

Algorithm for Arm Position Reconstruction from Optical Motion Capture Data with Noisy or Missing Data

Meredith Moore
Neuroscience and Computer Science
Drake University
Des Moines, IA, 50311
Meredith.Moore@drake.edu

Helene Moorman¹, James Gao², and
Jose Carmena^{1,3}
¹Helen Wills Neuroscience Institute
²Vision Science Program
³Electrical Engineering and
Computer Science
University of California Berkeley
Berkeley, CA, 94720

Abstract

To better understand the intricacies of motor control, two types of data are necessary: neural data, and movement data. Optical motion capture recordings were taken of a rhesus macaque monkey performing natural free movement tasks. An algorithm was implemented in order to reconstruct the arm positions of the monkey. This solution takes advantage of fundamental distance constraints that must hold true for markers attached to an articulated structure. Validation data with functions to add noise or missing data was created to assess the accuracy of the algorithm. Simulations were run to evaluate the effectiveness of this algorithm on data sets that have inconsistencies in the markers. The algorithm was sensitive to the addition of noise to the data as well as missing data. If the datasets meets specific conditions regarding the amount of noise and missing data, the algorithm can be considered reliable for reconstructing arm positions from optical motion capture data.

Introduction:

Motor control has been explored for decades with the primary motivation of learning more about how the brain controls movement. In order to analyze motor control, two types of data are necessary: movement and neural data. There are several different ways of collecting motor output data—these techniques include the use of exoskeletons, joysticks, and video paired with image processing software[1]. Motion capture technologies, however, drastically improve the ease and flexibility of measuring motor outputs.

Motion capture, the process of recording the movement of objects or people, has made a significant impact on many contemporary fields[2]. It is used in the military, entertainment industry, athletic development, as well as medicinal applications. In the entertainment world—largely filmmaking and video game development—motion capture refers to recording actions of human actors and using that information to animate digital models in 2D or 3D computer animation. Motion capture technology has been available for researchers for a few decades and has given new insights into many disciplines. It has provided vast amounts of new information on gait analysis[3].

In the world of computer graphics, specifically animation, the surroundings are specifically designed to incorporate commercial motion tracking equipment with minimal missing data—markers that are not always successfully recorded. The same motion-tracking technology, however, has been shown to be very effective in a scientific research environment. Unlike movie sets, often times these scientific research facilities are not designed with motion tracking functionality in mind. In these cases, data recorded from motion tracking equipment is not as robust and reliable as in the cases where the set up is specifically designed for motion tracking. Because of this, motion capture data sets recorded for scientific purposes are more likely to be wrought with missing and or noisy data[3].

Specifically, when recordings were taken from a Rhesus macaque monkey performing free arm movement tasks in a laboratory setting, the raw experimental data had many inconsistencies stemming from noise from the markers as well as obstructed markers leading to missing data. The movements that the monkey was making were very quick, and relatively concise. This is believed to have led to noise in the marker recordings. There were various obstructions in the room that led to markers being obstructed from view of the camera over the time of the experiment. Figure 1 shows an excerpt from the raw data demonstrating the inconsistencies in the dataset. The x coordinate of one marker was mapped over time, and there are at least 2 gaps of around 10 seconds where there is no data being recorded.

Using the algorithm developed in Kirk et al.'s paper[4], in order to reconstruct the joint positions of the monkey, only 3 markers need to be recorded for any given frame, two on one segment and at least one on the other. For datasets involving such

inconsistencies, the flexibility of this algorithm is a perfect match for extracting the joint positions from raw, noisy data.

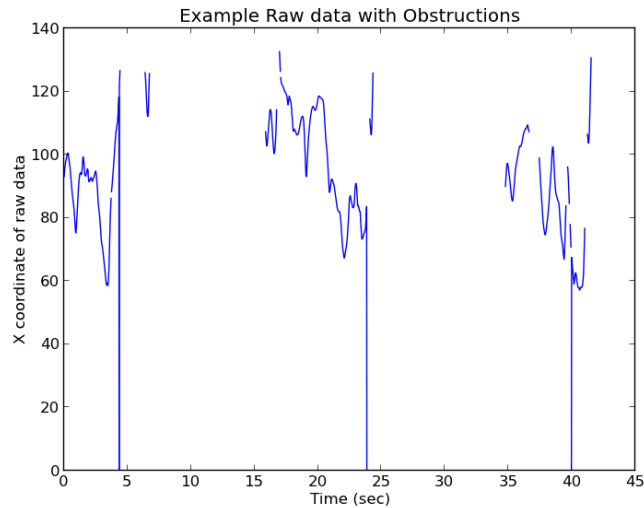


Figure 1: Example of Raw Data Inconsistencies. Figure 1 shows the x-coordinate of an arbitrary marker over time. The breaks in the line correspond to data that was not collected.

Methods:

The Phasespace Improv motion capture system was employed to track arm movements while a rhesus macaque monkey was completing natural arm movements to reach for food offered at random points by the experimenter. The monkey was allowed to take whatever path it desired to reach out, grab the food, and bring it back to its mouth. The animal performed the task for one hour with its right arm. Figure 2 demonstrates the experimental set up. The monkey's head and shoulder were fixed in place, but its arm was free and allowed to move in any direction of its choosing. The monkey was wearing a motion-tracking sleeve made up of 16 markers from which motion capture data was recorded by the motion tracking system. While performing the task, a chronic multi-electrode array was recording single unit spikes and local field potentials from both hemispheres of the animal's brain.

The Phasespace Improv motion tracking system records at 60 Hz, but upsamples through interpolation to 240 Hz. The first step was to remove the interpolated data to be left with only the real data recorded directly from the monkey rather than interpolated from the system. Data that was sorted, and if there was no data associated with a particular marker at any particular time, the data supplied by the system was masked and not used in the calculations of reconstructed arm positions.

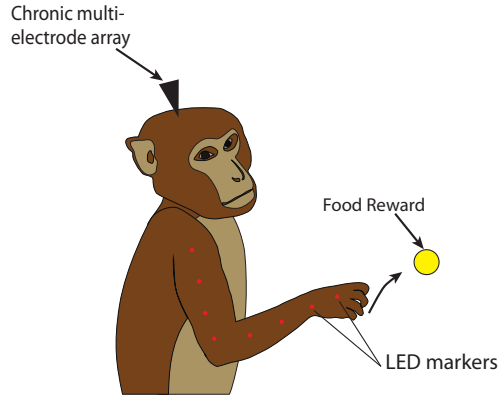


Figure 2: Experimental Setup: The animal wore an LED sleeve and performed free arm movements to obtain a food reward.

The algorithm implemented is split into two distinct steps (one using the subset of time samples and the other using all of the samples) for runtime purposes. The first step is a complex optimization, and is very computationally and energetically expensive. The output of this first step can be used to make the second step much less complex.

After the data was preprocessed, a subset of frames was selected to use for the first step of the algorithm. These frames were chosen specifically so there was no missing data included. The algorithm implemented was developed by Kirk et al. with the aim of reconstructing the joint positions of the monkey[4]. Taking advantage of fundamental distance constraints that must hold true for markers attached to an articulated structure—a structure that has a joint and obeys the laws of movement according to that joint, this algorithm is used to locate the joint position from motion capture data. Essentially, this algorithm is built upon the idea that the distance from each marker to the joint position does not change over time, as articulated in Figure 3.

To find the joint positions across the time samples, the idea that the distance between the markers and the joint position does not change is taken advantage of by using a minimization on the mean variance between the marker positions and the joint position. This process is repeated until the joint position is found that minimizes the variance in distance between the joint position and marker positions for each time sample. This step is the incredibly energetically expensive step that was mentioned above, and is only performed on a subset of the time samples.

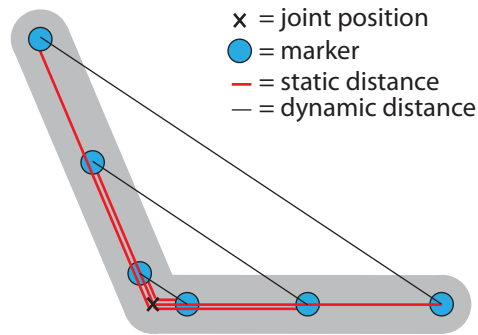


Figure 3: Articulated Structure of an Arm: The distances in red do not change regardless of the position of the arm, while the distances in black are dynamic distances.

The arbitrary solution to this step in the algorithm is to go infinitely far from the marker position because as distance goes to infinity, variance goes to zero. To encourage the algorithm to find the correct joint position rather than the arbitrary position, a distance penalty is added to the variance value [4]. The joint position of the subset of frames was then determined by a minimization function from Scipy's optimization module which minimized the variance of the distance from the marker to the joint position

After the estimated joint location was determined for the subset of frames, the distance from each marker to the joint location for each time sample was calculated. With these mean distances, in theory the geometric principle of trilateration, as described in Figure 4, can be used to more efficiently locate the joint position in the rest of the frames. But this principle can only be used if the distances are exact and since the data being used contained such inconsistencies, a more indirect route was taken to locate the joint positions. The distance from each marker to the joint is static, and known from the first step of the algorithm. Relying on this fact, a linear regression was used to find the set of joint locations that minimize the error between the known distance and the distance from the marker to the possible joint position in question. This optimization leads to the output of the estimated joint positions for all of the frames.

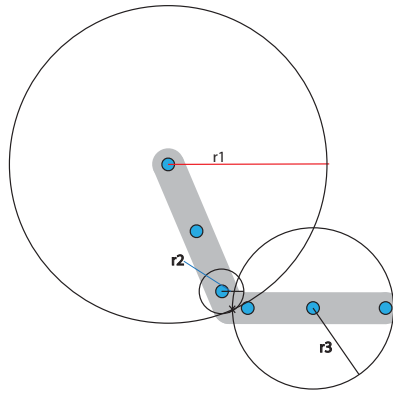


Figure 4: Geometric principle of trilateration as applied to an articulated structure.

The output of this function was the main objective of this project, but using raw data alone, there is no way to test the accuracy of the function. Two sets of data are necessary to calculate the accuracy of a function like this: the real positions of the joint, and the marker positions of the LEDs in order to find the generated positions of the joint. The function described above outputs the generated positions, but to determine how effective this algorithm is, the real position of the joint also needs to be known. For this reason, validation data was generated.

In order to generate validation data, a simulated arm with shoulder fixed at the origin was used. To find the ‘real joint position,’ random points on a sphere with a radius of the distance from the monkey’s shoulder to elbow were chosen. Another set of random points on a sphere were chosen with a new radius of the distance from the monkey’s elbow to wrist, and then each of these points was adjusted to fit the frame of reference established by the elbow position (see Figure 4). After the shoulder, elbow, and wrist positions were known, simulated marker data could be generated. For each set of arm positions, simulated markers were placed at even increments along the segments of the simulated arm. Three markers were placed on each segment of the arm. This set of simulated marker positions represented the data that was gathered by the motion capturing system.

In order to make the simulated dataset more realistic, functions were written to add Gaussian noise to the marker positions as well as take out a designated percentage of the data. The noise was added to simulate the degree of inaccuracy in the positions recorded by the motion tracking system, and to account for the obstructions in the data, the function was added to take out a specified percentage of the data.

The effect of added noise and missing data was tested by creating a set of simulated data with varying levels of noise and missing data—consisting of the simulated elbow and wrist positions, as well as the positions of the simulated markers, and measuring the accuracy of our method for estimating the joint positions. The accuracy was measured by finding the distance between the simulated elbow positions and the estimates of those positions generated by our analysis.

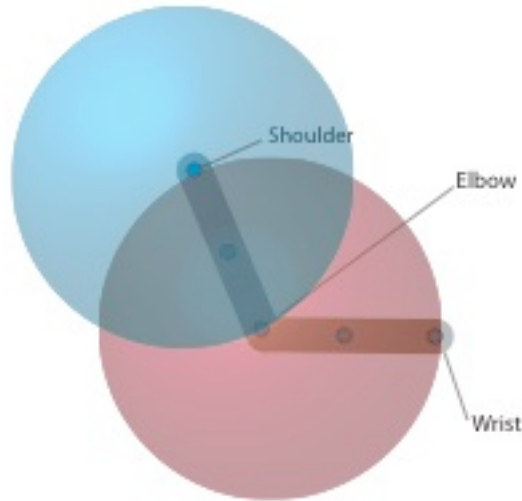


Figure 5: Process used in generating validation data. The blue sphere represents the possible elbow locations, while the red sphere chooses an arbitrary elbow position, and gives the set of possible wrist positions.

Results:

The degree to which manipulation of the noise in the data affected the accuracy of the elbow position generated was recorded, and the results are shown in Figure 6. The accuracy of the algorithm is very susceptible to inaccuracies in the recorded data. When 2.0 mm of Gaussian noise was added to the marker positions, there was an average of 30 mm of error between the real elbow position and the elbow position reconstructed from the input data.

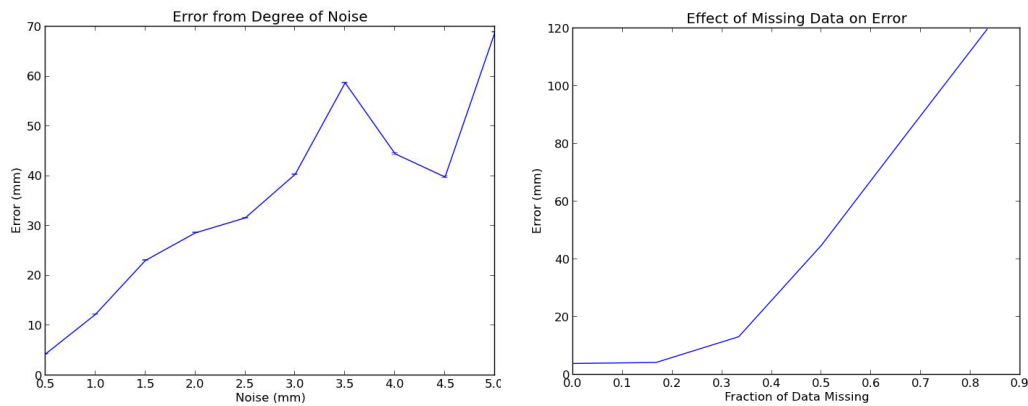


Figure 6: Degree of inaccuracies expected upon inconsistencies in the data manifesting in inaccurate marker recordings, or missing data. the error was measured as the distance between the generated elbow position and the elbow position.

Missing data affected the accuracy of the algorithm to a lesser extent than manipulation of the amount of noise. If the same threshold of acceptable error is upheld (30 mm), then 40% of the data can be missing.

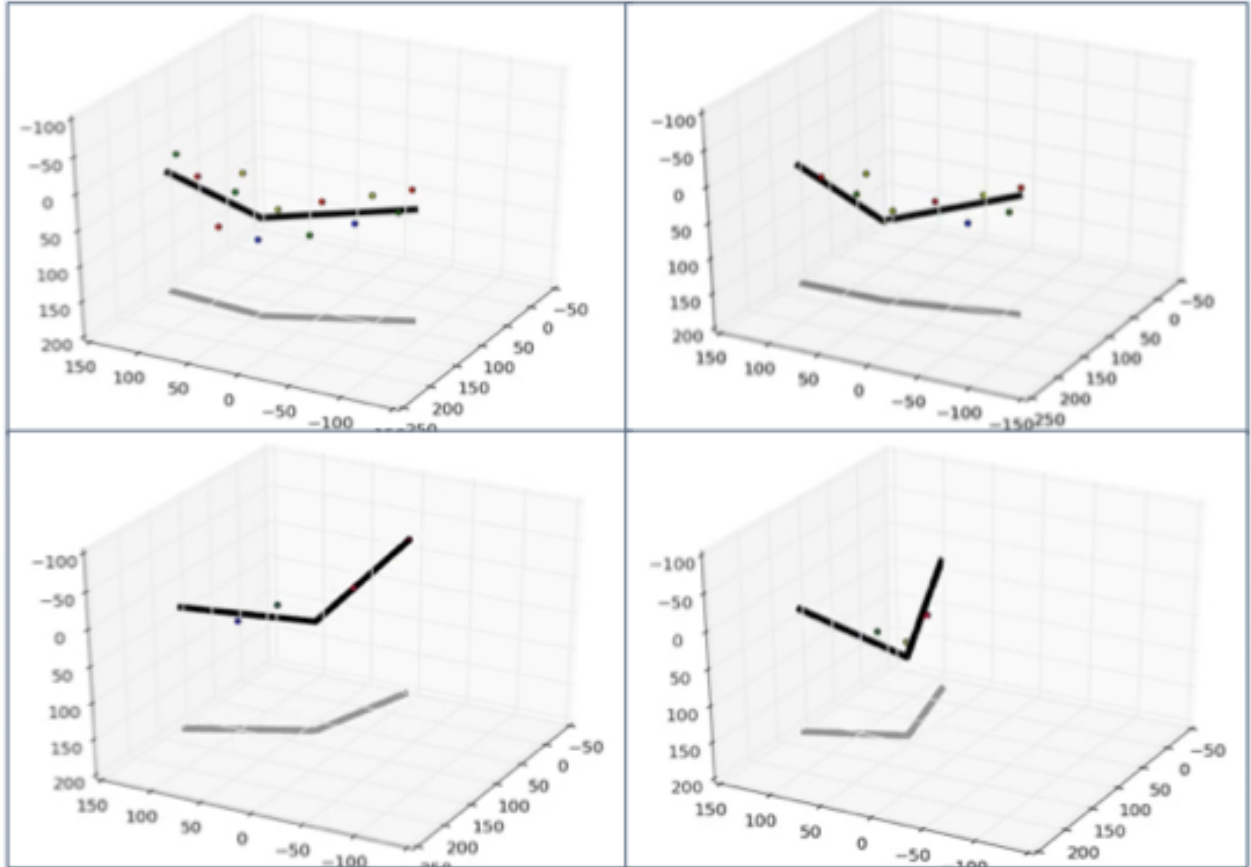


Figure 7: Example arm position reconstructions from the algorithm applied to experimental marker position data. The raw data input in the form of marker positions are shown as the multicolored dots, and the articulated structure reconstructed by the algorithm is shown in 3D by the black line (the grey line is a shadow).

Discussion:

The dataset that was being modeled by the validation data had about 20% of time samples missing, and less than 1 mm of error in the recorded marker positions according to the specifications of the Phasespace Imrov motiontracking system. When simulated data matching the raw data input as closely as possible, the error was 10.16 mm. This is within the upper threshold of acceptable error (30 mm) in this case, and the algorithm can be considered an effective way to reconstruct the figure from optical motion capture data.

In order to improve the efficacy of the validation to simulate the raw data input, the markers could be placed non-collinearly. The collinear markers give the algorithm less information to use in the ‘trilateration’ (regression) step. This problem could be fixed

by using staggered markers. The validation data could also be made more realistic by adjusting the distribution of the missing data. In the raw input data, the missing data is not distributed completely random, as certain arm positions correlate to more missing data (when the monkey brought the food to his mouth, many of the markers were missing). The validation data could be made to more accurately reflect the simulated input data.

References:

- [1] A. Miri, E. Azim, and T. M. Jessell, "Edging toward entelechy in motor control," *Neuron*, vol. 80, no. 3, pp. 827–834, Oct. 2013.
- [2] B. Bodenheimer, C. Rose, S. Rosenthal, and J. Pella, "The Process of Motion Capture: Dealing with the Data," in *Computer Animation and Simulation '97*, P. D. D. Thalmann and A. P. D. M. van de Panne, Eds. Springer Vienna, 1997, pp. 3–18.
- [3] L. Mündermann, S. Corazza, and T. P. Andriacchi, "The evolution of methods for the capture of human movement leading to markerless motion capture for biomechanical applications," *J. NeuroEngineering Rehabil. JNER*, vol. 3, pp. 6–11, Jan. 2006.
- [4] A. G. Kirk, J. F. O'Brien, and D. A. Forsyth, "Skeletal parameter estimation from optical motion capture data," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, 2005, vol. 2, p. 1185 vol. 2–.