Desktop Motion Capture Using Color Recognition

NATHAN BLACK
CSC 590
Advisor: Franz Kurfess

1 Introduction

This paper introduces a system for creating key frame animations from poses captured from a single camera. The work presented here is an extension of a system that uses a color glove for real time hand tracking. The basic idea being extended is that the pose estimation problem can be simplified by the addition of color data to the object being tracked. The main contributions of this paper are the extension of the color tracking scheme to the full human figure, and optimizations specific to key frame animation.

Wang and Popović convincingly show that reasonably accurate pose estimations can be performed in real time by querying a database of known poses sampled from a movement space. One of the attractive features of their system is that is relatively low cost and suitable for consumer product applications, requiring only a single camera setup [19]. In principle most of the features of our system mirror their work.

The main components of our system are a camera and a miniture, posable human figure. Images sampled from the camera are used to reconstruct a pose, which is then applied to a 3D model. The posable figure becomes an input channel for manipulating and animating the on screen model.

Our original contribution is an adaptation of Wang and Popović's system to the task of key frame animation, and accuracy improvements related to the chosen domain. The original system calculates the pose every frame, which causes significant jitter even with the application of temporal smoothing. We show that reduction in jitter can be achieved by leveraging the assumption that the captured poses will be used for animation frames rather than real time input. Specifically, we use threshold values to require a chosen distance from the previous pose before performing a recalculation.

We also modify the distance metric to be asymmetric, more heavily weighting the query image to database image distance. We hypothesize that will make the system more robust against loss of color information due to occlusion of the model by the user's hands.

Finally, we attempt to improve the system's accuracy in estimating distance using known geometric properties of the model.

Implementation of our system will follow several stages. First, we leverage the availability of existing high quality motion capture data to generate our movement space. Using known techniques, we take a uniform sample of poses in this space. We use the motion capture data to pose a 3D representation of our figure. Once the poses have been generated, we skin the model with a color pattern according to the algorithm proposed by Wang and Popović. After the model is skinned, we generate tiny rasterized images from each pose. These rasterizations can then be used for nearest neighbor comparisons from data captured by the camera.

The second phase is programming the image processing pipeline. This involves de-noising the captured input image, classifying the image according to color region, and normalizing the classified image. By normalization we mean cropping the image and scaling to match the size of the tiny rasterized images in the database.

Finally we implement the nearest neighbor comparison of images using our modified distance metric and a chosen subset of optimizations proposed by Wang and Popović.

2 Research Validation

There are two main types of validation we wish to use against our system. The first is qualitative. We want to find out how closely we can match the correct pose. Second is performance. We want to perform our calculations as quickly as possible, ideally at real time interactive rates.

While these two goals are mutually conflicting, they provide guidelines for what can be considered acceptable results. We are trying to build an interactive system. many applications, such as applying captured poses to animations. The system must be fast enough to facilitate interactive manipulation. While we would accept less than real time performance for pose capture, we would like real time performance to be achievable for the system in the near future. Thus we will use real time rates as a basis for comparison in performance testing. Specifically, we will try to match Wang and Popović's published results of ten updates per second [19].

On the other hand, it is necessary to be reasonably accurate because otherwise the data will not be useful for building animations. Here again we will draw upon published results for comparison.

2.1 Hypothesis

Drawing on previous work, we apply a database search method of pose estimation for the human hand to pose estimation of the human figure. It seems likely that we can achieve similar results in terms of speed and accuracy. There are several differences, however, that we can hypothesize. First, there may be less variance, on average, between individual frames of captured data when looking at the full human figure. This is due to the fact that the human body moves less quickly than the hand moves by itself. In particular, most of the body stays still during the majority of motions we are likely to track. On the other hand, there is a greater variation in the possible articulations of the body in comparison to the hand by itself, even if we do not allow articulation of the fingers in our model. Given this growth of the movement space, we want to increase the size of the pose database as much as necessary to achieve sufficient accuracy to cover the full range of body movements. We suspect that this increase will be necessary to achieve the required accuracy.

The literature gives us an estimate we can use as a starting point for the size of our database. Given a database of 100,000 poses, we can try to reproduce similar performance and accuracy to the hand tracking system. Since there is evidence that increasing the database size increases accuracy of predictions [19], we will attempt to verify by varying the number of allowed comparisons in our database query.

We think that previous successes can be duplicated for poses that are difficult to distinguish given only silhouette data, such as similar back facing and front facing poses. However, motions involving articulations of the extremities may be difficult to accurately match given a sparse database. If the results are less than adequate, we can attempt to improve accuracy through the use of inverse kinematics and by increasing the size of the database.

Based on results from the literature, we expect accuracy to increase as we increase the size of our pose database, roughly proportional to the log of the number of added images. We expect accuracy gains from inverse kinematics to be complementary to improvements achieved by increasing the size of the database. We should, however, be able to define a threshold at which introducing inverse kinematics give a better accuracy improvement than increasing the size of the database given the same performance tradeoff.

When building the pose database we will only sample from half subset of the available motion capture data. For evaluation, we select animations from the other half of the data.

3 Independent and Dependent Variables

For the method we wish to test, the control we have over the results hinges on control of the pose matching database. Increasing the number of poses in the database increases both the accuracy of the estimated pose, and the time to calculate the estimation. Thus the independent variable N is the number of poses in the database, t the average time to perform a pose estimate and v the error in the estimate are dependent variables.

4 Measures

We can measure error distance of pose estimation as follows. We map the motion capture data to a rigid body model. We then render the model animation using the data from the motion capture to define the animation sequence. The frames of the animation are then fed into the pose estimation program as input. During the evaluation step, we compare the estimated pose to the known data from the motion capture. The root mean square error distances between corresponding vertex pairs from the original motion capture and the output pose estimate define the error of the estimate. This error will be defined and measured on a per frame basis and averaged for the entire animation.

We derive our data set from Carnegie Mellon University's motion capture project. The data is available in several formats and we choose a convenient representation. Because the data in this project was generated using high quality capture from multiple cameras, for the purposes of the experiment we consider the data canonical.

There are several advantages to this approach. First, it is relatively easy to generate data for the test in this manner, rather than having to capture motions, existing data can be reused. Second, if we choose a good enough dataset, it may be reusable in future efforts in this area of research. This would lead to results of various approaches being more directly comparable.

5 Experiment Protocol

The pose lookup database is fixed at the start of the experiment. Several animation sequences are generated from data as described above. It is critical that data from the test sequence was not included in the database, which would skew the results, making them artificially more accurate. The animation sequence is fed into the pose estimation system as input.

The data is input using several methods in sequence. In the first run, the animation is fed in without alteration. In the second run, random noise is added to the animation to simulate noise from camera capture. In both cases, the animation is displayed on a screen and captured by a camera.

During each run the system generates as output an estimate of the pose represented by each frame of the animation. A frame by frame comparison is performed by the test system to calculate the mean square error between corresponding points in the motion capture data and the estimation system output. Both the estimation output and the error are saved as the results of the test run. In addition to the accuracy measure, the time in milliseconds for each calculated frame is saved.

In place of a smoothing algorithm, we attempt to reduce jitter by introducing a distance threshold. We take both accuracy and time measurements with the threshold comparison turned on and off, respectively, and compare this with the results of temporal smoothing described by [19].

In addition to quantitative measures, we will include a visual qualitative comparison. For this purpose we render the results of our pose estimation and overlay against the captured images of the figure from two separate camera positions. This will give us an idea of the system's performance in depth estimation accuracy and a qualitative assessment of jitter.

6 Related Work

This paper introduces a system for pose recognition. It is primarily an extension of real-time hand tracking research done by Wang and Popović, which demonstrates a system that can achieve interactive rates while tracking the human hand. Their methodology is unique in that it uses color patterns to speed up the process of visual recognition by the machine. In contrast, other systems use multiple cameras, computationally expensive inference algorithms, or cumbersome gloves with multiple sensors [19]. A demonstrated application of their system is real time animation of figures [18].

Their system uses data derived from a sensor unit to build a database of hand poses. Images captured in real time are then compared in real time to images in the database, and the closest match is used for pose estimation. A single camera is used to capture two dimensional images of a human hand wearing the glove[18].

This color technique had been employed in a more limited sense by [5] to recognize sign language gestures. Wang and Popović demonstrate that their system can accomplish the same task in real time.

Pattern recognition has been used to track deformable surfaces in real time.[10]. Other efforts have focused on the use of fiducial markers to estimate pose data. Such systems in practice typically require a large proportion of an objects area to be covered by markers in order to capture a pose, increasing with the number of possible joint articulations [8].

Mohan, et al. introduce bokodes as a method of pose capture. Bokodes are essentially a miniaturization of fiducial markers that are indistinct to the human eye, but recognizable by an out of focus camera. By embedding discreet light emitting markers in ordinary clothing, positional as well as orientational data can be reliably captured. This method requires the use of a second, out of focus camera to capture input from the markers from the same position as the original image. In principle, only the out of focus camera would be needed for input to a system that was used only for animating figures [14].

Mohan, et al. have demonstrated the stability and accuracy of bokodes compared to traditional fiducial markers. However, they point out that individual bokodes can only be recognized from a maximum viewing angle of 20 degrees.

The advantages of the fiducial marker systems, including bokodes, are that they provide not only spatial data, but also feature identification. Because each fiducial marker can be distinctly recognized, poses of objects corresponding to heirarchical models can be accurately estimated. The system described by [18], by contrast, treats each pose as a monolithic entity, trading accuracy for speed of recognition.

Several motion tracking systems have been built from consumer products, while several others are in commercial development. Lee demonstrates input taken from tracking of human fingertips using reflected infra red light input to a Nintendo Wiimote. In this method, the fingertips are tracked as points, though conceivably some inverse kinematics could be applied to estimate a pose of the whole hand. The system described by Lee can track a maximum of six points per Wiimote device but does not identify the points [13]. The Sony Playstation Eye provides the hardware base for several motion input systems. Currently in development is the Playstation Motion Controller, which tracks a color-changing sphere to detect motion, including depth through simple calculations. Microsoft has demonstrated an unreleased project codenamed Natal which is capable of multiple targets and reproducing the pose of a human figure.

References

[1] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, pages 34–47, 2001.

[2] R.T. Azuma et al. A survey of augmented reality. *Presence-Teleoperators and Virtual Environments*, 6(4):355–385, 1997.

[3] D.A. Bowman, D. Koller, and L.F. Hodges. Travel in immersive virtual environments: An evaluation of viewpoint motion control techniques. In *IEEE, Proceedings, 1997 Virtual Reality Annual International Symposium.*

[4] D.A. Bowman, E. Kruijff, J.J. LaViola, and I. Poupyrev. *3D user interfaces: theory and practice.* Addison-Wesley Boston, 2004.

[5] B. Dorner. *Chasing the colour glove: Visual hand tracking.* PhD thesis, Simon Fraser University, 1994.

[6] K. Eng. Miniaturized Human 3D Motion Input.

[7] K. Eng. A Miniature, One-Handed 3D Motion Controller. *Lecture Notes in Computer Science*, 4662:574, 2007.

[8] M. Fiala. Artag revision 1, a fiducial marker system using digital techniques. *National Research Council Publication*, 47419.

[9] M. Fiala. ARTAG Rev2 Fiducial Marker System: Vision based Tracking for AR. In *Workshop of Industrial Augmented Reality*, 2005.

[10] T. Heap and F. Samaria. Real-time hand tracking and gesture recognition using smart snakes. *Proceedings of Interface to Real and Virtual Worlds*, 1995.

[11] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, volume 99, pages 85–94. San Francisco, CA, 1999.

[12] J.C. Lee. Hacking the nintendo wii remote. *IEEE Pervasive Computing*, pages 39–45, 2008.

[13] J.C. Lee. Wiimote Project. http://johnnylee.net/projects/wii/, 2009.

[14] A. Mohan, G. Woo, S. Hiura, Q. Smithwick, and R. Raskar. Bokode: imperceptible visual tags for camera based interaction from a distance. In *ACM SIGGRAPH 2009 papers*, page 98. ACM, 2009.

[15] W. Piekarski and B.H. Thomas. Using ARToolKit for 3D Hand Position Tracking in Mobile Outdoor Environments.

[16] D. Schmalstieg. The Studierstube Augmented Reality Project.

[17] D. Schmalstieg, A. Fuhrmann, G. Hesina, Z. Szalavári, L.M. Encarnaçao, M. Gervautz, and W. Purgathofer. The studierstube augmented reality project. *Presence: Teleoperators & Virtual Environments*, 11(1):33–54, 2002.

[18] R.Y. Wang. Real-Time Hand-Tracking as a User Input Device.

[19] R.Y. Wang and J. Popović. Real-time hand-tracking with a color glove. In *ACM SIGGRAPH 2009 papers*, page 63. ACM, 2009.

[20] S. Zhai. User performance in relation to 3D input device design. *ACM Siggraph Computer Graphics*, 32(4):50–54, 1998.

[21] S. Zhai, P. Milgram, and A. Rastogi. Anisotropic human performance in six degree-offreedom tracking: an evaluation of 3D display and control interfaces. *IEEE Transactions on Systems, Man and Cybernetics*, 27:518–528, 1997.