Structural and Temporal Analysis for Time-Varying-Meshes

by

© Ning Sung Lee 37-66979

A thesis submitted to the

Graduate School of Engineering

in partial fulfilment of the

requirements for the degree of

Master of Engineering

Department of Electronics Engineering

University of Tokyo

2008 August 18

Abstract

This thesis proposes algorithms for structural and temporal analysis for Time-Varying-Meshes (TVMs), through hierarchical skeleton-based mesh segmentation and surface curvature matching. The motivation is to understand and retrieve structural information while establishing spatio-temporal correspondence in TVMs. With that, we can also achieve *markerless* motion transfer from humans onto synthetic models. The thesis also introduces a parts-based skeleton extraction method using geodesic distances to handle badly-defined meshes that are common in TVMs. Thus the algorithm enables tracking of highly deformable models with arbitrary genus.

The proposed framework is a recursive system that iterates between hierarchical segmentation on minimum distance satisfaction and skeleton realignment to refine a kinetic model for each TVM frame. Results show that the segmentation is stable and successful motion tracking is achieved from TVMs.

We also explore the use of surface curvatures in Iterated Closest Point (ICP) matching for more refined motion understanding on the segmented results. Results show that processing time can be reduced by 80% while retaining the same level of accuracy. Multi-temporal registration, where the data in the previous and the following frames are used in the 3D shape registration, is used to increase robustness against potential bad registration as ICP is sensitive to outliers and deformation. It achieved an angle disparity reduction of average 29% with two additional time registrations.

Acknowledgements

I would like to dedicate this work and my thanks to all my loved ones, family and friends for their patience and support for my education. In specific, I would like to express my gratitude to my advisor, Professor Aizawa, for the knowledge imparted and guidance that I have received over the last two years. I would also like to thank Professor Yamasaki and all members of the Aizawa-Yamasaki Laboratory for the numerous hours spent in discussion and the knowledge shared. Also, specifically to highlight the members of the 3D MEXT group, Han, Xu, Tadano, Takashi, Kasai, Nakagawa, whom I have worked closely with in this thesis. I would also like to thank Chaminda, Rene, Ovgu and Nishioka for their constant support, joy and laughter. Not to forget, friends I have in Japan for their encouragement and company.

This work is financially supported by Ministry of Education, Culture, Sports, Science and Technology of Japan under the "Development of Fundamental Software Technologies for Digital Archives".

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In recent years, 3 dimensional (3D) computer graphics has become more prevalent in our lives. 3D computer graphics are used extensively in entertainment such as games and movies, medical imaging and prototyping. We have notable success from entertainment movies such as *The Matrix*, *Lord of the Ring* and *Pirates of the Caribbean*, which are mix of 3D computer graphics and live actors. In addition, we also have fully animated films such as *Finding Nemo* and *Ice Age* having equal success. The growth of 3D computer graphics in the entertainment industry shows the popularization and demand of realistic representations amongst consumers. 3D computer graphics are used to simulate reality and to compensate the experience that 2D graphics is not able to achieve.

Moreover, 3D graphics are used in particular to simulate the real world, to imitate human behavior and natural movements. From the past, artists and animators are

employed to animate every single action, including facial features, and motion capture systems are used to simulate realistic movements in animated models. However, there are limitations in this approach. Humans are very sensitive to motion and human images and are able to detect subtle yet weird movements. Therefore, absolute realistic animation is difficult to achieve through simulated motion and models, one example is cloth animation.

Various methodologies and technologies have been widely researched into *digitizing humans* and their motion. Motion capture is a technology used to capture human motion and digitizing the data for animating human characters realistically. The system setup comprises of optical markers that are placed strategically on performers and cameras to capture the positions of markers in space [4]. *Final Fantasy: The Spirits Within* and *Titanic* [5] also utilized motion capture for visual effects. However, motion capturing systems are intrusive and are not perfect; it is unable to fully capture the surface dynamics of cloth, skin and hair. If we were to capture the naturalistic deformation of the skin, we have to attach many optical markers on the skin [6], and as for movement of hair it is also almost impossible to attach markers.

Besides research on motion capture systems, there has been research into *digitizing humans* that capture the surface information of a human subject. *Digitized humans* capture realism in its entirety and compensate the deficiencies of motion capture systems. Though traditional 3D scanning applications, such as laser scanning techniques [7, 8], have been able to produce an accurate digital model of the scene geometry down to the accuracy of μmeter scale, they are unable to capture dynamic scenes. This is due to the delay in data collection of different parts of the same subject. Texture capture is also compromised with the use of laser scanning

technologies.

There has been an emergence in another methodology to *digitize* humans. Time-Varying-Meshes (TVMs) or 3D video, in contrast to scanning techniques, are captured passively and *markerlessly* using synchronized cameras without restrictions on the subject. They are digital representations of dynamic scenes. TVMs are considered for sports analysis and entertainment purposes due to the dynamic surface capture and representation. Structural information and motion data are inherently captured but not explicitly represented in the surface animation. Therefore, to extract motion, we would need to analyze structure and extract motion from TVMs.

In this thesis, structural and temporal analysis of Time-Varying-Meshes is explored. We analyze each TVM frame for structural information and temporal correspondence for extraction of motion. We intend to extract human motion without the use of physical markers, thus allowing the layman, or non-professionals to digitize their movements easily [9]. In another words, to transfer motion from the subjects into synthesized models for animating purposes. By establishing correspondence between frames, we can apply the extracted information in motion analysis, motion search applications and also to be used in compression.

As each TVM frame is created independently, each frame has different number of vertices, connectivity and color, therefore this is no correspondence between frames. The lack of spatio-temporal correspondence across frames pose great challenges to compute motion. Furthermore, TVMs consists of highly deformable models that changes in topology and have arbitrary genus.

In this thesis, we utilize a pre-defined kinetic model to segment each 3D model into a structure that is consistent throughout the entire TVM sequence. Segmentation

reduces the problem by partitioning each frame into smaller meaningful parts that are easier to process. By having a consistent structure, correspondence across frames is established. Segmentation results in a piece-wise pseudo-rigid model that can be used in motion tracking.

Our approach for structural and temporal analysis of Time-Varying-Meshes (TVMs) uses hierarchical skeleton-based mesh segmentation and surface curvature matching. The thesis also introduces a parts-based skeleton extraction using geodesic distances to handle intersected meshes that are common in TVMs. The segmentation algorithm does not require surface registration across frames. However, the thesis also examines the use of surface curvature matching for more refined motion understanding of the segmented results. This is to improve the use of Iterative Closet Point matching algorithm, which is sensitive to outliers and deformation.

Chapter 2 first summarizes other related works with regard to motion capture, segmentation and motion extraction of TVMs. The chapter also includes a brief introduction to the creation technology and properties of the TVMs used in this thesis. An overview of the motion extraction system is described in Chapter 3 along with the terminology and assumptions. The proposed algorithm consisting of three main parts, initialization, segmentation, pose estimation or skeleton realignment, is further elaborated in Chapter 4. Chapter 5 includes experimental results of the proposed algorithm. Chapter 6 explores the use of surface curvatures for matching and chapter 7 shows the experimental results of matching. Finally, Chapter 8 concludes the thesis.

Chapter 2

Background

2.1 Related Works

For motion capture, physical skeletal movements are captured using physical optical markers. Motion capture systems comprise of optical markers placed strategically on performers and cameras to capture positions of these markers. By sampling the positions of such optical markers across time, the system translates the captured data into motion of these specified markers. The captured data is also called motion capture data. The capture data typically represent positional information of joints of humans in space and the motion with respect to time. Optical markers, which could be retro reflective or light-emitting have to be attached securely and non-moving on the performers during recording [1]. With that, the performers are required to wear tight-fitting suits 2.1 on which the markers are placed and to be properly seen [1]. With the increase in the number of markers, animating muscle deformation is also possible [6], but increase of markers compromises on the comfort level of the performers. Moreover,

markers distort the surface texture of the subject.

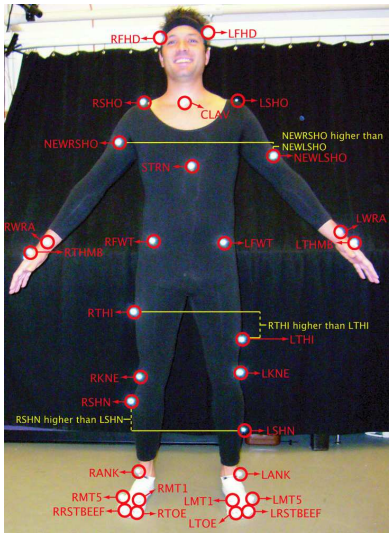


Figure 2.1: Motion capture performer in tight-fitting suits and markers [1].

Time-varying-meshes (TVMs), on the other hand, are *markerless*. In contrast to computer generated 3 Dimensional (3D) models, TVMs do not have fixed vertices or fixed topology, therefore lack spatio-temporal correspondence. Related works on motion extraction from TVMs establishes correspondence across time using various properties. As no markers were used in mesh construction, we need to extract motion through model and motion analysis techniques. These works can be broadly categorized into two types, non-structural methods that calculate movement or dense correspondence between frames and structure-based methods that uses body shape modeling for each TVM *frame*.

Non-structural motion extraction techniques includes Vedula et al.'s [10] 3D scene flow and Starck et al.'s [11] surface correspondence using a MAP-MRF framework allowing sparse to dense matching. Both methodologies describe methods for es-

tablishing dense surface correspondence across time. However, these methods lack structural correspondences. With a user-defined skeletal model, Starck et al. [12] applied the spherical matching algorithm to obtain motion tracking from dense correspondences. However, spherical matching only deals with genus-0 models. The algorithm parametizes the models in a sphere and correspondence is obtained by 2D matching across the spheres. For genus-n models, that are common in TVMs, mesh cutting is required to convert such models into genus-0 models. Though there are techniques [13, 14] that can convert automatically, in the case of TVMs, the cut positions are critical. Yamasaki et al. [15] has proposed a method that allow detection and identify regions that are to be cut. However, it might not be practical to cut surfaces for each frame manually for the entire TVM sequence.

In this thesis, we are interested in structural understanding and obtaining a uniform human structure for motion tracking across frames. Therefore we consider structure-based techniques. Typically, segmentation or kinetic models are used for structural analysis [16, 17, 18, 19, 20, 21, 22]. Pekelny et al. [17] uses user-defined segmentation to estimate pose in future frames through the use of Iterated Closest Point (ICP) [23] matching, or 3D surface registration between frames for motion estimation. The initial segmented model is motion compensated to estimate the segmentation in the future frames. Pekelny et al. [17] also define a piece-wise rigid model in their research and estimate pose for models that have less surface dynamics. There is a need for segmentation of the mesh into a piece-wise rigid model because 3D surface registration [23] assumes of matching between rigid and similar models. However, even with a piece-wise structure, each segment could deform with time, causing a drift matching across frames and correspondence loss [12]. Knoop et al. [19] also uses

ICP matching for 3D shape correspondence and tracking. ICP is commonly used for 3D registration [24, 25] in high resolution non-deformed meshes [26, 27, 28], however in this thesis, we also explore the use of ICP for our deforming meshes. However, instead of only using one segmented frame [17], we segment all frames in the entire sequence before ICP matching. Unlike Pekelny et al. [17], which used heuristics to overcome ICPs vulnerability to outliers, our algorithm does not impose constraints on the matching process. We characterize the mesh using curvatures that are local parameters of the mesh. This would be further discussed in section 6.

Mesh decomposition or segmentation is a very important procedure as it is a process of mesh simplification, and broadly used in pose estimation and structure analysis [17, 18, 16, 29]. As discussed, segmentation allows us to divide the mesh into smaller pieces or piece-wise pseudo rigid model that enables structure analysis and establishing correspondences. Research on automatic segmentation and skeleton extraction is generally applied to data that are well-defined, static or motion-restricted models [18, 30]. Mesh decomposition for well-defined and static models includes fuzzy clustering and cuts for mesh decomposition [18] and space sweeping with use of skeletons [31]. Hierarchical convex decomposition [18] calculates the probability of each vertex on the mesh model from pre-defined patches of interest. The algorithm utilizes geodesic distances and angle differences to segment the mesh into various parts. In contrast to computer generated models, which are feature specific and genus-0, TVMs are noisy and ill-defined models. Therefore deep concavities utilized in [18] are not available for clean segmentations. Space sweeping would not be applicable for genus-n models.

As for TVMs, previous works have explored the use of other features such as

voxels, "3D pixel" representations of 3D models [32] and original 2D video captures of the mesh [33, 34]. Mikić et al. [32] describes a recursive framework for human model acquisition of TVMs using voxels. However, in [32], the models were of subjects in tight suits, with minimal cloth motion. Our TVM data, on the other hand, capture motion without any constraint on the clothes. By having no constraints on the TVM data, it increases the variability of the data, causing the models to difficult to process. In addition, we aim to segment 3D polygonal meshes. Two of the earlier works on *markerless* human motion transfer and pose estimation by Cheung [33, 34] models the acquired 3D shape into a kinematic model and extracts motion data. The algorithm uses the silhouette information obtained from the original video sequences to estimate pose and kinematics of the subjects in the video sequence. Similar to Cheung's proposed algorithm, we estimate pose in each frame and track the obtained structure across time. However, in this thesis, we estimate pose from 3D models without the use of the original video sequences.

Generally algorithms require an initialized model or information on the relative structure for structure analysis and pose estimation. There have been research on extraction of skeleton structures from 3D without prior structural information. An example is Tadano et al.'s [35] geodesic distance based skeleton extraction algorithm, which retrieves human pose from each TVM frame without the use of any initial structure. However, the resultant skeleton is unstructured and varies in topology and highly dependent on noise, gaps or artefacts in the mesh. Therefore it fails to correctly estimate poses from genus-n models, though Tadano et al. also propose tracking of previously extracted skeletons to estimate poses from such models. Xiao et al. [36] also utilizes geodesic distance to construct a reeb-graph for segmentation

of the human body. It demonstrates the feasibility of mesh segmentation into basic components through use of geodesic distances. In the paper, it is stated that the Reeb-Graph contains O-type, λ-type and Y-type graphical patterns, and TVMs only contain λ-type graphical structures. By distinguishing the λ-type from O-type and Y-type graphical patterns in the Reeb-Graph structure, it could segment the model into a basic structure of the legs, arms and the body. However, the paper does not seem to handle genus-n models and is unable to extract the head.

From the research discussed as far, there is insufficient information to construct an appropriate model of badly-defined models just from geodesic distances. Tadano et al. [35] uses a prior model and matches the entire model onto the extracted unstructured reeb-graph. However, predicting the path in a search space did not seem sufficient to accurately track the models. From these previous works, we come to understand that to handle motion extraction of TVMs, we need a prior structure to successfully estimate the pose of a genus-n model.

We discussed automatic segmentation and structure extraction for TVMs, sequences of highly deformable objects. In this thesis, we extract motion from full human models with high dynamics that are non-rigid and have limited surface details. In contrast with previous works [32, 18, 17], our subjects are of real humans with high cloth dynamics. In our proposed algorithm, we first utilize a pre-defined skeleton for mesh segmentation to segment each frame into a pseudo piece-wise rigid kinetic structure. Our proposed algorithm also uses hierarchical approach similar to Katz et al.'s [18] and Xiao et al. [36] for intermediate understanding to achieve a stable segmentation and handle arbitrary genus models. Mesh decomposition utilizes a kinetic chain structure [18, 37, 16, 19] to simplify the mesh model to a piece-wise

pseudo-rigid model. This kinetic chain structure is pre-defined and fixed in length and achieves pose estimation of ill-defined models where reeb-graph-based skeleton extraction fails. In this thesis, we use geodesic distance with hierarchical decomposition to segment genus-n models.

2.2 Mesh Creation Methodology

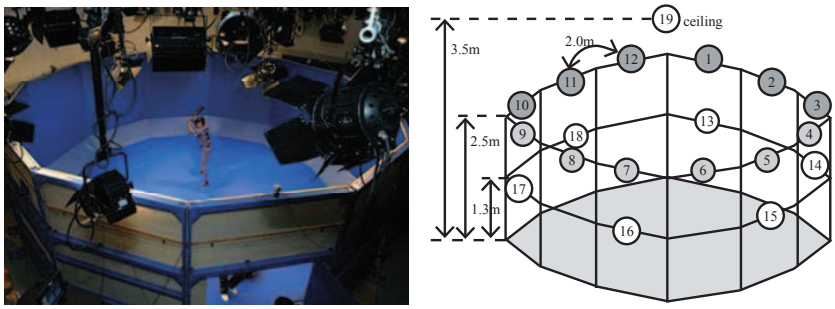


Figure 2.2: Camera setup for TVM capture.

In recent years, there has been a lot of development and research in this area [38, 39, 40]. This is due to the flexibility, low-cost and range capabilities of stereo camera systems. Geometric-based passive visualization methods, more commonly known as multi-view stereo, are characterized by its use of information acquired from images of a given scene or object. Such images are acquired from multiple camera systems such as Tomiyama et al.'s [2] camera system where 19 fire-wire cameras are arranged in a ring as shown in Figure 2.2. Other examples include Kanade et al.'s [41] 3-dimensional dome, made from 51 cameras mounted on a geodesic dome 5 meters in diameter and Starck et al.'s [42] 8 High-Definition video-camera system. Four views

of a scene at a specific time instance is shown in Figure 2.3 (top). In this paper, two main techniques, volume intersection and multi-baseline stereo, will be further described in the following sections. These techniques are used for construction of the models used in this paper.

Figure 2.3: (Top) Two views taken at the same time, (bottom) two views with background removed.

2.2.1 Volume Intersection

The volume intersection method describes a fast and robust method used to generate a volume using silhouettes. Through the method, *visual hull*, a surface defining the volume of the object, is created. In recent years, visual hulls are commonly used as an initial surface [3, 2, 43, 42, 44] to restrict search and also to reduce computation time. It is made from voxels, volumetric representations of geometry by using a regularly

sampled 3D volume with discrete occupancy function [45]. Volumetric reconstruction is used to create a volumetric representation of the scene or object that is consistent with all images or simply, a model that matches the original when projected back on to the images. It represents the volume directly, with no explicit reference to any image. Therefore it is view independent.

This method uses shapes taken from silhouettes of the subject in a particular scene, and carves a visual hull of maximum possible volume. Firstly, for each image view of the subject, a silhouette of the subject is created by removing the shape of the subject from the background. Background subtraction methods such as chromakey processing [2] are used to separate the subject from its background, see Figure 2.3 (bottom). A visual cone is then formed from the camera's optical center as the vertex, and inversely projected onto the world's coordinates, see Figure 2.4. The silhouettes form the limits of the visual cone. More specifically, the visual cone is formed by determining if the voxels in space are visible (in the visual cone) or not. A visual cone is formed for every camera view and the intersected regions of all the visual cones from available cameras define the visual hull. In another words, the visual hull consists of voxels that are consistently in the visible regions.

The volume intersection method is a robust method used to generate a stable approximate model at a low computation cost [3, 2, 43, 42, 44]. However, this method is highly conservative thus the resultant visual hull takes the form of the upper bound on the surface. The volume of the visual hull is an over-estimation of the real volume and envelopes the real surface. Another limitation of this method is poor performance with capturing concave features in a subject. The picture on the left in Figure 2.5 shows caves, such as the eyes, in a real object that do not appear in silhouettes. The

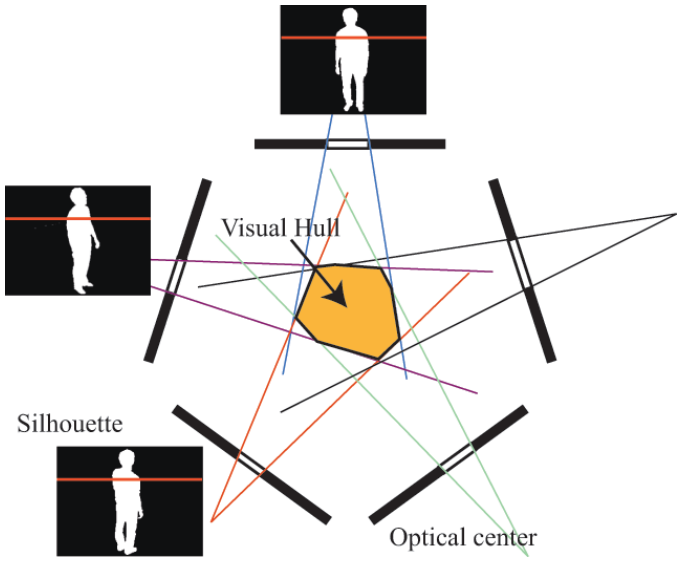


Figure 2.4: Volume intersection method [2].

algorithm would fail in this situation as it only retrieves information from silhouettes of real objects. Volume intersection method is also limited by the use of silhouettes. Silhouettes are not necessary accurate as background subtraction is not easy to perform. The picture on the right in Figure 2.5 shows the result of a bad implementation of background subtraction where the background is not fully removed.

2.2.2 Multi-baseline Stereo

The second technique is multi-baseline stereo. Similar to how humans perceive depth, discrepancies of an object from two views are used to estimate depth of objects. It was first introduced as a window-based binocular stereo matching method. This class of methods uses a window based matching algorithm to match images of different views and retrieves correspondence between the images. By drawing correspondence, this technique recovers depth information for the acquired set of images. Subsequently,

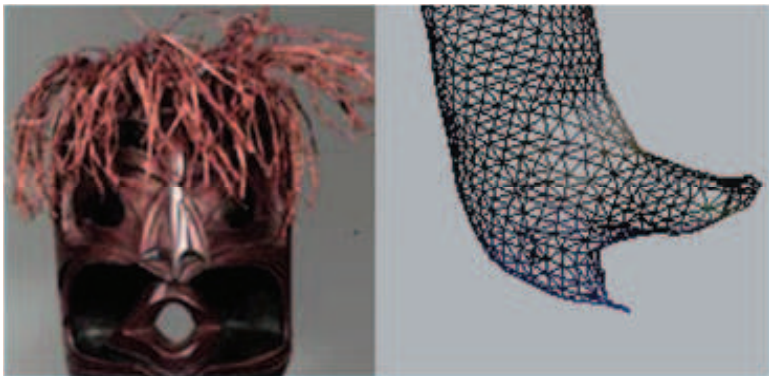


Figure 2.5: Limitations of volume intersection method. (Left) Example of topology that fails [3]. (Right) Example of bad background subtraction.

the set of 2.5D depth maps are merged to form a 3D model from which new views can be generated.

In order to calculate depth information, images of different views are used. First consider two parallel cameras with the same focal length F at a distance of B apart from each other. The perpendicular distance z to a point in world coordinates is related to the difference in the location of the point in the images d and is defined by:

$$d = \frac{BF}{z} \tag{2.1}$$

Various cost-based matching algorithms [46] are then used to find correspondences across images. The most common window-based matching costs include squared intensity differences and absolute intensity differences [41], which correspond to the color intensity difference between two neighborhoods. Other methods include normalized cross correlation (NCC) of two neighborhoods which compares the intensity distributions and is invariant to changes of the mean intensity value and of the dynamic range [3] of the neighborhoods. The depth with the minimum error is calculated after correspondence between images is found. Depth is calculated for each pixel in

the image with a generalized form of equation 2.1:

$$\frac{d}{BF} = \frac{1}{z} = \varsigma \tag{2.2}$$

In Kanade et al.'s [41] algorithm, depth information is calculated for each pixel in each view. The depth maps are then merged. This is done by fixing a reference view and building the model by filling holes that appear with new views.

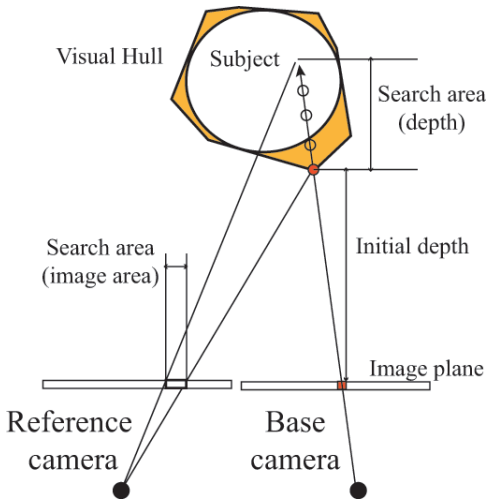


Figure 2.6: Correspondence detection with a restricted search area [2].

Multi-baseline stereo fails with uniform appearances where matching becomes ambiguous and with surfaces that are tangential to the image plane. In addition, the method is view dependent as a reference view is selected for reconstruction. Therefore, there is a possibility of obtaining different results when different reference views are used. Models derived using multi-baseline stereo also tend to be noisy as depth of each pixel is calculated independently. Kanade et al. [41] also states that as depth of every pixel is constructed, the algorithm is slow and resultant models could be noisy.

In recent years, several modifications have been proposed to improve the multi-

baseline stereo methods. One of the adaptations by Tomiyama et al. [2] used in the construction of the TVM data used in this paper. The adaptation uses an initialized surface to limit the search space for finding image correspondence as shown in Figure 2.6.

2.3 Time-Varying-Meshes

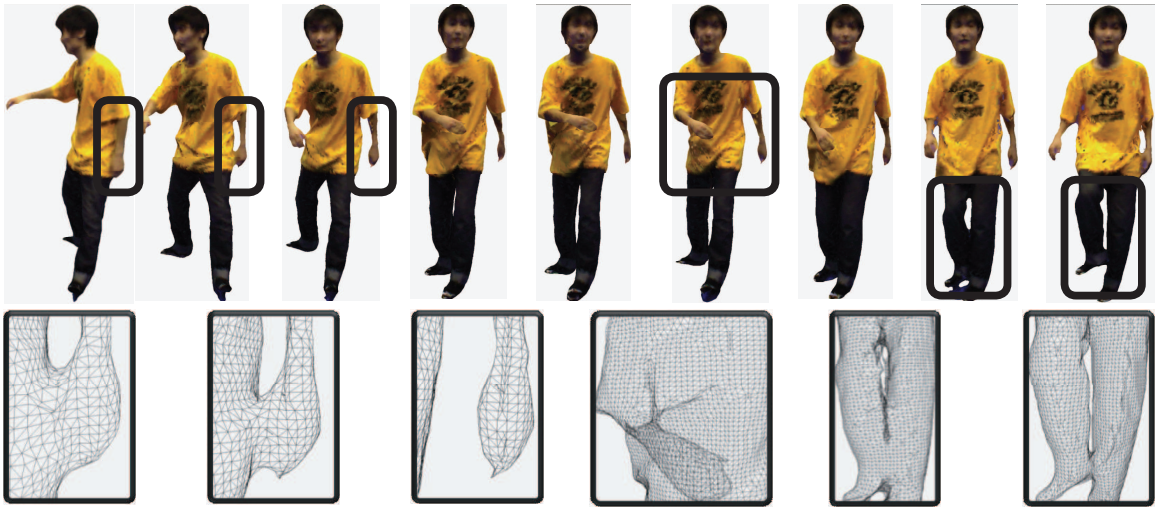


Figure 2.7: Nine frames of TVM sequence, *hip-hop* with inconsistencies highlighted.

Time-Varying-Meshes (TVMs), commonly associated with free-viewpoint video or 3D video are sequences of three-dimensional (3D) meshes, depicts a 3D scene which changes with respect to time as shown in Figure 2.7. Each TVM *frame* is a 3D polygon mesh model, with low level data such as geometry including positional and connectivity information for points in space and color information. More explicitly, a 3D mesh model is a set of coordinates (x, y, z) in 3D space. It contains connectivity

information between vertices as the mesh is represented as triangular patches. Textual mapping is provided as color information (r, g, b) per vertex. Each *frame* is a dense mesh model with highly detailed external surface outline of a given model. Each *frame* also captures the pose of the object of interest in each time step. Therefore, a sequence of TVM *frames* capture the motion of the object. TVMs are able to recreate accurate representations of real motion by simulating motion of the entire surface across time.

However, due to independent generation of each frame, the frames of a TVM are characterized by arbitrary genus-n models with inconsistent topology and properties. In another words, different frames have different number of vertices, connectivity and color as highlighted in Figure 2.7. Therefore there is no spatio-temporal correspondence between each frame. Even though motion is inherently captured through the changes in the meshes across time, it is not explicitly represented in a TVM. Needless to say, no high-level information such as structural information is captured either.

The sequences used in this paper are generated using dense stereo and volumetric intersection methodologies described by Tomiyama et a [2] and have a frame rate of 10 frames per second.

Chapter 3

Overview

3.1 Algorithm Overview

This thesis is split into two parts. The first is structural analysis and temporal correspondence and the second part focuses on finer surface correspondence analysis.

We first approach the problem of structural analysis by segmenting a TVM frame with respect to a pre-defined mesh, by calculating the distance correspondence between the mesh and the structure. This thesis uses a pre-defined structure for segmentation to handle genus-n and badly defined models. To obtain temporal correspondence or motion, we estimate pose from each frame with respect to the segmentation results, which includes volume information. As the estimated kinetic structure used for segmentation may be erroneous, leading to a sub-optimal pose estimation, we recurse between segmentation and skeleton realignment to refine the extracted pose. A sequence of poses gives rise to motion.

A flow chart of the proposed algorithm is shown in Figure 3.1. There are three

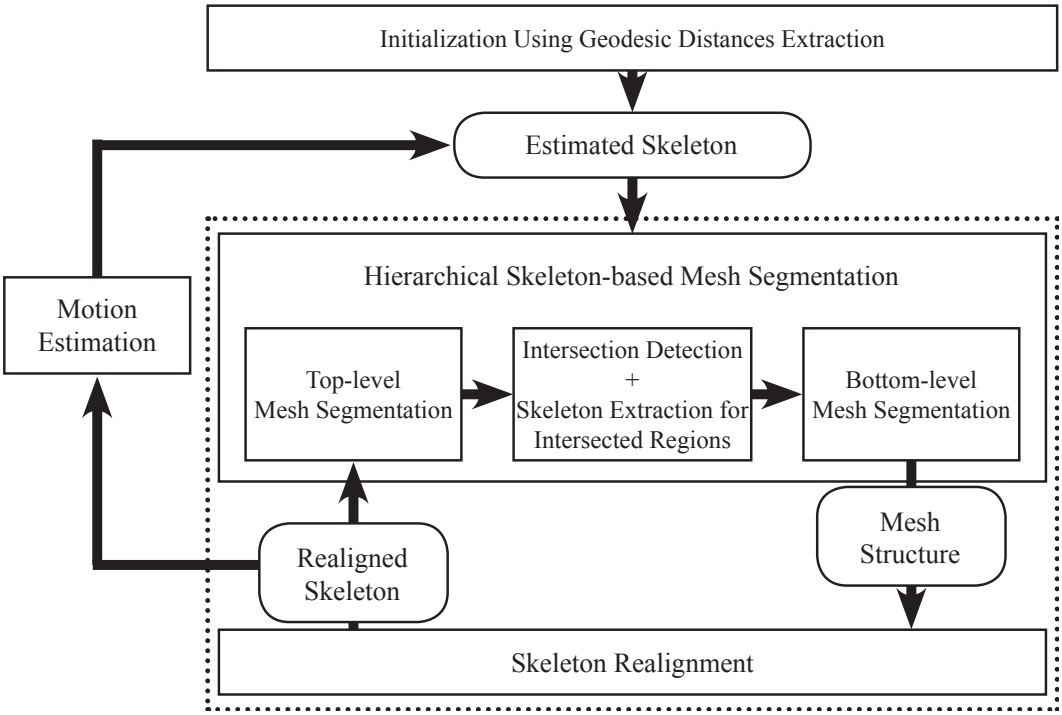


Figure 3.1: Flow chart of structure and temporal analysis algorithm.

main components; an initialization step for acquisition of an estimated kinetic model of a TVM frame, followed by a recursive mechanism between hierarchical mesh segmentation and skeleton realignment. For initialization frames where no prior segmented frames exist, Tadano et al.'s [35] algorithm is used for estimating an initial pose from a genus-0 and well-defined frame (Section 4.1). Subsequently, hierarchical segmentation (Section 4) is used to segment the mesh into defined parts consistent in each TVM frame of a sequence. The model is first segmented into six parts: head, left arm, right arm, left leg, right leg and body; each part is then further decomposed into smaller portions independently with consideration of mesh intersections. Thereafter, the skeleton is adjusted to fit the resultant segmentation (Section 4.3). This process is necessary as the initial skeleton may not be optimal. By recursively performing

segmentation and realignment, a refined skeleton is obtained. Color refinement, a step that is useful for TVMs with many badly-defined frames is also discussed. The kinetic model of the next frame is determined using the estimated poses of the previous frames. No motion, linear, and non-linear motion estimators are used and determined empirically.

Segmentation also allows us to break the problem into a piece-wise pseudo-rigid structure. Each segmented part is pseudo-rigid, thus used in 3D matching to estimate finer transformations between each body part. As Iterated Closet Point (ICP) is computationally expensive and sensitive to outliers, we filter vertices of high surface curvatures determined by normals for matching. Then corresponding segmented parts of the previous and following frames are matched to the current frame. The skeleton that is defined along with the segmented part is refined along with a third virtual marker, used for tracking rotation about the skeletal part.

3.2 Skeleton Representation

The proposed algorithm leverages on the properties of a human to describe each TVM frame and uses the skeleton or kinetic model to segment the mesh models. As the skeleton is a good representation of various body parts, it is used to segment the model into desired parts. Kinetic models [47, 37, 17, 34, 33, 16, 19] are fixed structures defined by the use of several rigid segments connected at flexible joints. The acquired skeleton defines the number of body parts that are to be segmented from the TVM frame. In this paper, we are considering a standardized segmentation of each human mesh model into the six top level parts. Each extension is subsequently segmented

resulting in a total of 15 body parts. In this paper we consider human proportions as described in *Vitruvian Man* by Leonardo da Vinci when defining the skeleton of each TVM. Some important ratios considered are: a human head is about 1/7 of his height, the pelvis is midway between feet and neck and the length from the finger tip to tip approximates to full height of the body.

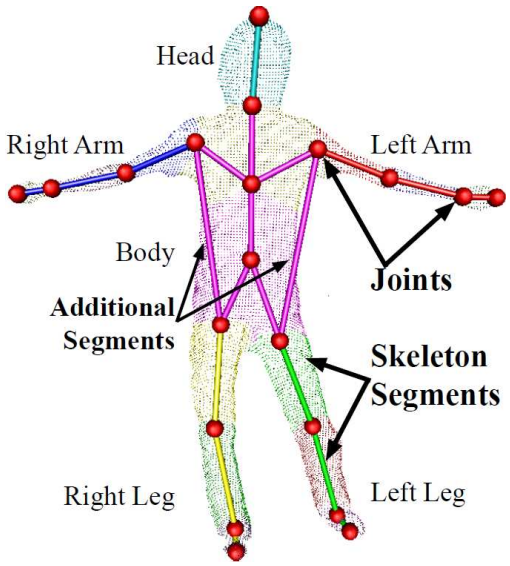


Figure 3.2: Kinetic model with 15 body segments defined by 21 skeleton segments. Red spheres represent joints.

As shown in Figure 3.2, the parts are split into the head, left and right hands, lower arms, upper arms, left and right feet, lower legs and upper legs, mid and top body. They are connected through flexible joints, such as the tip of the hand, wrists, elbows, shoulders, ankles, knees, hips and neck.

The skeleton represents connectivity and pose of each body part and other constraints such as hierarchical information of each segment, such as broadly categorizing the body parts into six portions, body, head, left and right arms and left and right

legs. Each limb is then subdivided into three smaller pseudo-rigid parts. Each segment part is accompanied by two joints placed at the ends of the parts to indicate the rotation of the pseudo-rigid parts. The parts are pseudo-rigid due to the surface deformations of the clothes worn by subject. Each joint contains three degrees of rotational freedom and define the connection between segments. No constraints on the degrees of freedom are placed on the joints unlike Carranza et al. [37].

The defined joints cannot encode joint angle limitations or indicate self-intersection. Therefore some constraints such as the skeleton must lie within the mesh cavity, are imposed to resolve these issues while estimating pose from segmentation. Two additional skeleton segments are also included to define the large body and take volume into consideration for segmentation.

These body parts are fixed to provide correspondences across frames and allow the algorithm to be more robust to the inconsistencies of TVM. The skeleton segment lengths are fixed and defined by the user. The ratios described earlier are consulted when calculating the skeleton lengths. However, the interior lengths such as the skeleton segment joining the head and the shoulders and belly to hips are flexible.

3.3 Terminology and Assumptions

First we define some terms used throughout the rest of the thesis. Each TVM is a sequence of 3D models, $M(t)$, where $t = 1, 2, ..., K$. Figure 2.7 shows nine frames of a *hip-hop* sequence. Each *frame* or 3D model is a set of vertices, $M(t) = \{ V_1, V_2, ...V_N \}$ where N is the total number of vertices in $M(t)$ and $V_N = (v_x, v_y, v_z)$. The kinetic model or skeleton is defined to have 16 skeleton parts or $Bone_{sp}$. In

our predefined model, $sp = \{$ *lhand*, *llowarm*, *lupparm*, *rhand*, *rlowarm*, *rupparm*, $lfeet$, *llowleg*, *luppleg*, $rfeet$, *rlowleg*, *ruppleg*, *head*, *chest*, *body* $\}$ or $sp = 1 \ldots 15$ respectively. Each *bone* is connected to other *bones* at *joints*, $J_{sp,1}$ and $J_{sp,2}$ each having 6 degrees of freedom. The *bone* has a length of length $|J_{sp,1}J_{sp,2}|$. Each *bone* is surrounded by a piece of surface or $Skin_{sp}$, In addition, hierarchical information is used in this thesis. The kinetic model has six top-level *subparts*, namely the limbs, head and body. Each *subpart* could contain one or more *subparts* or *bones*. In this paper, each subpart contains one or more *bones*. For example, $Subpart_{larm} = \{$ $bone_{lhand}, bone_{llowarm}, bone_{lupparm}$ $\}$.

The following assumptions regarding the skeletal structure of the human body are used:

1. Each *bone* is rigid.

2. Each *skin* is one continuous surface unless mesh intersection is detected.

3. Each *bone* lies within the *skin*.

3.4 Genus-n Models

Badly-defined models are defined as models that have surfaces that are not represented. Stereovision methods [2, 41] are still limited to the problems of having insufficient information of the scene to construct a well-defined model. As the TVMs used in this thesis are constructed via the visual hull, we are faced with the problem of surfaces that appear to be contiguous but are not, as described in section 2.2.1. If two legs are touching each other, distinct outlines will not represented in the visual

hull, thus the separability of legs would not be captured in the digitized 3D model. This results in unrepresented connected mesh between the legs.

Genus-n models are models which require n cuts to form plane graph that can be flattened into a place with only a single boundary. A cut a separates the mesh into two by removing edges thus creating boundaries. Examples of genus-0 and genus-1 models using simple geometric shapes are as follows: spheres are genus-0 whereas toruses are genus-1. In Figure 3.3 examples of a genus-0 and well-defined model, genus-1 and genus-1 and badly-defined models of the TVM data used are shown.

(a) Genus-0 (b) Genus-1 (c) Badly defined

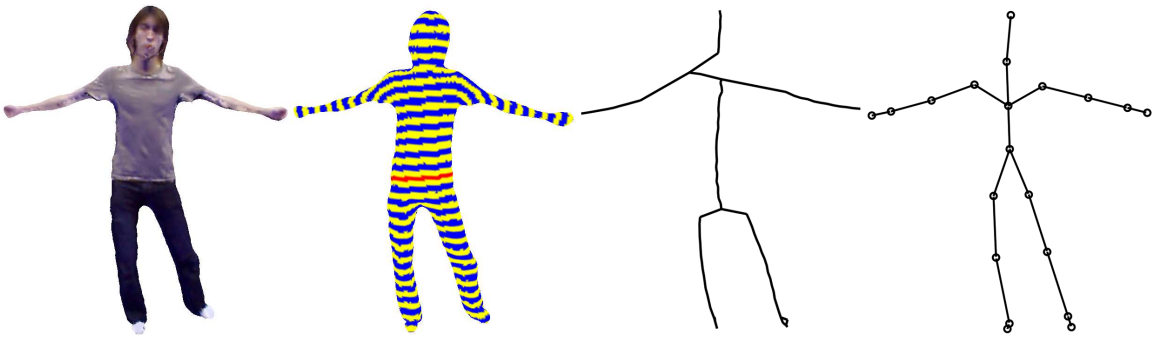
Figure 3.3: Examples of genus-0, genus-1 and badly-defined models.

Chapter 4

Hierarchical Skeleton-based Mesh Segmentation

Mesh decomposition or segmentation is a very important procedure as it is process of mesh simplification, and broadly used in pose estimation [17, 18]. This procedure establishes spatio-temporal correspondences, thus also extracting motion. As TVMs are sequences of low resolution models with no distinct structure and varying in form, we want to work with smaller meaningful components. Human perception is sophisticated, mesh segmentation using our eyes is visually simple but difficult to implement. 3D mesh decomposition is applicable and beneficial for various applications, such as compression, motion search and motion estimation. In research involving 3D models, segmentation is usually manual segmentation may be difficult as the models are complex. The following section describes the algorithm for mesh segmentation and pose estimation.

4.1 Skeleton Estimation and Initialization



(a) TVM frame (b) Geodesic distance (c) Reeb-graph skeleton (d) Structured skeleton

Figure 4.1: Skeleton initialization using geodesic distance-based skeleton extraction.

As an initialization step, we utilize Tadano et al.'s [35] geodesic distance based skeleton extraction algorithm to extract the skeletons of the initial frames. As the algorithm does not utilize any pre-defined structure, the resultant skeleton is an unstructured skeleton. In our algorithm we use the resultant unstructured skeleton to define our proposed skeleton structure as described. For Tadano et al. [35]'s algorithm, the geodesic distance from the central cross section of the model is calculated, as shown in Figure 4.1. Thereafter the center of mass for each band of mesh with the same geodesic distance is calculated (Figure 4.1b). The structured skeleton in Figure 4.1d is refined using the unstructured skeleton in Figure 4.1c and constraints as described by the *Vitruvian Man* in Section 3.2.

As seen in Figure 4.1, the algorithm is fairly accurate in extracting the skeleton in the model. However, the algorithm fails when the model is of genus-n or *badly-defined*, see Figure 4.2. This is because the extracted skeleton is highly correlated to

the mesh.

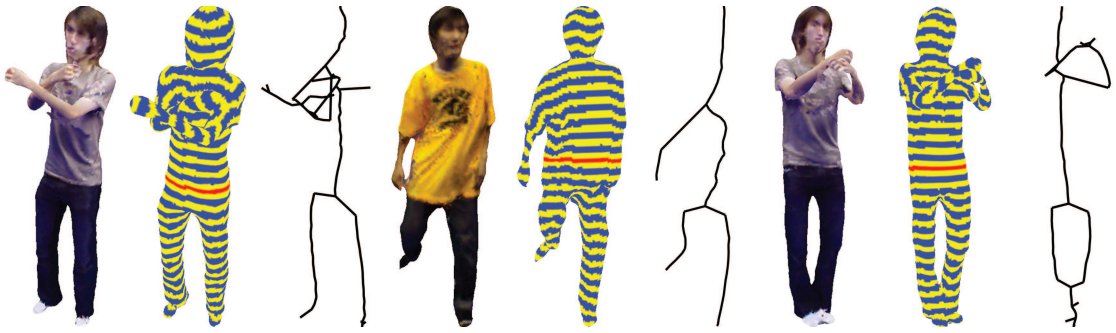


Figure 4.2: Examples of models which reeb-graph extraction fails to extract a good skeleton.

The skeletons of the first frame of a TVM is initialized with the use of Tadano et al.'s [35] skeleton extraction algorithm. For subsequent frames, various types of motion estimators are used instead as [35] fails for models with connected parts. In general, the pose from the previous frame is sufficient. However, if the movement of the subject is large, a non-linear Bezier motion estimator is used instead as shown in equation 4.1, where p_n corresponds to a parameter in frame n.

$$p_0 = f(0)$$

$$p_1 = f(1)$$

$$3(p_1 - p_0) = f'(0)$$

$$3(p_3 - p_2) = f'(1)$$

and rewritten as

$$f(t) = (1 - 3t + 3t^2 - t^3)p_0 + (3t - 6t^2 + 3t^3)p_1 + (3t^2 - 3t^3)p_2 + (t^3)p_3 \qquad (4.1)$$

The first and second derivatives of motion are taken into consideration at the end of the curve, therefore it retains C" continuity. With that, Bezier curves serve as a good estimator for motion. However, Bezier curves are convex hull problems, thus only four control points are used at each point of interpolation to prevent over-smoothing of human motion.

4.2 Hierarchical Skeleton-based Mesh Segmentation

In this section, the algorithm for hierarchical segmentation of a TVM frame is further elaborated. As described in the previous section, a predefined skeleton extracted using [35] is structured into a common kinetic model. The acquired model defines correspondence and the number of parts to be segmented. Hierarchical segmentation and distance correspondence are used to segment a 3D mesh into a piece-wise rigid structure. This method utilizes minimum distance calculation for computing correspondence between vertices and skeleton. The mesh is first segmented into a top-level hierarchy of six *subparts* and each subpart is decomposed further independently of the other subparts.

Skeleton-based distance segmentation consists of three steps are independently repeated for the segmentation of each subpart:

1. Calculation of distance correspondence between all vertices and all *bones*.

2. Assignment of each vertex to the nearest *subpart* or *bone*.

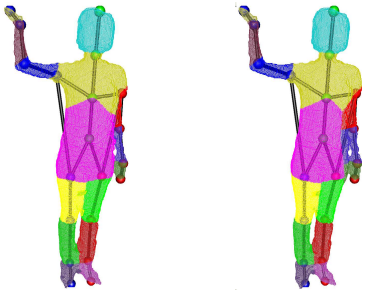
3. Filtering of wrongly assigned vertices and reassigning these vertices to the nearest neighbor.

4.2.1 Distance Computation

For each decomposition step, the distances from each vertex V_i to each skeleton part, $bone_{sp}$ is first computed. Distance, $d_{i,sp}$, is defined as the shortest inward distance from vertex V_i to $bone_{sp}$. The shortest distance is given by the perpendicular distance from vertex to bone. However, in the situation where the foot of the perpendicular line V_{perd} lies outside the bone, $J_{sp1}J_{sp2}$, the distance between the nearest J_{sp} and vertex V_i is computed instead. An inward ray is defined as the direction that is opposite to the vertex normal, $\vec{n}_{vi}$. This is based on the assumption that the skeleton lies within mesh cavity, thus we consider only inward rays to prevent mislabeling of vertices that are near but do not belong to *sp* as depicted in Figure 4.3. In this step, correspondence between the mesh and the skeleton is established.

$$d_{i,sp} =\mid V_i Bone_{sp} \mid= \begin{cases} \mid J_{sp,1}V_i \mid & \text{if } |J_{sp,1}V_{perd}| < |J_{sp,2}V_{perd}| \text{ and } |J_{sp,1}J_{sp,2}| < |J_{sp,2}V_{perd}| \\ \mid J_{sp,2}V_i \mid & \text{if } |J_{sp,2}V_{perd}| < |J_{sp,1}V_{perd}| \text{ and } |J_{sp,1}J_{sp,2}| < |J_{sp,1}V_{perd}| \\ \mid V_{perd}V_i \mid & \text{if } |J_{sp,1}V_{perd}| < |J_{sp,1}J_{sp2}| \text{ and } |J_{sp,2}V_{perd}| < |J_{sp,1}J_{sp2,}| \\ N.A. & \text{if } V_i\vec{Bone}_{sp} \cdot \vec{n}_{vi} > 0 \end{cases}$$

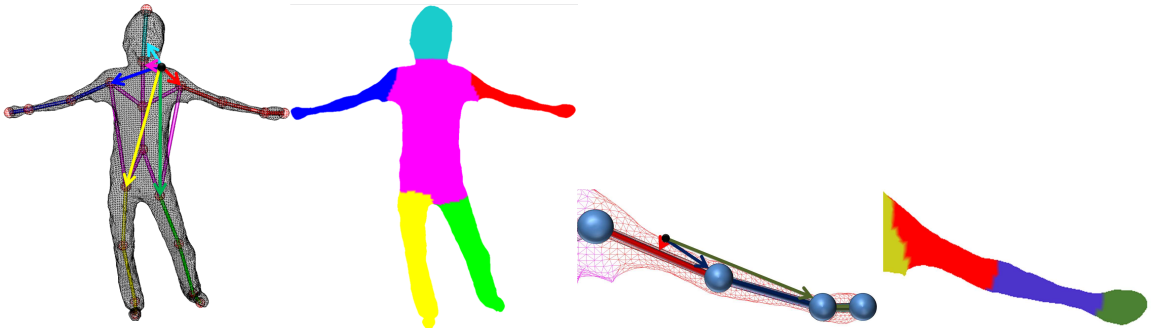
Each arrow in Figure 4.4a represents the distance between vertex and each *subpart*. The cyan arrow is pointing away from the mesh, therefore it is considered as an outward ray. This distance is ignored. As the pink arrow is the shortest ray from vertex, the vertex is to be labeled as the body. Similarly, Figure 4.4b illustrates the



(a) (b)

Figure 4.3: With and without consideration of inward rays

segmentation of a *subpart*.



(a) Top-level segmentation. (b) Independent second-level segmentation.

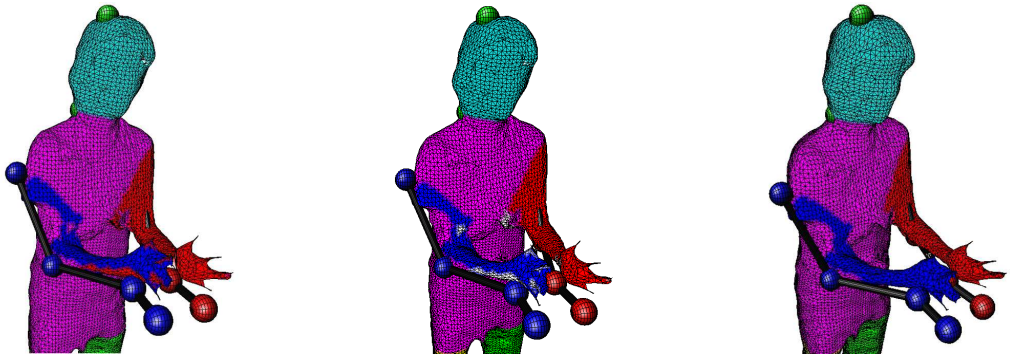
Figure 4.4: Hierarchical skeleton-based segmentation.

4.2.2 Labeling and Filtering

Labeling of vertices is the process of sorting each vertex V_i into segments. Each vertex, V_i is labeled to the part with the shortest inward ray, as shown in equation 4.2.

$$Label_i = sp \; where \; min(d_{i,sp}) \tag{4.2}$$

Each vertex is assigned irregardless of its neighbors, as such, some of the vertices could be mislabeled. Filtering is the process of identifying such vertices, as shown in Figure 4.5. With the assumption that each $skin_{sp}$ is a continuous mesh, only vertices in the largest patch is kept labeled as *sp* and the other vertices are unassigned, $Label_i = 0$. The unassigned points, shown in white in Figure 4.5b are then relabeled according to the *sp* of the neighboring vertices.



(a) Labeling vertices. (b) Filter big islands. (c) Reassigned to neighbor.

Figure 4.5: Labeling and filtering.

In this filtering process, we are able to identify patches that might not belong to the mesh. The edge vertices of a patch with unassigned vertices are first considered. For each edge vertex, we reassign the label of the vertex according to its neighbor. We encourage the uniformity of labels of the vertices in a triangular face, therefore as in case 1 (Figure 4.6a), the vertex is assigned to the most common label of surrounding patches. A face is labeled if two of the vertices have the same *sp*. If different labels have the same number of faces, the vertex is assign arbitrarily. A unique case is illustrated in Figure 4.6b where there is only one unique label. In case 2 (Figure 4.6c),

there are no labeled faces. Therefore we assign the most common label amongst the neighboring vertices. Case 2 is only used if no labeled triangular faces are found for all vertices in the patch of unassigned vertices.

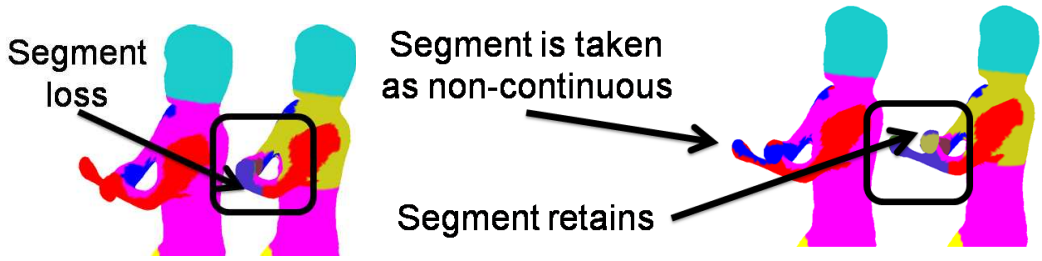
(a) Case 1. (b) Unique example. (c) Case 2.

Figure 4.6: Reassignment of vertices to its neighbors. Colored circles represent labeled vertices and white circles represent unassigned vertices. The vertex in the middle is relabeled as red.

4.2.3 Intersection Detection

As TVM models are not necessarily well-defined, a top-down approach for segmentation can ensure a cleaner decomposition. With hierarchy, it is possible to understand intersections of different parts. When intersections of two segments occur, the possibility that a segment is not a continuous mesh is considered, in contrast to the stated assumption in section 3.3. Independent decomposition of subparts also reduces errors resultant from a possible bad initial skeleton approximation. When intersections of two segments occur, it is identifiable by the labels of neighboring vertices. For example, the left arm is only connected to the body, therefore the neighbors of the left

arm should only be labeled as the body. In the case where the neighbors consists of labels other than body, like the right arm, we assume that intersection of the left and right arms has occurred. In another words, the left and right arms are joined by mesh therefore the mesh is *badly-defined*.



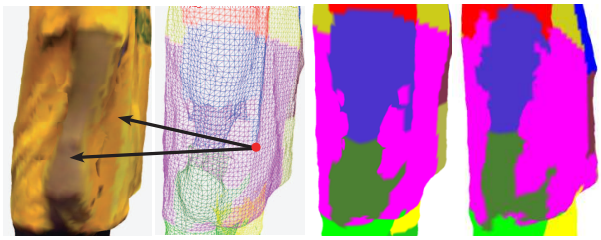
(a) (b)

Figure 4.7: (Left) Without intersection consideration. (Right) With intersection consideration.

With a good skeleton estimation, intersection detection is included to prevent loss of segments as shown in Figure 4.7a. To do so, two intersected regions are treated as one top-level region instead of two then decomposed into the desired parts. However, if the estimated skeleton is erroneous, we seize this opportunity to extract the skeletons of the two subparts that are touching each other. The approach is a derivation from Tadano et al.'s skeleton extraction algorithm using geodesic distances [35], see section 4.3.1.

4.2.4 Color-based Segmentation Refinement

The kinetic structure is a representation of pose and connectivity but do not contain any information on volume of shape of the surrounding mesh. The lack of constraints on the volume, makes it difficult to accurately estimate poses when the mesh is not well-defined over a long period of time. There is then a tendency for the movement of the skeleton to be undesirable. Meshes that are *badly-defined*, especially genus-0 meshes without proper definition of certain parts increases the volume of segmented parts, *sp*. It is difficult to be fully dependable solely on distance-based segmentation. In order to control the movement of the skeleton, or estimations of pose of each frame, color is used to more accurately segment the mesh by defining regions of connected mesh that can be restrained by color. However, it is only useful for defining regions with distinct color differences.



(a) Comparing color (b) Before (c) After

Figure 4.8: Color refinement.

Color is used to refine each iteration of the mesh decomposition. For color refinement, color histogram in the RGB domain is computed for each past decomposed segment *sp*. $Hist_{R,sp}$, $Hist_{G,sp}$, $Hist_{B,sp}$ for frames of time $1 \cdots (t-1)$ is calculated for each segment *sp*.

For growth and reduction of the segmented regions, only edge vertices, whose neighbor is wrong is considered. As a standard structure is given by the kinetic chain, it is possible to determine which neighbors are not legitimate. For example, from the structure, we understand that the left hand is connected to the left lower arm, therefore for vertices that are connected to the body or any part other than the left lower arm, it is an edge vertex that is potentially mislabeled. For these edge vertices, we compute the probability of the color from the histograms of *sp* from the past. By comparing the probabilities, the $label_i$ is changed to the segment which has the highest probability, higher than that of the original segment. Edge vertices are considered for mesh reduction and as for mesh growth, the neighbors of these edge vertices are considered instead. Though it could aid in refinement of the segmented *skin*, the use of color leads to jagged edges as seen in Figure 4.8c, it is used in the *hip-hop* sequence.

4.3 Skeleton Realignment

Using the estimated skeleton, we have identified regions on the mesh of the corresponding structure. As the initial skeleton is estimated, we realign the skeleton with respect to the newly decomposed model with the assumptions stated.

However, an erroneous initial skeleton estimate could lead to a sub-optimal segmentation. In this thesis, the proposed algorithm handles erroneous estimates to a certain extent. Several constraints are determined experimentally, are also considered for handling cases where the skeleton may be poorly predicted. As described earlier, skeleton lengths are fixed and adjusted to fit within the mesh cavity. The process of

segmentation and realignment is recursed as shown in Figure 4.9 to produce the best possible pose estimation of the frame.

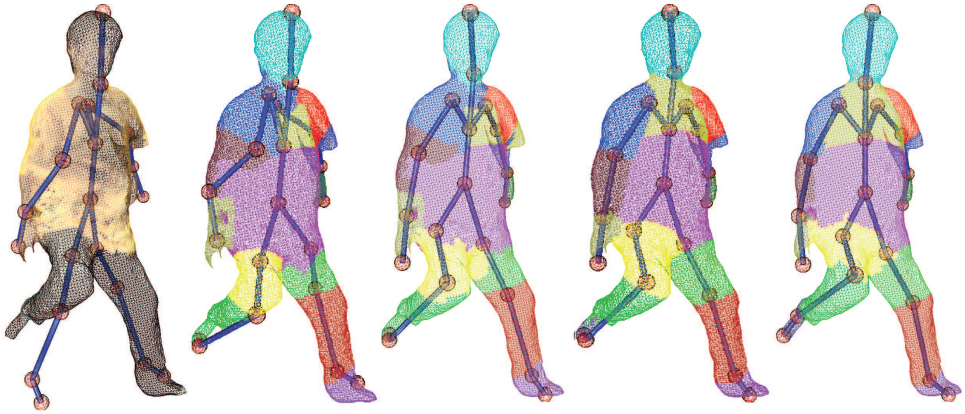


Figure 4.9: Recursive refinement of skeleton and segmentation.

The realignment is calculated as follows, for each segmented part, the center of mass and the center of mass of the edges form a vector used to realign the associated *bone*. Several heuristics are followed, for example, the tip of the limbs (*hands*, *feet* and *head*) is found by locating the point where the vector would cut the mesh nearest to the tip joint. In addition, we consider the restriction that the shoulder-to-chest lengths and hip-to-belly lengths should be approximately equal. Furthermore, the shoulder joints positions should not deviate too much from previous frames. This is to handle sub-optimal segmentations due to erroneous *bone* estimations. However for intersected mesh, there is a large possibility of bad segmentation so we use geodesic distances to extract the separate skeletons of the intersected parts. We detect intersected parts using the method described in section 4.2.3.

4.3.1 Skeleton Extraction for Intersected Regions

The proposed algorithm utilizes geodesic distance to extract the skeleton. However, geodesic distance is subjected to the errors of the surface, therefore we need to consider intersections and structural properties. Unlike Tadano et al. [35], skeleton of each subpart is extracted separately.

1. Calculate geodesic distance of all vertices in segmented part from edge vertices (Red ring in the left of Figure 4.11) and quantize into bands.

2. Extract skeleton by calculating the center of mass for each band. (Right of Figure 4.11)

3. Separate joined skeleton portions by seeking the branch which is similar in direction as the previous segment. (Left of Figure 4.12)

4. Retrieve the longest path from start till end of the skeleton.

5. Merge skeleton segments that are similar and estimate skeleton of fixed structure. (Right of Figure 4.12)

Geodesic distance is a property that enables us to more accurately calculate the relative distance between two points on the mesh model. It is defined as the shortest surface distance between the points. Unlike euclidean distance, geodesic distance differentiates two points that have small euclidean distance but are actually far away if you were to trace the patch from one point to the other along the mesh. This property is missing when intersection occurs, so we consider both parts that are joined together at the same time.

In Figure 4.10, geodesic distance between P and Q, $Geodist(P, Q)$ is indicated by the red line and euclidean distance, $Dist(P, Q)$ by the blue line. To calculate geodesic distance, the weight between two adjacent vertices is defined by the euclidean distance along the edge and geodesic distance between two points is the shortest path along the edges connecting these points. If the two points are not connected by any path, or the two vertices belong to two patches, the geodesic distance is infinite.

$$Dist(P, Q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2)} \tag{4.3}$$

$$Weight(P, P_{adj}) = Dist(P, P_{adj}) \tag{4.4}$$

$$Geodist(P, Q) = min(\sum Weight(P_{between}, Q_{between})) \tag{4.5}$$

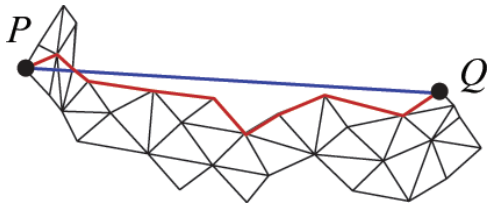


Figure 4.10: Geodesic distance and euclidean distance.

The calculated distances are then quantized into bands. The center of masses of each band is calculated. If the band can be separated into two of more unconnected patches, the center of mass for each patch is calculated. Then, the center of masses are joined to form a unstructured skeleton.

Once the skeletons are separated accordingly, segmentation is done as described in the earlier section. Results are shown in Figure 4.13. In the dataset, there are other types of intersections between the subparts. The skeleton extraction results of

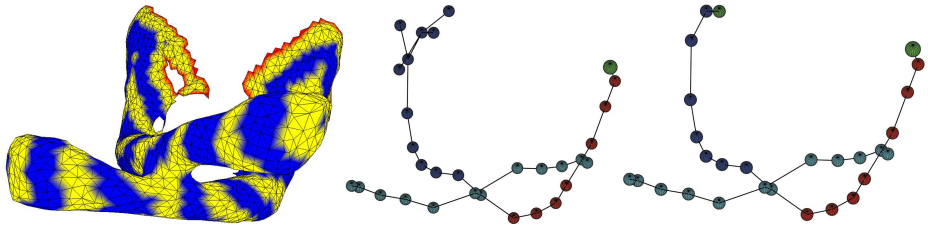


Figure 4.11: Extraction of unstructured skeleton from joined mesh.

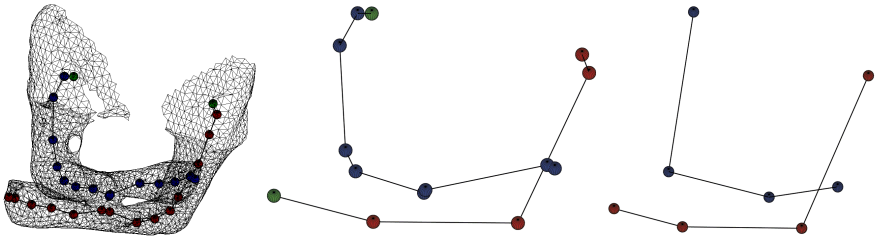


Figure 4.12: Separation of joined skeleton and estimation of structured skeleton.

various intersections such as left with right arm, left with right leg, arm with head and arm with leg are shown in Figure 4.14.

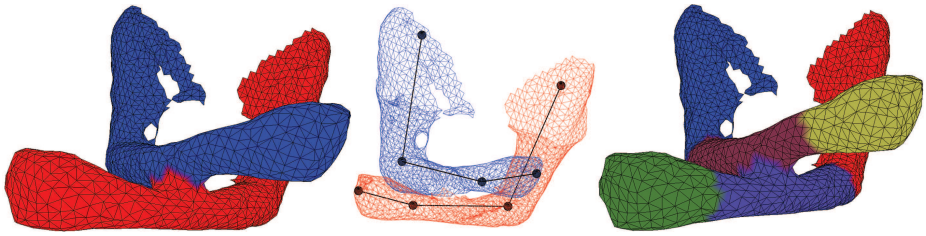


Figure 4.13: Segmentation of joined mesh using estimated skeleton.

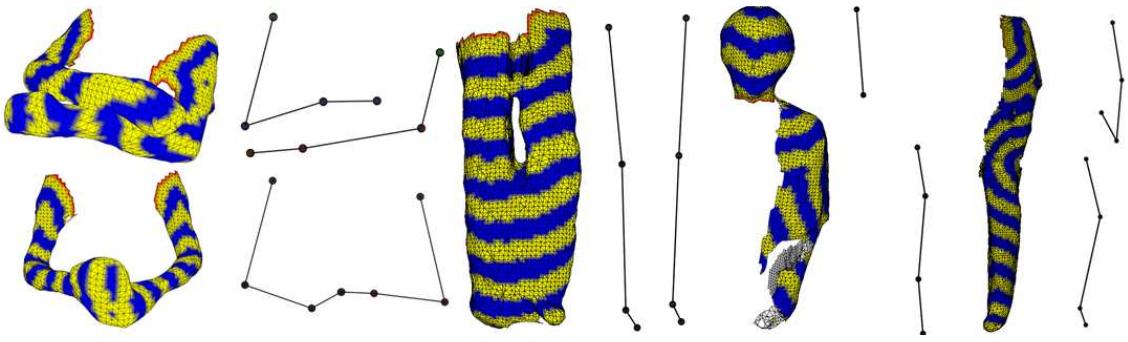


Figure 4.14: Refinement of skeleton and segmentation.

Chapter 5

Experiments and Results for Mesh Segmentation and Motion Transfer

5.1 Experimental Setup

In this thesis, the models used are constructed from 22 IEEE1394 cameras with XGA resolution (1024 × 768 pixels) and are constructed using Tomiyama et al.'s [2] implementation. We have several TVMs capturing motion of high dynamics such as *hip-hop*, *exercise*, *japanese dance*, *announcer* and *running* sequences. No restrictions and no markers were placed on the clothes. Currently, our TVMs are 10 frames per second.

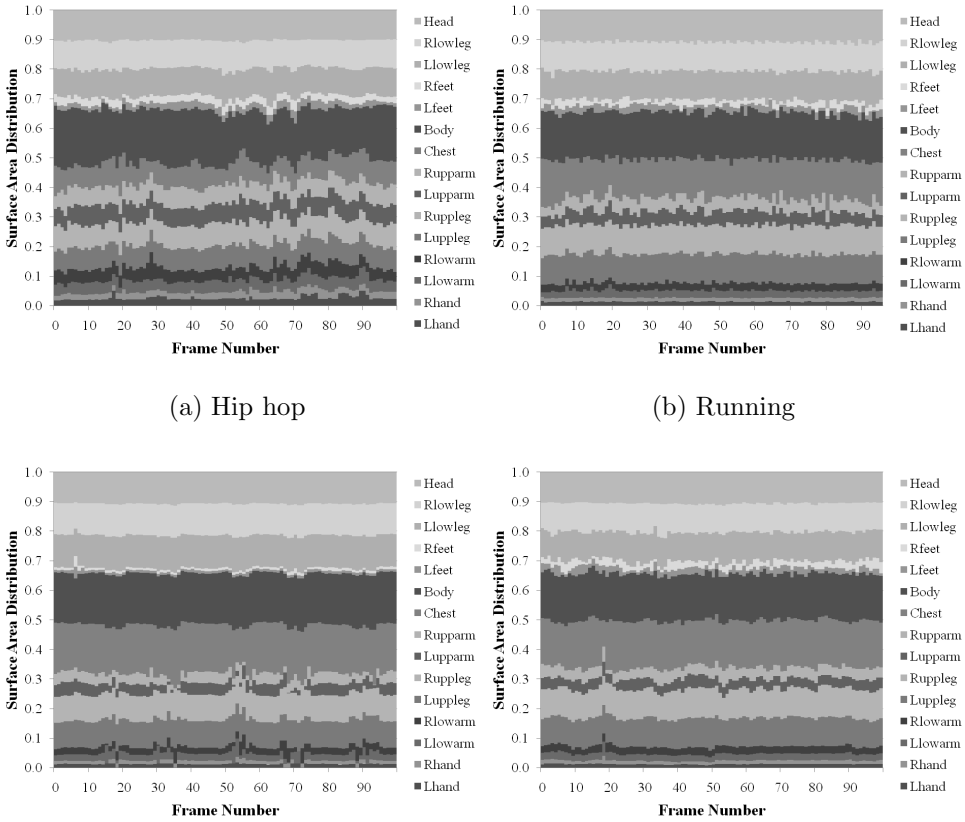
For experimental purposes, we used sequences *hip-hop*, *running*, *arm swing* and *exercise* for evaluation purposes. For each sequence, approximately 100 frames are segmented and tests are conducted. Table 5.1 shows the properties of the TVMs to be evaluated. Sequence *hip-hop* has high motion dynamics and complex surface

animation resulting from clothes worn by the subject. On the other hand, sequences *exercise*, *arm swing* and *running* have complex surface intersections and motion, but less complex cloth animation. In addition, sequences *exercise*, *arm swing* and *running* are of the same subject. As we do not have motion capture data or ground truth of

Table 5.1: Properties of TVM data

Sequence (No. of frames)	Genus-0	Genus-n	*Badly-defined*	Total
Exercise	78	22	9	100
Running	93	3	26	96
Arm Swing	98	2	39	100
Hip-hop	55	45	48	100

the motion, it is difficult to compare the accuracy of our results objectively. Presently, there is no standardized methodology for evaluating segmentation of TVMs. Therefore we consider other factors such as surface area distribution. In this thesis, the stability of our algorithm is evaluated by calculating the ratio of surface area of each segmented part to the total surface area of each TVM frame. Ratio is used in order to normalize the results for a fair comparison across frames as the surface area of each TVM frame in different sequences varies. Similar to other research [32, 34], we compare the surface area distribution of segments between different sequences of the same subject to evaluate the accuracy of our algorithm.



(a) Hip hop

(b) Running

(c) Arm Swing

(d) Exercise

Figure 5.1: Surface area distribution per frame for segmentation results.

5.2 Results

Figures 5.1a, 5.1b, 5.1c and 5.1d show the surface area distribution of each segmented frame of sequences *hip-hop*, *running*, *arm swing* and *exercise* respectively. Distinct bands displayed in the figures, especially for head, hands, lower arms and lower legs, shows stability in the proposed algorithm. The irregularities, for example in frames 18-20 of Figure 5.1d are due to the intersection of legs and that of hands. Intersections cause discrepancies in the surface area as only a portion of the actual surface area is available. Such discrepancies are good indicators of models which are badly-defined,

where pose estimation was difficult. For *exercise*, *arm swing* and *running* sequences, color-based refinement was not used during realignment. Whereas, it was used for *hip-hop* as models are highly deformable and more variation in movement of surface. Color-based refinement is necessary in models where the mesh of a segment completely overlaps with another. However, color-based refinement results in jagged edges, which are not visually pleasing.

From the figures in Figure 5.1, we observe that the head has a very stable segmentation and in Figure 5.2, the actual surface area of the head of *hip-hop*, *running*, *arm swing* and *exercise* are shown. Sequences *running*, *arm swing* and *exercise* capture the same subject. For the *running* sequence, a fluctuation of the surface area can be observed. The wave follow a frequency of about three frames per cycle. Interestingly, it corresponds to the frequency at which the subject lifts his leg while running. A similar fluctuation can be observed in sequence *exercise* in frames 61-80. The window corresponds to the period where the subject was jogging for two seconds in the sequence. As it can be seen, the upper and lower bounds of this window matches that of the *running* sequence. The results from different sequences of the same subject having average surface area difference of 40 cm^2, which is 0.2% of the total surface area indicate the proposed algorithm is accurate. The surface area of the segmented head of *hip-hop* is less than that of the other sequences reaffirms that the subject is different.

Figure 5.7 shows the visual results for segmentation of four different sequences, along with the pose estimated model. In this figure, we have *badly-defined*, genus-1 and genus-2 models. We have shown that our proposed algorithm is able to extract structural information from models that are difficult to attain if geodesic distance is

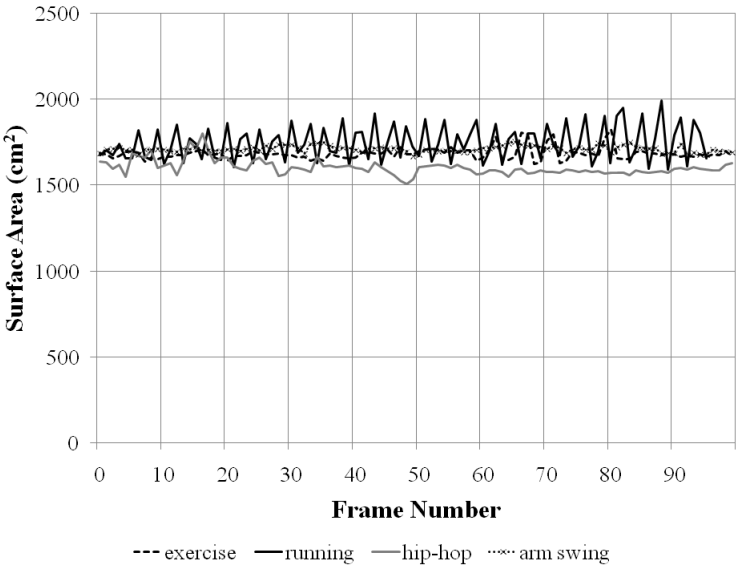


Figure 5.2: Surface area of head in sequences.

used.

As we estimate pose of each frame, we are able to transfer the pose or motion for the entire sequence into other models. Figure 5.3 shows the results of motion transfer from TVM sequences to a synthetic model, a random model used in Poser, while Figures 5.4, 5.5 and 5.6 shows a time span of tracking results of sequences *hip-hop*, *exercise* and *announcer* respectively.

5.3 Limitations

The proposed algorithm is good for estimating pose independently as each frame can be regarded independently with a reasonably good estimation. However, there are still limitations of the algorithm with respect to the initialization or estimation of the skeleton structure. We need a estimated kinetic model for segmentation. Though

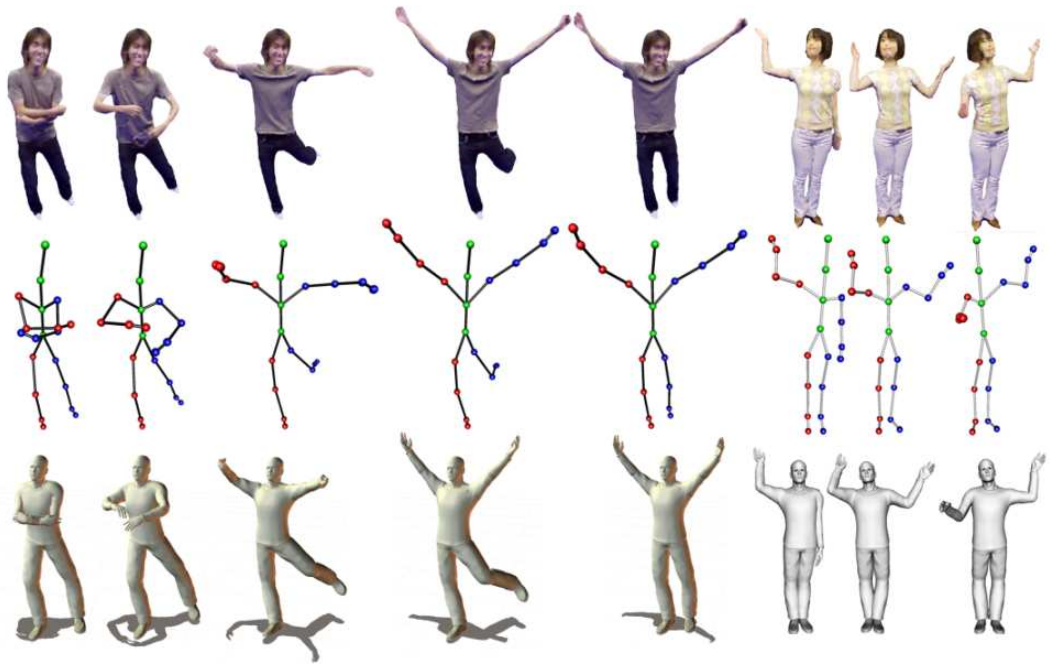


Figure 5.3: Motion transfer results from *exercise* and *announcer*.

we have heuristics to realign the erroneous skeletons, there are instances where it is difficult to segment and estimate poses. Such examples occur when two *bones* of the same *subpart* are connected, for example bent knees or elbows having small degree of separation. Also, if the mesh is *badly-defined* for an extended number of frames, the algorithm causes the skeleton to drift. These problems are present as there is not enough surface information for skeleton fitting.

In particular, it is especially difficult for automatic segmentation of frames with largely intersected meshes. Experiments do show that if the mesh is connected for a brief moment, with the proposed motion interpolation methodology it is possible to recover the pose and continue tracking for subsequent frames. However, for situations where the meshes are ill-defined for a long sequence, the estimated skeleton is unable to accurately estimate the pose and recover.

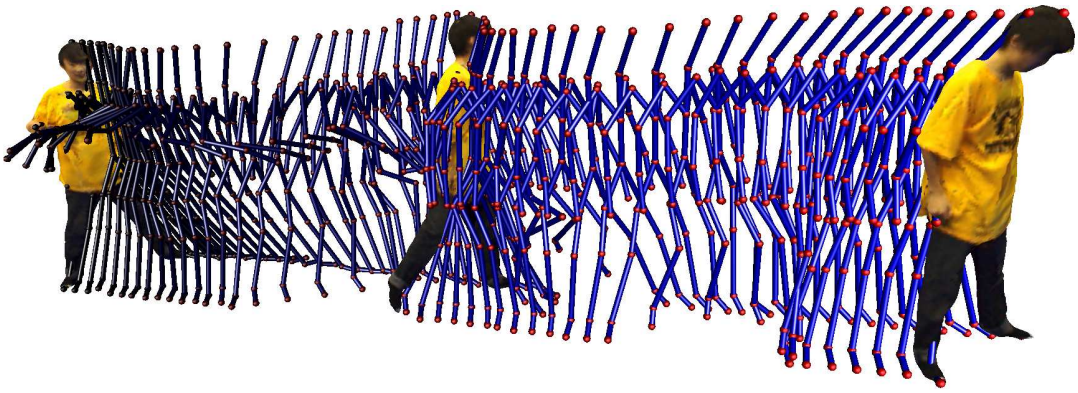


Figure 5.4: Tracking results from *hip-hop*.

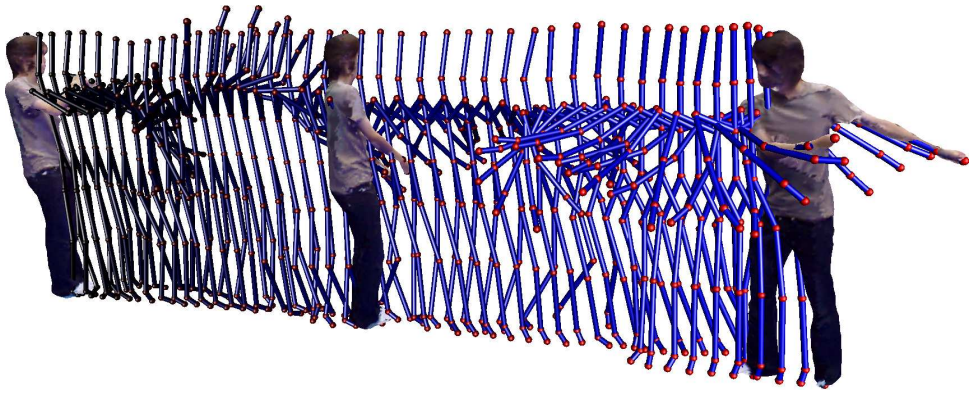


Figure 5.5: Segmentation results from *exercise*.

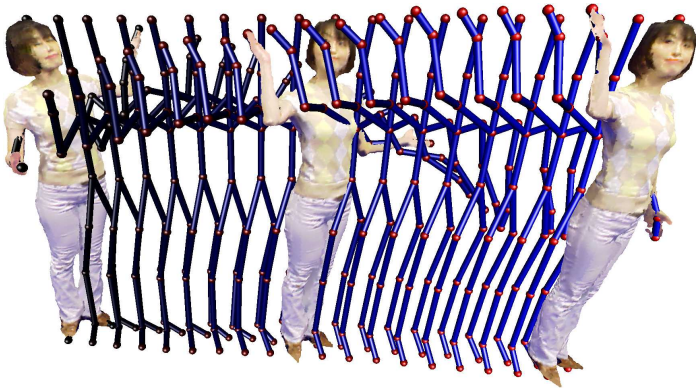


Figure 5.6: Segmentation results from *announcer*.

Figure 5.7: Segmentation results from *hip-hop*, *exercise*, *announcer* and *Japanese dance*.

Chapter 6

Surface Curvature Matching

Although the algorithm is able to track poses of humans reasonably well for most of the frames, the extracted information is still insufficient for motion capture. Only two degrees of freedom of rotation are recovered though the estimation of kinetic poses. Therefore, we explore the use of the segmented parts for finer surface correspondences and rotation parameters. We attained a piece-wise rigid model from the previous part of this thesis. Motion extraction through distance based segmentation with the use of a kinetic model is sufficient for motion search applications where a rough estimate of the pose is sufficient. But surface matching gives us more understanding of the surface changes or lost and surface correspondences.

However, the commonly used algorithm for 3D surface registration [24, 25], Iterated Closet Point (ICP) [23] matching is computationally expensive and sensitive to outliers. Since our data is highly deformable, first we segment the mesh into piece-wise rigid components. However, it is still insufficient as the surfaces are deformable. Local parameters such as surface curvatures are used for matching using the classical

pair-wise ICP matching algorithm. The most challenging issue in our algorithm is the changing topology of a non-parametric motion model. Therefore, our algorithm uses previous and following time frames other than the immediate ones for registration for robustness against bad registration.

As the models are dense point clouds, a modified ICP with distance calculation is used in the algorithm. Before surface curvature matching, each frame is segmented into desired rigid parts as described in Section 4. Each segmented body part has two motion vectors or six degrees of freedom (x, y, z, θ , ϕ, γ) and are resolved by surface curvature matching used along with a "virtual marker" to resolve the last degree of freedom. Matching with normal weighted Euclidean distance calculation is used in the algorithm.

6.1 Matching Feature Filter

As we are dealing with a large number of points, there is a possibility of noise and low processing speed. In order to increase accuracy and speed, we first filter the mesh to obtain feature vertices with large curvatures. These vertices of ridges or obtrusions serve as good matching features. Surface curvature defined at vertex i, c_i, is calculated by average of the dot product of vertex i and its immediate neighbors, where n_i is the vertex normal at vertex i, K_i is the number of neighboring vertices at i and $j = 1...K_i$.

$$c_i = \sum_{j=1}^{K_i} \frac{n_i \cdot n_{i,j}}{K_i} \tag{6.1}$$

Surface curvature at vertex i, c_i, ranges from -1 to 1 and c_i of value close to 1 represent vertices of smooth surfaces. Large surface deviations or near-cones are represented by

low c_i values. To reduce the number of vertices used for matching, the lower values of c_i are selected. By selecting the lower 20 percentile, we reduce the number of vertices used for matching by 80%. Figure 6.1 shows three models with 5%, 10% and 20% of its vertices filtered and are shown in red. The vertices with the lower 5, 10 and 20 percentile of c values are selected. The range of c values are only kept constant across segments that are to be matched.

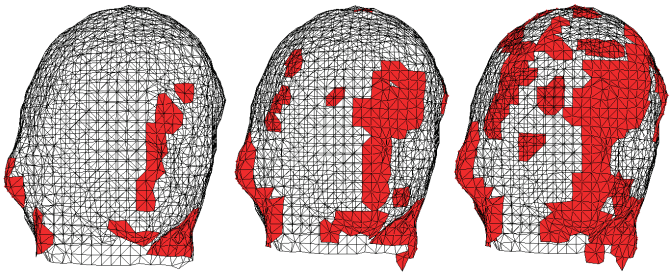


Figure 6.1: Three models with 5%, 10% and 20% of its vertices filtered.

6.2 Normal Weighted Euclidean Error

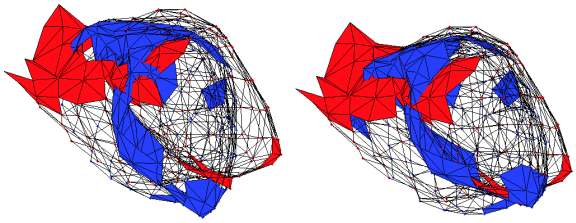


Figure 6.2: (Left) Matching with euclidean distance error. (Right) Matching with normal weighted euclidean distance error.

The motion vectors are resolved using a modified ICP registration algorithm. As

compared to Besl et al. [23]'s algorithm, we find the corresponding vertices between meshes before evaluating every possible iteration. This is done to determine a better estimation though at the cost of computation. Termination criterion is defined at the point where disparity between two segments is stabilized. Disparity between two segments is typically calculated as the sum of minimum Euclidean distance between vertices (Equation 6.2) of one segment to the other corresponding segment given by the following, where $i = 1 \cdots K_n$, $j = 1 \cdots K_{n+1}$ and K_n and K_{n+1} are the number of vertices in segment of frame n and $n+1$, $skin_{sp,n}$ and $skin_{sp,n+1}$ respectively.

$$error = \sum min(\mid p_i - p_j \mid) \tag{6.2}$$

However, due to imperfections in segmentation, we need to consider loss of vertices. Distance only calculations could lead to a possibility of mismatch of surface curvatures as shown in Figure 6.2 (left). In this paper, we introduce a different error calculation method where vertex normals are considered applied in our modified matching algorithm. Error is given by an offset of the dot product of normals of the closest vertices. Normals are taken into consideration as the surface is described through the direction of vertex normals.

$$error = \sum (1 - n_i \cdot n_j) min(\mid p_i - p_j \mid) \tag{6.3}$$

Figure 6.2 (right) shows a better match when vertex normals are considered. In equation 6.3, error is reduced when vertices have similar normals but penalized when normals are of different directions.

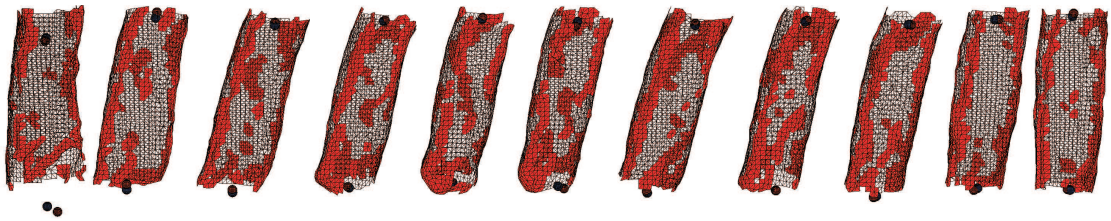


Figure 6.3: Ten frames of segmented lower leg with 20% of the vertices filtered and refined markers using multi-temporal registration.

6.3 Multi-Temporal Registration

As ICP is sensitive to outliers and deformable surfaces, there is possibility of bad registrations. Pair-wise registrations are commonly used [17], however, in this paper we consider multi-temporal registrations. Pekelny et al. [17] uses heuristics of setting thresholds to detect outliers when searching for corresponding pairs of points while computing disparity between segments. Our algorithm does not impose constraints on the matching process. Instead, temporal information is used to gain robustness against bad registrations. Frame n is registered across a number of previous and future time steps, frame $n-t$ to frame $n+t$. By taking the previous and following frames into consideration, the effect of a bad registration with the immediate neighbor can be reduced. Three markers are tracked for each segments, thus tracking three degrees of freedom of rotation. J_{sp1} and J_{sp2} are defined as two of the markers and a third "virtual marker", J_{sp3} arbitrarily defined at frame 1. Each J_{sp} of frame n is determined arbitrarily by a non-linear weighted average of the corresponding J_{sp} in each registered time steps. This is given by,

$$J_{sp,n} = \sum_{k=-t+1}^{t-1} (\frac{1}{2^{|k|+1}} J_{sp,n+k}) + \frac{1}{2^t}(J_{sp,n-t} + J_{sp,n+t}) \tag{6.4}$$

As time, k, increases, introduction of error due to deformation of surface is present, therefore non-linear weights are used. The refinement of the "virtual markers" are shown in Figure 6.3.

Chapter 7

Experiments for Surface Curvature Matching

7.1 Experiment Setup

Tracking was done for the *head* where the ground truth could be easily determined. This is achieved by estimating the position of a "virtual marker" in the next frame and motion is resolved through our proposed algorithm. Dataset used in this experiment is the sequence, *exercise* where the subject turns 360 degrees. So for 90 frames, the head rotates 360 degrees over 40 frames and the ground truth (position of right ear) is marked by hand. Rotation is calculated as the angle between corresponding "markers" in aligned consecutive frames. Tracking is achieved by estimating the position of a virtual marker in the next frame from the previous frame. For these experiments, only the first frame is initialized and the "markers" are tracked for 90 frames. Two angles were calculated, one, referred as *actual* angle, is the difference of angle between first

frame and current frame. This is to evaluate the overall visual impact of our tracking system. The other, *relative* angle, is the change of angle between consecutive frames. Errors while calculating the *actual* angle could be cumulative, however, the *relative* angle is error-independent and allows us to understand accuracy of our algorithm.

7.2 Results

First, we evaluate our proposed matching feature filter. We compared the results of our proposed feature selection and that of randomly selected vertices of equivalent size for one time step.

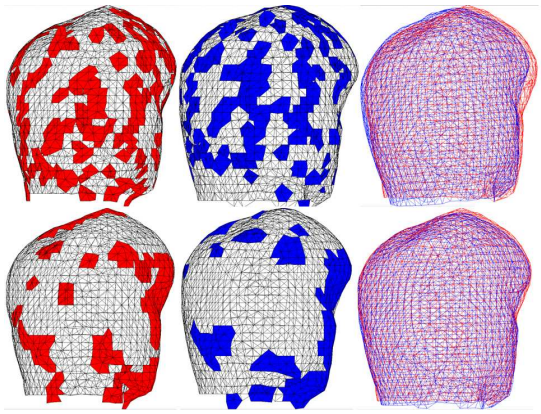


Figure 7.1: (Top) Random vertices selected and (bottom) large surface curvature vertices selected for matching.

Table 7.1 shows the times taken for one ICP iteration for 5%, 10%, 20% and 100% of the original segment. As the number of vertices used for matching is reduced, the times taken for ICP reduces in proportion to N^2. In order to evaluate and justify the effectiveness of use of large gradient vertices for time reduction, we compared the

Table 7.1: Time Comparisons

Percentile	Average number of vertices	Time taken (sec)
5	72	1.5
10	145	3.3
20	295	8.5
100	1495	92.1

results of our proposed feature selection and that of randomly selected vertices of equivalent size. Corresponding vertices are defined as the vertices in the next frame with the shortest Euclidean distance. The corresponding vertices of the randomly selected vertices are used for calculations of the minimal distance. In Figure 7.1, we see the setup where the top models have 10% randomly selected vertices, while the bottom models, show vertices selected through our algorithm. The matched figures on the right shows that our proposed algorithm yield better "visual" results. Figure 7.2

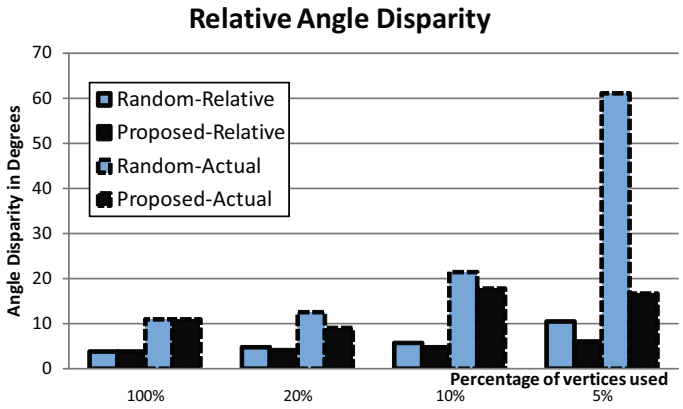


Figure 7.2: Average angle disparity between random selection and proposed surface gradient filter.

shows the average angle disparities for *actual* and *relative* angles. The results show a low average angle disparity even when the number of vertices is reduced to 5%. This shows that our matching features filter is fast and accurate.

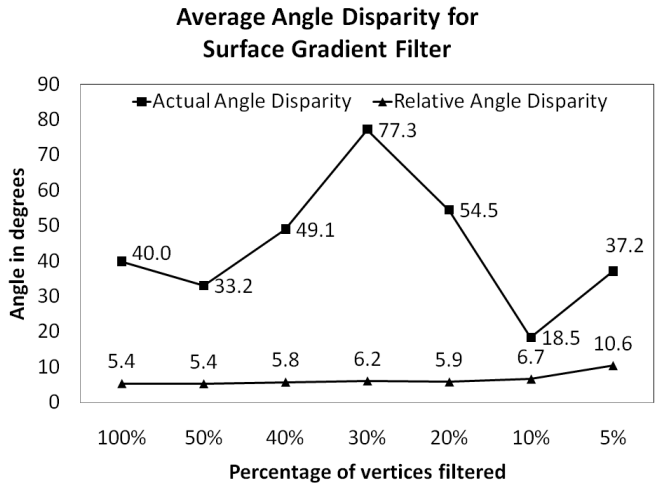


Figure 7.3: Average angle disparity results of matching for 100%, 50%, 40%, 30%, 20%, 10% and 5% selection of vertices.

Figure 7.3 shows the results of comparing the use of our proposed algorithm for 50%, 40%, 30%, 20%, 10% and 5% of the total number of vertices. Results show that a reduction of 50% of vertices still yields the same accuracy rate with better visual results. Accuracy of the pair-wise matching between frames is kept, for up to a reduction of 80% of the vertices used. However, this experiment also shows that with too many vertices selected, there are noise present, whereas when number of vertices selected is too small, there is not enough information for matching.

Experiments with 100%, 20%, 10% and 5% and multi-temporal registration for $k = 1...10$ were conducted and results consolidated in Figure 7.4. The dotted and

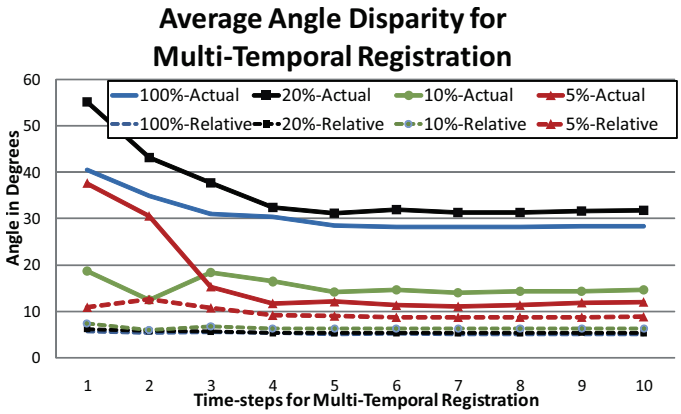


Figure 7.4: Average *relative* and *actual* angle disparity for multi-temporal registrations.

solid lines represent averaged *relative* and *actual* angle disparities respectively. From the plot, it is notable that with multi-temporal registrations, average angle disparity decreases as k increases though it stablises. An average of 29% reduction is achieved by two additional time steps, however, there is not much difference when the matching is good. In addition, our algorithm is able to achieve similar results when feature points used are reduced by 80%. Interestingly, between 20% and 10% use of vertices, there is a significant drop in angle disparity. Though 100% and 20% use are very stable when the segment is not rotating, the accumulation of error when the segment rotates 360 degrees is more significant than when 10% and 5% of the vertices are used. Error is due to the deformation of the segment.

In Figure 7.5, motion compensated frame 1 and frame 50 and 90 are shown. The visual results show the accumulation of error when 20% and 100% of the vertices are used. From the results we see a compromise between using more vertices for a more stable but erroneous result.

In Figure 7.7 shows the filtering results of other body parts, the lower leg and

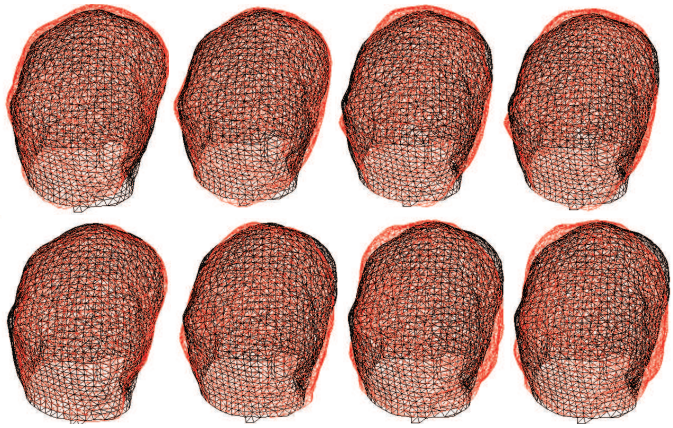


Figure 7.5: (Left) Motion compensated frame 1 (black) and frame 50 (red) for use of 5%, 10%, 20% and 100% vertices. (Right) Motion compensated frame 1 (black) and frame 90 (red) for use of 5%, 10%, 20% and 100% vertices.

hand. Figure 7.6 compares the results of addition of rotation parameter found through matching. Figure 7.8 shows the visual results of motion transfer for the head for 20% vertices with time step = 1, 20% filtered vertices with time step = 5, 10% vertices with time step = 1 and 10% filtered vertices with time step = 2. As the results show, there is an accumulation of error after a 360 degree rotation where there is large deformation in the head. However with multi-temporal registration, it generally yields better results.

(a) Without Matching (b) With Matching

Figure 7.6: Results of with and without matching for sequence *announcer*.

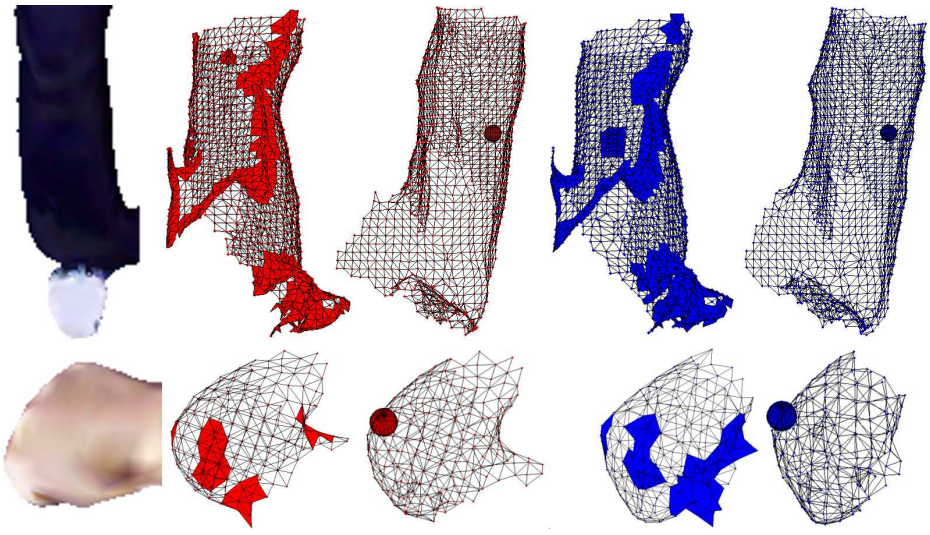


Figure 7.7: (From left) Original model with color information, filtered vertices with marker for frames 1 and 2. Row 1 and 2 show results from right lower leg and left hand.

Figure 7.8: (From left column) Original model with color information, 20% vertices with time step = 1, 20% filtered vertices with time step = 5, 10% vertices with time step = 1 and 10% filtered vertices with time step = 2.

Chapter 8

Conclusions and Future Work

A segmentation and motion tracking approach from structural and temporal analysis of Time-Varying-Meshes has been proposed in this thesis. The thesis consists of two main parts; first, pose estimation of each TVM frame using hierarchical mesh segmentation and skeleton realignment, and second, refinement for surface correspondence between the segmented parts. The framework in the first part consists of three steps, initialization, followed by hierarchical mesh segmentation that utilizes distance calculation and skeleton realignment according to the structural information attained. The segmentation algorithm is evaluated to be stable for a long span of sequences and able to analyze the structure of badly-defined models given a reasonable estimation of the pose. *Markerless* motion tracking is possible as the structure is kept constant for each frame. When we compare between two sequences with the same subject, the average surface area difference is 40 cm^2, which is 0.2% of the total surface area.

We also presented an iterated closet point matching algorithm with use of vertex normals to resolve motion. Results show 80% higher speed with equivalent accuracy

is achieved with the use of 20% of vertices having large surface gradients filtered as matching features. With multi-temporal registration, the effect of bad registration can be reduced to an average of 29% for two additional time steps. However, the benefits using multi-temporal registration are less significant when the matching results are good. Experiments show good tracking results with an angle of four degree difference on average from the ground truth with only 20% use of the vertices.

In this thesis, the skeleton structure is pre-defined and does not adjust automatically. For future work, flexible lengths could be calculated from the structure that is obtained. Color could also be included in the calculation of correspondence between the predefined structure and the mesh. Given a good estimate of the pose, the algorithm is able to segment *badly-defined* and models of arbitrary genus. However, the accuracy and stability of the algorithm is affected by the estimation of the skeleton. Also, limitations include sequences that contain continuous sequences of *badly-defined* models.

Currently, the matching algorithm is used to extract rotation about bones for each segmented part. It is limited by the deformations present in the mesh. For future work, we can consider quantizing the local parameters, geodesic distance and surface curvatures for more accurate results. Matching could also be used to refine segments that have incomplete surface representations.

Bibliography

[1] Carnegie mellon motion capture database. http://mocap.cs.cmu.edu.

[2] K. Tomiyama, Y. Orihara, M. Katayama, and Y. Iwadate. Algorithm for dynamic 3d object generation from multi–viewpoint images. In *Proceedings of SPIE*, volume 5599, pages 153–161. SPIE, 2004.

[3] C. Hernandez and F. Schmitt. Silhouette and stereo fusion for 3d object modeling. In *Computer Vision and Image Understanding*, volume 96, pages 367–392, 2004.

[4] L.Y. Chang, N.S. Pollard, T.M. Mitchell, and E.P. Xing. Feature selection for grasp recognition from optical markers. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 2944–2950, 29 2007-Nov. 2 2007.

[5] M Izani, Aishah, A Eshaq, and Norzaiha. Analysis of the keyframe animation and motion capture case studies. In *Research and Development, 2003. SCORED 2003. Proceedings. Student Conference on*, pages 177–182, Aug 2003.

[6] S. Park and J. Hodgins. Capturing and animating skin deformation in human motion. In *ACM Transaction on Graphics SIGGRAPH*, volume 25, pages 881–889, July 2006.

[7] B. Francois. Review of 20 years of range sensor development. *Journal of Electronic Imaging*, 13(1):231–240, Jan 2004.

[8] A. Banno and K. Ikeuchi. Shape recovery of 3d data obtained from a moving range sensor by using image sequences. In *Tenth IEEE International Conference on Computer Vision (ICCV'05)*, volume 1, pages 792–799, 2005.

[9] M. Tamai, W. Wu, K. Nahrstedt, and K. Yasumoto. A view control interface for 3d tele-immersive environments. In *IEEE International Conference on Multimedia and Expo*, pages 1101–1104. IEEE, June 2008.

[10] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *IEEE 7th International Conference on Computer Vision ICCV.*, pages 722–729, September 1999.

[11] J. Starck and A. Hilton. Correspondence labelling for wide-timeframe free-form surface matching. In *IEEE 11th International Conference on Computer Vision ICCV.*, pages 1–8, 2007.

[12] J. Starck and A. Hilton. Spherical matching for temporal correspondence of non-rigid surfaces. In *IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1387–1394. IEEE, 2005.

[13] D. Steiner and A. Fischer. Cutting 3d freeform objects with genus-n into single boundary surfaces using topological graphs. In *7th ACM Symposium on Solid Modeling and Applications*, pages 336–343, 2002.

[14] D. Tamal, K. Li, J. Sun, and D. Cohen-Steiner. Computing geometry-aware handle and tunnel loops in 3d models. In *ACM Transaction on Graphics SIGGRAPH*, volume 27, 2008.

[15] T. Yamasaki, Y. Hamazaki, and K. Aizawa. Interactive refinement and editing for time-varying mesh. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pages 285–288, 2008.

[16] C. Theobalt, E. Aguiar, M. Magnor, H. Theisel, and H. Seidel. Marker-free kinematic skeleton estimation from sequences of volume data. In *ACM Symposium on Virtual Reality Software and Technology (ACM VRST)*, 2004.

[17] Y. Pekelny and C. Gotsman. Articulated object reconstruction and markerless motion capture from depth video. In *Eurographics Computer Graphics Forum*, volume 27, 2008.

[18] S. Katz and A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. In *ACM Transactions on Graphics SIGGRAPH*, volume 22, pages 954–961, 2003.

[19] S. Knoop, S. Vacek, and R. Dillmann. Sensor fusion for 3d human body tracking an articulated 3d body model. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1686–1691, 2000.

[20] N. Werghi. Segmentation and modeling of full human body shape from 3-d scan data: A survey. In *IEEE Transactions on Systems, Man and Cybernetics*, 2007.

[21] P. Sand, L. McMillan, and J. Popovic. Continuous capture of skin deformation. In *ACM Transactions on Graphics SIGGRAPH*, volume 22, pages 578–586, 2003.

[22] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *European Conference on Computer Vision*, 2002.

[23] P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, February 1992.

[24] T. Zinsser, J. Schmidt, and H. Niemann. A refined icp algorithm for robust 3-d correspondence estimation. In *International Conference on Image Processing ICIP*, volume 2, 2003.

[25] Z. Zhang. Iterative point matching for registration of free-form curves. Technical report, INRIA, Sophia Antipolis, 1992.

[26] J. Salvi, C. Matabosch, D. Fofi, and J. Forest. A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing*, 25:578–596, 2007.

[27] A. Hamza and H. Krim. Geodesic matching of triangulated surface. *IEEE Transactions on Image Processing*, 15:2249–2258, 2006.

[28] C. Chen and I. Stamos. Semi-automatic range to range registration: a feature-based method. In *International Conference on 3-D Digtal Imaging and Modeling*, pages 254–261, 2005.

[29] O. Au, C. Tai, H. Chu, D. Cohen-Or, and T. Lee. Skeleton extraction by mesh contraction. In *ACM Transaction on Graphics SIGGRAPH*, volume 27, 2008.

[30] Z. Ji, L. Liu, Z. Chen, and G. Wang. Easy mesh cutting. In *Computer Graphics Forum*, volume 25, pages 283–291, 2006.

[31] X. Li, T. Toon, T. Tan, and Z. Huang. Decomposing polygon meshes for interactive applications. In *2001 Symposium on Interactive 3D Graphics*, pages 35–42, 2001.

[32] I. Mikic, M. Trivedi, E. Hunter, and P. Cosman. Human body model acquisition and tracking using voxel data. *International Journal of Computer Vision*, 53:199–223, 2003.

[33] K.M.G Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–77–I–84, 2003.

[34] K.M.G Cheung, S. Baker, J. Hodgins, and T. Kanade. Markerless human motion transfer. In *2nd International Symposium on 3D Data Processing, Visualization and Transmission 3DPVT*, pages 373–378, 2004.

[35] R. Tadano, T. Yamasaki, and K. Aizawa. Fast and robust motion tracking for time-varying mesh featuring reeb-graph-based skeleton fitting and its application to motion retrieval. In *IEEE International Conference on Multimedia and Expo*, pages 2010–2013, July 2007.

[36] Y. Xiao, P. Siebert, and N. Werghi. Topological segmentation of discrete human body shapes in various postures based on geodesic distance. In *17th International Conference on Pattern Recognition*, 2004.

[37] J. Carranza, C. Theobalt, M. Magnor, and H. Seidel. Free-viewpoint video of human actors. In *ACM Transaction on Graphics SIGGRAPH*, volume 22, 2003.

[38] D. Bradley, T. Popa, A. Sheffer, W. Heidrich, and T. Boubekeur. Markerless garment capture. In *ACM Transaction on Graphics SIGGRAPH*, volume 27, 2008.

[39] C. Stoll, C. Theobalt, N. Ahmed, H. Seidel, and S. Thrun. Performance capture from sparse multi-view video. In *ACM Transaction on Graphics SIGGRAPH*, volume 27, 2008.

[40] D. Vlasic, I. Baran, W. Matusik, and J. Popovic. Articulated mesh animation from multi-view silhouettes. In *ACM Transaction on Graphics SIGGRAPH*, volume 27, 2008.

[41] T. Kanade, P. Rander, and P. Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE Multimedia*, 4:34–47, Jan./Mar. 1997.

[42] J. Starck and A. Hilton. Surface capture for performance-based animation. In *IEEE Computer Graphics and Applications*, 2007.

[43] J. Starck, G. Miller, and A. Hilton. Volumetric stereo with silhouette and feature constraints. In *British Machine Vision Conference*, 2006.

[44] G. Vogiatzis, P. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 391–398, 2005.

[45] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and D. R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithm. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 519–528, 2006.

[46] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspon-dence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[47] R. Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 108:4–18, 2007.

Published Works

[1] **N. Lee**, T. Yamasaki and K. Aizawa. Motion Tracking of Time-Varying Mesh Through Surface Gradient Matching With Multi-Temporal Registration. In *ACM Transaction on Graphics SIGGRAPH POSTER SESSION*, 2008.

[2] **N. Lee**, T. Yamasaki and K. Aizawa. Motion Tracking Through Surface Gradient Matching for Time-Varying Mesh. In *Meeting on Image Recognition and Understanding MIRU*, pages 411, 2008.

[3] **N. Lee**, T. Yamasaki and K. Aizawa. Hierarchical Mesh Decomposition and Motion Tracking For Time-Varying-Meshes. In *IEEE International Conference on Multimedia and Expo*, pages 1565–1568, 2008.

[4] **N. Lee**, T. Yamasaki and K. Aizawa. Hierarchical Mesh Decomposition and Motion Tracking for Time-Varying-Meshes. In *Institute of Electronics, Information and Communication Engineers General Conference*, 2007.

[5] **N. Lee**, T. Yamasaki and K. Aizawa. Mesh-Segmentation Based Frame Interpolation for Time Varying Mesh. In *12th Image Media Processing Symposium*, pages 81–82, 2007.