viewpoint videos using visual hulls [143, 144] or to fit a synthetic body model [106]. Visual quality of reconstructed shape models can be improved by considering photometric information [115, 145] and also by using laser scanned models as templates that are deformed and tracked over temporal sequences [46, 31, 48]. Interestingly, these shape tracking strategies provide temporally coherent 3D models that carry therefore motion information. In addition to geometric and photometric information, considering shading cues allows to recover finer scale surface details as in [146, 147] .

More recent approaches have proposed to recover both shapes and motions, as reviewed in Chapter 2. They follow various directions depending on the prior information assumed for shapes and their deformations. For instance in the case of human motion, a body of work assumes articulated motions that can be represented by the poses of skeleton based models, *e.g.* [31, 91, 57]. We base our system on a different class of techniques aiming at more general scenarios, with less constrained motion models simply based on locally rigid assumptions in the shape volume, as the algorithms presented in Chapters 4 and 5. This has the advantage that a larger class of shapes and deformations can be considered, in particular motions of humans with loose clothes or props. The techniques also have the significant advantage that they allow to track dense volumetric cell decompositions of objects, thereby allowing for consistent captured motion information to be associated and propagated with each cell in the volume, a key property to build our animation generation framework on.

In the following, we provide first an overview of our system which tessellates input shapes in regular polyhedral cells, recover cells trajectories, then uses these trajectories

as input for creating physics-based and procedural animations. Then we present briefly the volumetric shape tessellation (§6.4), we detail how the physics-based model combines acquired trajectories and simulation (§6.5), and we provide examples of resulting visual effects (§6.6). To our knowledge, this is the first attempt to propose such an end-to-end system and framework to generate animations from real captured dynamic shapes.

6.3 System Overview

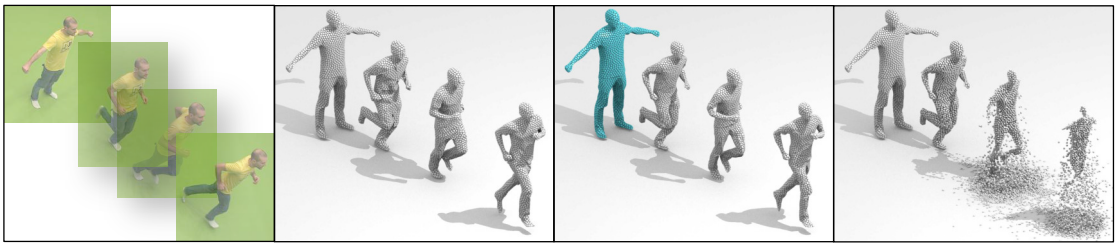


Figure 6.2 – From video-based shape capture to physic simulation. The approach uses multiple videos and Voronoi tessellations to capture the volumetric kinematic of a shape motion which can then be reanimated with additional mechanical effects, for instance volumetric erosion with gravity in the figure.

We generate a physically plausible animation given a sequence of 3D shapes as well as user specifications for the desired effect to be applied on the animation. 3D reconstructed shapes are transformed into temporally consistent volumetric models using centroidal Voronoi tessellations and shape tracking. Kinematic and physical constraints are then combined using rigid body physics simulation. The approach involves the following main steps depicted in Figure 6.2.

Video based acquisition Input to our system is a sequence of 3D shapes reconstructions of a dynamic scene. Traditional and probably most common dynamic scenes in graphics are composed of human movements; however our system can consider a larger class of shapes since only local rigidity is assumed to get temporally consistent shape models. Shape reconstructions are assumed to be obtained using a multi-camera system and represented as a surface mesh. Our own apparatus, the Kinovis platform, is composed of 68 calibrated and synchronized cameras with a resolution up to 2048×2048 pixels. The acquisition space is about $8m \times 4m$ and the camera frame rate can go up to 50 fps at full resolution. The outputs of this step are surface meshes with around $15k$ points.

Volumetric Representation. Input shapes are tessellated into polyhedral cells. This volumetric representation is motivated by two aspects of our animation goal: first, the representation is well suited to physical simulation; second, volumetric deformation models are more flexible than skeleton based models, hence enabling non rigid shape deformations. Still, they allow for locally rigid volumetric constraints, a missing feature

with surface deformation models when representing shapes that are volumes. We adopt centroidal Voronoi tessellations that produce regular and uniform polyhedral cells.

Tracking. In this step, incoherent volumetric shape models of a temporal sequence are transformed into coherent representations where a single shape model is evolving over time. This provides kinematic information at the cell level that will further be used in the simulation. We use a the volumetric tracking method of Chapter 4[1] that finds the poses of a given template shape at each frame. The template shape is taken as one of the volumetric models at a frame. The approach uses a volumetric deformation model, instead of surface or skeleton based model, to track shapes. It optimizes the pose of the template shape cells so as to minimize a distance cost to an input shape model while enforcing rigidity constraints on the local cell configurations.

Simulation. The tracked cell representation is both suitable for tracking and convenient for solid based physics. We embed the tracked volumetric model in a physical simulation, by considering each cell to be a rigid solid object in mechanical interaction with other cells and scene objects. We ensure cohesion of cells by attaching a kinematic recall force in the simulation, and offer various controls as to how the scene may deform, collide, or rupture during contacts and collisions. This simple framework allows for a number of interesting effects demonstrated in §6.6.

6.4 Volumetric Shape Modeling

In order to perturb captured moving shapes with simulated mechanical effects, we resort to volumetric discretizations. They enable combined kinematic and physical constraints to be be applied over cells using rigid body simulations. To this goal, we partition shapes into volumetric cells using Voronoi tessellations. Ideally, cells should be regular and uniform to ease the implementation of local constraints such as physical constraints for simulation or local deformation constraints when tracking shapes over time sequences. Volumetric voxel grids [148, 149], while efficient, are biased towards the grid axes and can therefore produce tessellations with poor quality. Other solutions such as Delaunay tetrahedrizations, *e.g.* [150, 151], can be considered however they can present badly shaped cells such as slivers. Moreover, they can not always guarantee a correct topology for the output mesh since the boundary of a tetrahedral structure can always present non manifold parts. In this work, we consider centroidal Voronoi tessellations (CVTs) to model shapes and their evolutions for the tracking stage as well as for the animation stage. Resulting cells in CVTs are known to be uniform, compact, regular and isotropic [1] .

6.5 Combined Animation

The first stages of the system have already been presented in Chapter 5, so we only describe the last stage, which produces the combined animation. The template representation of the subject now being consistently tracked across the sequence, we can use

[1]The volumetric tracking-by detection method of Chapter 5 could be used as well.

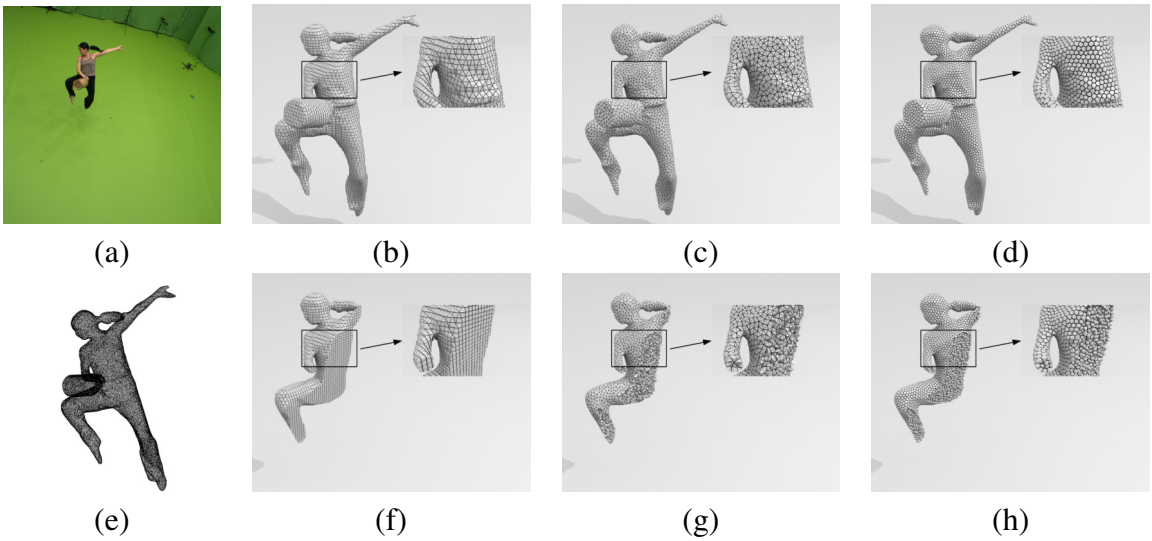


Figure 6.3 – (a, e) Input multi-camera observations and surface reconstruction (15k points). (b,f) Tessellations generated using voxels [152]. (c,g) Tetrahedrisations generated using Delaunay refinement [153] . (d,h) Clipped centroidal Voronoi tessellations (14455 sites).

the tracked cells as input for solid physics-based animation. We have purposely chosen CVTs as a common representation as they can be made suitable for shape and motion capture as we have shown, while being straightforwardly convenient for physics-based computations. In fact CVT cells are compact, convex or easily approximated by their convex hull. This is an advantage for the necessary collision detection phase of physics models, as specific and efficient algorithms exist for this case [154, 155]. We here describe the common principles of our animation model, with more specific applications being explored and reported on in the following sections.

As our animation framework is solid-based, we base our description on commonly available solid-based physics models, *e.g.* [156]. Each CVT cell is considered a homogeneous rigid body, whose *simulated* state is parametrized by its 3D position, rotation, linear momentum and angular momentum. The cell motion is determined by Newton's laws through a differential equation involving the cell state, the sum of forces and sum of torques that are applied to the cell.

The animation is thus obtained by defining the set of forces and torques applied at each instant, and iteratively solving these differential equations to obtain a new cell position and orientation for a target time step, using one of many available techniques. For our demonstrator, we use the simple and efficient off-the-shelf Bullet Physics engine [157, 158].

6.5.1 Ordinary Applied Forces

We classically apply the constant gravity force $F_g = Mg$. We apply additional external forces or constraints as needed for the target application, as will be detailed in the coming sections. Additionally, contact forces such as collisions are handled with scene

objects, as well as between different cells of the CVT. For this purpose the physics engine first needs to detect the existence of such contacts. It relies on a hierarchical space decomposition structure, such as an AABB-tree, for broad-phase collision detection, *i.e*. coarse elimination of collision possibilities. A narrow-phase collision test follows, between objects lying in the same region of space as determined by the AABB-tree traversal. In this narrow phase the full geometry of objects is examined, *e.g*. using the GJK algorithm [154] for pairs of convex polyhedra. Once the existence of a contact is established, various strategies exist to deal with the collision, *e.g*. by introducing impulse repulsion forces to produce a collision rebound. We follow the common approach of modeling contacts as a linear complementary problem (LCP) popularized by [159], which derives contact forces as the solution of a linear system that satisfies certain inequality constraints. These constraints are typically formulated using a constraint Jacobian over the combined state spaces of rigid bodies. [158] expose the specific variant applied in the context of the Bullet Physics engine.

6.5.2 Physical Modeling of Kinematic Control

To relate the physical simulation to the acquired non-rigid poses of the model, we need to introduce coupling constraints. Our goal is to allow the model to materialize and control the tradeoff between the purely kinematic behavior acquired from visual inputs for the cell, and the purely mechanically induced behavior in the simulation. First it is important to note that the temporal discretization used for acquisition and for simulation and rendering of the effects are generally different. Consequently the first stage in achieving our goal is to compute a re-sampling of the pose sequence, to the target simulation and rendering frequency, using position and quaternion interpolation. The poses so obtained are here referred as the acquired cell center location $\hat{x}^a(t)$ and cell rotation $R^a(t)$. Second, we formulate the coupling by introducing a new kinematic recall force, in the form of a damped spring between the acquired cell poses and the simulated cell poses:

$$F_r(t) = k.(\hat{x}^a(t) - \hat{x}(t)) - \lambda.\frac{\mathrm{d}}{\mathrm{d}t}(\hat{x}^a(t) - \hat{x}(t)), \tag{6.1}$$

where k and λ are respectively the rigidity and damping coefficients of the spring, which control the strength and numerical stability of the coupling.

6.6 Visual Effects

This section presents various animation results whose potential is best illustrated in the supplemental video[2]. These animations are based on three captured sequences of variable nature and speed. RUNNER shows a male character running in a straight line, during 3 motion cycles. This animation lasts 2.5 s. In BIRDCAGEDANCE a female dancer moves while holding a bird cage. This sequence is 56 s long and shows a complex sequence of motions which would be difficult to synthesize without sensors. Finally, in SLACKLINE a male acrobat evolves on a non rigid line above the ground, for 25 s.

[2] https://hal.inria.fr/hal-01255337

Figure 6.4 – Time persistence on the RUNNER sequence: a slower copy of the shape that erodes over time is generated at regular intervals.

The input sequences of temporally inconsistent 3D meshes are made respectively of 126 (RUNNER), 2800 (BIRDCAGEDANCE) and 1240 (SLACKLINE) temporal frames.

Parameters Centroidal Voronoi tessellations were computed using 5000 cells for each shape by applying the algorithm of Wang *et al.* [130] with ten iterations, except for the template shape where 50 iterations were applied in order to get a higher cell regularity in the resulting animations. We use a temporal window of 10 frames for the tracking, and cluster the 5000 cells into 200 patches. Sixty iterations of the EM-ICP algorithm were applied.

We list below examples of visual effects that we were able to generate using the proposed approach. First, we present animations combining tracking results with solid dynamics simulation (§6.6.1). Then we present other visual effects that also exploit the volumetric tracking information (§6.6.2 and §6.6.3).

6.6.1 Asynchronous Kinematic Control Deactivation

In order to show the effect of gravity while keeping the dynamic motion of the input sequence, we deactivate kinematic control forces independently for each cell. After being deactivated, a cell usually falls to the ground, since it follows a trajectory determined only by gravity, collision forces and its initial velocity. Asynchronous cell deactivations result in an animation combining cells that follow their tracking trajectory and cells that fall. By choosing diverse strategies for scheduling cell deactivations, a wide variety of animations can be obtained.

We explore here three possible deactivation strategies that are based on different criteria. Note that while these effects are straight-forward to produce with our volumetric framework, it would be difficult to obtain them if only surface or skeleton-based tracking was available.

Rupture under Collisions

Collisions with obstacles sometimes deviate a cell from its theoretical trajectory, which results in an increase of the recall force magnitude (see Eq. 6.1). This phenomenon can be detected and used for simulating the rupture of the material: when the recall force magnitude of a cell is above a given threshold, we deactivate the recall force (for this cell only). This makes the rupture looks like the consequence of the collision.

Figure 6.6 shows a heavy pendulum that hits the subject and makes a hole in it. Under the intensity of the collision, several cells are ejected (rupture) and fall to the ground.

Heat Diffusion

In order to make the cell deactivation both temporally and spatially progressive, we rely on a diffusion algorithm. We diffuse an initial temperature distribution inside the volume according to the diffusion equation. Deactivation is triggered when the cell temperature is above a given threshold.

Heat diffusion in a CVT The CVT provides a graph structure on centroids, which is a subgraph of the Delaunay tetrahedralization of the cells centroids. The heat diffusion on a graph structure is expressed by the heat equation:

$$(\partial/\partial t + \mathbf{L})\mathbf{F}_t = 0$$

where $\mathbf{F}_t$ is the column vector of centroids temperatures at time t, and $\mathbf{L}$ is the combinatorial graph Laplacian matrix (note that a geometric Laplacian is not necessary since centroids are regularly distributed in space). Given an initial temperature distribution $\mathbf{F}_0$, this equation has a unique solution

$$\mathbf{F}_t = \mathbf{H}_t\mathbf{F}_0$$

where $\mathbf{H}_t = e^{t\mathbf{L}}$ is the heat diffusion kernel, which can be computed by means of the spectral decomposition of $\mathbf{L}$.

We compute the temperature evolution on the BIRDCAGEDANCE sequence for an initial temperature distribution where all cell temperatures are zero, except for the dancer's head top cells (which are set to 1). We observe in Figure 6.1 that cells fall progressively across time, from the upper to the lower body parts. Note that the cage cells remain kinematically controlled since heat is not transferred between different connected components.

Morphological Erosion

The discrete cell decomposition of shapes allows to apply morphological operators. To illustrate this principle, we have experimented erosion as shown in Figure 6.2 . In this example, cells are progressively eroded starting from the outside. The morphological erosion is performed by deactivating each cell after a delay proportional to the distance between the cell centroid and the subject's surface. The distance is computed only once for each cell, on the template shape. Figure 6.2 shows the erosion animation on the RUNNER sequence. Note that the operation progressively reveals the dynamic of the inner part of the shape.

6.6.2 Time Persistence

In this example, we experiment time effects over cell decompositions. To this aim, dynamic copies of the model are generated at regular time intervals. These copies are equipped with deceleration and erosion effects over time and create therefore ghost avatars that vanish with time (see Figure 6.4 and the accompanying video). The benefit of the tracked volumetric representation in this simulation is the ability to attach time effect to the model behavior at the cell level, for instance lifetime and deceleration in the example.

6.6.3 Morphing

Our dynamic representations allows to apply volumetric morphing between evolving shapes, enabling therefore new visual effects with real dynamic scenes. To this purpose, cells of the source shape are first matched to the target shape. Second, each cell is individually morphed to its target cell at a given time within the sequence. Time ordering is chosen such that cells in the source shape are ordered from the outside to the inside, and associated with the cells of the target shape ordered from the inside to the outside. Cells are transformed from the source to the destination by interpolating their positions and using [160] to morph their polyhedral shapes. Figure 6.5 shows the dynamic morphing of the RUNNER sequence onto the BIRDCAGEDANCE sequence.

6.6.4 Run Time Performance

Our approach has been tested on a dual Intel Xeon E5-2665 processor with 2.40 GHz each. For each animation, the Poisson runs in 0.67 s per frame on average, and the volumetric decomposition for each frame runs in 6.27 s, except for the template model for which it runs in 25.52 s since more iterations are applied. Our tracking algorithm runs in 45 s per frame on average. The physical simulation usually needs about 350 ms per simulation step on a single thread. The morphing runs on multiple threads in about 2.35 s.

Figure 6.5 – Tracking result of the RUNNER and the BIRDCAGEDANCE sequences (middle) and combination with volumetric morphing with 5000 cells (bottom).

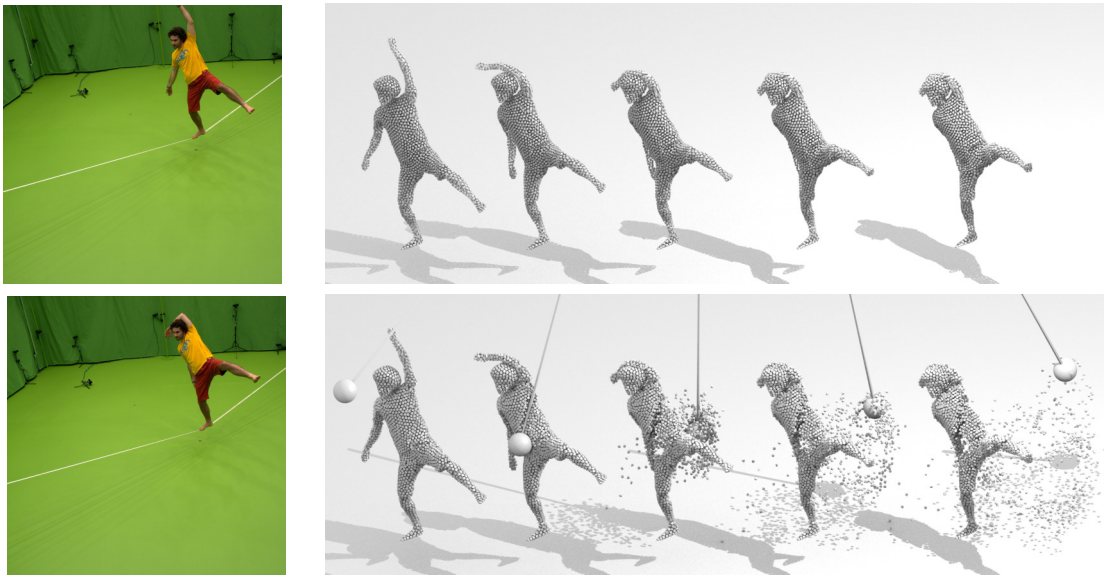


Figure 6.6 – Input SLACKLINE Multi-camera observations (left), tracking result of the SLACKLINE sequence (top) and combination with the effect of collision with a pendulum (bottom).

6.7 Limitations

As shown in the previous examples, our approach generates plausible results for a variety of captured and simulated motions. However, a few limitations must be noted. First, the true captured shape must be volumetric in nature, since we tessellate its interior into 3D cells. Thin shapes such as clothes may cause some cells to be flat, leading to volume variation among cells and ill-defined cohesion constraints that would cause difficulty to the tracking model.

Regarding the physical simulation, our current demonstrator is limited to rigid body interactions, but could be extended to other physical models such as soft body physics and fluid simulation. Since a CVT provides neighboring information, soft body simulation could be achieved by introducing soft constraints between neighboring cells. This would lead to animations where cell sets behave more like a whole rather than independent bodies.

6.8 Conclusions

This chapter presents how volumetric shape tracking can be useful to the creation of shape animations. Precisely, our framework allows video-based animations to be combined with physical simulation to create unprecedented and plausible animations. The interest is to take benefit of both modalities and to bring complementary properties to computer animations. Our approach relies on volumetric tessellations within which kinematic constraints, driven by the captured volumetric motion, can be associated to mechanical forces or procedural tasks at a cell level. Because of its simple and unified nature, relying on centroidal Voronoi tessellations, the system opens various new opportunities to reconsider how animations can be generated and automated from raw captured 3D data. Moreover, it paves the way for other effects such as soft-body and fluid mechanics. In particular, the captured subject's dynamics could be imitated by conditioning soft-body mechanics by the local rigidity properties learned during tracking.

Chapter 7

Conclusion and Perspectives

7.1 Conclusion

Unified framework We want to emphasize that all our contributions form a unified volumetric framework built around the centroidal Voronoi tessellation, which makes our global contribution consistent. The deformation model of Chapter 3 which learns intrinsic deformation properties has been extended to volumetric shapes in Chapter 4 and used in the Chapters 5 and 6. The two proposed volumetric association models (in Chapters 4 and 5) can be used alternatively. Finally, CVT tracking results can be used directly for generating cell-based volumetric animations.

Volumetric representations for shape tracking Our main contribution is to propose and evaluate a new framework for volumetric shape tracking from visual data. In particular, this framework takes benefit from the volumetric nature of the problem. We compare our approach to surface-based equivalents and analyse the differences. The main conclusion of our work is that when the observed shape has a volumetric nature, volumetric representations can be leveraged for all stages of the tracking problem (the deformation model, the association model), as well as for generating volumetric animations combining simulated and acquired physics. More precisely, volumetric shape representations are advantageous for enforcing local volume rigidity constraints, for establishing shape correspondences with proximity-based or detection-based methods, and for combining captured and simulated mechanics. To the deformable shape tracking problem, the main benefit is a robustness increase.

Temporal evolution and redundancy Should we treat time samples independently, or should we leverage temporal information? Intuitively, transferring information between frames should provides higher quality results, but it comes with the risk of transferring and propagating errors. This intuition has been confirmed by our experiments. However, we've shown that these two strategies are not incompatible, as they can be used for distinct part of a tracking algorithm. On one hand, we have proposed a framework for transferring information at the deformation model level in a rather safe way thanks to the accumulation of information over a temporal window and a prior that limits the deviation from the template pose. On the other hand, we have introduced an

association model which is fully independent for each time sample. These two models are combined in Chapter 5 and lead to a robust shape tracking algorithm. The conclusion with respect to this question is that transferring information between time samples is advantageous for the quality of the results, but it requires a mechanism for limiting error accumulation.

7.2 Perspectives

In light of our work, we propose several perspectives for future works, ranked from shortest to longest term.

Real-time tracking While some applications of 4D modeling require a high quality independently of processing time, other applications require real-time processing. Examples are telepresence, live streaming and interactive augmented or virtual reality applications [161, 162]. The goals of this thesis is to improve the quality of the results and we did not focus on reaching low processing times. Our algorithm implementation runs three order of magnitude slower than real time, but it could be improved by optimizing the implementation or changing parts of the algorithms. We present here several directions for decreasing processing times.

The bottlenecks of our implementation are the shape tessellation, the correspondence estimation (ICP) and the construction of the linear system at each Gauss-Newton iteration. Recent advances have decreased the CVT computational time by considering point clouds [2], but are not sufficient for real-time processing. Real-time performance can be reached with other volume discretizations such as cubic volumetric grids, but it may decrease the quality of the tracking results.

Computing the EM-ICP correspondences is a highly parallel task that can be accelerated with a GPU. Dropping the probabilistic association model could also decrease processing times. The approaches that track a shape from depth images have a very fast data-term which integrates in the image domain the depth difference between the captured depth and the depth obtained when rendering the deformed template [27]. This association can be computed very fast with GPU rendering. This association model could be adapted to mesh tracking approaches, by simulating several depth images of the input mesh.

The linear system construction could be accelerated by several ways. The algorithm loops over correspondence residuals for accumulating partial derivatives. This computation could be parallelized (on GPU). Using deterministic correspondences would highly decrease the number of residuals, but it would also impact the robustness of the tracking algorithm.

After optimizing those steps, resolving the linear system of Gauss-Newton iterations may become a bottleneck. If the size of the sparse linear system is large, which happens when using a large number of patches, replacing the sparse Cholesky factorization by the conjugate gradient algorithm as an iterative method improves performance [89]. Replacing the Gauss-Newton algorithm by the Levenberg-Marquardt method may also accelerate the convergence [83, 163, 89], as the latter provides a hybrid solution between

the Gauss-Newton method and a gradient descent.

Non-uniform volumetric representations The regularity and homogeneity of the CVT is a great advantage for shape tracking, since it releases the tracking algorithm to take into account meshing irregularities. However, there are situations where shape tracking may benefit from heterogeneous shape decompositions. If we could locally adapt the cell radius to the level of non-rigidity or the level of shape detail, we could improve the precision of the shape tracking without increasing the number of cells. For example, fingers requires small cells for being precisely represented, while inner torso cells may be larger without impacting shape and deformation detail. Such a strategy has already proven useful for surface mesh-based tracking of human motion: Collet *et al*. [14] uses smaller mesh primitives for the face than for the rest of the body, as human perception is more sensitive to detail on this area. It provides a higher plausibility of the resulting animation while it keeps the tracking approach computationally reasonable. Therefore, there is an interest in the research of CVT-like decompositions with locally controllable cell size.

Physics-based tracking In Chapter 6, we have used mechanical simulation for creating visual effects from captured shapes in motion. Reciprocally, we could leverage physics for solving the tracking problem. Mechanical simulation provides a model for temporal shape evolution which could be used for motion smoothing or prediction. This may results in tracking results that are physically more plausible. An even more ambitious goal is to infer physical constraints from motion. For these applications, volumetric shape representations seem more relevant than surface-based ones since they represents better the mechanical nature of volumetric shapes.

Topology and template inference Not only template-based tracking makes the assumption of topology-preserving motion, but it also requires a template with correct topology. This can be a strong limitation, since such a template can be complicated or impossible to obtain directly. Several works go beyond template tracking and propose to discover the shape geometry and topology iteratively, by integrating the shape and topology information over several temporal observations. The interest is not only to recover the correct topology, but also to recover shape detail by filtering the observation noise out. Those methods usually estimate a deformation between the current template and the observation at time t, then use this deformation to fuse the current observation and template in a new template. This idea has been implemented with surface mesh representations [164] or with implicit surfaces defined by a sampled signed distance field [165]. While those early works are limited to simulated or simple data, recent works [27, 89, 166] runs in real time and are able to track and fuse shapes undergoing large and fast deformable motions as well as topology changes, such as a man who unzip and takes off his jacket. While most of these works target depth or RGBD data, for which the TSDF is an efficient fusion tool, only little work propose to solve the same problem with multi-view video data. Given the advantages of passive acquisition systems, this is a promising research direction.

Joint model of shape, motion and appearance One of the great challenges of 4D modeling is to combine several signals in the same model. Especially, reconstructing shape, appearance and motion together in a joint optimization of a consistent model is particularly interesting: since those signals are inter-dependent, ambiguities of one signal could be compensated by leveraging the other signals.

Diversity of capture situations Practical 4D shape capture being highly constrained, there is a large interest in making 4D modeling applicable to a wider diversity of scenes and environments. Increasing the diversity of capture cases includes the generalisation of shape tracking to other types of motions. Most works in shape tracking in computer vision assume the shape deformation is locally consistent and the elasticity is relatively small. However, there exists shape motions that do not fit these deformation models. For example, the growth of a plant over long periods of times is locally highly non-rigid, and it poses difficult topology issues. Tracking such a motion requires to elaborate new deformation and topology models.

Publications

This thesis has led to several publications summarized below.

Peer-reviewed international conferences

- B. Allain, J.-S. Franco, E. Boyer and T. Tung. On Mean Pose and Variability of 3D Deformable Models. In *Proceedings of the 13th European Conference on Computer Vision*, Springer, 2014

  HAL page: https://hal.inria.fr/hal-01016981

- B. Allain, J.-S. Franco and E. Boyer. An Efficient Volumetric Framework for Shape Tracking. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2015

  HAL page: https://hal.inria.fr/hal-01141207

- C.-H. Huang, B. Allain, J.-S. Franco, N. Navab, S. Ilic and E. Boyer. Volumetric 3D Tracking by Detection. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2016

  HAL page: https://hal.inria.fr/hal-01300191

Technical report

- B. Allain, L. Wang, J.-S. Franco, F. Hetroy-Wheeler and E. Boyer. Shape Animation with Combined Captured and Simulated Dynamics. *arXiv e-prints*, 2016

  arXiv page: http://arxiv.org/abs/1601.01232

  HAL page: https://hal.inria.fr/hal-01255337

Appendix A

This appendix provides computational details of Chapter 3.

Factorization of rigidity variable set posterior

The sets C and $\mathbf{Y}$ are independent given Θ thanks to the D-separatibilty property [104]. Therefore,

$$
\begin{aligned}
p(\mathrm{C}|\mathbf{Y}\Theta) &= p(\mathrm{C}|\Theta) && (7.1)\\
&= \frac{p(\Theta|\mathrm{C})p(\mathrm{C})}{\sum_{\mathrm{C}} p(\Theta|\mathrm{C})p(\mathrm{C})} && (7.2)\\
&= \frac{\prod_{t\in\mathcal{T}} p(\mathbf{T}^t|\bar{\mathbf{T}}\mathrm{C})p(\bar{\mathbf{T}})p(\mathrm{C})}{\sum_{\mathrm{C}} \prod_{t\in\mathcal{T}} p(\mathbf{T}^t|\bar{\mathbf{T}}\mathrm{C})p(\bar{\mathbf{T}})p(\mathrm{C})} && (7.3)\\
&= \frac{\prod_{(k,l)\in\mathcal{E}} \tilde{\lambda}_{kl}(\mathrm{c}_{kl},\Theta)}{\sum_{\mathrm{C}} \prod_{(k,l)\in\mathcal{E}} \tilde{\lambda}_{kl}(\mathrm{c}_{kl},\Theta)} && (7.4)\\
\text{where } \tilde{\lambda}_{kl}(\mathrm{c}_{kl},\Theta) &= p(\mathrm{c}_{kl}) \exp(-\mathcal{D}_{kl}(\bar{\mathbf{T}}_{k-l}, \mathbf{Id})) \prod_{t\in\mathcal{T}} \exp(-\mathcal{D}_{kl}(\bar{\mathbf{T}}_{k-l}, \mathbf{T}^t_{k-l}, \mathrm{c}_{kl})). && (7.5)
\end{aligned}
$$

The denominator of (7.4) also factorizes as

$$
\sum_{\mathrm{C}} \prod_{(k,l)\in\mathcal{E}} \tilde{\lambda}_{kl}(\mathrm{c}_{kl},\Theta) = \prod_{(k,l)\in\mathcal{E}} \sum_{\mathrm{c}_{kl}} \tilde{\lambda}_{kl}(\mathrm{c}_{kl},\Theta), \tag{7.6}
$$

so the full posterior expression factorizes as

$$
p(\mathrm{C}|\mathbf{Y}\Theta) = \prod_{(k,l)\in\mathcal{E}} \lambda_{kl}(\mathrm{c}_{kl},\Theta) \tag{7.7}
$$

with the help of the normalized coefficients $\lambda_{kl}(\mathrm{c}_{kl},\Theta) = \frac{\tilde{\lambda}_{kl}(\mathrm{c}_{kl},\Theta)}{\sum_{\mathrm{c}_{kl}} \tilde{\lambda}_{kl}(\mathrm{c}_{kl},\Theta)}$.

Simplification of the EM auxiliary function

The MAP-EM auxiliary function Q is defined by

$$Q(\Theta|\Theta^m) = \sum_Z p(Z|\mathbf{Y},\Theta^m)\ln p(\mathbf{Y},Z|\Theta) + \ln p(\Theta) \tag{7.8}$$

$$= \sum_Z p(Z|\mathbf{Y},\Theta^m)\ln p(\mathbf{Y},Z,\Theta). \tag{7.9}$$

Given the posteriors and joint probability

$$p(Z|\mathbf{Y},\Theta^m) = \prod_{t\in\mathcal{T}}\prod_{o\in\mathcal{O}_t}\gamma_{ot}(\mathrm{k}_o^t,\Theta^m)\prod_{(k,l)\in\mathcal{E}}\lambda_{kl}(\mathrm{c}_{kl},\Theta^m) \tag{7.10}$$

$$\text{and } p(\mathbf{Y},Z,\Theta) = \prod_{t\in\mathcal{T}}\prod_{o\in\mathcal{O}_t}\tilde{\gamma}_{ot}(\mathrm{k}_o^t,\Theta)\prod_{(k,l)\in\mathcal{E}}\tilde{\lambda}_{kl}(\mathrm{c}_{kl},\Theta), \tag{7.11}$$

the auxiliary function (7.9) expands as

$$Q(\Theta|\Theta^m) = \sum_{\mathrm{K}}\sum_{\mathrm{C}}\left(\prod_{t\in\mathcal{T}}\prod_{o\in\mathcal{O}_t}\gamma_{ot}(\mathrm{k}_o^t,\Theta^m)\prod_{(k,l)\in\mathcal{E}}\lambda_{kl}(\mathrm{c}_{kl},\Theta^m)\right)\sum_{t\in\mathcal{T}}\sum_{o\in\mathcal{O}_t}\ln\tilde{\gamma}_{ot}(\mathrm{k}_o^t,\Theta) \tag{7.12}$$

$$+\sum_{\mathrm{K}}\sum_{\mathrm{C}}\left(\prod_{t\in\mathcal{T}}\prod_{o\in\mathcal{O}_t}\gamma_{ot}(\mathrm{k}_o^t,\Theta^m)\prod_{(k,l)\in\mathcal{E}}\lambda_{kl}(\mathrm{c}_{kl},\Theta^m)\right)\sum_{(k,l)\in\mathcal{E}}\ln\tilde{\lambda}_{kl}(\mathrm{c}_{kl},\Theta) \tag{7.13}$$

$$= \sum_{t\in\mathcal{T}}\sum_{o\in\mathcal{O}_t}Q_{ot}(\Theta|\Theta^m) \tag{7.14}$$

$$+\sum_{(k,l)\in\mathcal{E}}Q_{kl}(\Theta|\Theta^m). \tag{7.15}$$

For each edge $(k,l)\in\mathcal{E}$, the elementary function $Q_{kl}(\Theta|\Theta^m)$ factorizes as

$$Q_{kl}(\Theta|\Theta^m) = \sum_{\mathrm{K}}\sum_{\mathrm{c}_{kl}}\sum_{\mathrm{C}\setminus\{\mathrm{c}_{kl}\}}\left[\prod_{t\in\mathcal{T}}\prod_{o\in\mathcal{O}_t}\gamma_{ot}(\mathrm{k}_o^t,\Theta^m)\prod_{(k',l')\in\mathcal{E}}\lambda_{kl}(\mathrm{c}_{k'l'},\Theta^m)\right]\ln\lambda_{kl}(\mathrm{c}_{kl},\Theta^m) \tag{7.16}$$

$$= \left[\sum_{\mathrm{K}}\prod_{t\in\mathcal{T}}\prod_{o\in\mathcal{O}_t}\gamma_{ot}(\mathrm{k}_o^t,\Theta^m)\right]\left[\sum_{\mathrm{C}\setminus\{\mathrm{c}_{kl}\}}\prod_{(k',l')\in\mathcal{E}\setminus\{(k,l)\}}\lambda_{k'l'}(\mathrm{c}_{k'l'},\Theta^m)\right]\sum_{\mathrm{c}_{kl}}\lambda_{kl}(\mathrm{c}_{kl},\Theta^m)\ln\tilde{\lambda} \tag{7.17}$$

where all but one factors simplify to the unit:

$$\sum_{\mathrm{K}} \prod_{t\in\mathcal{T}} \prod_{o\in\mathcal{O}_t} \gamma_{ot}(\mathrm{k}_o^t, \Theta^m) = \prod_{t\in\mathcal{T}} \prod_{o\in\mathcal{O}_t} \sum_{\mathrm{k}_o^t} \gamma_{ot}(\mathrm{k}_o^t, \Theta^m) = \prod_{t\in\mathcal{T}} \prod_{o\in\mathcal{O}_t} 1 = 1 \tag{7.18}$$

$$\text{and} \sum_{\mathrm{C}\setminus\{\mathrm{c}_{kl}\}} \prod_{(k',l')\in\mathcal{E}\setminus\{(k,l)\}} \lambda_{k'l'}(\mathrm{c}_{k'l'}, \Theta^m) = \prod_{(k',l')\in\mathcal{E}\setminus\{(k,l)\}} \sum_{\mathrm{c}_{k'l'}} \lambda_{k'l'}(\mathrm{c}_{k'l'}, \Theta^m) \tag{7.19}$$

$$= \prod_{(k',l')\in\mathcal{E}\setminus\{(k,l)\}} 1 \tag{7.20}$$

$$= 1, \tag{7.21}$$

leading to the simple expression

$$Q_{kl}(\Theta|\Theta^m) = \sum_{\mathrm{c}_{kl}} \lambda_{kl}(\mathrm{c}_{kl}, \Theta^m) \ln \tilde{\lambda}_{kl}(\mathrm{c}_{kl}, \Theta). \tag{7.22}$$

The same manipulation can be applied to $Q_{ot}(\Theta|\Theta^m)$, which leads to

$$Q_{ot}(\Theta|\Theta^m) = \sum_{\mathrm{k}_o^t} \gamma_{ot}(\mathrm{k}_o^t, \Theta^m) \ln \tilde{\gamma}_{ot}(\mathrm{k}_o^t, \Theta). \tag{7.23}$$

Finally, the auxiliary function simplifies to

$$Q(\Theta|\Theta^m) = \sum_{t\in\mathcal{T}} \sum_{o\in\mathcal{O}_t} \sum_{\mathrm{k}_o^t} \gamma_{ot}(\mathrm{k}_o^t, \Theta^m) \ln \tilde{\gamma}_{ot}(\mathrm{k}_o^t, \Theta) \tag{7.24}$$

$$+ \sum_{(k,l)\in\mathcal{E}} \sum_{\mathrm{c}_{kl}} \lambda_{kl}(\mathrm{c}_{kl}, \Theta^m) \ln \tilde{\lambda}_{kl}(\mathrm{c}_{kl}, \Theta). \tag{7.25}$$

References

[1] Q. Du, V. Faber, and M. Gunzburger, "Centroidal voronoi tessellations: Applications and algorithms," *SIAM review*, vol. 41, pp. 637–676, 1999. 14, 15, 25, 53, 55, 56, 91

[2] L. Wang, F. Hétroy-Wheeler, and E. Boyer, "On volumetric shape reconstruction from implicit forms," in *European Conference on Computer Vision*, 2016. 15, 100

[3] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006. 15, 22

[4] C. Cagniart, E. Boyer, and S. Ilic, "Probabilistic deformable surface tracking from multiple videos," in *European Conference on Computer Vision*, 2010. 15, 32, 34, 47, 48, 49, 61, 62, 64, 68, 77, 78

[5] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992. 15, 30, 31, 32, 69

[6] S. Granger and X. Pennec, "Multi-scale EM-ICP: A fast and robust approach for surface registration," in *European Conference on Computer Vision*, vol. 4, pp. 69–73, 2002. 15, 32, 42, 45

[7] R. Horaud, F. Forbes, M. Yguel, G. Dewaele, and J. Zhang, "Rigid and articulated point registration with expectation conditional maximization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 587–602, March 2011. 15, 32

[8] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon, "The Vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012. 16, 34, 70, 71, 72, 73

[9] J. P. Lewis, M. Cordner, and N. Fong, "Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM, 2000. 16, 26

[10] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel, "Laplacian surface editing," in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '04, pp. 175–184, 2004. 16, 27, 54

[11] M. Botsch, M. Pauly, M. Wicke, and M. Gross, "Adaptive space deformations based on rigid cells.," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 339–347, 2007. 16, 24, 27, 39, 40, 53, 57

[12] C. Cagniart, E. Boyer, and S. Ilic, "Free-from mesh tracking: a patch-based approach," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 16, 27, 31, 36, 38, 39, 40, 46, 54, 57

[13] E.-J. Marey, *Le Mouvement*. G. Masson (Paris), 1894. 19

[14] A. Collet, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, and S. Sullivan, "High-quality streamable free-viewpoint video," *ACM Transactions on Graphics*, vol. 34, 2015. 20, 101

[15] J. Pansiot and E. Boyer, "3d imaging from video and planar radiography," in *MICCAI 2016 - 19th International Conference on Medical Image Computing and Computer Assisted Intervention*, LNCS, Springer, 2016. 20

[16] Y. Liu, C. Stoll, J. Gall, H.-P. Seidel, and C. Theobalt, "Markerless motion capture of interacting characters using multi-view image segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 20

[17] A. Mustafa, H. Kim, J.-Y. Guillemaut, and A. Hilton, "General dynamic scene reconstruction from multiple view video," in *International Conference on Computer Vision*, 2015. 20

[18] B. G. Baumgart, "A polyhedron representation for computer vision," in *AFIPS National Computer Conference*, 1975. 21

[19] A. Laurentini, "The visual hull concept for silhouette-based image understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 150–162, Feb 1994. 21

[20] G. Cheung K.M., T. Kanade, J.-Y. Bouguet, and M. Holler, "A real time system for robust 3d voxel reconstruction of human motions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2000. 21

[21] J.-S. Franco and E. Boyer, "Efficient polyhedral modeling from silhouettes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, Mar. 2009. 21, 78

[22] M. Douze, J.-S. Franco, and B. Raffin, "QuickCSG: Arbitrary and faster boolean combinations of N solids," Research Report RR-8687, Inria - Research Centre Grenoble – Rhône-Alpes ; INRIA, Mar. 2015. 21

[23] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Eurographics Symposium on Geometry Processing*, 2006. 22

[24] Y. Furukawa and C. Hernández, "Multi-view stereo: A tutorial," *Foundations and Trends in Computer Graphics and Vision*, vol. 9, no. 1-2, pp. 1–148, 2015. 22

[25] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pp. 303–312, ACM, 1996. 24

[26] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *ISMAR*, IEEE, 2011. 24, 71

[27] R. A. Newcombe, D. Fox, and S. M. Seitz, "DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 24, 33, 71, 100, 101

[28] K. Zhou, J. Huang, J. Snyder, X. Liu, H. Bao, B. Guo, and H.-Y. Shum, "Large mesh deformation using the volumetric graph laplacian," *ACM Trans. Graph.*, vol. 24, pp. 496–503, July 2005. 24, 27, 53, 71

[29] E. de Aguiar, C. Theobalt, S. Thrun, and H.-P. Seidel, "Automatic conversion of mesh animations into skeleton-based animations," *Eurographics, Computer Graphics Forum*, vol. 27, no. 2, 2008. 24, 26

[30] I. Baran and J. Popović, "Automatic rigging and animation of 3D characters," *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 72:1–72:8, 2007. 26, 36, 76

[31] D. Vlasic, I. Baran, W. Matusik, and J. Popovic, "Articulated mesh animation from multi-view silhouettes," *ACM Transactions on Graphics*, vol. 27, no. 3, 2008. 26, 28, 36, 38, 54, 71, 89

[32] Y. Liu, J. Gall, C. Stoll, Q. Dai, H.-P. Seidel, and C. Theobalt, "Markerless motion capture of multiple characters using multi-view image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013. 26, 36, 46, 47, 48, 54, 60, 61, 62, 64

[33] T. Ju, S. Schaefer, and J. Warren, "Mean value coordinates for closed triangular meshes," *ACM Trans. Graph.*, vol. 24, pp. 561–566, July 2005. 26

[34] P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki, "Harmonic coordinates for character articulation," in *ACM SIGGRAPH 2007 Papers*, 2007. 26, 54

[35] Y. Lipman, D. Levin, and D. Cohen-Or, "Green coordinates," *ACM Trans. Graph.*, vol. 27, pp. 78:1–78:10, Aug. 2008. 26

[36] Y. Savoye and J.-S. Franco, "Cage-based tracking for performance animation," in *ACCV*, 2010. 26, 54

[37] E. Duveau, S. Courtemanche, L. Reveret, and E. Boyer, "Cage-based motion recovery using manifold learning," in *3DIMPVT 2012 - Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, IEEE, Oct. 2012. 26, 28

[38] E. Duveau, *Measure of 3D surface for osteo-muscular characterization of small vertebrates: application to the characterization of ageing in mice*. Theses, Université de Grenoble, Dec. 2012. 26

[39] J.-M. Thiery, J. Tierny, and T. Boubekeur, "Cager: Cage-based reverse engineering of animated 3d shapes," *Computer Graphics Forum*, vol. 31, no. 8, pp. 2303–2316, 2012. 26, 54

[40] M. Salzmann, F. Moreno-Noguer, V. Lepetit, and P. Fua, "Closed-form solution to non-rigid 3d surface registration," in *European Conference on Computer Vision*, 2008. 27, 54

[41] M. Perriollat, R. Hartley, and A. Bartoli, "Monocular template-based reconstruction of inextensible surfaces," *International Journal of Computer Vision*, vol. 95, 2011. 27

[42] S. Vicente and L. Agapito, "Balloon shapes: Reconstructing and deforming objects with volume from images," in *3DV*, 2013. 27

[43] A. Malti, R. Hartley, A. Bartoli, and J.-H. Kim, "Monocular template-based 3d reconstruction of extensible surfaces with local linear elasticity," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 27

[44] M. Alexa, "Differential coordinates for local mesh morphing and deformation," *The Visual Computer*, vol. 19, no. 2, pp. 105–114, 2003. 27

[45] O. Sorkine and M. Alexa, "As-rigid-as-possible surface modeling," in *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pp. 109–116, Eurographics Association, 2007. 27

[46] E. de Aguiar, C. Theobalt, C. Stoll, and H.-P. Seidel, "Marker-less deformable mesh tracking for human shape and motion capture," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 27, 54, 89

[47] C. Cagniart, *Motion Capture of Deformable Surfaces in Multi-View Studios*. Theses, Université de Grenoble, July 2012. 27

[48] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun, "Performance capture from sparse multi-view video," *ACM Transactions on Graphics*, vol. 27, no. 3, 2008. 27, 37, 53, 54, 55, 71, 89

[49] C. Schüller, L. Kavan, D. Panozzo, and O. Sorkine-Hornung, "Locally injective mappings," *Computer Graphics Forum (proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing)*, vol. 32, no. 5, pp. 125–135, 2013. 27

[50] C. Budd and A. Hilton, "Temporal alignment of 3d video sequences using shape and appearance," in *Conference on Visual Media Production*, pp. 114–122, 2010. 27, 54, 55

[51] J. Franco and E. Boyer, "Learning temporally consistent rigidities," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 27, 32, 36

[52] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," *ACM Transactions on Graphics*, vol. 23, no. 3, 2004. 28, 37, 89

[53] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "Scape: Shape completion and animation of people," *ACM Transactions on Graphics*, vol. 24, no. 3, 2005. 28, 29, 37

[54] N. Hasler, H. Ackermann, B. Rosenhahn, T. Thormählen, and H.-P. Seidel, "Multilinear pose and body shape estimation of dressed subjects from image sets," in *IEEE Conference on Computer Vision and Pattern Recognition*, (San Francisco, USA), IEEE, 2010. 28, 37

[55] C.-H. Huang, E. Boyer, N. Navab, and S. Ilic, "Human shape and pose tracking using keyframes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 28, 29

[56] C. Budd and A. Hilton, "Skeleton driven Laplacian volumetric deformation," in *CVMP*, 2009. 28, 71

[57] M. Straka, S. Hauswiesner, M. Ruether, and H. Bischof, "Simultaneous shape and pose adaption of articulated models using linear optimization," in *European Conference on Computer Vision*, 2012. 28, 89

[58] C.-H. Huang, E. Boyer, and S. Ilic, "Robust human body shape and pose tracking," in *3DV*, 2013. 28

[59] G. Pons-Moll, J. Romero, N. Mahmood, and M. J. Black, "Dyna: A model of dynamic human shape in motion," *ACM Trans. Graph.*, vol. 34, pp. 120:1–120:14, July 2015. 28, 29

[60] C. Budd, P. Huang, M. Klaudiny, and A. Hilton, "Global non-rigid alignment of surface sequences," *International Journal of Computer Vision*, 2012. 29

[61] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *The Annals of Mathematical Statistics*, vol. 3, pp. 1554–1563, 12 1966. 29

[62] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, 2008. 29

[63] N. D. Lawrence, "Gaussian process models for visualisation of high dimensional data," in *Advances in Neural Information Processing Systems 16*, pp. 329–336, MIT Press, 2004. 29

[64] V. Pavlovic, J. M. Rehg, and J. MacCormick, "Learning switching linear models of human motion," in *Advances in Neural Information Processing Systems 13* (T. K. Leen, T. G. Dietterich, and V. Tresp, eds.), pp. 981–987, MIT Press, 2001. 29

[65] T. Tung and T.Matsuyama, "Intrinsic characterization of dynamic surfaces," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 29, 37

[66] Q. Chen and V. Koltun, "Robust nonrigid registration by convex optimization," in *International Conference on Computer Vision*, 2015. 31

[67] J. Starck and A. Hilton, "Correspondence labelling for wide-time free-form surface matching," in *International Conference on Computer Vision*, 2007. 31

[68] K. Varanasi, A. Zaharescu, E. Boyer, and R. Horaud, "Temporal surface tracking using mesh evolution," in *European Conference on Computer Vision*, 2008. 31

[69] F. Cuzzolin, D. Mateus, and R. Horaud, "Robust temporally coherent laplacian protrusion segmentation of 3d articulated bodies," *International Journal of Computer Vision*, vol. 112, pp. 43–70, Mar. 2015. 31

[70] R. Litman, A. M. Bronstein, and M. M. Bronstein, "Stable volumetric features in deformable shapes," in *Shape Modeling International (SMI) Conference*, 2012. 31

[71] A. Rangarajan, H. Chui, and F. L. Bookstein, "The softassign procrustes matching algorithm," in *Information Processing in Medical Imaging*, pp. 29–42, Springer, 1997. 32

[72] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, series B*, 1977. 32, 43, 46, 60

[73] G. E. Hinton, C. K. I. Williams, and M. D. Revow, "Adaptive elastic models for hand-printed character recognition," in *Advances in Neural Information Processing Systems 4*, vol. 4, 1992. 32

[74] C. Stoll, N. Hasler, J. Gall, H.-P. Seidel, and C. Theobalt, "Fast articulated motion tracking using a sums of gaussians body model," *International Conference on Computer Vision*, 2011. 32

[75] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004. 33

[76] E. Boyer, A. M. Bronstein, M. M. Bronstein, B. Bustos, T. Darom, R. Horaud, I. Hotz, Y. Keller, J. Keustermans, A. Kovnatsky, *et al.*, "Shrec 2011: robust feature detection and description benchmark," in *Eurographics 3DOR Workshop*, pp. 71–78, Eurographics Association, 2011. 33, 72

[77] A. Zaharescu, E. Boyer, and R. Horaud, "Keypoints and local descriptors of scalar functions on 2d manifolds," *International Journal of Computer Vision*, 2012. 33, 71

[78] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *CGF*, vol. 28, no. 5, pp. 1383–1392, 2009. 33, 71

[79] M. Aubry, U. Schlickewei, and D. Cremers, "The wave kernel signature: A quantum mechanical approach to shape analysis," in *ICCV Workshops*, IEEE, 2011. 33, 71

[80] D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst, "Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks," *Computer Graphics Forum*, vol. 34, no. 5, pp. 13–23, 2015. 33

[81] P. J. Huber, *Robust Statistics*. Wiley, 1981. 33

[82] R. M. Haralick, H. Joo, C. Lee, X. Zhuang, V. G. Vaidya, and M. B. Kim, "Pose estimation from corresponding point data," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, pp. 1426–1446, Nov 1989. 33

[83] H. Li, R. W. Sumner, and M. Pauly, "Global correspondance optimization for non-rigid registration of depth scans," *Computer Graphics Forum, Proc. SGP*, vol. 27, no. 5, 08. 33, 100

[84] M. Zollhöfer, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, and M. Stamminger, "Real-time non-rigid reconstruction using an rgb-d camera," *ACM Transactions on Graphics*, vol. 33, no. 4, 2014. 33

[85] C. Zach, "Robust bundle adjustment revisited," in *European Conference on Computer Vision*, pp. 772–787, Springer International Publishing, 2014. 33

[86] D. Geman and G. Reynolds, "Constrained restoration and the recovery of discontinuities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 367–383, Mar 1992. 33

[87] J. Shotton, A. Fitzgibbon, M. Cook, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2011. 34, 71, 72, 73

[88] L. Wei, Q. Huang, D. Ceylan, E. Vouga, and H. Li, "Dense human body correspondences using convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 34, 72, 73

[89] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, P. Kohli, V. Tankovich, and S. Izadi, "Fusion4D: Real-time performance capture of challenging scenes," *ACM Transactions on Graphics*, vol. 35, pp. 114:1–114:13, jul 2016. 34, 48, 72, 100, 101

[90] L. Ballan and G. M. Cortelazzo, "Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes," in *3DPVT*, 2008. 36, 54

[91] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel, "Motion capture using joint skeleton tracking and surface estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 36, 54, 71, 89

[92] F. Mémoli and G. Sapiro, "Comparing point clouds," *SGP*, 2004. 36

[93] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Efficient computation of isometry-invariant distances between surfaces," *SIAM Journal on Scientific Computing*, vol. 28, 2006. 36

[94] M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. J. Guibas, "One point isometric matching with the heat kernel," *Comput. Graph. Forum*, vol. 29, no. 5, 2010. 36

[95] Y. Sahillioğlu and Y. Yemez, "3d shape correspondence by isometry-driven greedy optimization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 36

[96] T. Windheuser, U. Schlickewei, F. Schmidt, and D. Cremers, "Geometrically consistent elastic matching of 3d shapes: A linear programming solution," in *International Conference on Computer Vision*, 2011. 36

[97] Y. Zeng, C. Wang, X. Gu, D. Samaras, and N. Paragios, "A generic deformation model for dense non-rigid surface registration: a higher-order MRF-based approach," in *International Conference on Computer Vision*, 2013. 36

[98] D. Kendall, "Shape manifolds, procrustean metrics, and complex projective spaces," *Bulletin of the London Mathematical Society*, vol. 16, no. 2, pp. 81–121, 1984. 37

[99] H. Hufnagel, X. Pennec, J. Ehrhardt, N. Ayache, and H. Handel, "Generation of a statistical shape model with probabilistic point correspondences and em-icp," *IJCAR*, vol. 2, no. 5, 2008. 37

[100] J. Franco, C. Menier, E. Boyer, and B. Raffin, "A distributed approach for real-time 3d modeling," in *CVPR Workshop*, 2004. 37

[101] T. Matsuyama, S. Nobuhara, T. Takai, and T. Tung, *3D Video and its Applications*. Springer, 2012. 37

[102] M. Fréchet, "Les éléments aléatoires de nature quelconque dans un espace distancié.," *Annales de l'institut Henri Poincaré*, vol. 10, pp. 215–310, 1948. 38

[103] H. Pottmann, Q.-X. Huang, Y.-L. Yang, and S.-M. Hu, "Geometry and convergence analysis of algorithms for registration of 3d shapes," *Int. J. Comput. Vision*, vol. 67, pp. 277–296, May 2006. 39

[104] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. 42, 43, 44, 60, 105

[105] J. Starck and A. Hilton, "Spherical matching for temporal correspondence of non-rigid surfaces," in *International Conference on Computer Vision*, 2005. 46

[106] J. Carranza, C. Theobalt, M. A. Magnor, and H.-P. Seidel, "Free-viewpoint video of human actors," in *ACM SIGGRAPH 2003 Papers*, pp. 569–577, 2003. 53, 89

[107] M. Alexa, D. Cohen-Or, and D. Levin, "As-rigid-as-possible shape interpolation," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pp. 157–164, 2000. 53, 54

[108] W. Matusik, C. Buehler, and L. McMillan, "Polyhedral visual hulls for real-time rendering," in *Rendering Techniques 2001* (S. Gortler and K. Myszkowski, eds.), Eurographics, pp. 115–125, Springer Vienna, 2001. 54

[109] J.-S. Franco and E. Boyer, "Exact polyhedral visual hulls," in *British Machine Vision Conference (BMVC'03)*, vol. 1, (Norwich, United Kingdom), pp. 329–338, Sept. 2003. 54, 61

[110] R. Szeliski, "Rapid octree construction from image sequences," *CVGIP: Image Underst.*, vol. 58, no. 1, pp. 23–32, 1993. 54

[111] K. Kolev, T. Brox, and D. Cremers, "Fast joint estimation of silhouettes and dense 3d geometry from multiple images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012. 54

[112] S. Vedula, S. Baker, S. M. Seitz, and T. Kanade, "Shape and motion carving in 6d.," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2000. 54

[113] L. Guan, J.-S. Franco, E. Boyer, and M. Pollefeys, "Probabilistic 3d occupancy flow with latent silhouette cues.," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 54

[114] A. O. Ulusoy, O. Biris, and J. L. Mundy, "Dynamic probabilistic volumetric models," in *International Conference on Computer Vision*, 2013. 54

[115] J. Starck and A. Hilton, "Surface capture for performance-based animation," *IEEE CGA*, 2007. 54, 89

[116] B. Goldlücke and M. Magnor, "Space-time isosurface evolution for temporally coherent 3D reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2004. 54

[117] N. Magnenat-Thalmann, R. Laperrière, and D. Thalmann, "Joint-dependent local deformations for hand animation and object grasping," in *Proceedings on Graphics Interface '88*, (Toronto, Ont., Canada, Canada), pp. 26–33, Canadian Information Processing Society, 1988. 54

[118] Y. Furukawa and J. Ponce, "Dense 3d motion capture from synchronized video streams," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 54

[119] T. W. Sederberg and S. R. Parry, "Free-form deformation of solid geometric models," in *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, pp. 151–160, 1986. 54

[120] Y. Liu, W. Wang, B. Lévy, F. Sun, D.-M. Yan, L. Liu, and C. Yang, "On centroidal voronoi tessellation - energy smoothness and fast computation," *ACM Transactions on Graphics*, vol. 28, no. 101, 2009. 56

[121] R. W. Sumner, J. Schmid, and M. Pauly, "Embedded deformation for shape manipulation," *ACM Transactions on Graphics*, vol. 26, July 2007. 57

[122] B. Allain, J.-S. Franco, E. Boyer, and T. Tung, "On mean pose and variability of 3d deformable models," in *European Conference on Computer Vision*, 2014. 57, 60, 61, 62, 64, 68, 71, 77, 78, 83

[123] E. Rodola, S. R. Bulo, T. Windheuser, M. Vestner, and D. Cremers, "Dense non-rigid shape correspondence using random forests," in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2014. 70

[124] C.-H. Huang, E. Boyer, B. do Canto Angonese, N. Navab, and S. Ilic, "Toward user-specific tracking by detection of human shapes in multi-cameras," in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun 2015. 70, 71, 72, 75, 79, 81, 83, 85

[125] G. Pons-Moll, J. Taylor, J. Shotton, A. Hertzmann, and A. Fitzgibbon, "Metric regression forests for human pose estimation," in *BMVC*, 2013. 71, 75

[126] M. Klaudiny, C. Budd, and A. Hilton, "Towards optimal non-rigid surface tracking," in *European Conference on Computer Vision*, 2012. 71

[127] K. Fujiwara, K. Nishino, J. Takamatsu, B. Zheng, and K. Ikeuchi, "Locally rigid globally non-rigid surface registration," in *International Conference on Computer Vision*, IEEE, 2011. 71

[128] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *European Conference on Computer Vision*, Springer, 2010. 71, 73

[129] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, "A comprehensive performance evaluation of 3d local feature descriptors," *International Journal of Computer Vision*, 2015. 72

[130] L. Wang, F. Hétroy-Wheeler, and E. Boyer, "A hierarchical approach for regular centroidal voronoi tessellations," *Computer Graphics Forum*, 2015. 72, 78, 94

[131] A. Criminisi and J. Shotton, *Decision forests for computer vision and medical image analysis*. Springer, 2013. 72

[132] P. Viola and M. J. Jones, "Robust real-time face detection," in *International Journal of Computer Vision*, Springer, 2004. 73

[133] B. Allain, J.-S. Franco, and E. Boyer, "An efficient volumetric framework for shape tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2015. 78, 83

[134] Z. Popović and A. Witkin, "Physically based motion transformation," in *Proc. of SIGGRAPH*, 1999. 89

[135] V. B. Zordan and J. K. Hodgins, "Motion capture-driven simulations that hit and react," in *Proc. of SCA*, 2002. 89

[136] A. Sulejmanpasić and J. Popović, "Adaptation of performed ballistic motion," *ACM Transactions on Graphics*, vol. 24, no. 1, 2005. 89

[137] C. K. Liu, A. Hertzmann, and Z. Popović, "Learning physics-based motion style with nonlinear inverse optimization," *ACM Transactions on Graphics*, vol. 24, no. 3, 2005. 89

[138] A. Safonova, J. K. Hodgins, and N. S. Pollard, "Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces," *ACM Transactions on Graphics*, vol. 23, no. 3, 2004. 89

[139] Y. Ye and C. K. Liu, "Animating responsive characters with dynamic constraints in near-unactuated coordinates," *ACM Transactions on Graphics*, vol. 27, no. 5, 2008. 89

[140] X. Wei, J. Min, and J. Chai, "Physically valid statistical models for human motion generation," *ACM Transactions on Graphics*, vol. 30, no. 3, 2011. 89

[141] D. Casas, M. Tejera, J. Guillemaut, and A. Hilton, "Interactive animation of 4d performance capture," *IEEE Trans. on Visualization and Computer Graphics*, vol. 19, no. 5, 2013. 89

[142] T. J. Cashman and K. Hormann, "A continuous, editable representation for deforming mesh sequences with separate signals for time, pose and shape," *Comput. Graph. Forum*, vol. 31, no. 2, 2012. 89

[143] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan, "Image based visual hulls," in *Proc. of SIGGRAPH*, 2000. 89

[144] M. Gross, S. Würmlin, M. Naef, E. Lamboray, C. Spagno, A. Kunz, E. Koller-Meier, T. Svoboda, L. V. Goll, S. Lang, K. Strehlke, A. V. Moere, and O. Staadt, "blue-c: A Spatially Immersive Display and 3D Video Portal for Telepresence," *ACM Transactions on Graphics*, vol. 22, no. 3, 2003. 89

[145] T. Tung, S. Nobuhara, and T. Matsuyama, "Complete multi-view reconstruction of dynamic scenes from probabilistic fusion of narrow and wide baseline stereo," in *International Conference on Computer Vision*, 2009. 89

[146] D. Vlasic, P. Peers, I. Baran, P. Debevec, J. Popović, S. Rusinkiewicz, and W. Matusik, "Dynamic shape capture using multi-view photometric stereo," *ACM Transactions on Graphics*, vol. 28, no. 5, 2009. 89

[147] C. Wu, K. Varanasi, and C. Theobalt, "Full-body performance capture under uncontrolled and varying illumination : A shading-based approach," in *European Conference on Computer Vision*, 2012. 89

[148] W. E. Lorensen and H. E. Cline, "Marching Cubes: A high resolution 3d surface construction algorithm," in *Proc. of SIGGRAPH*, 1987. 91

[149] T. Ju, F. Losasso, S. Schaefer, and J. Warren, "Dual contouring of Hermite data," *ACM Transactions on Graphics*, vol. 21, no. 3, 2002. 91

[150] J. R. Shewchuk, "Tetrahedral mesh generation by Delaunay refinement," in *Symposium on Computational Geometry (SoCG)*, ACM, 1998. 91

[151] C. Jamin, P. Alliez, M. Yvinec, and J.-D. Boissonnat, "CGALmesh: a generic framework for delaunay mesh generation," *ACM Transactions on Mathematical Software*, 2014. 91

[152] E. V. Chernyaev, "Marching Cubes 33: Construction of topologically correct isosurfaces," Tech. Rep. CN 95–17, CERN, 1995. 92

[153] CGAL, "Computational Geometry Algorithms Library." http://www.cgal.org/. 92

[154] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE Jour. on Robotics and Automation*, vol. 4, 1988. 92, 93

[155] R. Rabbitz, "Fast collision detection of moving convex polyhedra," in *Graphics Gems IV* (P. S. Heckbert, ed.), Academic Press, 1994. 92

[156] D. Baraff, "An introduction to physically based modeling: Rigid body simulation," in *SIGGRAPH Course Notes*, 1997. 92

[157] Bullet, "Bullet physics library." http://bulletphysics.org/. 92

[158] E. Catto, "Iterative dynamics with temporal coherence," in *Proc. of Game Developer Conference*, 2005. 92, 93

[159] D. Baraff, "Fast contact force computation for nonpenetrating rigid bodies," in *Proc. of SIGGRAPH*, 1994. 93

[160] J. R. Kent, W. E. Carlson, and R. E. Parent, "Shape transformation for polyhedral objects," in *Proc. of SIGGRAPH*, 1992. 96

[161] B. Petit, J.-D. Lesage, E. Boyer, and B. Raffin, "Virtualization gate," in *SIGGRAPH 2009 - Emerging Technologies*, ACM, 2009. 100

[162] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou, V. Tankovich, C. Loop, Q. Cai, P. A. Chou, S. Mennicken, J. Valentin, V. Pradeep, S. Wang, S. B. Kang, P. Kohli, Y. Lutchyn, C. Keskin, and S. Izadi, "Holoportation: Virtual 3d teleportation in real-time," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, (New York, NY, USA), pp. 741–754, ACM, 2016. 100

[163] M. Dou, J. Taylor, H. Fuchs, A. Fitzgibbon, and S. Izadi, "3D scanning deformable objects with a single RGBD sensor," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2015. 100

[164] A. Letouzey and E. Boyer, "Progressive shape models," in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, June 2012. 101

[165] M. Dou, H. Fuchs, and J. M. Frahm, "Scanning and tracking dynamic objects with commodity depth cameras," in *Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on*, 2013. 101

[166] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger, "Volumedeform: Real-time volumetric non-rigid reconstruction," in *European Conference on Computer Vision*, 2016. 101