Real-Time Tracking of Multiple People Using Unlabelled Markers and Its Application in Interactive Dance[*]

Daniel Whiteley[‡], Gang Qian[†‡], Thanassis Rikakis[†], Jodi James[†], Todd Ingalls[†], Siew Wang[†], Loren Olson[†]
[†]Arts, Media and Engineering Program and
[‡]Department of Electrical Engineering
Arizona State University
Tempe, AZ, 86287-8709

Abstract

This paper describes an approach to tracking the global locations of multiple dancers using marker-based motion capture system from only unlabeled 3D marker data for the application of interactive dance. This algorithm is based on the fact that in the presence of multiple people, marker labeling is poor or even infeasible, while reconstruction of the marker 3D coordinates is still reasonable. The tracking of dancers is done through treating dancers with markers as point clouds that have noticeable characteristics. Such characteristics include fixed marker set, all the markers being within an average wingspan, and that the markers on a dancer are closely bounded together. Using these characteristics, a blurring process was used to find the location of the dancers, and a mean shift algorithm was implemented to track them. Satisfactory results have been obtained using the proposed method.

1 Introduction

Movement-based interactive dance has recently attracted great interests in the performing arts. In traditional dance, dancers need to follow the background music with pre-choreographed steps and movements. Although improvisation is possible, the background music or visual effects will not change automatically according to the dancers movements. In a movement-based interactive environment, the dancers have full control of the background music and visual display through their movements. A multitude of cues related to human moments have been used to drive such interactive systems. Camerri et al. [1] have used the Laban movement qualities [2] to guide the background music synthesis. Woo [3]

[*] This material is based upon work supported by the National Science Foundation under CISE-RI No.0403428.

extracted limb movement in terms of speed and trajectory in nine cubes centered at human body. In [4], a number of static human body gestures (poses) were used to drive the interactive system.

In this paper, a multiple people tracking algorithm is presented using a marker-based 3D motion capture system in the application of interactive dance. Unlike most of the movement-based interactive system, this system looks at the spatial relationship of multiple dancers (including those with markers and without markers) and uses these spatial relationships as cues for dancers to communicate with the interactive system.

The choreography we are working with involves how multiple dancers interact between each other as the metric that interfaces with the multimedia elements. The choreographer has created a work which interacts with the music and animation that would be played during the show. This interaction is driven by the motion capture data of the dancers on stage, which is interpreted into metrics. The metrics used in this piece are as follows:

- *Proximity*: How close two dancers are to each other.
- *Translational correlation*: How similar two dancers are traveling through the capture volume
- *Grouping*: If the dancers are separated into close groups of duets or trios, or if they are separated
- *Activity rate correlation*: How similar are the choreographed motions between two dancers in terms of body-centered limb movement rates.

A marker-based motion capture system with 16 digital cameras is used to provide real-time marker positions in global space. In the performance, seven dancers were present with four of them with sixteen markers on each person's upper body. Marker positions are shown in Figure 1. When the marker labels are correctly recovered, the above quantities can be easily derived. However, this approach has revealed some complications and challenges in reliable people tracking in real-life performance.

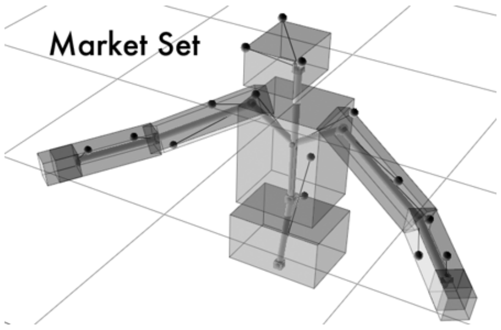

Figure 1: Marker set used in multiple people tracking

1.1 Challenges

Upon implementing this system during a trail run of the performance, it was found that not enough labeled markers could be recovered in order to compute accurate metrics for use. Three factors were identified to be the causes of the poor labeling.

The first part was the change in the capture volume. A 16-camera system was used in a proscenium theater where the performance was actually to be held. The theater was larger than the capture space used previously with this system, had lighting fixtures which interfered with the cameras, and the stage's marley was too reflective to infrared, which all contributed to an imperfect camera setup. This leads to a noisier and less reliable data retrieval that affected labeling.

Second, there were multiple subjects that were entering and exiting the capture space. For instance, when three out of the four dancers exited the stage, the system became confused looking for all the dancers, including the absent ones, which caused difficulty in labeling.

Lastly, in many instances of the dance, the dancers were in very close proximity, where there was a lot of occlusion, marker crossovers, and trajectory overlaps, which all lead to mislabeled or unlabeled markers. As a whole, these factors contributed to a situation such wherein every metric calculated was inaccurate. It was observed, though, that the reconstructed marker coordinates were more stable, but unlabeled. This leads to an investigation of the use of the unlabeled data together with existing clustering methods.

Since we could no longer fully rely on the labeled data, a two-layered approach was devised. The top layer is the original system that was created, which used the labeled data. The sub-layer though, will use the stream of unlabeled reconstructed points. Instead of having the skeletons of specific dancers, the markers will be divided into clustered point clouds, which will be tracked. The metrics of proximity, translational correlation, and grouping can all be derived from tracking the cluster centers. Activity rate correlation between two people is derived by looking at the marker distribution in the tracked marker blobs related to the two dancers.

2 Tracking People Using Mean Shift

An efficient method of searching and tracking point clusters is necessary to implement this method in real-time. These clusters must also accurately approximate where the real dancers are actually located. A mean shift algorithm was found to achieve both of these goals.

2.1 Mean shift: an introduction

The sample mean of samples S under a kernel K(x) centered at t, with sample weights w(s), can be found using this equation:

$$m(t) = \frac{\sum_{s \in S} K(s-t)w(s)s}{\sum_{s \in S} K(s-t)w(s)} \tag{1}$$

where m(t) is the new sample mean [6]. Mean shift is the process of repetitively shifting the center t to the sample mean. In [5], they prove that if the kernel K(x) has a profile k(||x||²), and k is convex and monotonically decreasing, then the center t will converge onto a single point.

2.2 Design of the kernel

Using the properties of the dancers' marker sets, mean shift can be used to converge onto the location of each dancer. These properties are:

a) The dancers markers are within arm's length of the center of the torso,
b) Most of the markers are close to the torso, and
c) Each dancer has the same number of markers.

To take property (a) into account, a flat kernel can be used as shown in the equation below:

$$F_\lambda(x) = \begin{cases} 1 & if \ \|x\| \le \lambda \\ 0 & if \ \|x\| > \lambda \end{cases} \tag{2}$$

The threshold λ is the average arm's length of a dancer.

Since most of the markers are close to the dancer's body, more consideration, or weight, should be placed on the markers that are closest to the center. To account for that, a Gaussian kernel as shown below can be used:

$$G_\beta(x) = e^{\frac{-\|x\|^2}{\beta}} \tag{3}$$

The parameter β was set as twice the average variance of the distribution of the dancer's marker set. The final kernel used in the tracking algorithm was a combination of the two kernels, giving us:

$$(F_\lambda G_\beta)(x) = \begin{cases} e^{\frac{-\|x\|^2}{\beta}} & if \|x\| \le \lambda \\ 0 & if \|x\| > \lambda \end{cases} \tag{4}$$

2.3 Weighting the markers

It is also possible that the markers of multiple dancers to be mixed together. To reduce the error incurred by this mixing, a weight was given to each marker that was the probability that a maker is "owned" by any particular cluster center. Since the true ownership of the marker is not known, it is equally likely for any cluster center that is within arm's length of the marker to own it. Thus, the probability that a marker is part of a cluster is the inverse of the number of centers that are in range of that marker. This probability can be calculated using the flat kernel, as shown below:

$$w_F(s) = p(s \in C_t) = \frac{1}{\sum_{t \in T} F_\lambda(s-t)} \tag{5}$$

Where T is the set of cluster centers, S is the set of markers, and C_t is the set of markers that are in arm's length of center t.

This weight, however, was not sufficient for marker tracking. Using this weight alone, the centers tended to merge when the dancers became close to each other. This was obviously undesirable. *K*-means would prevent this merging, but the sharing of markers would be eliminated, which is also undesirable. In [7], a clustering algorithm is discussed which is used to maximize the entropy. In [6], the author observed that this "max entropy" algorithm is a mean shift algorithm with a Gaussian kernel $G_\beta(x)$ and a weight $w_G(s)$ equal to:

$$w_G(s) = \frac{1}{\sum_{t \in T} G_\beta(s-t)} \tag{6}$$

It was also mentioned that as β goes to zero, the algorithm degenerates into *k*-means clustering [7]. Since it is desired that the centers share markers without merging together, a modified version of the max entropy algorithm can be used. Instead of using p(s) in equation (5) as the marker weight, the weight can be calculated using:

$$w_{FG}(s) = \frac{1}{\sum_{t \in T} (F_\lambda G_\beta)(s-t)} \tag{7}$$

In the final implementation, a truncated kernel $(F_\lambda G_\beta)(x)$ and weight $w_{FG}(s)$ were used in the tracking. The parameter β was incrementally decreased until the cluster centers stopped merging.

2.4 Creation of clusters through blurring

In order to track the dancers' locations, their initial locations first need to be found. In [6], the author mentions a deterministic way to determine the natural clusters in the data is through a blurring process. The blurring process is simply the mean shift algorithm, where the cluster centers and the data points are one and the same (i.e. the center set T is the same as the data set S, and both evolve with each iteration of mean shift). Once the process has converged, the data set will be tightly packed into clusters, with all of the data points located closely to the center of that cluster. This process is fully deterministic, which is more favorable than random seeding used in most clustering methods. Using a blurring process with a truncated Gaussian kernel and no additional weights has shown to do well in finding the initial location of new dancers. The radius λ and the Gaussian parameter β were both the average distance between markers on a dancer.

At times, though, this blurring process cannot initially separate two dancers when they are close together. As well, in the middle of the performance, a dancer can enter the stage closely behind another dancer, and the markers of that dancer will be falsely included with the markers of the dancer already identified on stage. The solution to this problem is to "split" this cluster when the dancers are well enough separated to be clustered separately. "Splitting" the cluster is simply re-clustering all of the markers of that cluster using the blurring process. The measures used to indicate whether a cluster needs to be split were the variance of the markers and the effective number of markers. The effective number of markers for a cluster is defined as the statistical expected number of markers for that cluster, which is calculated using the formula below:

$$E[n(C_t)] = \sum_{s \in S} F_\lambda(s-t)\, p(s \in C_t) = \sum_{s \in S} \frac{F_\lambda(s-t)}{\sum_{t \in T} F_\lambda(s-t)} \qquad (8)$$

2.5 The clustering/tracking algorithm

The algorithm is iterated for each frame of motion capture data, where a frame is the time period between updates of the marker positions. The full algorithm for finding and tracking the locations of the dancers is thus:

1) Use mean shift formula in equation (1), with the truncated Gaussian in equation (4) and the weight in equation (7) to converge on the current cluster centers using the new marker position data.
2) Calculate the effective number of markers for each cluster using equation (8). If the effective number of a cluster is too small, delete that cluster.
3) If one or more cluster centers were deleted in step 2, go to step 1.
4) Find all markers that are not within arm's length of any center. These are considered to be "orphan" markers (markers without a cluster to go to).
5) Check the effective number of markers and the marker variance of each cluster. If both of these measures are above a threshold, delete that center and include all the markers that are within arm's length of that center in the set of orphan markers.
6) Perform blurring on the orphan marker set to find new clusters. The mean shift algorithm used in the blurring process has a truncated Gaussian kernel in equation (4) but no additional weights.
7) All new clusters that contain whose marker count is more than half of the number of markers in a dancer's marker set is included in the set of dancer locations. The rest that do not fill this requirement are discarded.
8) Perform mean shift using the new set of cluster centers.

3 Experimental results

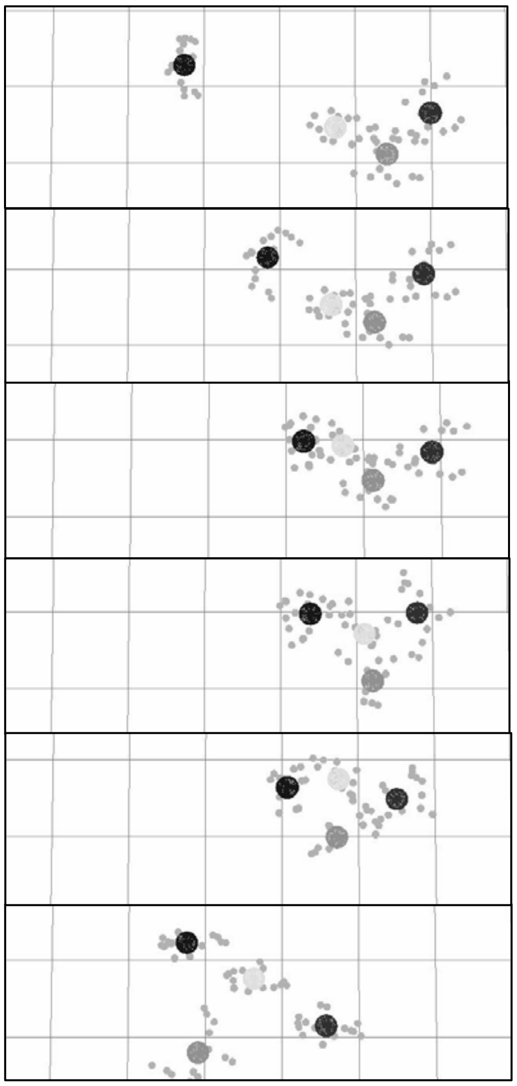

Figure 2: Tracking of four dancers. In close quarters, they still maintain their identity.

It was required that the algorithm detects and tracks the dancers entering and exiting as point clouds, and be able to track their positions while they are in the space. As well, this process must also happen within the time period between frames so that it can be continuously tracked in real time. The results of the algorithm were fairly successful. Using a C++ implementation on a 2GHz Celeron machine, the processing of the algorithm took only a 15ms. This translates into a frame rate over 60 frames per second, which is the frame rate necessary for real time. The results of tracking the dancers' locations were done qualitatively. Once in the space, they were continuously tracked, even when close together. Figure 2 shows the clustering and tracking results of four dancers over roughly 200 frames (approximately 7 seconds). The top-down views are shown here. During the tracking process, each dancer is identified using a sphere of unique gray level. As it can be seen in the figures, although sometimes all four of the dancers tightly group together, they can still be correctly tracked. On rare occasion, however, where the cluster centers changes which dancer it tracks. This seems to occur most when the dancers are tightly grouped together, where it is hard to differentiate between dancers using the point cloud characteristics that have been previously defined. Future plans to correct this error include creating the cost from a prediction using a constant velocity/acceleration model of the dancer instead of simply using the past frame.

4 Conclusions/Future Work

A multiple people tracking algorithm using a mean shift algorithm has been developed and implemented. Though not perfect, it shows great promise in tracking multiple dancers real-time in fast and complex movement as well as compensating for dancers going on and off the stage. Further work to be done is to improve the tracking algorithm by accounting for kinematics. This system will then be implemented in calculating the metrics needed in the interactive dance program. Also, further research is being done on using the clusters as a way to segment the points for improved tracking and labeling.

References

[1] Camurri, A., Hashimoto, S., Ricchetti, M., Ricci, A., Suzuki, K., Trocca, R. and Volpe, G., "EyesWeb: Toward Gesture and effect Recognition in Interactive Dance and Music Systems". Computer Music Journal. 24(1), 57-69, 2000

[2] Moore, C.-L., Yamamoto, K. *Beyond Words: Movement Observation and Analysis*. Gordon and Breach Science Publishers, New York, 1988.

[3] Woo, W., Kim, N., Wong, K., and Tadenuma M. "Sketch on Dynamic Gesture Tracking and Analysis Exploiting Vision-based 3D Interface". Proc. SPIE PW-EI-VCIP'01, vol. 4310, pp. 656-666, 2000

[4] G. Qian, F. Guo, T. Ingalls, L. Olson, J. James and T. Rikakis, "A Gesture-Driven Multimodal Interactive Dance System," in Proceedings of the International Conference on Multimedia and Expo, Taibei, Taiwan, China, June 27-30, 2004

[5] D. Comaniciu, V. Ramesh, and P. Meer, "Real-Time Tracking of Non-Rigid Objects using Mean Shift", *IEEE Conf. Computer Vision and Pattern Recognition (CVPR'00)*, vol. 2, pp. 142-149, Hilton Head Island, South Carolina, 2000.

[6] Y. Cheng, "Mean Shift, Mode Seeking and Clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790-799, August 1995

[7] K. Rose, E. Gurewitz, and G.C. Fox, “Statistical mechanics and phase transitions in clustering,” Physical Review Letters, vol. 65, pp. 945-948, 1990

Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF).