

Real-time 3D Motion Capture

Thanarat Horprasert, Ismail Haritaoglu
David Harwood, Larry Davis
Computer Vision Laboratory
University of Maryland,
College Park, MD 20742 USA

{thanarat,hismail,harwood,lsd}@umiacs.umd.edu

Christopher Wren
Alex Pentland
MIT Media Laboratory
20 Ames Street,
Cambridge MA 02139 USA
{cwren, sandy}@media.mit.edu

Abstract

We describe a real-time 3D computer vision system for detecting and tracking human movement and providing a person with control over the movement of a virtual puppet. Multiple cameras observe a person; silhouette analysis and template matching achieve real-time 3D estimation of human posture. A dynamics/kinematics model of human body and Kalman filter are utilized to help the tracking process as well as to interpolate some joint locations. All estimation and rendering processes run in real-time on a PC based system.

1. Introduction

This research note describes our ongoing research and a demonstration at SIGGRAPH98 Emerging Technologies. The project is the result of collaboration among ATR's Media Integration Research Laboratory, the University of Maryland's Computer Vision Laboratory, and Massachusetts Institute of Technology's Media Laboratory.

Our project is a real-time 3D computer vision system for detecting and tracking human movement. It provides a

person with control over the movement of a virtual computer graphics character. A version of the previously developed W^4 system [1], extended to operate on color images, is run on each of the multiple cameras observing a person. Its silhouette analysis and template matching achieve real-time 3D estimation of human postures. The estimated body postures are then reproduced in a 3D graphical character model by deforming the model according to the estimated data. Dynamics/kinematics model of human motion [2] and Kalman filters are utilized to help the tracking process as well as to interpolate some 3D joint locations (i.e. elbows). The system runs on a network of Dual-Pentium 400MHz PCs at 20-30 frames per second (depending on the size of person whom the system observes).

2. System Overview

The block diagram of the system is shown in Figure 1. A set of color CCD cameras observes a person. Each camera is connected to a PC running the W^4 system. W^4 performs background subtraction, silhouette analysis and template matching to detect and track 2D locations of salient body parts, e.g., head, torso, hands, feet, etc. At a central

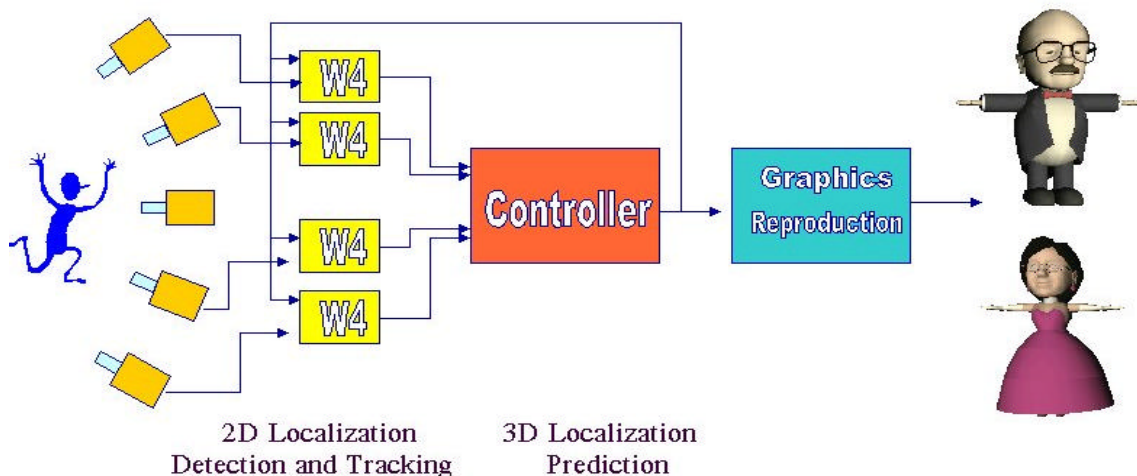


Figure 1. The block diagram of the real-time 3-D motion capture system.

controller, the 3D localization of these body parts is obtained by triangulation and optimization processes. A dynamics/kinematics model of human motion is also used to smooth the motion trajectory as well as to predict the locations of the body parts for the next frame. Those predictions are then fed back to the W^4 systems to control tracking.

To interact, a person enters the demonstration area and momentarily assumes a fixed posture that allows the system to initialize (i.e. locate the person's head, torso, hands, and feet). They then are allowed to move freely in the area. The trajectories of their body parts are used to control the animation of a cartoon-like character. Whenever the tracking fails, the person can reinitialize the system by assuming a fixed posture at the center of the demonstration area.

3. W^4 and Its Extension in Background Subtraction

W^4 [1] combines shape analysis and robust techniques for tracking to detect people, and to locate and track their body parts (head, hands, feet, and torso). W^4 consists of five computational components: background modeling, foreground object detection, motion estimation of foreground objects, object tracking and labeling, and locating and tracking human body parts. The original version of W^4 , which is designed for outdoor visual surveillance systems, operates on monochromatic video sources taken from a stationary camera. In this project, we extended W^4 to work on color images by developing a new method of background subtraction that is robust and efficiently computed.

3.1 Background Modeling and Foreground Detection

The background scene is statically modeled by representing each pixel by its mean color (μ_R, μ_G, μ_B) computed over a sequence of frames observed during the background learning period. The global standard deviations of each color band ($\sigma_R, \sigma_G, \sigma_B$) are also determined. After the background is modeled, each pixel in the image is classified into four categories - original background (B), shaded background (S), highlighted background (H), or moving foreground object (F)- by a new method of pixel classification.

For a pixel, x , we define $\alpha(x)$ to be value that brings a new color observation closest to the background color at a pixel. For each pixel x , we compute $\alpha(x)$ by minimizing

$$\phi_x(x) = \sum_{c=R,G,B} [(\alpha(x) \mu_c(x) - I_c(x)) / \sigma_c]^2$$

where $\mu_c(x)$ is the mean background color (R,G,B) values of the pixel x , $I_c(x)$ is the observed color (R, G, B) value of

the pixel x in an incoming image, and σ_c is average standard deviation of color (R, G, B) over the entire image.

Next, we define three kinds of distortion, the observed color distortion (D_c) representing the normalized distance between I and μ , the estimated color distortion (D_{ec}) representing the normalized distance between I and $\alpha\mu$, and the brightness distortion (D_α) representing the normalized distance between $\alpha\mu$ and μ .

$$D_c(x) = \sum_{c=R,G,B} [(I_c(x) - \mu_c(x)) / \sigma_c]^2$$

$$D_{ec}(x) = \sum_{c=R,G,B} [(I_c(x) - \alpha(x) \mu_c(x)) / \sigma_c]^2$$

$$D_\alpha(x) = \sum_{c=R,G,B} [(\alpha(x) \mu_c(x) - \mu_c(x)) / \sigma_c]^2$$

Base on these values, the pixel is classified into one of the four categories {B, S, H, F} by the following two-stage decision procedure.

Stage I: If the observed color distortion is small ($D_c(x) < \epsilon_c$), that means both color and brightness of the observed pixel are similar to the expected values of the pixel. The pixel is classified as an ordinary background pixel (B). This efficient test allows us to quickly identify most background pixels. If the test fails, then the pixel is one of the {S, H, F}; we determine its class based on the other distortion measures.

Stage II: If the estimated color distortion is too high ($D_{ec}(x) > \epsilon_{ec}$), the pixel is classified as a foreground pixel (F). Otherwise, the observed color, I_c , is similar to the background color μ_c , but may be either darker or brighter. This can be determined from α ; it is darker if $\alpha < 1$, otherwise it is brighter. Nevertheless, the pixel still cannot be classified as either a shaded background (S) or a highlighted background (H). We must first test the brightness distortion. If the brightness distortion is sufficiently high ($D_\alpha(x) > \epsilon_\alpha$), we classify it as a foreground pixel (F). Otherwise it is classified as either a shaded background or a highlighted background pixel depending on whether it is darker or brighter than the expected value. $\epsilon_c, \epsilon_{ec}$, and ϵ_α are constant essentially set at the 3σ level during training. .

3.2 Silhouette Analysis and 2D Body Part Localization

After the background scene is modeled and the foreground region is segmented. A geometric cardboard human model of a person in a standard upright pose is used to model the shape of the human body and to locate the body parts (head, torso, hands, and feet)[1]. Those parts are then tracked using template-matching methods. After

predicting the locations of the head and hands using the cardboard model, their positions are verified and refined using dynamic template matching. Their color templates are then updated, unless they are located within the silhouette of the torso. In

this case, the pixels corresponding to the head and hand are embedded in the larger component corresponding to the torso. This makes it difficult to accurately estimate the

controller then computes the 3D localization of the body part by performing a least square triangulation over that set of 2D data which has confidence values higher than a threshold. We treat each body part separately; i.e. at a certain frame, the 3D position of the right hand and the left

hand may be obtained from triangulation of different subsets of the cameras. The camera calibration and

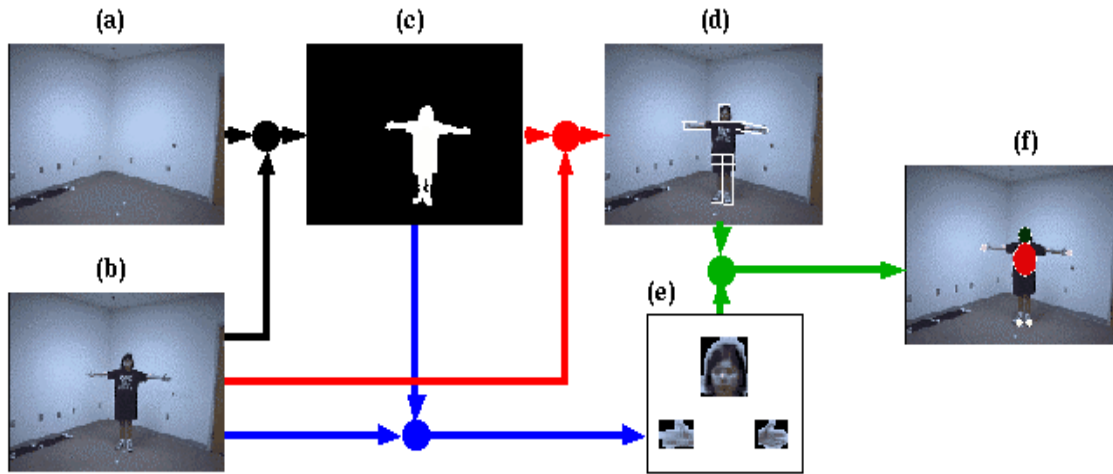


Figure 2. 2-D body part localization process triangulation are implemented following methods proposed nization method.

median position of the part, or to determine which pixels within the torso are actual part pixels. In these cases, the parts are tracked using correlation, but the templates are updated only using skin color information and the location prediction comes from the 3D controller. Figure 2 shows the body part localization algorithm. First, the background scene is modeled (2.a). For each frame in the video sequence (2.b), the foreground region is segmented (2.c) by the new method of pixel classification. Based on the extracted silhouette and the original image, the cardboard model is analyzed (2.d) and salient body part templates are created (2.e). Finally, these parts are located by a combined method of shape analysis and color template matching (2.f).

4. 3D Reconstruction and Dynamic Model of Human Motion

By integrating the location data from each image, the 3D body posture can be estimated. First, the cameras are calibrated to obtain their internal and external parameters. For each frame in the sequence, each instance of W^4 sends to a central controller not only the body part location data but also a corresponding confidence value that indicates the level of confidence of its 2D localization for each particular part. The confidence value is obtained from the similarity score of the template-matching step. The

To constrain the body motion and smooth the motion trajectory, a model of human body dynamics [2] developed by MIT's Media Lab was first employed. The knowledge that the system will be tracking a human body provides many powerful constraints. Humans only move in certain ways: the human body is made up of physical objects that must obey the laws of Newtonian mechanics; the parts of the body are joined together in specific, well-understood ways. The knowledge that the system will be tracking a human body provides many powerful constraints on the possible observations. A general tracking mechanism, when incorporated into a recursive framework with explicit models of these types of constraints, becomes a much more robust system.

The general form of such a system is the Kalman filter. Each generalized tracker reports its current result to the estimator, along with a measure of its confidence in that result. These reports are treated as noisy observations by the estimator, which then refines its current estimate of the true state of the tracked body. As with Kalman filters, this estimate, along with the motion model, allows the estimator to predict future observations with bounded confidence. These predictions represent prior knowledge for the trackers. They not only allow the tracker to take

advantage of the motion models contained in the estimator, but also indirectly, to take advantage of the observations made by other sensors in the system. In the system described here, all the other sensors are cameras with varying viewpoints, but this need not be the case.

Explicit models of the non-linear dynamics of the human body are computationally expensive and difficult to build. This implementation of the framework represents an experiment with a computationally light-weight version that utilizes several linear Kalman filters tracking individual body parts. Given the rich literature on Kalman filter design, this system required much less development time than the full dynamic model. These individual filters are then linked together by a global kinematic constraint mechanism. The linear Kalman filters approximate the low-level, dynamic constraints while the global constraint system maintains the kinematic constraints. We found that this optimization provides sufficient predictive performance while making the system computationally more accessible and easier to construct.

5. Conclusion

A real-time 3D motion capture system has been implemented in C++ and runs under the Windows NT operating system. Currently, for 320x240 resolution color images, it runs at 20-30 Hz on a PC that has dual 400 MHz PentiumII processors. Figure 3 illustrates examples of our system in three different posture; in each frame, the two images shows two views of the body parts located by W^4 in 2D, and the middle image shows the reproduced cartoon character. Figure 4 shows the demonstration area at SIGGRAPH. The cameras were placed in a semi-circle arrangement pointing toward the dancing area. A projector was placed next to the dancing area and displayed the animated graphical character.

Acknowledgement:

We would like to thank Ross Cutler for his hardware advice.

References

[1] W^4 : Who? When? Where? What? A Real Time System for Detecting and Tracking People. Ismail Haritaoglu, David Harwood, and Larry Davis. Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, April 1998.

[2] Dynamic Modeling of Human Motion. Christopher R. Wren and Alex P. Pentland. Proceedings of the Third

IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, April 14-16, 1998. MIT Media Laboratory Perceptual Computing Section TR #451.

http://vismod.www.media.mit.edu/cgi-in/tr_pagemaker#TR451

[3] Three-Dimensional Computer Vision, A Geometric Viewpoint. Olivier Faugeras. The MIT Press, Cambridge, Massachusetts, 1993.

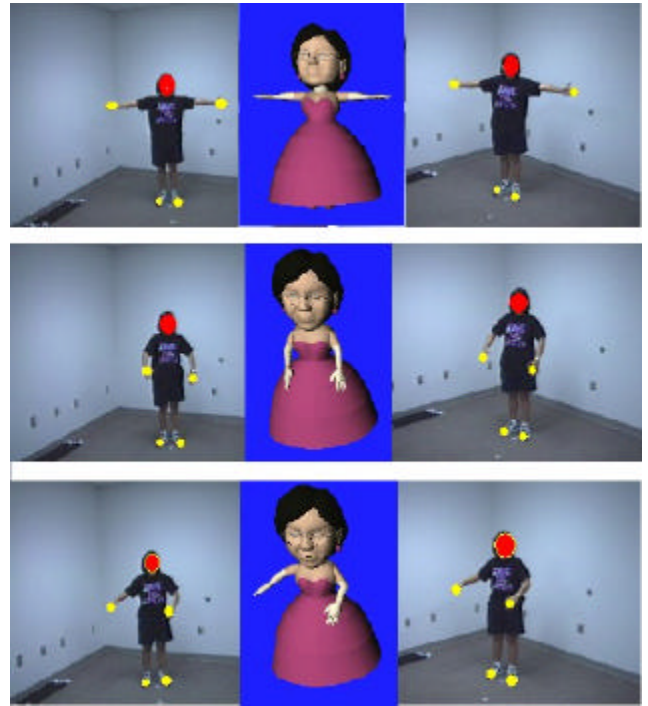


Figure 3. Our result on some key frames on a video sequence.

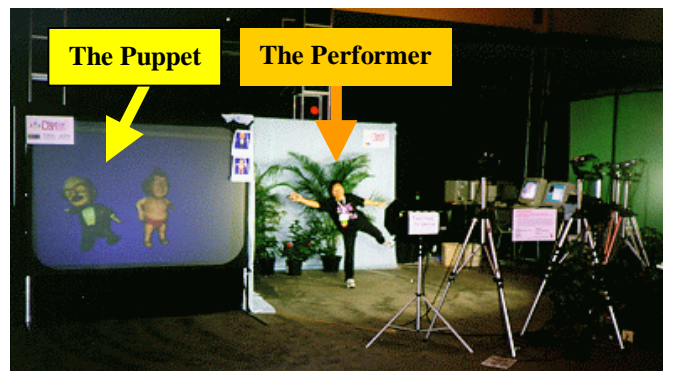


Figure 4. A snap-shot of the demonstration site at SIGGRAPH98