



All Theses and Dissertations

2012-10-24

Motion Capture of Character Interactions with a Rope

Bryce Zachary Porter

Brigham Young University - Provo

Follow this and additional works at: <https://scholarsarchive.byu.edu/etd>



Part of the [Computer Sciences Commons](#)

BYU ScholarsArchive Citation

Porter, Bryce Zachary, "Motion Capture of Character Interactions with a Rope" (2012). *All Theses and Dissertations*. 3374.
<https://scholarsarchive.byu.edu/etd/3374>

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Motion Capture of Character Interactions with a Rope

Bryce Z. Porter

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Master of Science

Michael Jones, Chair
Parris Egbert
Quinn Snell

Department of Computer Science

Brigham Young University

October 2012

Copyright © 2012 Bryce Z. Porter

All Rights Reserved

ABSTRACT

Motion Capture of Character Interactions with a Rope

Bryce Z. Porter

Department of Computer Science, BYU

Master of Science

We present a solution to animating interactions between characters and thin, non-rigid bodies using a passive optical motion capture system. Prior work in human body motion capture can accurately recreate human motion but this work is not adequate because it does not allow for interactions with a non-rigid body. Prior work in face and cloth motion capture handles non-rigid planes but rope is better handled with a curved spline rather than a curved plane. The segmented motion is in the form of un-indexed motion capture data. After segmenting the motion of the thin, non-rigid body and the human character the separated motion capture data can be recreated individually. The recreated motion streams are then recombined using 3D modeling and animation software. The presented solution also improves techniques for recreating thin, non-rigid body motion from un-indexed motion capture data. Using the linear combination of two predicted marker positions our solution can accurately track motion capture markers through each frame of the motion capture data. This also allows our solution to invent marker positions when gaps are present in the motion capture data. Our improvements allow users to reconstruct the motion of both a human character and a thin, non-rigid body simultaneously from motion capture data gathered in a mixed motion capture session.

Keywords: rigid body motion capture, non-rigid body motion capture, motion capture, animation

ACKNOWLEDGMENTS

I would like to express my appreciation to all those who have supported and assisted me with my MS thesis. I am grateful for all the time and assistance that my mentor, Dr. Mike Jones, has provided me throughout this thesis work. He spent many hours sharing his expertise and advising me on how to improve my thesis. I would also like to thank his family for their help and patience in capturing the motion capture data and for providing our motion capture actors. I would like to acknowledge the time and knowledge that Jie Long has shared with me during our motion capture sessions. I also appreciate the time my committee members have spent advising and sharing their insights with me to help improve this thesis. Lastly, I would like to thank my family and friends for their support, patience and encouragement as I worked towards my goal.

Table of Contents

List of Figures	v
1 Introduction	1
1.1 Thesis Statement	9
2 Related Works	10
2.1 Human Character Motion Capture	11
2.2 Non-rigid Body Motion Capture	12
3 Methods	15
3.1 Motion Reconstruction	15
3.2 Motion Capture Data	16
3.3 Segmentation of Motion Capture Data	18
3.4 Initialization of Segmentation algorithm	19
3.5 Marker Tracking	21
4 Results	26
5 Conclusion	31
References	33

List of Figures

1.1	Examples of motion capture. Gollum in Lord of the Rings Trilogy(top left, image courtesy of screenrant.com), Davy Jones in Pirates of the Carribean: Dead Man’s Chest (bottom left, image courtesy of trupsm.blogspot.com), Train conductor from Polar Express(right, image courtesy of wardomatic.blogspot.com)	2
1.2	Markerless motion capture used to bring characters to life in the video game L.A. Noire. Image courtesy of www.theurbanshogun.com	3
1.3	Rigid-body motion consisting of a translation as the arm is lowered and a rotation about a fixed pivot as the arm is extended.	4
1.4	Motion capture of a human character jumping rope. Note the conspicuous lack of a rope in the animation. Video provided by mocapdata.com	5
1.5	The process of reconstructing a mixed motion capture session	7
3.1	OptiTrack twelve camera setup	16
3.2	OptiTrack Flex camera(left), jump rope wrapped with reflective tape(right) .	17
3.3	Three frames ($f^1, f^2, and f^3$) of a motion capture session C . Each frame consists of q provisional markers where q varies from frame to frame	17
3.4	Three frames of a motion capture session overlaid on each other. A marker trace t_j is shown in gray as it tracks a marker from frame to frame	18
3.5	Rope marker selection	20
3.6	Calculating of V_j^i, N_j^i and \tilde{V}_j^i	23

3.7	Three frames in which the marker segmentation algorithm must overcome gaps in the motion capture data. In the final frame the actual rope marker is found again	24
4.1	Screenshots of the original motion, segmented rope and character markers and the reconstructed rope with human markers	27
4.2	Screenshots of the original motion compared to reconstructed motion with gaps in the data	27
4.3	Handling multiple consecutive gaps in the data	28
4.4	Gap repair error as the gap size varies	29

Chapter 1

Introduction

Passive optical motion capture systems are used ubiquitously in computer graphics to animate human characters and other objects. Motion capture systems vary in the way the motion of the actors is captured. The systems that our research focused on are called passive optical motion capture. Passive optical systems use cameras and markers to capture the motion of the actors and objects they interact with. The system is comprised of an arena with mounted cameras that view the scene in infra-red in a circle around the arena. Each actor or object in the scene is outfitted with markers that reflect infra-red light back toward the cameras with very little of that light scattering. The markers create a reflection that is much brighter than the surrounding background. Infra-red is used to eliminate as much of the other objects, and background in the arena as possible so that the cameras only see the markers. Infra-red light is not visible to the human eye so the flashing strobes used to create frames are not distracting to the human subjects. Using structure from motion and the different views of the same scene it is possible to calculate a 3D position for each reflective marker in the scene.

Motion capture systems are popular because of their ability to accurately recreate the motion of the subject. In industries such as movies and video games motion capture has become an integral part of creating believable and interesting characters. Figure 1.1 shows several examples of motion capture techniques used in movies. In the Lord of the Rings trilogy[7], shown top left in Figure 1.1, they use a passive optical motion capture system to animate the character Gollum. In Pirates of the Caribbean: Dead Man's Chest[21], shown



Figure 1.1: Examples of motion capture. Gollum in Lord of the Rings Trilogy(top left, image courtesy of screenrant.com), Davy Jones in Pirates of the Carribean: Dead Man’s Chest (bottom left, image courtesy of trupsm.blogspot.com), Train conductor from Polar Express(right, image courtesy of wardomatic.blogspot.com)

bottom left, they use facial markers to animate the octopus-like face of Davy Jones. Lastly, the train conductor from Polar Express[27] is shown with a combination of facial markers and body markers used to animate both the skeletal movement and the facial expression of the character. Figure 1.2 shows an example of markerless motion capture from the video game L.A. Noire[4]. L.A. Noire uses motion capture to realistically animate facial expressions, and to enhance the gameplay. Using motion capture to animate characters adds believability to computer generated characters and brings them to life for the audience.

Our research focuses on accurately recreating the interactions between a human character and a thin, non-rigid body. Capturing character interactions with a non-rigid body is a difficult problem because you cannot make the same assumptions about the kinematics of the motion as you do in rigid-body motion capture. In rigid-body motion capture you make the assumption that the character’s limbs, torso and head are all rigid-bodies connected by joints. A rigid-body is a solid object with a fixed size in which deformations can be neglected. The markers are placed on the actor in order to define rigid-bodies. As force is applied to a rigid-body the markers that define that body maintain their structure, meaning there is a fixed distance between each marker. Figure 1.3 shows an example of rigid body motion in the



Figure 1.2: Markerless motion capture used to bring characters to life in the video game L.A. Noire. Image courtesy of www.theurbanshogun.com

context of a person lowering and extending his arm. The red dots represent markers placed on the forearm. The markers form a plane that defines the rigid-body and any motion of a plane can be considered as a translation plus a rotation. The blue lines show a translation that happens as the arm is lowered. The green lines show a rotation that also takes place in this movement as the arm is extended. The arm pivots about the elbow joint which acts as the center of rotation for this rigid-body. Solving for just the rotation and translation that the rigid-body undergoes is a much easier problem than solving a complete affine transformation and any deformations that may have occurred. In non-rigid body motion capture there is deformation in the bodies that make up the target object. This means you cannot assume that its structure will be maintained throughout the movement. Therefore the motion cannot be reduced down to just a translation and rotation. We present a solution to capturing the interactions between a thin, non-rigid body and a human character simultaneously.

Figure 1.4 is three frames of a video showing a skeleton animated from motion capture of a human jumping rope. The rope is missing from the animation. This is partly because there is currently no solution to reconstructing rope motion without limitations on how the

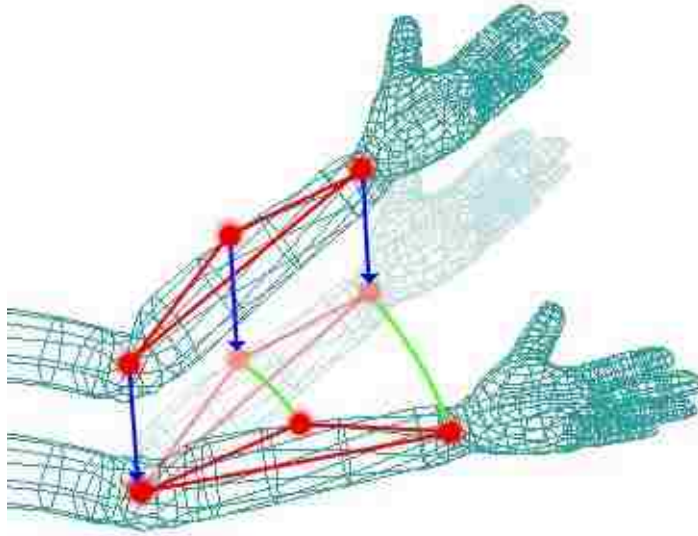


Figure 1.3: Rigid-body motion consisting of a translation as the arm is lowered and a rotation about a fixed pivot as the arm is extended.

rope moves and the speed at which the rope can move. The other reason is that there is no solution for capturing and recreating rope and human motion simultaneously. In order for an animator to create a scene with a human character interacting with a thin, non-rigid body they would need to capture the motion of the character and the thin, non-rigid body separately. The animator would then need to coerce the animation for the thin, non-rigid body to be complementary to the motion of the human character. Removing the need to manually animate thin, non-rigid bodies can save valuable time for animation studios. For example, if an animator is required to animate a character jumping rope in a scene it could take a significant amount of time to manually adjust the rope as it moves through the animation. This system would fit in with current work flows used in animation studios and would eliminate the need to manually animate the rope in the scene.

This problem is important to any animation or video game studio that uses motion capture systems to animate computer generated characters. These studios use motion capture data to simplify the animation process while preserving the subtleties of character interactions.

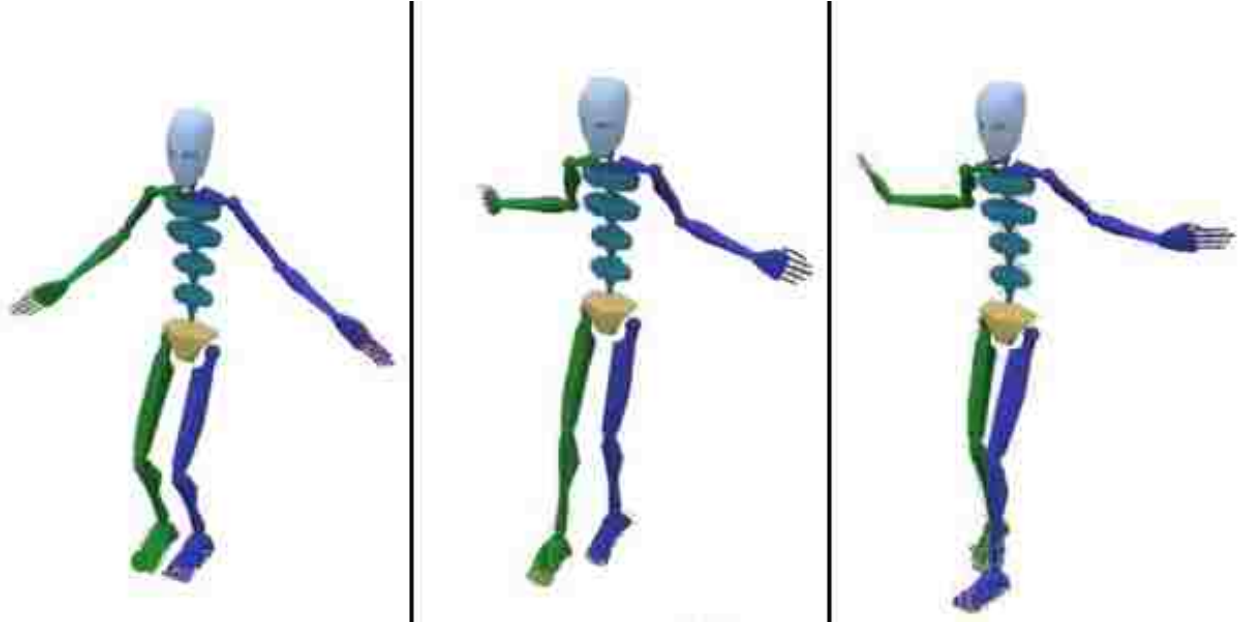


Figure 1.4: Motion capture of a human character jumping rope. Note the conspicuous lack of a rope in the animation. Video provided by mocapdata.com

This research will also benefit the motion capture research community by exploring the domain of capturing character interactions with a non-rigid body.

Our work builds upon prior research in character and non-rigid body motion capture. The majority of research in non-rigid body motion capture has been focused on capturing facial and cloth motion([24], [15], [17], [11], [19]). Both facial and cloth motion capture build a mesh that represents the motion of a surface. There has also been some work done with capturing the motion for a thin, non-rigid body [10], which is fundamentally different because it is more accurately represented by a curved spline than a curved surface.

By using a robust marker tracking algorithm we can segment the motion capture data into markers on the actor and markers on a segment of rope allowing us to reconstruct the motion of a character separately from the motion of the thin, non-rigid body. This approach invents visually plausible data when gaps occur in the motion capture data even when the velocity of the thin, non-rigid body is rapidly changing. Another improvement relative to prior systems for recreating the motion of a thin, non-rigid body is that our solution does

not make the assumption that the thin, non-rigid body is fixed to a stationary object at one end. This allows for a greater range of interactions with the thin, non-rigid body.

The most common approach to human motion capture is to use a rigid body model ([18], [16], [28]). Assuming that a human body is a collection of rigid bodies connected by joints simplifies the problem of determining skeletal structure from marker positions. This research is applicable to our research because we need to be able to reconstruct both the motion of a thin, non-rigid body and a human character. Despite significant research into motion capture of both rigid, and non-rigid bodies there is still a difficult problem of capturing interactions between a human character and non-rigid bodies.

The contribution of this work is that the novel marker segmentation algorithm simplifies the process of reconstructing interactions between a character and a thin, non-rigid body. Our system allows the user to capture and reconstruct the motion of a human character and thin, non-rigid body simultaneously. This reduces the time needed to record the motion capture data and preserves the subtlety of the character interactions. Our algorithm segments the mixed motion capture data so that the motion of the human character and thin, non-rigid body can be reconstructed independently. This significantly improves the process of reconstructing the interactions between a character and a thin, non-rigid body, as well as the perceived realism of the reconstructed motion. The marker tracking algorithm allows the thin, non-rigid body motion to be reconstructed without the assumption that it is attached at one end to a stationary object. This allows for more complex interactions with a human character.

The marker tracking algorithm is robust enough to properly segment the motion even in the presence of noise and large gaps in the motion capture data. After separating the marker positions of the rope from those of the character we can solve for the motion of each individually. The resulting recreated motion is recombined for a final result that is visually plausible and closely follows the motion that was originally captured.

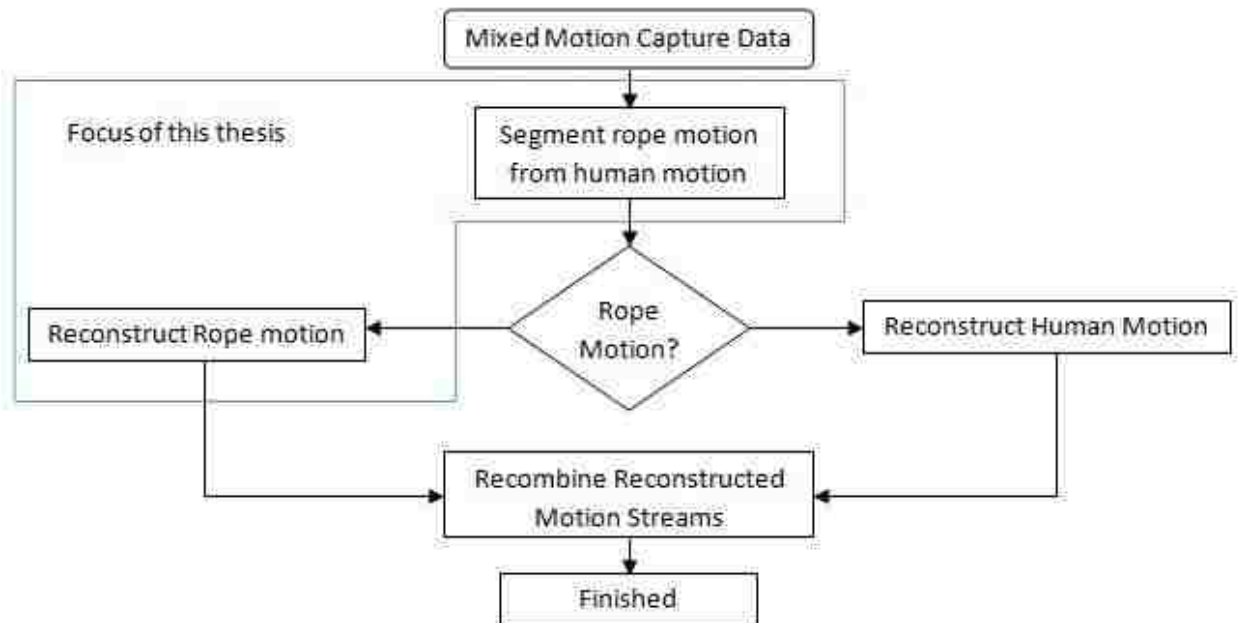


Figure 1.5: The process of reconstructing a mixed motion capture session

Figure 1.5 shows the complete process of reconstructing the motion capture data with a character interacting with a thin, non-rigid body. The process starts with un-indexed motion capture data of the character interactions captured during the motion capture session. Our solution takes the motion capture data in as input and begins by segmenting the motion capture data. After the motion capture data is segmented using the presented marker tracking algorithm we can reconstruct the motion of the rope and character separately. The rope motion is reconstructed using the marker tracking algorithm presented in this work which extends Long’s gap repair algorithm [10]. The marker tracking algorithm creates temporary markers in order to track a marker from frame to frame, but the temporary markers can also be used to fill gaps. Long’s gap repair algorithm is a post segmentation process that fills gaps in the rope data using a midpoint displacement algorithm. Our algorithm extends this idea by combining a midpoint displacement estimated position with a velocity based estimated position. Using velocity is an important extension to Long’s algorithm because it allows for gap repair even with high speed interactions and invents more accurate data to repair gaps. Velocity gives the gap repair algorithm more accuracy because the velocity describes how

an object moves over time. Given the change in time and a marker's current location and last known velocity you can predict the marker's next location. The character motion is reconstructed using commercially available skeletal solving and animation software. The final step in reconstructing the data is to recombine the two motion streams using commercial animation software.

The marker segmentation algorithm presented consists of two parts. The first part is an initialization phase in which the user selects the markers that belong to the thin, non-rigid body. After the selection is made the algorithm initializes marker traces, which are used to track the marker throughout the motion capture data. This initialization phase also gathers data about the structure of the thin, non-rigid body for use in the second step of the marker segmentation algorithm.

The second part of the marker segmentation algorithm uses assumptions about the rope motion and the structure of the rope to follow each rope marker throughout the motion capture session. The algorithm calculates an estimated marker position for each marker trace at every frame in the motion capture session. The first estimated position calculated uses the marker's last known velocity. If an actual marker is found at this estimated marker position then it is added to the marker trace. If no marker is found then this is most likely a gap in the motion capture data.

In order for this algorithm to succeed in segmenting the motion capture data it must be able to repair gaps and eliminate noise in the motion capture data. To do this our algorithm relies on neighboring markers that have known marker locations. The algorithm uses a weighted combination of two estimated marker positions to predict the position of a marker in a specific frame. The first estimated position is based on the weighted average of neighboring marker velocities. The other estimated position is based on the position of the neighboring markers. The weight used to combine the two estimated positions is related to how many frames have passed without being able to find an actual marker position near the predicted position. This method is also used to invent plausible motion capture data when there are

gaps in the motion capture data. This contribution is not only beneficial for tracking markers but also for reconstructing the motion of a thin, non-rigid body. Being able to invent visually plausible data where gaps exist in the motion capture data is important for recreating the motion of a thin, non-rigid body. This builds on previous work to reconstruct the motion of a thin, non-rigid body. Long's gap filling algorithm uses midpoint displacement to invent new data. We add to this by combining the estimated position using a midpoint displacement with an estimated position using velocity. This allows our system to segment and reconstruct without the assumption that the rope is attached at one end. Using a weighted combination of estimated marker positions is a method for tracking and inventing motion capture markers for use in segmenting and recreating motion from a mixed motion capture session.

This solution makes it possible to capture the interactions between a human character and a thin, non-rigid body using a passive optical motion capture system. A video comparison of the reconstructed motion and the actual motion will show the visual plausibility of the reconstructed interactions. We also test the accuracy of the reconstructed rope motion by holding back data to create gaps then using our algorithm to invent data to fill the gap. We start by holding back ten frames of data and progressively hold back more data until the algorithm fails. The algorithm is considered to fail if it loses track of all of the markers. If an invented marker is over half a meter from the actual marker it is considered lost. Then the distance between the invented data and the actual data is calculated as a measure of how accurate the gap filling algorithm is.

1.1 Thesis Statement

Using a weighted combination of estimated marker positions will allow us to invent and track motion capture markers for use in segmenting and recreating motion from a mixed motion capture session.

Chapter 2

Related Works

Computer graphics is a broad research area that can be split into three main topics: modeling, rendering, and animation. Modeling focuses on the creation and representations of geometry. Rendering is the process of taking a geometric representation and turning it into an image by applying lighting, projecting the 3D geometry and applying any other desired effects. Animation is the process of creating a sequence of rendered images to produce the illusion of motion when the images are viewed in rapid succession. Our research falls under the animation category because it focuses on capturing and reconstructing motion.

There are three main strategies for creating animations: simulation, key framing, and motion capture. All of the methods are commonly used in movie production and video games. Simulations create animations by displaying the results of numerical iterative approximations to differential equations or similar models that describe the behavior of a system. Simulations are widely used in film and video games to simulate fluids, fire, and other special effects ([20], [6]). Some simulations can run in real-time because they are simplified approximations of physical systems. However, simulations are also difficult to control because they typically are configured and then run with very little ability to direct them. Key framing is a process of constructing poses at different frames and then using the posed frames to fill in the motion for frames in between the posed frames. This can be a very time intensive process but it does give the artist greater control over the animation. We investigate the process of using motion capture to create animations. Motion capture involves recording the motion of real-world objects as they move and then applying the captured data to computer generated objects.

The motion capture technique is most commonly used to animate human motion in films and games. Our research further investigates the process of using motion capture to create animations.

The process of reconstructing motion capture data can be split into three steps: data collection, data processing, and model reconstruction. For both rigid body and non-rigid body motion capture the majority of the research focuses on optimizing the data collection and processing and the quality of the reconstructed motion. However, there is still the difficult problem of capturing the interactions between rigid and non-rigid bodies simultaneously that requires more research. In addition motion capture of thin, non-rigid bodies is still a largely unexplored topic.

2.1 Human Character Motion Capture

There is an extensive body of literature on human character motion capture. Here we highlight representative works from two well-studied approaches to human character motion capture. In human character motion capture there are two main approaches. One approach is to extract a skeletal or kinematic model from motion capture data. In this approach a model is extracted from the data. [12] describes a system for denoising human motion capture by learning a series of filtering bases from pre-captured motion capture data. [16] builds a kinematic model from optical motion capture data using a markov random field. [18] describes a unique system for extracting a skeleton model. The system uses outward facing cameras mounted on the body to capture motion data. Structure from motion is used to estimate the skeletal structure. The pose of each camera is estimated through non-linear optimization.

Another approach to human character motion capture is to apply motion capture data to animate a pre-defined skeleton. In this approach the data is fit to a model. In [23] least squares optimization is used to match motion capture data to a skeleton. [28] presents

a method for using a force-based physical model to map data to a pre-defined skeleton. All of the above mentioned works focus on reconstructing motion for a rigid body.

An important aspect of motion capture is being able to deal with missing data or noise in the data. Liu and McMillan present a system for estimating the position of missing markers for human motion capture[9]. The system uses a machine learning approach to solve the problem of inventing data for gaps in motion capture data. This system creates a global model using motion sequences as a training set. Then for new sequences with missing data the system gets a rough estimation of the missing data using the global model. That rough estimation is refined using a hierarchy of local linear models.

All of these approaches work well at reconstructing the motion of a human character. However, they are limited to human motion only. Our work hopes to further this research by allowing the capture of rigid body motion simultaneously with a thin, non-rigid body.

2.2 Non-rigid Body Motion Capture

Non-rigid body motion capture is fundamentally different from recording the motion of a rigid body because it is not possible to assume that the markers on a single body remain in the same relative positions. There has been extensive research done with a focus on recording and reconstructing the motion of non-rigid bodies because of these fundamental differences. Most of the research in non-rigid body motion capture is centered around reconstructing facial or cloth motion. Our work builds on research into reconstructing thin, non-rigid body motion. Thin, non-rigid body motion is different from other non-rigid body motion because it deals with deforming lines rather than with deforming surfaces.

Some work in non-rigid body motion capture for planes is quite general. [1] presents a generalizable model of time-varying spatial data that can be used in a wide variety of applications. Motion capture data is an example of time-varying spatial data. The presented model is used to capture not only the data but also temporal correlations found in the data. The generalized model can be used to label the data, then denoise and fill gaps in the data.

In [2] a method is presented for shape completion which the authors define as generating a complete mesh from a limited set of data. Using a training set of poses the system learns a pose deformation model which is used to produce a realistic surface mesh with muscle deformation based on the pose. The pose deformation model is combined with a supplemental model of variations based on body shape. The result is a 3D surface model with realistic muscle deformations for different poses and varying body shapes.

Specialized algorithms for capturing facial motion have also been developed. These algorithms leverage specific assumptions about facial geometry and behavior. [13] describes a method for capturing and reconstructing facial deformations by training a polynomial displacement map and then applying that map to a mesh. The result is high resolution facial expressions with large muscle deformation, wrinkles and skin pores. [11] reconstructs facial expressions by building a surface-oriented deformation paradigm. This allows their system to adapt a surface mesh created from one set of motion capture data to several different target models. In [19] motion capture data is combined with an anatomical model to create a model of the underlying facial structure including muscle tissue and skeletal structure. [5] introduces a markerless motion capture system that produces a facial mesh using a multi-view stereo approach. The mesh produced is then deformed by tracking its vertices throughout the motion capture session. In [3] a marker based motion capture setup is used to reconstruct large-scale facial motions along with a two camera system that captures medium-scale facial motions such as wrinkles. In [14] Park and Hodgins present a system for animating skin deformations such as bulging and stretching using a dense marker set, combined with a hole filling and smoothing algorithm.

Another area of research in non-rigid body motion capture is reconstructing cloth motion. [17] reconstructs cloth motion using optical flow from a calibrated, multi-camera setup. In [24] White, Crane, and Forsyth use specially textured cloth with a synchronized video camera to capture the motion of cloth. [15] recreates cloth motion by using stereo images to build rope geometry and parameterize cloth motion. [22] presents a system for

animating clothing that fits closely to the character. This system uses an example-based wrinkle synthesis technique that generates wrinkles based on the character’s pose.

Although many researchers focus on cloth and facial motion when researching non-rigid body motion capture there is a need to research thin, non-rigid body motion capture. Thin, non-rigid bodies present a different problem from normal non-rigid body motion capture because a thin, non-rigid body is more appropriately represented as a curved spline rather than a curved surface.

[26] and [25] both describe methods for extracting 3D curves from 2D images in continuous time frames. Our research uses motion capture data rather than 2D images because it is less computationally intense and does not require relying on segmenting the image to extract the target object.

A finite element model can also be used to simulate the motion of a thin, non-rigid body [8] by using a parameter system to describe the mechanics and structure of the thin, non-rigid body. The model can be robust enough to simulate physical forces such as tension, shear, bending, torsion, contact and friction. This approach is more computationally intensive because it is simulating complex physical interactions but may be a viable alternative to motion capture.

Our system builds on an unpublished system that uses a clustering algorithm to create motion traces for each marker as it moves from frame to frame [10]. That system also uses an algorithm for handling noise and gaps in the motion capture data when reconstructing the motion of a rope. We use a modified version of this method to recreate the motion of the thin, non-rigid body after it has been separated from the motion of the human character that is interacting with it.

Chapter 3

Methods

3.1 Motion Reconstruction

Our system reconstructs the motion of the interactions between a human character and a thin, non-rigid body from un-indexed motion capture data in the presence of noise. Our solution consists of multiple steps to take un-indexed motion capture data of both a human character and a thin, non-rigid body to create a final animation of the reconstructed interactions. The process begins by capturing the human character motion and the rope motion simultaneously and exporting the motion capture data to a .csv file. This is the input for our solution which separates the rope motion capture data from the human motion capture data. Gaps in the data are handled by using temporary markers. Temporary markers allow us to continue to track markers from frame to frame with minimal error propagation. Noise at the separation layer is not removed but is merely considered as a possible rope marker if it is within the bounds of where a marker is predicted to be in the next frame. We rely on the rope and human reconstruction algorithms to de-noise the separated motion capture data rather than trying to clean up the data as we are separating motion traces.

After the motion capture data has been separated into rope motion and human character motion we export the motion streams so that they can be reconstructed independently. We use a modified version of Long’s algorithm [10] to reconstruct the motion of the rope. Our improvements to Long’s algorithm allow the rope motion to be reconstructed without making the assumption that the rope is hanging with one end attached to a stationary object. This allows the thin, non-rigid body to move more freely and have more complex interactions with

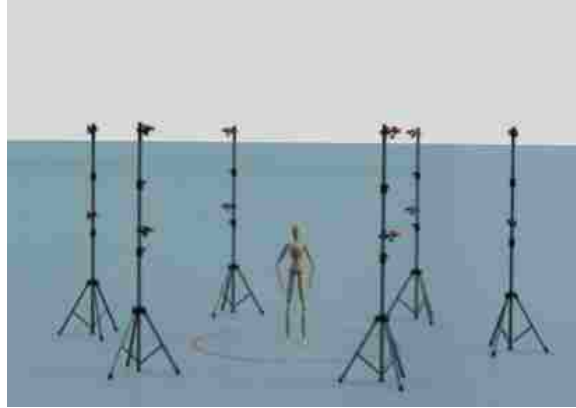


Figure 3.1: OptiTrack twelve camera setup

the human character than has been previously possible. With the human motion capture data we use a commercially available software package to reconstruct the motion and animate a pre-defined skeleton. After the human motion and rope motion have been reconstructed separately they are recombined using commercially available 3D modeling and animation software, such as Maya.

3.2 Motion Capture Data

We use a passive optical motion capture system comprised of 12 OptiTrack Flex cameras mounted on light poles. The light poles form a circle arena where the motion is recorded. The actor to be captured in the motion capture session wears a set of highly reflective markers. The thin, non-rigid body is also wrapped in intervals with reflective tape similar to the reflective markers.

Motion recorded by the system is then exported as a set of un-indexed marker positions to a .csv file. Our solution takes the un-indexed marker positions and separates the motion of the thin, non-rigid body from the motion of a human character. This section presents the notation used to describe the motion capture data. A motion capture session is comprised of n frames of capture data

$$C = [f^1, f^2, \dots, f^n]$$



Figure 3.2: OptiTrack Flex camera(left), jump rope wrapped with reflective tape(right)

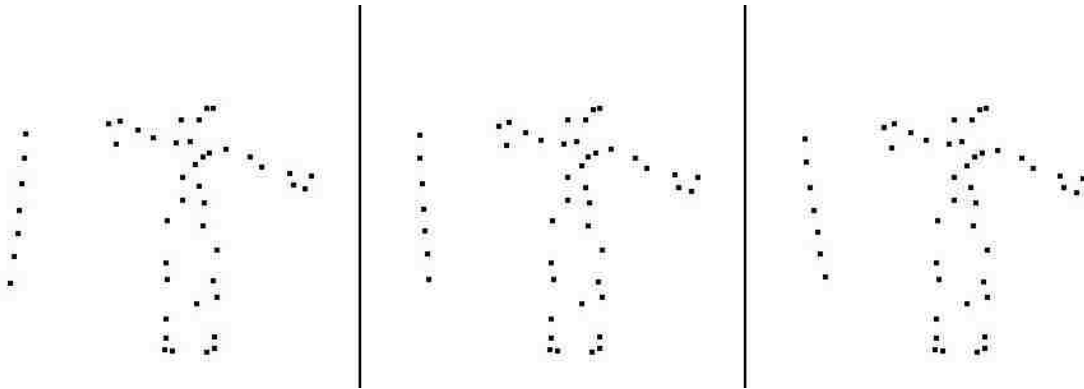


Figure 3.3: Three frames (f^1 , f^2 , and f^3) of a motion capture session C . Each frame consists of q provisional markers where q varies from frame to frame

and each frame consists of an unordered set of q provisional marker locations:

$$f^i = [p_1^i, p_2^i, \dots, p_q^i]$$

Each provisional marker location, $p_j^i (1 < j < q)$, is a vector of the 3D position of the j th marker at frame i . A provisional marker location might represent the position of an actual marker or might be generated by transient noise. A marker trace, t_j , is a set of marker positions that traces the motion of a specific marker throughout the capture session.

$$t_j = [m_j^1, m_j^2, \dots, m_j^i]$$

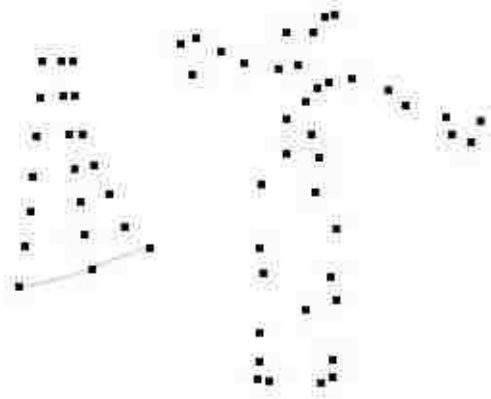


Figure 3.4: Three frames of a motion capture session overlaid on each other. A marker trace t_j is shown in gray as it tracks a marker from frame to frame

This means that m_j^i is the position of the j th marker at frame i . Assembling provisional marker positions across frames into traces is called labeling. Within a single trace there is the possibility for a gap in the sequence of markers. Gaps are created when markers are occluded in a frame or no provisional marker locations can be found to assign to a marker trace. When a provisional marker location cannot be found to assign to trace j in frame i then marker $m_j^i = temp_j^i$ denotes that there is a temporary marker in the trace at frame i . Noise is also present in the recorded motion capture data. Noise occurs when transient reflections in the capture area are interpreted as being marker locations. For our solution noise is partially eliminated as the rope and the human motion are being segmented and reconstructed. As the motion capture data is being segmented the algorithm is also separating the rope motion from any noise that may have been generated from the rope markers.

3.3 Segmentation of Motion Capture Data

Segmenting the motion of the rope from the human character captured in a mixed motion capture session is an important step in recreating the motion. We use motion traces to track

a particular marker from frame to frame. At each frame f^i we attempt to find a provisional marker location p_j^i that is in close proximity to the next estimated position for a marker trace t_j^i . When no provisional marker location can be found for t_j^i then a temporary marker is added to the marker trace at the estimated position.

$$m_j^i = \begin{cases} p_j^i & \text{if } p_j^i \text{ is close to the next estimated position} \\ temp_j^i & \text{a temporary marker at the estimated position} \end{cases}$$

The temporary marker allows the algorithm to continue tracking a marker despite the presence of noise and gaps in the motion capture data. Before fully segmenting the motion capture data there is a short initialization phase where the user identifies the rope markers that will be segmented from the other markers.

3.4 Initialization of Segmentation algorithm

The setup for the marker segmentation algorithm is a short process where the user identifies the rope markers and the motion traces are setup so that each rope marker can be tracked from frame to frame. The initialization takes place in a small portion of the motion capture data when the rope is hanging with little or no movement. This makes it easier to identify rope markers and to gather information required to fill gaps and filter out noise in the motion capture data. The user makes a simple selection of the markers that belong to the rope as shown in Figure 3.5. This requires that the rope markers are selectable separately from the human character. In scenes where the character interacts directly with the rope it is recommended that the rope be hanging from a stationary object for the short initialization phase, after which the character can pick up the rope and interact with it. This is not unlike starting a human character from a T-pose. The selected markers are used to initialize a set of traces T :

$$T = [t_1, t_2, t_3, \dots, t_j]$$

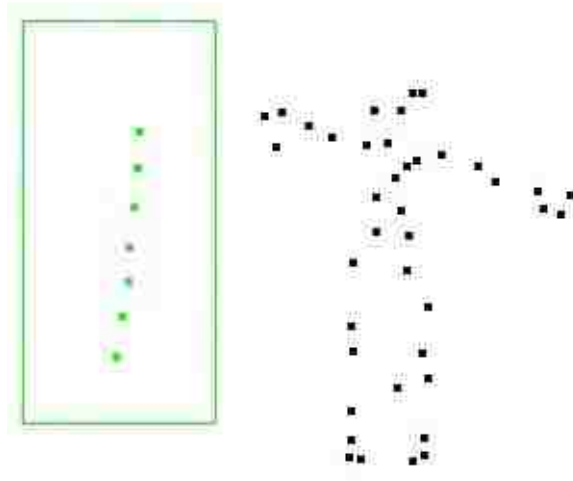


Figure 3.5: Rope marker selection

The next step in the initialization phase is to begin tracking the selected markers through the next few frames in order to record the approximate distance between consecutive marker traces.

In frame i ($0 \leq i \leq n$) the provisional marker p_j^i is appended to a marker trace t_j^i if it is the closest to the average position of t_j^i and within a threshold γ , which is the maximum allowable distance a provisional marker can be from a marker trace and still be considered part of the trace. If no provisional marker is found then a temporary marker is added to the trace at the average position. After all n frames have been processed any trace that has more temporary markers than the threshold, $maxTempMarkers$, is discarded and any provisional markers that were added to that trace are considered non-rope markers. All the remaining traces are then sorted by the average position's y-values so that T is ordered from highest rope marker trace to lowest marker trace. In the initialization phase only the position of provisional markers and the average position of the traces are used to segment the data.

3.5 Marker Tracking

After the initialization phase the marker tracking algorithm begins to match provisional markers to marker traces using the last known velocity of the marker, V_j^i or a linear combination of two predicted positions, \tilde{V}_j^i and N_j^i . \tilde{V}_j^i is a weighted average of the last known velocities of two neighboring markers that are not temporary markers. The final estimated position, N_j^i , uses a mid-point displacement method using two of the closest neighboring marker traces. V_j^i is calculated as follows:

$$V_j^i = m_j^{i-1} + v_j^{i-1} \Delta t \quad (3.1)$$

Where v_j^{i-1} is the velocity of the marker at t^{i-1} :

$$v_j^{i-1} = (m_j^{i-2} - m_j^{i-1}) / (t^{i-2} - t^{i-1}) \quad (3.2)$$

Figure 3.6 demonstrates the calculation of V_j^i . As the marker moves from frame to frame we use the positions from the previous two frames to calculate the velocity of the marker. Velocity is our primary means of predicting the next position of the marker because it can accurately track markers from frame to frame even when the marker is moving quickly and is changing direction rapidly. After V_j^i has been calculated we try to find a provisional marker p_j^i that is closest to V_j^i and is within the maximum distance threshold γ . If no provisional marker is found then a temporary marker is added to the trace at V_j^i . The temporary marker allows the algorithm to continue to calculate predicted positions and potentially overcome gaps in the motion capture data.

V_j^i is a fast approximation of where a marker should be based on prior knowledge. Due to noise and gaps in the data V_j^i is not sufficient to track a particular marker throughout a motion capture session. However, simply adding temporary markers can also lead to permanently losing track of a marker. If a gap occurs while the marker is changing direction, the temporary markers based solely on the last known velocity will cause the trace to continue

in the opposite direction of the actual marker. Also the combination of noise and gaps in the data can cause the temporary markers to move to a position that is far from the actual marker that should be added to the trace.

When no provisional marker p_j^i is found using the predicted position V_j^i it is either due to gaps in the data or a rapid change in velocity. When this occurs we rely on the neighboring markers that are not temporary markers and use a weighted average of those marker's last known velocities to estimate a marker position. This estimated position is \tilde{V}_j^i and is calculated as follows:

$$\tilde{v}_j^i = \beta v_{j+n}^i + (1 - \beta)v_{j+m}^i \quad (3.3)$$

Where v_{j-n}^i and v_{j+m}^i are neighboring marker velocities whose last known velocities are not based on temporary marker locations. \tilde{v}_j^i is the linear combination of the neighboring velocities. We use \tilde{v}_j^i to calculate \tilde{V}_j^i . β is a weight based on the relative positions of the neighboring markers. Using equation 3.1 we can derive the estimated position based on neighboring velocities. \tilde{V}_j^i is a much more stable way of tracking a marker through gaps because it ensures that marker traces do not stray too far from each other. This allows the markers to be flexible and deform as they should without letting them deform too much. Figure 3.6 shows how \tilde{V}_j^i is derived. The velocities of two neighboring markers are calculated and combined to estimate the velocity, \tilde{v}_j^i , of the marker trace t_j^i . The estimated velocity, \tilde{v}_j^i , is then used to calculate the estimated position \tilde{V}_j^i . However, this is still susceptible to failure when multiple consecutive marker traces have gaps simultaneously.

In order to prevent the temporary markers from causing a marker trace to lose the actual marker it is tracking we combine another predicted position with \tilde{V}_j^i . This ensures that the marker trace does not stray too far from the actual position of the marker and from neighboring marker traces. If the number of consecutive temporary markers is over a threshold then a predicted position from neighboring marker traces N_j^i is calculated and combined with V_j^i . In order to calculate N_j^i a directional vector \hat{d} is first computed and

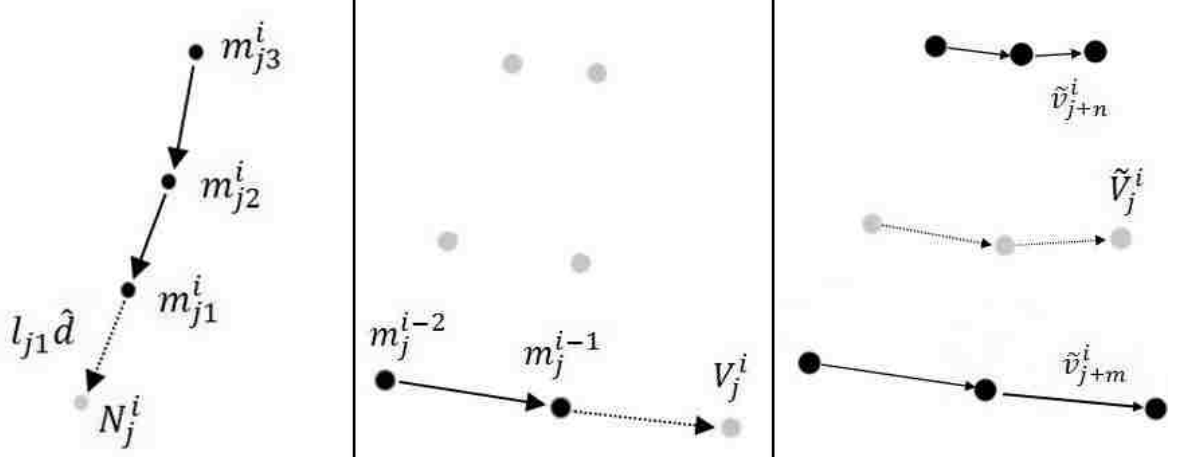


Figure 3.6: Calculating of V_j^i , N_j^i and \tilde{V}_j^i

normalized:

$$\hat{d} = \text{Norm}[\frac{(m_{j1}^i - m_{j2}^i) + (m_{j2}^i - m_{j3}^i)}{2}] \quad (3.4)$$

Then N_j^i is calculated by moving the average distance between t_j^i and t_{j1}^i , which is calculated in the initialization phase, along the directional vector \hat{d} starting from m_{j1}^i :

$$N_j^i = m_{j1}^i + l_{j1} \hat{d} \quad (3.5)$$

Where l_{j1} is the average distance. This estimated position is important to the algorithm because it maintains the structure of the rope. It ensures that the rope markers do not stray from the collection of rope markers that have known positions. By using three neighboring markers, m_{j1}^i , m_{j2}^i and m_{j3}^i , the algorithm takes into account the structure of that portion of the rope as it moves. This allows the algorithm to find rope markers even while the rope is moving freely without being attached at one end. One downside to this approach is that because the rope is a non-rigid body and not anchored at either end the structure of the rope can change rapidly and in several locations. This is why we cannot completely rely on N_j^i to recover missing markers and fill gaps in the data.

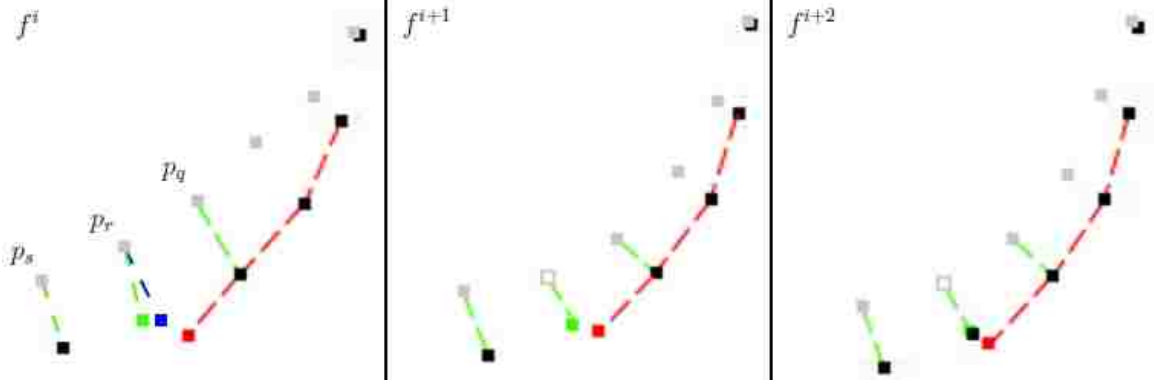


Figure 3.7: Three frames in which the marker segmentation algorithm must overcome gaps in the motion capture data. In the final frame the actual rope marker is found again

After \tilde{V}_j^i and N_j^i have been calculated they are combined based on how many consecutive temporary markers have been added to the marker trace.

$$F_j^i = \alpha \tilde{V}_j^i + (1 - \alpha) N_j^i \quad (3.6)$$

Where α is one over the number of consecutive temporary markers. This means that the more consecutive temporary markers the trace has the less we rely on \tilde{V}_j^i because it could have strayed too far from the actual rope markers. This is an important part of segmenting the data as well as inventing new data to fill gaps. Maintaining the structure of all parts of the thin, non-rigid body across gaps will result in better looking reconstructions of the motion. This algorithm reliably tracks un-indexed markers from frame to frame throughout a motion capture session. After fully segmenting the rope markers from the human markers it is possible to reconstruct the motion streams for each independently.

Figure 3.7 is an illustration of how gaps in the data are handled by the marker segmentation algorithm. This shows how the estimated positions invent new data to find markers at gaps in the data. The gray markers are the markers from the previous frame. If the gray marker has white in the center that indicates that it is a temporary marker in that frame. The blue marker indicates an estimated position using V_j^i which is used only as an

initial estimated position and is not used after the first frame of missing data. The green markers are estimated positions using \tilde{V}_j^i these estimated positions are combined with the estimated position using N_j^i to invent the temporary marker for that frame. The markers in red show the estimated position using N_j^i . In Figure 3.7 the focus is on the marker p_r as it moves through a gap in the motion capture data. Markers p_q and p_s are neighboring markers that do not have gaps in the frames shown. In frame f^i the algorithm attempts to find a provisional marker to add to the marker trace t_r . The estimated position V_r^i is calculated first, V_r^i is shown in blue. When no provisional marker is found then \tilde{V}_r^i , shown in green, and N_r^i , shown in red, are both calculated. After calculating the two estimated positions, \tilde{V}_r^i and N_r^i , we combine them using Equation 3.6 to calculate an estimated position F_r^i . If no provisional marker is found at location F_r^i then a temporary marker is added to trace t_r at the estimated position F_r^i . In frame f^{i+1} the marker p_r is now shown as a temporary marker, grey with a white center. In frame f^{i+1} , V_r^{i+1} is not calculated because it would be based on a temporary marker and becomes unreliable even after one frame. The estimated position F_r^{i+1} is calculated and the algorithm tries to find a provisional marker at F_r^{i+1} . Again no provisional marker is found and a temporary marker is added to trace t_r at frame f^{i+1} . In frame f^{i+2} the process from frame f^{i+1} is repeated by calculating F_r^{i+2} . In this frame however F_r^{i+2} is within the maximum distance, γ , from provisional marker. Now that a provisional marker has been found it can be added to trace t_r . After a provisional marker is found that fits the marker trace then the gap has ended and the algorithm proceeds using the last known velocity to continue to track the marker from frame to frame.

Chapter 4

Results

The results presented in this paper indicate that the interactions between a human character and thin, non-rigid body can be accurately reconstructed from un-indexed motion capture data. Additionally, the motion can be reconstructed from data where the human motion and rope motion were captured simultaneously. The figures shown in this section demonstrate the visual plausibility of the reconstructed interactions. Figure 4.1 shows the reconstruction of a motion capture session where the character is jumping rope. Comparison with still frames taken during the capture session shows that the markers were accurately segmented so that the human motion and rope motion could be reconstructed individually. This also shows the robustness of the tracking and reconstruction algorithms for the rope markers even with complex interactions such as jumping rope.

Interactions such as jumping rope are complex because the rope is not attached to a stationary object. Without the assumption that the rope is attached at one end the rope has more freedom to move, stretch, compress and twist. Also the interaction between the character and the rope can be very quick and the rope can change directions in a short time frame. With rapidly changing velocities a gap in the data becomes very challenging to repair because the actual markers will move far from their last known locations quickly. Another aspect that makes this interaction complex is that the rope markers are in close proximity to the human character throughout the interaction. Collisions are problematic as this could cause gaps. In addition markers may be swapped after collisions causing the algorithm to lose the rope marker as it follows the character marker it collided with.

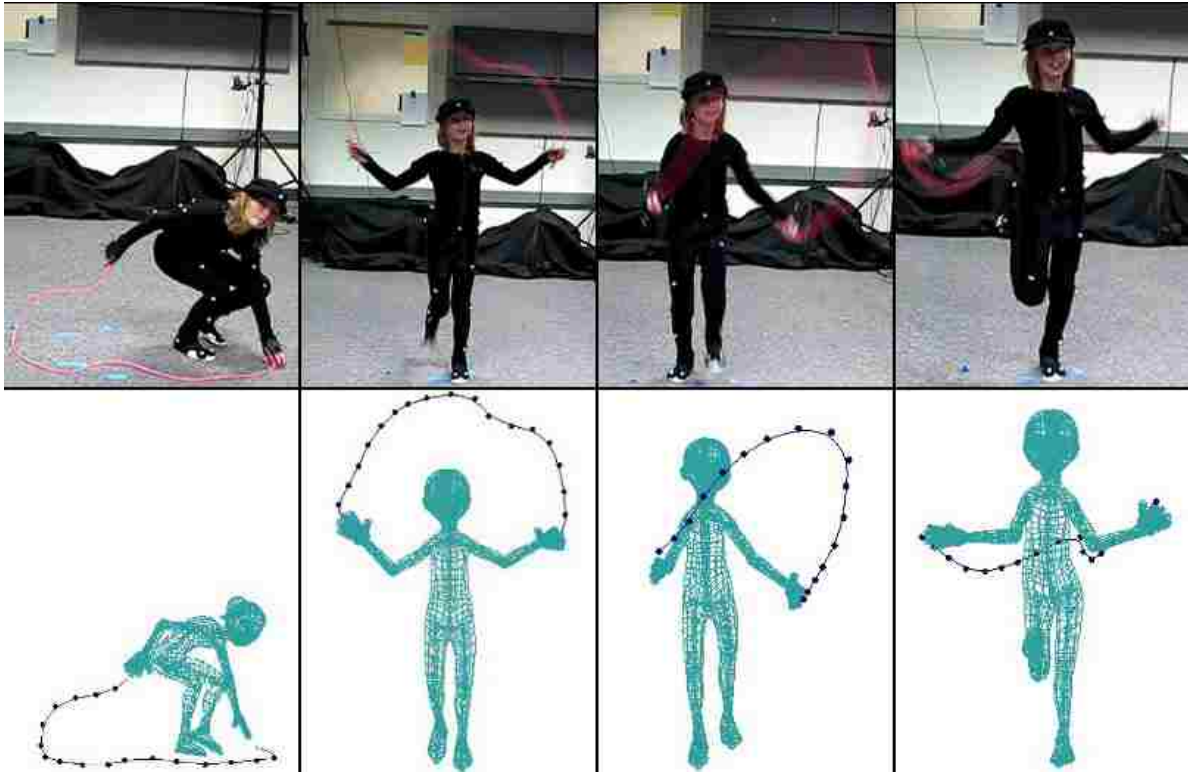


Figure 4.1: Screenshots of the original motion, segmented rope and character markers and the reconstructed rope with human markers

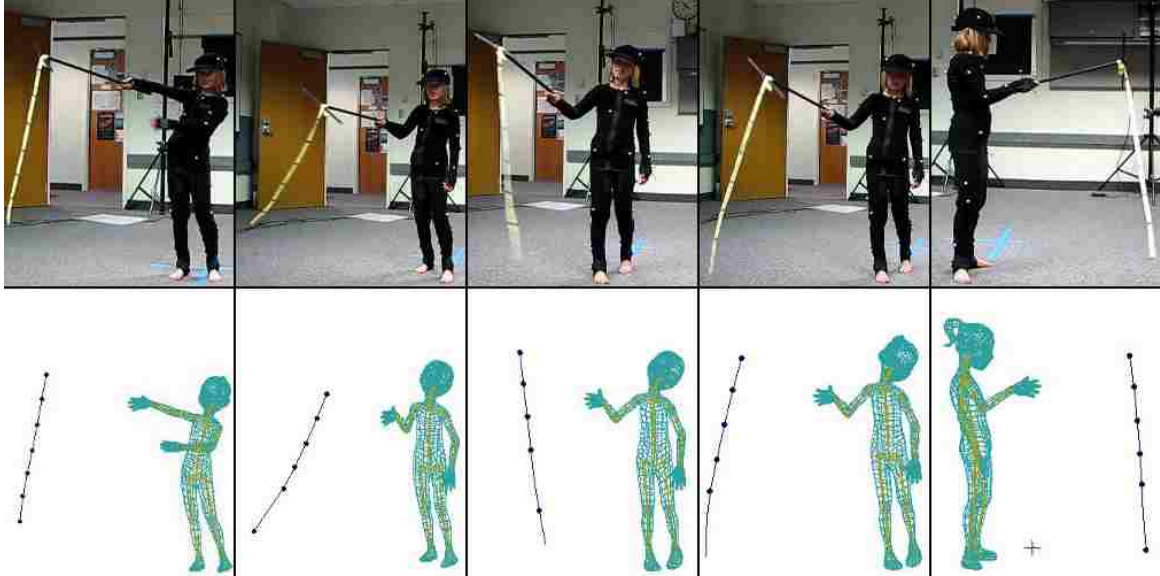


Figure 4.2: Screenshots of the original motion compared to reconstructed motion with gaps in the data

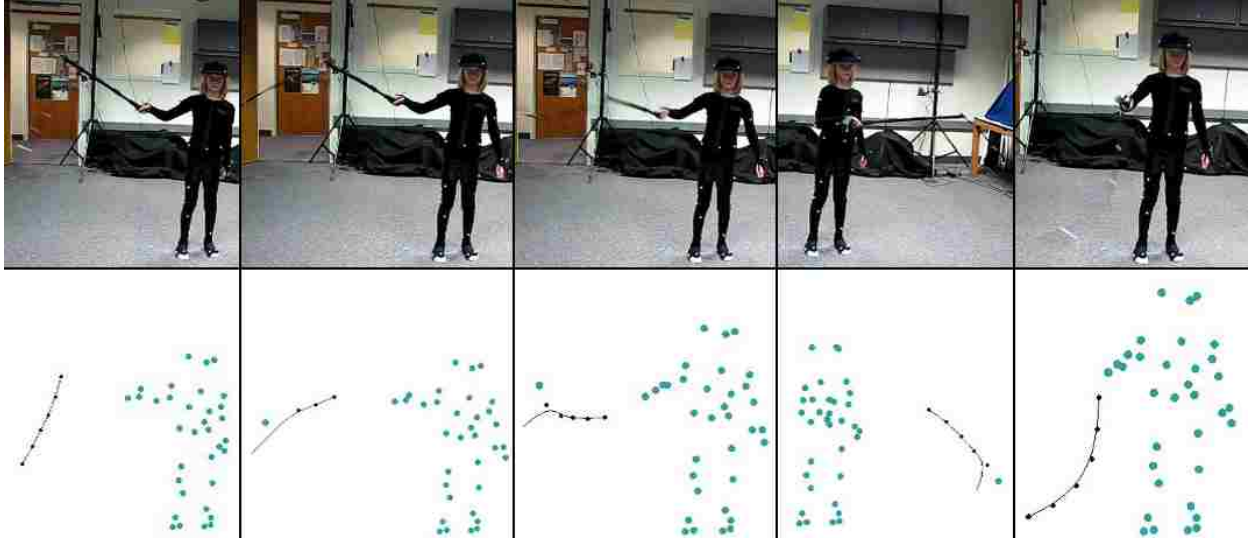


Figure 4.3: Handling multiple consecutive gaps in the data

Figure 4.2 demonstrates the accuracy of the reconstruction even when gaps are present in the rope marker data. The image set on the far left shows the reconstructed motion without any gaps in the data as a reference. The other image sets show either one or two missing markers as the motion capture session progresses. Note that regardless of whether or not the marker gaps have multiple consecutive gaps the rope structure is accurately estimated. The segmentation algorithm is robust enough to segment even when multiple markers are temporarily missing. Figure 4.2 also demonstrates the accuracy of the reconstructions of both the human character and the rope even with noise and gaps in the motion capture data.

Figure 4.3 shows a situation where this algorithm temporarily fails and later finds the rope markers. This figure shows a span of approximately 200 frames. In the second set of images of Figure 4.3 several consecutive markers disappear in the same frame. When multiple consecutive markers go missing at the same time then we suddenly have very little information about the structure of the rope and the velocities of the markers on the rope. Without much information about where the markers currently are our algorithm has a hard time finding the markers again. Eventually more markers may be found which helps constrain where the rest of the markers could be. As shown in the third and fourth image sets the

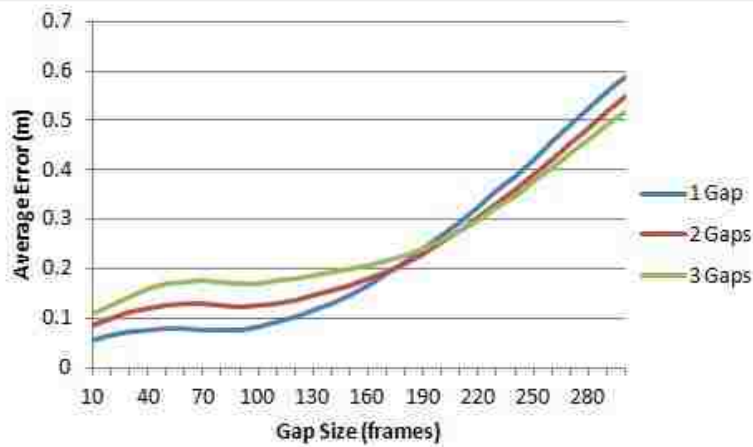
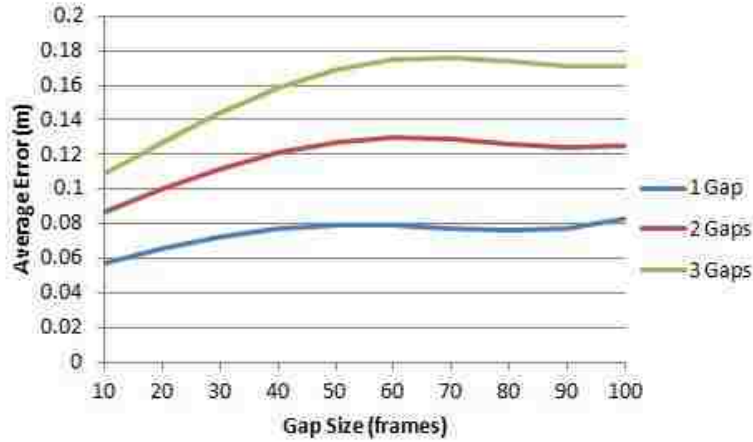


Figure 4.4: Gap repair error as the gap size varies

more markers that are recovered the more constrained the still missing marker locations can be and the easier it is to find the rest of the missing markers.

In order to test the accuracy of our method we created gaps in the known data and used our algorithm to fill those gaps then measure the error in the reconstruction. This not only shows the accuracy of our algorithms gap repair but also demonstrates the effectiveness of the marker segmentation. The test starts by choosing a beginning frame and creating a gap of size n . The gap size, n , starts at ten and increases in size until the algorithm fails. After a gap is created we use the system to repair the gap. When the gap has been repaired we calculate the distance from the actual marker that was held back to the invented marker. Figure 4.4 presents the average error across the length of the gap with a variable number

of consecutive markers missing for the ribbon swinging motion capture session shown in Figure 4.2. The algorithm performs very well with small to medium size gaps. With the error on average being less than a tenth of a meter across gaps of size ten to thirty frames. This is because using a linear combination of estimated positions is a reliable and accurate method for tracking and inventing data. Using the last known velocity and the velocities of known neighboring markers to predict the position of a missing marker is important for the accuracy of the reconstructed motion. Using the velocities of neighboring markers is a reliable way to track and invent data for a thin, non-rigid body. The algorithm relies on the assumption that the markers on a thin, non-rigid body will move mostly in a cohesive group, where markers follow each other. Based on this assumption we can assume that one marker will have similar velocities to its neighbors. However, with larger gaps and with several consecutive markers missing the algorithm begins to fail. This is because we know less about immediate neighbors and must make assumptions based on markers that are farther apart and less likely to move in a similar pattern. This means that the less that is known about the current structure of the rope the harder it is to invent data to reconstruct the motion.

Chapter 5

Conclusion

This work presents a system for reconstructing the interactions between a human character and a thin, non-rigid body. The results from this system are visually plausible and closely match the original motion. This paper introduces a novel marker segmentation algorithm that quickly and accurately segments the thin, non-rigid body markers from the character markers. The marker segmentation is robust enough to handle higher speed interactions as well as noise and gaps in the motion capture data.

Complex interactions such as tying knots or interactions where the thin, non-rigid body markers come in direct contact with the character's markers are still difficult cases that this system does not handle well because there are rope shapes or changes to rope structure that are smaller than the interval at which markers are placed on the rope. These smaller changes to the rope are undetectable to our system. Sudden changes in velocity when there are multiple consecutive gaps in the data also provide a challenge that the methods presented struggle with because using linear interpolation to estimate positions and velocities during segmentation and gap repair relies on having a sufficient number of known neighboring marker locations. When several markers are missing simultaneously the algorithm knows significantly less about the structure of the rope during the gap.

The presented system could be introduced as part of a work flow for use in film and video games because our system segments and does gap repair in the same process. For example, in 1.1 it shows the train conductor from Polar Express talking into an intercom. This system could be used in this scenario to animate the wire attached to the intercom so

that it moved and reacted realistically to the actor. Using this system would improve the visual plausibility of the character interactions with the intercom and reduce the time the animators would spend manually animating that part of the scene. Our system could become part of a motion capture data toolkit for navigating, cleaning, segmenting and repairing motion capture data. When used together with commercial software to reconstruct human motion our system simplifies the process of reconstructing human interactions with a thin, non-rigid body.

This work provides a viable framework for reconstructing the interactions between a character and a thin, non-rigid body that can be built upon to allow for more complex interactions and to handle more types of interactions. The presented system uses a linear combination of multiple estimated positions to quickly and accurately track markers through a motion capture session. However, there may be better non-linear blending functions that would more accurately track the motion of a marker. There may also be more ways of estimating the position of a marker that could be combined with the estimated positions presented in this paper that may make the marker segmentation algorithm more robust under extreme noise and gaps in the data. More robust marker segmentation algorithms may provide better results in reconstructing more complex interactions. More robust algorithms might also have the potential to be able to segment and repair not only thin, non-rigid bodies but also effectively segment and repair rigid body motion capture.

References

- [1] Ijaz Akhter, Tomas Simon, Sohaib Khan, Iain Matthews, and Yaser Sheikh. Bilinear spatiotemporal basis models. *ACM Transactions on Graphics*, 31(2):17:1–17:12, 2012.
- [2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. *ACM Transactions on Graphics*, 24(3):408–416, 2005.
- [3] Bernd Bickel, Mario Botsch, Roland Angst, Wojciech Matusik, Miguel Otaduy, Hanspeter Pfister, and Markus Gross. Multi-scale capture of facial geometry and motion. *ACM Transactions on Graphics*, 26(3):33:1–33:12, 2007.
- [4] Team Bondi. *L.A. Noire*. DVD-ROM, 2011.
- [5] Y. Furukawa and J. Ponce. Dense 3d motion capture for human faces. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1674 –1681, 2009.
- [6] Christopher Horvath and Willi Geiger. Directable, high-resolution simulation of fire on the gpu. *ACM Transactions on Graphics*, 28(3):41:1–41:8, 2009.
- [7] Peter Jackson. *The Lord of the Rings: The Two Towers*. Film, 2002.
- [8] W.G. Jiang, M.S. Yao, and J.M. Walton. A concise finite element model for simple straight wire rope strand. *International Journal of Mechanical Sciences*, 41(2):143–161, 1999.
- [9] Guodong Liu and Leonard McMillan. Estimation of missing markers in human motion capture. *The Visual Computer*, 22:721–728, 2006.
- [10] Jie Long. Motion capture of rope anchored at one end. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2011.
- [11] Manuel Sanchez Lorenzo, Manuel Schez, Lorenzo James, James D. Edge, Scott A. King, and Steve Maddock. Use and re-use of facial motion capture data. In *Proceedings of Vision, Video, and Graphics*, pages 135–142, 2003.

- [12] Hui Lou and Jinxiang Chai. Example-based human motion denoising. *IEEE Transactions on Visualization and Computer Graphics*, 16:870–879, 2010.
- [13] Wan-Chun Ma, Andrew Jones, Jen-Yuan Chiang, Tim Hawkins, Sune Frederiksen, Pieter Peers, Marko Vukovic, Ming Ouhyoung, and Paul Debevec. Facial performance synthesis using deformation-driven polynomial displacement maps. *ACM Transactions on Graphics*, 27(5):121:1–121:10, 2008.
- [14] Sang Il Park and Jessica K. Hodgins. Capturing and animating skin deformation in human motion. *ACM Transactions on Graphics*, 25(3):881–889, 2006.
- [15] David Pritchard and Wolfgang Heidrich. Cloth motion capture. In *Proceedings of ACM Special Interest Group on Computer Graphics*, volume 22, pages 263–271, New York, NY, USA, 2003. ACM.
- [16] Stjepan Rajko and Gang Qian. Real-time automatic kinematic model building for optical motion capture using a Markov random field. In *Proceedings of the IEEE International Conference on Human-Computer Interaction*, pages 69–78, Heidelberg, Germany, 2007. Springer-Verlag.
- [17] Volker Scholz and Marcus A. Magnor. Cloth motion from optical flow. In *Proceedings of 9th International Fall Workshop on Vision, Modeling and Visualization*, pages 117–124, 2004.
- [18] Takaaki Shiratori, Hyun Soo Park, Leonid Sigal, Yaser Sheikh, and Jessica K. Hodgins. Motion capture from body-mounted cameras. *ACM Transactions on Graphics*, 30(4):31:1–31:10, 2011.
- [19] Eftychios Sifakis, Igor Neverov, and Ronald Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Transactions on Graphics*, 24(3):417–425, 2005.
- [20] Jos Stam. Stable fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pages 121–128, New York, NY, USA, 1999. ACM.
- [21] Gore Verbinski. *Pirates of the Caribbean: Dead Man’s Chest*. Film, 2006.
- [22] Huamin Wang, Florian Hecht, Ravi Ramamoorthi, and James O’Brien. Example-based wrinkle synthesis for clothing animation. *ACM Transactions on Graphics*, 29(4):107:1–107:8, 2010.

- [23] Gaojin Wen, Zhaoqi Wang, Shihong Xia, and Dengming Zhu. From motion capture data to character animation. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 165–168, New York, NY, USA, 2006. ACM. ISBN 1-59593-321-2.
- [24] Ryan White, Keenan Crane, and D. A. Forsyth. Capturing and animating occluded cloth. *ACM Transactions on Graphics*, 26(3):34:1–34:8, 2007.
- [25] Marcel Worring, Pia Pfluger, Arnold W.M. Smeulders, and Adriaan B. Houtsmuller. Measurement of 3d-line shaped objects. *Pattern Recognition Letters*, 15(5):497–506, 1994.
- [26] Laurent Younes. Optimal matching between shapes via elastic deformations. *Image and Vision Computing*, 17(1):381–389, 1999.
- [27] Robert Zemeckis. *The Polar Express*. Film, 2004.
- [28] Victor Brian Zordan and Nicholas C. Van Der Horst. Mapping optical motion capture data to skeletal motion using a physical model. In *Proceedings of the ACM SIG-GRAPH/Eurographics Symposium on Computer Animation*, pages 245–250, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.