Rotation Points From

Motion Capture Data Using a Closed Form Solution

by

JONATHAN KIPLING KNIGHT

M.A. Applied Mathematics, Cal. State Fullerton, 1995

B.S. Physics, Cal Poly, San Luis Obispo, 1987

A thesis submitted to the Graduate Faculty of the

University of Colorado at Colorado Springs

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2008

Acknowledgements

First and foremost I would like to acknowledge my wife Kiki for her patience during this research. I would also like to thank Gene Johnson for his support during the ups and downs of my life during this major event in my life. He kept telling me to continue. I would also like to thank my advisor Dr. Semwal for allowing me the freedom and guidance to pursue the ideas laid down in this paper.

This thesis for the Philosophical Doctor degree by

Jonathan Kipling Knight

has been approved for the

Department of Computer Science

by

_____
Sudhanshu Kumar Semwal, Chair

_____
Robert Carlson

_____
C. Edward Chow

_____
Jugal Kalita

_____
Charles M. Shub

_____
Date

Knight, Jonathan Kipling (Ph.D., Computer Science)

Rotation Points from Motion Capture Data Using a Closed Form Solution

Thesis directed by Professor Sudhanshu Kumar Semwal

Four new closed-form methods are present to find rotation points of a skeleton from motion capture data. A generic skeleton can be directly extracted from noisy data with no previous knowledge of skeleton measurements. The new methods are ten times faster than the next fastest and a hundred times faster than the most widely accepted method. Two phases are used to produce an accurate skeleton of the captured data. The *first phase*, fitting the skeleton, is robust even with noisy motion capture data. The formulae use an asymptotically unbiased version of the Generalized Delogne-Kása (GDKE) Hyperspherical Estimation (first estimator: UGDK). The second estimator takes advantage of multiple markers located at different distances from the rotation point (MGDK) thereby increasing accuracy. The third estimator removes singularities to allow for cylindrical joint motion (SGDK). The fourth estimator incrementally improves an answer and has advantages of constant memory requirements suitable for firmware applications (IGDK). The UGDK produces the answer faster than any previous algorithm and with the same efficiency with respect to the Cramér-Rao Lower Bound for fitting spheres and circles. The UGDK method significantly reduces the amount of work needed for calculating rotation points by only requiring 26N flops for each joint. The next fastest method, Linear Least-Squares requires 236N flops. In-depth statistical analysis shows the UGDK method converges to the actual rotation point with an error of $O(\sigma/\sqrt{N})$ improving on the GDKE's biased answer of $O(\sigma)$. The *second phase* is a real-time algorithm to draw the

skeleton at each time frame with as little as one point on a segment. This speedy method, on the order of the number of segments, aids the realism of motion data animation by allowing for the subtle nuances of each time frame to be displayed. Flexibility of motion is displayed in detail as the figure follows the captured motion more closely. With the reduced time complexity, multiple figures, even crowds can be animated. In addition, calculations can be reused for the same actor and marker-set allowing different data sets to be blended. The main contributions in this dissertation are the new unbiased center formulae; the full statistical analysis of this new formula; and the analysis of when the best measurement conditions are to initiate the formula. The dissertation further establishes the application of these new formulae to motion capture to produce a real-time method of drawing skeletons of arbitrary articulated figures.

CONTENTS

TABLES

FIGURES

PREFACE

The research that will be presented in this paper is the culmination of a story. The five-year journey involves the typical parts of a novel with protagonist, antagonists, heartaches, and triumphs. The protagonist is, of course, me – the author. The antagonists in this story were the many hurdles that I stumbled over to finally realize what my actual goal was for the research. The story of this research is a good example of what it means for the research topic itself to tell you are going in the wrong direction. Perseverance has declared a victory in the research only by finally exposing a vital and missing piece of the puzzle that builds up the topic at hand.

So, what is the topic? The research started out five years ago as an idea that current technology for articulated motion analysis and display was much too slow and unrealistic. I started the research, albeit naively, by thinking that a combination of physics and genetic algorithms could infuse a sense of realism and speed into analyzing the motion. So I sat down and coded up the infrastructure needed to build up this combination.

First, I needed a computer representation of an articulated figure. To keep physics in the equation, I decided to make each segment of the figure solid. I found that, for physics related simulations, this is the only way to go since calculations of inertial moments are fairly simple (cf. Chapter 7.2). A few other authors use this approach for high-end physical simulations but, for the most part, most use triangulated surfaces for their "solid" figures. It is usually difficult to calculate moments of inertia from only the surface data. A tetrahedral solid mesh was chosen to simplify building up an arbitrary shape. With a solid shape, and its mass, the rotational and translational physics are fully calculable from the Newton-Euler differential equation. As I progressed further in the

research, I found it increasingly time-intensive to calculate projected motions based on solving the second-order differential equation on an arbitrary articulated figure.

This suggested that I pursue an alternative approach. So, what if I relied more on motion-capture data and infuse only a little bit of physics? I then looked at reading in and using raw motion-capture data. Little did I know that motion-capture data was ex-cruciatingly hard to acquire both on your own and on the Internet. The people and companies that did any kind of work with capturing motion data were usually quite willing to *sell* it. Only a few free examples were found that were of any use. A file reader was built from scratch for the most popular format since no one had open source code. I struggled for a year with reading the quirky data format with the small amount of data I found until a goldmine was found. Carnegie-Melon University Graphics Laboratory had just started placing a large cache of raw and analyzed motion-capture data on the Internet through a government grant. I subsequently downloaded gigabytes of raw binary data for a plethora of motions captured by the students at this lab. One problem solved – gathering data to analyze.

Now I set about trying to analyze the raw data so that I may use it to better analyze the underlying articulated motion. Starting with the simplest issue, I started trying to display the motion on a 3-D scene. As it turns out, the raw data was very inconsistent in storing which measurement units it was using. Sometimes it was in meters, sometimes it was in feet, sometimes it was stored in feet and the format said it was meters. Once you figure out what scale the data is in, you can display the points on the screen.

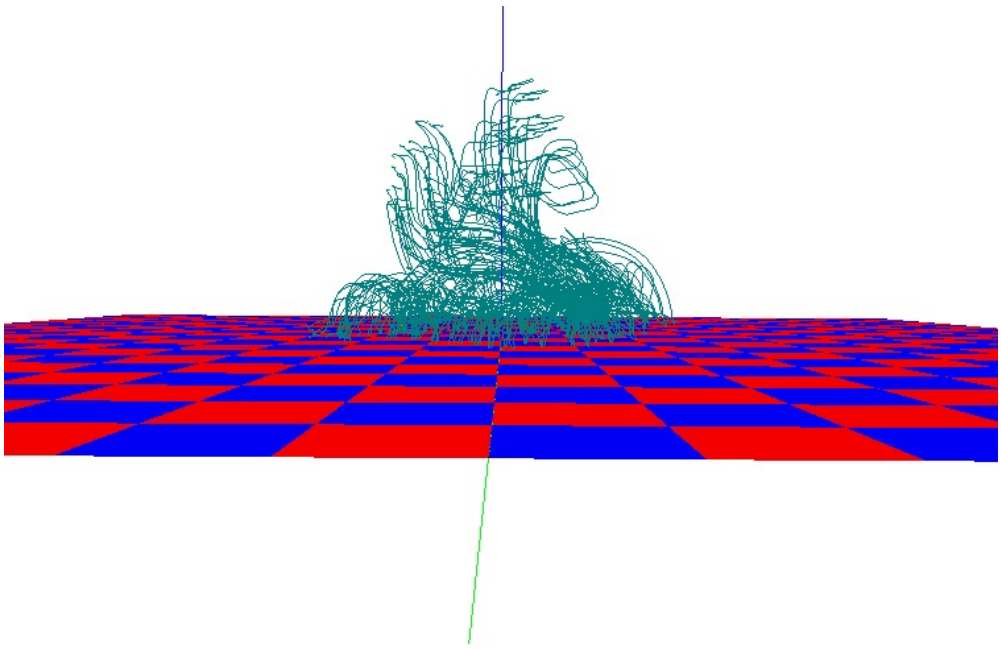

Figure 1 Display of Motion-Capture Data

Now, all I had was a bunch of squirrelly lines on the screen (cf. Figure 1). Clearly, the lines had to be associated and grouped. Each line was attached (i.e. correlated) to a particular segment on the articulated figure. This can be done in one of two ways – autocorrelation and precognition. I chose the latter since autocorrelation is extremely time intensive. Precognition may sound like cheating but nine times out of ten, the user will know ahead of time which bits of data are associated with which segment on the articulated figure.

What do I have so far? I have a bunch of lines that each segment can follow in time. As data shows marker location at each time frame, each segment has some surface positions pin-pointed in space. In addition, exact marker location on segments usually were not known or given. That leaves an unsavory taste in your mouth when the segment can be placed almost anywhere relative to these points. A systematic approach must be

thought of that allows the conclusive attachment of the segments to the data. This implies that the data must contain both relative position and orientation of the segment so the model can be sized and oriented into position. This satisfies an individual segment, but not the whole articulated figure. For that I needed the rotation points in between the segments. What this boils down to is that I needed to draw a stick figure based solely on the motion-capture data. I could then attach whatever shape model I wanted onto the stick figure. I did a search for existing methods and found a few examples in the literature (cf. Chapter 8). The most popular was to perform a least-squares fit and a few still did non-linear fitting of the data. They all had one thing in common – they all assumed the marker moved on a sphere around the rotation point of the joint. This means that, to draw a stick figure, I had to solve for the center of a sphere at every joint. A little research showed there were only three major algorithms to solve for the center of a sphere by data points sitting on the sphere. The most popular was the Maximum-Likelihood Estimator (MLE) that solved the problem in a non-linear fashion. This had the undesirable effect of being excruciatingly slow and sometimes not producing an answer at all. The next most popular type was linear least-squares solutions that involve fairly slow (but faster than MLE) pseudo-inverse matrix solutions. This was the preferred method by most authors concerned with speed. The third type of solution was a small set of approximations that were closed-form solutions to the best-fit sphere. These formulae were unpopular and usually used only in the case to start off a non-linear search for the real answer. They were unpopular because of biases introduced.

I was unsatisfied with any of these solutions found in the existing literature. They would get me to the answer, but they were too slow or quirky. I decided to sit back and

analyze a case study of a particular joint movement. To proceed with analyzing, the raw (x,y,z) data was loaded into a spreadsheet for the step-by-step analysis to see how the rotation point could be retrieved from the data. A column of distances from an unknown point (the rotation point) was made for each data point. Then, thinking about it – what would happen if the unknown point were truly at the center of rotation? Then, the distances to the data points would all be about the same. What this translates into is that the standard deviation from the mean of the distances would be minimized. This can be easily done in Excel with the Solve tool. Thinking about it further, that this is exactly what the Maximum Likelihood Estimator (MLE) solution is – minimizing the standard deviation of the distance to the center of rotation. I looked carefully at the mathematics to see if a closed form solution can be found. The MLE does not solve to a closed form. I then altered the column to be the square of the distance. The Solve tool still got nearly the same answer as the MLE. I looked at the math for this minimizing and lo and behold, a closed form solution dropped out on the floor. So the minimum of the standard deviation of the *square* of the distance to the center of rotation can be solved with a fairly simple formula.

I found this new formula easy to use and very fast compared to the other two techniques. I was all ready to name my new discovery but then I thought it was too good to be true. I tried desperately to find an author in the existing literature that used this formula. I could not find a single author that even implied the existence until a month later. I found an author [119] who had published three months earlier in some obscure electronics conference (yes 3 months!). His formula was a different arrangement of the same solution I had developed. So, at least I knew what to call the formula – Generalized

Delogne-Kása Estimator (GDKE). Apparently this formula had been used off and on in many industries since 1972 [56][57] in only its two-dimensional form (i.e. circle). This new formula can handle any dimensional sphere. The formula had not risen in popularity because of a certain flaw. In a particular arrangement of data, the estimator would be expected NOT to be the answer. The flaw would not show itself if the data points were distributed evenly over the entire sphere. Unfortunately, this ideal case is not too common in the real world. Therefore, the formula ended up only being useful when the measurement system was extremely accurate – another not so common real world case.

Fine, someone beat me to a fast and flawed formula that solves my problem. Not good enough for me, so I set out to remove the flaw and retain the speed. The first thing to do to remove the flaw is to exactly identify it. The original paper from 2004 [119] did not explain in detail what the flaw was, just that the bias was about the size of the measurement error. As it turns out, the multiplication factor can amplify the bias far beyond the simple measurement error. It took over two years to work out the exact relationship of the bias. It involves the probability analysis of multidimensional random variables multiplied together up to six times. It was slow going since the equation grew rather large. Keeping track of all of the variables became unwieldy, as even Mathematica couldn't handle the math. Finally, the flaw was identified and a systematic equation was produced that effectively removed the flaw. The results were a simple modification to the GDKE that made the formula guarantee to converge to the true center and radius as more raw data was thrown at it. The gap in technology for building a skeleton from motion capture data was identified and filled. This paper goes into excruciating detail to prove the capabilities of this new method for skeletal extraction.