

# 3D Head Tracking Using Motion Adaptive Texture-Mapping

Lisa M Brown  
IBM T.J. Watson Research Center  
[lisab@us.ibm.com](mailto:lisab@us.ibm.com)

## Abstract

*We have developed a fast robust 3D head tracking system based on rendering a texture-mapped cylinder. In order to handle the variable frame-to-frame motion changes, the system uses motion templates, which adapt to the current size of the motion increment. The relationship between measurable pixel energy and tracking error is used to design the parameters of the adaptive algorithm. To speedup processing and decouple rotational and translational motion, 2D positional information of the neckline is utilized. The confidence at each point is computed based on the amount of information used in creating the texture map and re-rendering the face. The system can also handle large out-of-plane rotations via additional templates. If the tracker fails, it can recover using an independent face detection routine. We compare the results of our approach with the extensive results of a closely related technique.*

**Keywords:** 3D head tracking, real-time motion estimation, robust face modeling.

## 1. Introduction

Several applications in computer vision will benefit from an accurate and fast 3D head tracking system. These include pose-invariant face recognition[1-8], facial expression analysis[9-10], input routines for head and facial animation[1-13], facial compression systems for teleconferencing[14], and human computer interfaces[15-16]. Although recently, several investigators have developed methods to perform 3D head tracking[17-19], there is still a need to improve the accuracy and reliability of such systems. This is the task we have undertaken.

In particular, we are interested in 3D head tracking as part of a larger, multi-scale human tracking system being developed at IBM, called PeopleVision[20]. This multi-camera system is being designed to determine a range of

human activities, from how many people are in a particular space, to who is there and what are they doing. From the viewpoint of this project, head tracking needs to be designed so that, on the one hand, it is useful for pose-invariant (or dynamic) face recognition, and on the other hand, can exploit larger scale information such as the absolute position of the head or the relative position of the head to the shoulders.

From the point of view of face recognition, commercial systems are, by and large, still pose dependent. Visionics can recognize people for pose less than 20 degrees. Recent research has made significant strides towards improved pose invariance. Wen Yi Zhao and Chellapa [1], Chung et. al. [2], and Senior[3] have improved the robustness to face recognition to small changes in pose or lighting. Demir et al.[4] achieve pose invariance based on matching the pose of the input with a similar pose recorded in the database. Feng et.al.[5] and Kousani[6] achieve pose invariance by determining the pose of the input image, normalizing this image to a frontal pose and testing against the original database of frontal images. All of these systems are limited, either by a small range of pose, an extensive database or in the latter case, in the accuracy of 'normalized' pose.

Two very recent works have pushed the state of the art forward. Georghiades et. al.[7] generate synthetic poses and lighting conditions with large variations from a small number of images of each subject. These are used to build a representation of the statistical space spanned by each subject under all lighting conditions and poses. Okada et. al. [8] create a PCMAP of each face in the database from a large number of images of each subject. The pose of the input test image is determined by finding the optimal PCMAP using a classification method. The work of both Georghiades and Okada are appearance or view-based and extend the current statistical foundation of face recognition. Their emphasis is on stand-alone face recognition systems. These approaches rely on statistical information and ignore the geometric reality. They are effective for the current state-of-the art in face recognition.

Our approach to head tracking is an attempt to bridge the gap between the view-based statistical techniques and the geometric model and feature-based methods. We would like to ultimately integrate our system into multi-scale human tracking and at the same time achieve dynamic face recognition.

## 2. Background

An excellent survey of face tracking systems can be found in [21]. Systems are categorized by their algorithm characteristics, ranging from: color blob, motion blob, depth blob, edge, feature, template and optic flow. The first three types refer to systems which track color, motion or depth clusters, respectively. Systems are also differentiated according to their recovery algorithms, which are similarly classified. Toyama makes two interesting conclusions. First, certain cues, like color and motion are useful for positional tracking and recovery since they can be computed quickly, but lack the precision and robustness necessary for full 3D tracking. Second, 3D tracking is predominantly based on either feature or template algorithms. Furthermore, template techniques tend to be more robust while feature-based methods can be more efficiently implemented.

The method we have developed primarily uses templates. As a template approach, the dense comparison or corresponding pixels between image and template makes the method robust against individual variation and small non-rigid motion. It can also be made to be robust against illumination variations by the use of subject-independent illumination templates. On the other hand, we are able to track efficiently by exploiting widely available texture mapping hardware and a straightforward least squares computation. In addition, we use an independent estimate of 2D position to speedup the processing.

Our three-dimensional head tracker is an extension of the method designed and thoroughly tested by La Cascia et. al.[22]. This technique is based on mapping the face onto a cylindrical model and estimating the change in pose with respect to the incremental difference in the re-rendered texture maps. We have studied the cases in which the accuracy of the system degenerates and have designed our system to handle these problems. The primary sources of error in the original system are due to (1) the inability to distinguish rotations around the horizontal axis and vertical translations or similarly rotations around the vertical axis and horizontal translations, (2) very large rotations around the vertical axis, and (3) large frame to frame pose changes. Error sources of the first and third kind are reduced via adaptive templates. Errors of the first kind are also reduced by using an independent measure of two-dimensional pose based on feature tracking along the neckline. The

information to distinguish small rotations from their counterpart translations is not sufficient using frame-to-frame differences but can be roughly estimated using neckline features and then refined using the full 3D tracker. When the source of error is due to large rotations around the vertical axis the original system fails because the face is not truly cylindrical. This is corrected by updating the motion templates when rotation around the vertical axis is sufficiently large. We show several examples where the original method fails but is robust using adaptive/additional templates and 2D positional information. The data used was created by La Cascia et. al. and is publicly available at [23].

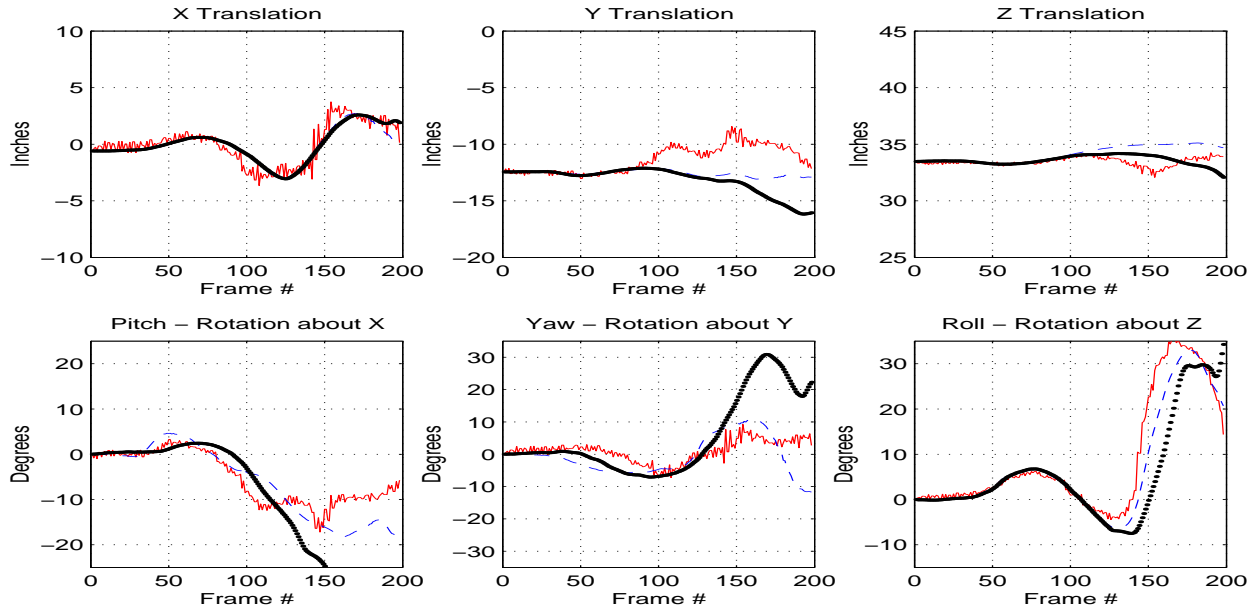
## 3. Motion Adaptive Templates

Our system is based on the creation of a set of motion templates. Initially, a frontal image of the person to be tracked is acquired using a face detection algorithm. This image is used to create a texture map for a cylindrical model of the head. This texture map is then used to render the face at small perturbations in pose, in particular along each of the six degrees of freedom of the motion: three translations and three rotations. Each motion template is the difference image between the original frontal image and the re-rendered texture map of the head at a slightly different pose, based on a small change in one of the motion parameters.

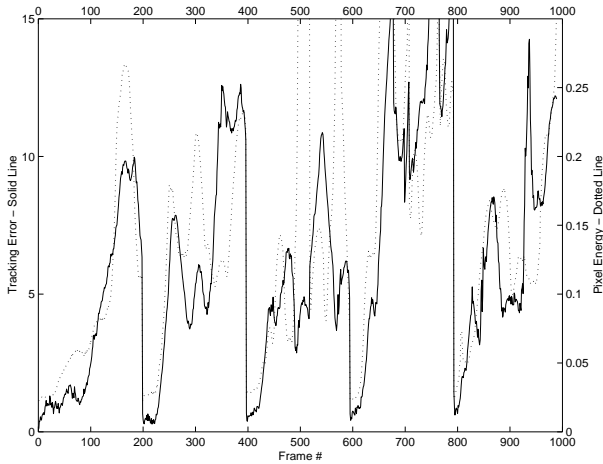
Each subsequent frame of the video sequence or camera input is used to create a texture map for a cylindrical model of the head, based on the previous estimate of the pose. The difference between the frontal view of this texture map and the frontal view of the original (frontal) texture map is used as a measure of the motion that occurred between the previous frame and current one. The system determines the best linear combination of templates that accounts for this motion. This linear combination is an accurate estimate of the motion as long as the motion is small, the motions are not too tightly coupled, the head moves rigidly and the head and face behave approximately like a cylinder

We compute the best linear combination using least squares based on the normal equations computed by a straightforward Gauss-Jordan elimination. Originally, we performed the fit using SVD which although, in general, is more robust, was in this case unnecessary and significantly slower. The majority of the time required to compute the motion parameters was the time required by the least squares fit calculation.

We found our system to be highly sensitive to the size of the perturbations in the motion templates. Following the notation used by La Cascia et.al., we create 4 motion templates for each of the six motion parameters, changing



**Figure 1. Results of 3D head tracker using two sets of motion template perturbation deltas. Light solid line is the ground truth. Dark solid line represents the results with the smaller deltas which lose the track by frame 150. Dashed line shows the results with larger set which successfully tracks the head.**



**Figure 2. Tracking Error is compared with Pixel Difference Energy over 1000 frames of head tracking.**

the  $k^{th}$  parameter by  $\pm\delta_k$  and  $\pm 2\delta_k$ . In La Cascia et.al., the  $\delta$ 's were set so that corresponding difference images for different motion parameters would have the same energy. This was based on the work on view-based active appearance models performed by Cootes et.al.[]. It was also verified in our own analysis. However, even if this relative perturbation size is maintained, the absolute value of these  $\delta$ 's can alter which video sequences can be tracked. In Figure 1, the video sequence “vam2.avi” is

tracked using two sets of deltas. For one set (the larger) the head is tracked successfully. For the smaller set, the track is lost by frame 150. For a different sequence, the smaller set of deltas is required to successfully track the head while the larger set causes it to fail.

Based on the relationship between template energy and the relative perturbation size, we decided to investigate the relationship between tracking error and pixel difference energy (between the current frame mapped frontally and the original frontal frame.) We define the tracking error as the sum of the rotational and translation errors. These are each defined as a function of the Mahalanobis distance between the estimated and measured position and orientation respectively[22]. For example, the error in translation is given by:

$$e_{t,i}^2 = [x_{t,i} - \tilde{x}_{t,i}]^T \Sigma^{-1} [x_{t,i} - \tilde{x}_{t,i}]$$

where  $x_{t,i}$  and  $\tilde{x}_{t,i}$  are the estimate of the 3D position and the ground truth, and  $\Sigma$  is the covariance computed over the entire set of ground truth data.

In Figure 2, this relationship is plotted over 1000 frames for the original head tracking system. The solid line represents the tracking error whose units are shown on the left hand side. The dotted line represents the pixel energy; units are on the right. The two values appear linearly related, in particular for the smaller tracking error where the system is successful. We conclude from this

relationship, that the perturbation size needs to adapt to the tracking error so that the system can correct for motion changes of difference sizes. If the motion is too large, or the system has failed to track correctly, templates created with larger perturbations are necessary to bring the system back to the smaller tracking error size.

#### 4. Confidence Maps

Each frame is used to create a texture map and then projected frontally. The frame is inversely projected onto the texture map based on the current estimate of the pose, given by the 6 motion parameters. If the pose is frontal, then the creation of the texture map is simply the reverse of the projection and the information given in the image has uniform confidence. However, as the pose in the current frame varies, the confidence map is defined as the product of the two processes, the inverse projection onto the texture map, and the forward projection onto the image.

In both processes, we define the confidence to be the ratio of the area of an element in the input over the size of the area of an element of the output. In other words, our confidence is greatest if the input used to compute the output has the greatest relative area. For forward projection, image points derived from texture map points, which are viewed obliquely, have the highest confidence because a large area of the texture map is evaluated. For inverse projection, texture map points derived from image points, which represent points viewed frontally, have the highest confidence.

For the moment, let us consider this ratio for the frontal (forward) perspective projection. As seen in Figure 3, we define the image plane by,

$$\vec{\mathbf{P}} \cdot \vec{i} = d$$

where  $\vec{\mathbf{P}}$  is the unit normal to the plane,  $\vec{i}$  is an arbitrary point on the plane,  $d$  is a constant. Let  $\vec{t}$  be a point on the texture-mapped cylinder. Let  $\vec{\mathbf{R}}$  be the unit radius vector from the axis of the cylinder to  $\vec{t}$ .  $\vec{\mathbf{R}}$  is therefore normal to the surface of the cylinder. Construct the unit vector,

$$\vec{\mathbf{U}} = \frac{(\vec{i} - \vec{v})}{|\vec{i} - \vec{v}|}$$

that points from the viewpoint,  $\vec{v}$ , to the image point,  $\vec{i}$ , and consequently to a texture-mapped cylinder point,  $\vec{t}$ . The line,

$$L(k) = \vec{v} + \vec{\mathbf{U}}k$$

is the line that runs from the view point to the image point, intersecting the cylinder. We find the point of intersection,

$$\vec{t} = \vec{v} + \mathbf{U} t_0.$$

by solving for  $t_0$  such that  $\vec{t}$  lies on the cylinder,

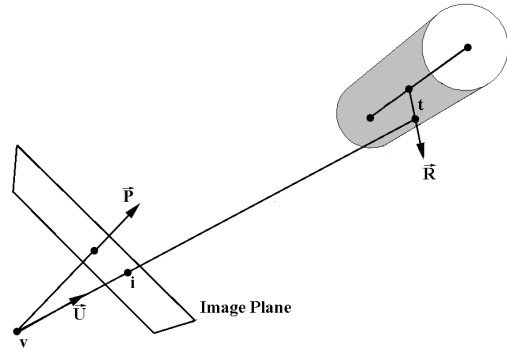
$$\vec{t} = [r \cos \theta, h_i, r \sin \theta]$$

where  $r$  is the radius of the cylinder,  $h_i$  is in the range of  $(0, h)$  where  $h$  is the height of the cylinder.

The cylinder point,  $\vec{t}$ , is at a distance,  $t_0 = |\vec{t} - \vec{v}|$  from the viewpoint. Similarly, the image point is located at a distance  $i_0 = |\vec{i} - \vec{v}|$ . The ratio of the area-element on the texture-mapped cylinder to the projection of that element on the image plane is then given by the following formula:

$$\frac{\partial A_t}{\partial A_i} = \left( \frac{t_0}{i_0} \right)^2 \left( \frac{|\vec{\mathbf{U}} \cdot \vec{\mathbf{P}}|}{|\vec{\mathbf{U}} \cdot \vec{\mathbf{R}}|} \right).$$

The area grows with the square of the distance and inversely with the projection of the surface normal to the view direction.



**Figure 3.**

The result of this computation is shown in Figure 4b. The intensity is scaled to improve the visualization. The largest confidence values occur where the face is viewed most obliquely. We also mask this image with an ellipse with major axis equal to half the height of the cylinder and minor axis equal to the radius. In this way, we limit our measurements to facial pixels.

In addition to the confidence values due to the frontal projection, we need to compute the confidence due to the inverse projection. This is simply the inverse of Equation 1, with two additional details. First, the cylinder is rigidly transformed since the inverse projection depends on the current estimate of the pose. This can be implemented by performing the transformation with respect to the viewpoint and the image plane, thereby allowing us to use the same equations as before.

The second detail is that the confidence measures need to be computed with respect to each point in the final frontal image so the cumulative confidence can be computed as the product. In particular, for each point in the frontal image, we determine the point on the cylinder to which it corresponds (when viewed frontally.) We then compute the distances between the viewpoint and the image point, the viewpoint and the cylinder, and the viewing directions with respect to the current pose. An example of this computation is given in Figure 4a. In this case, the current pose has rotated  $30^\circ$  around the horizontal axis. Notice how the greatest confidence values occur near the top where the face is viewed most perpendicularly to the camera.

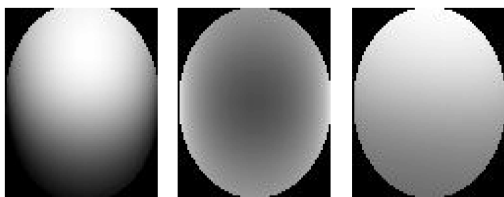


Figure 4a-c.

The final confidence map is shown in Figure 4c. This map is used as the weighting system in the least squares computation.

## 5. Additional Templates

One of the primary sources of error in the original system were due to very large rotations around the vertical axis. These errors cause a problem for the system because the face is not truly cylindrical. In particular, when a person turns their head sharply left or right, their nose and its three dimensional shape can be seen more clearly and at the same time it obscures other parts of their face. The motion templates created based on the original frontal image do not provide this information. The difference image cannot be accounted for using the original templates. In addition, the amount of useful information with high confidence from both images is low since what is frontal in one is along the side of the other.

We have found that we can reduce the error in tracking due to large rotation around the vertical axis by creating new motion templates, either on the fly when the rotation grows too large or beforehand using additional footage processed offline. For the sequence “jim1.avi”, the original tracker failed by frame 60 as shown in Figure 5 by the dark dotted line. The ground truth based on a magnetic tracker is shown as a solid noisy line. Using additional motion templates when the rotation exceeded 10 degrees allowed the system, depicted as a dashed line, to continue to track. We also tested the utility of adding additional templates when the rotation exceeded 5 degrees shown as the dotted line. As can be seen in the results, the difference is negligible.

## 6. 2D Positional Information

The remaining source of error is due to the inability of the original system to differentiate small rotations from their visually similar translations. However, since we envision our system as part of a larger human tracking project, we have available information regarding the 2D positional location of the neckline. This is computed using a standard Lucas-Kanade feature tracker.

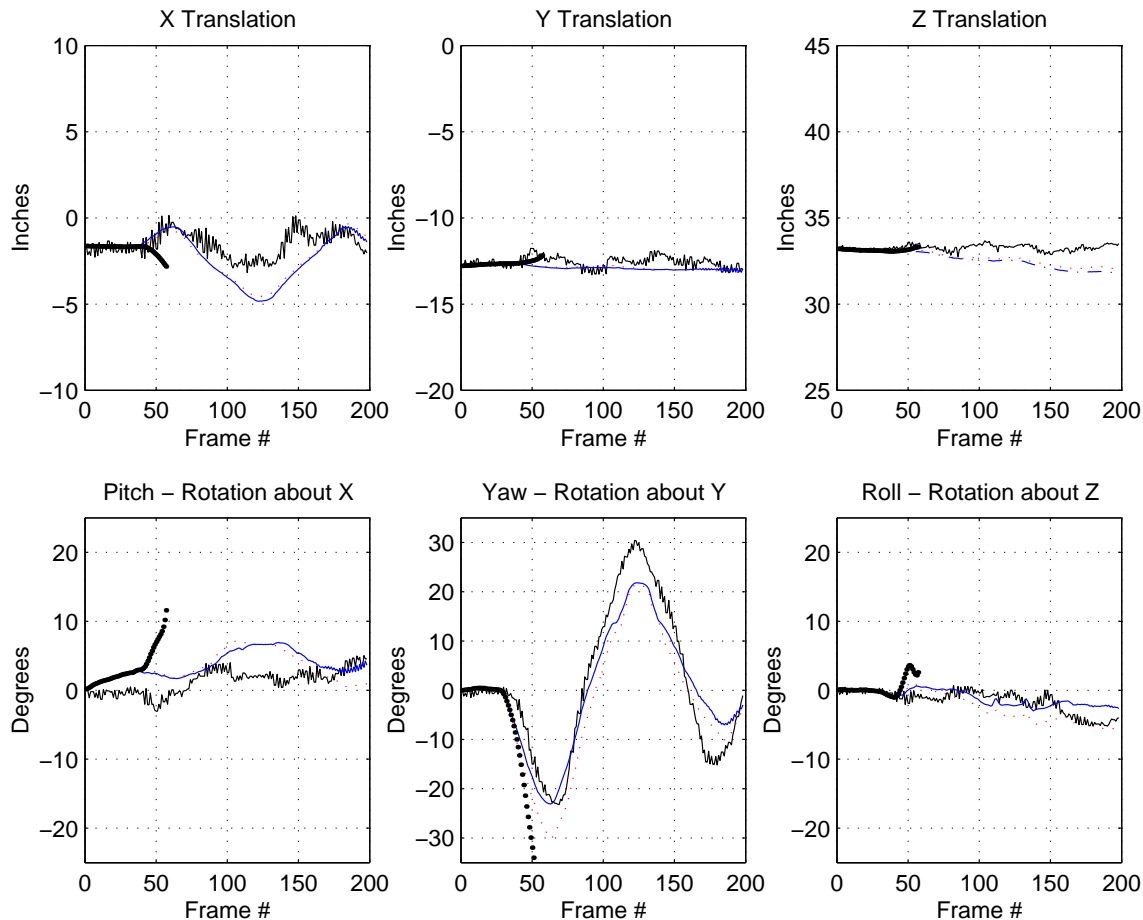
This information is used in conjunction with the 3D head tracker estimates to derive a more robust 3D positional estimate. The x and y estimates computed by the neckline feature detector are averaged with the current estimates computed by the tracker prior to the least squares fit.

In Figure 6, we show the results of the head tracker algorithm on the challenging “vam4.avi” sequence, a sequence which includes rotation along two axes, coupled rotation about the vertical and translation along x, and large fast rotational motion. Our system is able to cope with coupled rotation/translation seen in frames 30-100. It also handles simultaneous rotations around y and z in frames 50-100. But it fails to track when the rotational motion around the y-axis is too fast and is coupled with horizontal translation and rotation around the z-axis.

## 7. Recovery

A face detection scheme based on a template search across an image pyramid [24] was integrated into the system in two ways. It is initially used to detect the face in the sequence, in order to initialize the size and location of the cylinder. It is also used to recover the pose when the tracker fails.

An image pyramid is constructed representing the image at different resolutions, with neighboring scales differing



**Figure 5. Results of 3D head tracking system augmented with additional templates. Dark dots, represent original system which lost the track by frame 60. The solid noisy line is the ground truth. The dashed line represents the system with additional templates every 10 degrees of yaw. The nearly identical dotted line represents the system with additional templates every 5 degrees of yaw.**

by a factor of 1.2. The size of the pyramid can be limited by domain constraints on the sizes of faces to be expected. The face detector is applied at each location in the image pyramid to determine if the surrounding square represents a face. The face detector applied at each such candidate region consists of a hierarchy of binary classifiers, each seeking to filter out a proportion of the non-faces but retain the true faces.

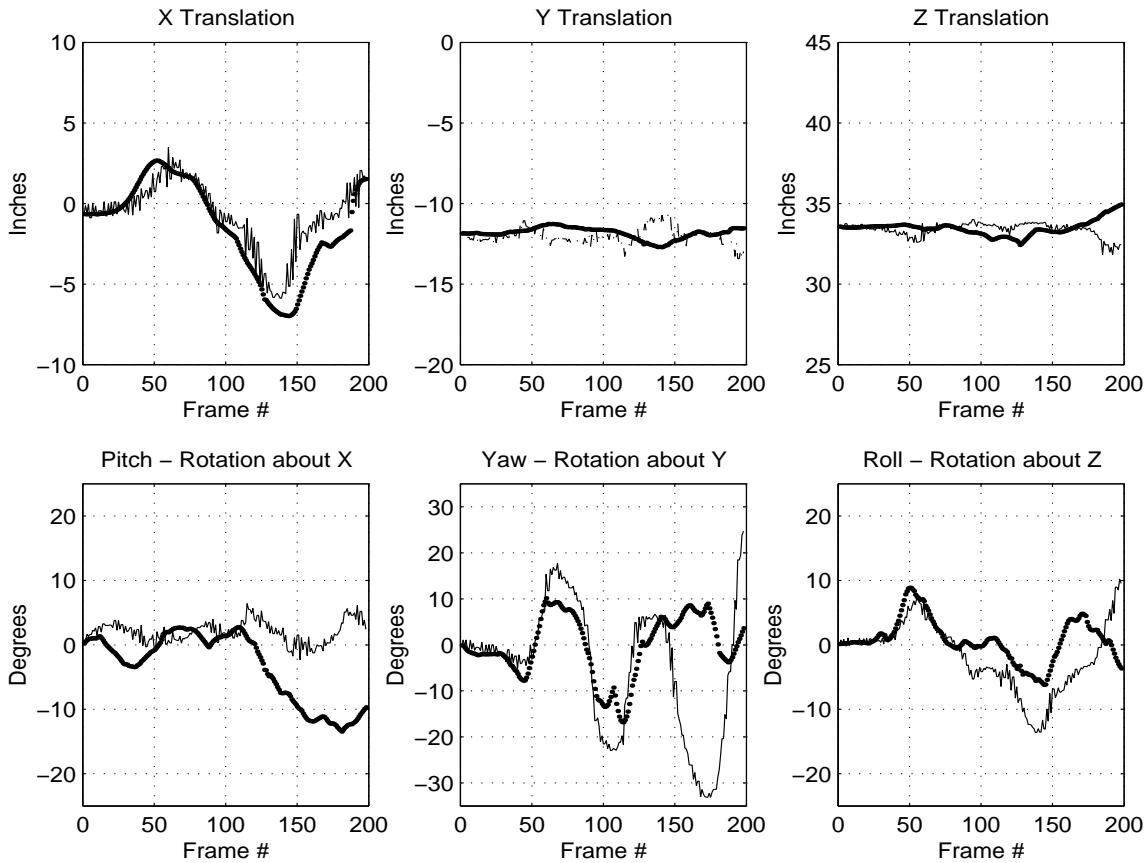
The first detector is a skin tone filter that determines if the pixels of the candidate region have coloring consistent with skin. Each pixel is independently classified as skin-tone or not, and the candidate region is rejected if it contains an insufficient proportion of skin-tone pixels (typically 60-80%). Subsequently a linear discriminant trained on face and non-face exemplars is applied to the gray-level pixels. The linear discriminant is fast and removes a significant proportion of the non-face images. Next a Distance From Face Space [25] measure is used to

further filter the exemplars and finally a combination of the two scores is used to rank overlapping candidates that have been retained, with only the highest scoring candidate being retained.

The pose parameters of the face detections are refined by a local search to maximize the combined score over the space of small perturbations in scale, location and rotation about the z-axis. Some rotation about the x and y-axes is accommodated but not estimated by extending the search to small perturbations in aspect ratio. Tracking is carried out by a similar local search, after predicting the face's new location with a constant-velocity model and the current frame rate.

## 8. Conclusions

We have demonstrated the efficacy of improving 3D head tracking based on texture mapping onto a cylinder



**Figure 6. Results of 3D Head Tracker integrated with 2D pose information from a face detection algorithm. Light solid line represents ground truth; dark dotted line represents the results of the tracker.**

using confidence maps derived from the geometry, adaptive motion templates based on the pixel energy, additional templates for large rotations and external 2D positional information Confidence maps derived from the geometry of the mapping enable the system to measure only the overlapping and related portions of two frames which may have a very large pose difference. Adaptive motion templates based on the pixel difference energy improve the system's ability to tolerate large motion and small tracking errors. Additional templates for large vertical rotations compensate for the 3D structural properties of the head. Two dimensional position information enable the system to de-couple small rotations from their visually similar translations and simultaneously speed up the computation. We would like to more extensively evaluate the benefits of this methodology through a more thorough analysis of a large data set. Ultimately we would like to integrate this system into our multi-scale people tracking project using information about a person as they approach the camera and determining the identity of the person as they travel through the perceptual environment.

## 9. Bibliography

- [1] Wen Yi Zhao, Chellapa, R. "SFS Based View Synthesis for Robust Face Recognition," *Proc. 4<sup>th</sup> IEEE Conf. On Automatic Face and Gesture Recognition*, Grenoble, France, March 2000, p277-284.
- [2] Chung, K-C, Kee, S.C., Kim, S.R., "Face Recognition Using Principal Component Analysis of Gabor Filter Responses," *Proc. of the Int'l Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems*, Corfu, Greece, Sept. 1999, p53-57.
- [3] Senior, A.W., "Recognizing Faces in Broadcast Video," *Proc. of the Int'l Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems*, Corfu, Greece, Sept. 1999, p105-110.
- [4] Demir, E., Akarun, L., Alpaydin, E., "Two Stage Approach for Pose Invariant Face Recognition," *2000 IEEE Int'l Conf. On Acoustics, Speech and Signal Processing*, Vol 4, Istanbul, Turkey 5-9, June 2000, p2343-4.
- [5] Feng, G.C., Yuen, P.C., "Recognition of Head and Shoulder Face Image Using Virtual Frontal View Image," *IEEE Trans. Syst. Man Cybern. & Syst. Humans*, Vol 30, No. 6, Nov 2000, p871-82.

- [6] Kouzani, A.Z., He, F., Sammut, K., "Towards Invariant Face Recognition," *Information Sciences*, Vol. 123, (2000), p75-101.
- [7] Georgiades, A.S., Behumeur, P.N., and Kriegman, D.J. "From Few to Many: Generative Models for Recognition Under Variable Pose and Illumination," *Proc. 4<sup>th</sup> IEEE Conf. on Automatic Face and Gesture Recognition*, Grenoble, France, March 2000, p277-284.
- [8] Okada, K., Akamatsu, S., von der Malsburg, C., "Analysis and Synthesis of Pose Variations of Human Faces by a Linear PCMAP Model and its Application for Pose Invariant Face Recognition System," *Proc. 4<sup>th</sup> IEEE Conf. on Automatic Face and Gesture Recognition*, Grenoble, France, March 2000, p142-149.
- [9] Tian, Y-L, "Facial Expression Analysis", PAMI
- [10] Chen, T., "Audiovisual Speech Processing," *IEEE Signal Processing*, Vol. 18, No. 3, May 2001, p9-21.
- [11] Vetter, T., "Synthesis of Novel Views from a Single Face Image," *Int'l J. Computer Vision*, Vol. 28, No.2, 1998, p103-116.
- [12] Takacs, B. Fromherz, T., Tice, S., Metaxas, D., "Digital Clones and Virtual Celebrities: Facial Tracking, Gesture Recognition and Animation for the Movie Industry," *Proc. of the Int'l Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems*, Corfu, Greece, Sept. 1999, p169-176.
- [13] Goto, T., Kshirsagar, Magnenat-Thalmann, N., "Automatic Face Cloning and Animation," *IEEE Signal Processing*, Vol. 18, No. 3, May 2001, p17-25.
- [14] Shin, M.C., Dmitry G. Kim, Carlos, "Estimation of the MPEG-4 FAPs Using Point and Curve Features," *IEEE Workshop on Human Modeling Analysis and Synthesis*, Hilton Head Island, SC, June 2000, p59-64.
- [15] Darrell, T., Gordon, G., Woodfill, J., Harville, M., "A Virtual Mirror Interface using Real-time Robust Face Tracking,"
- [16] Sherrah, J., and Gong, S., "Fusion of 2D Face Alignment and 3D Head Pose Estimation for Robust and Real-time Performance," *Proc. of the Int'l Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems*, Corfu, Greece, Sept. 1999, p24-30.
- [17] Jebara, T.S., and Pentland, A., "Parameterized Structure from Motion for 3D Adaptive Feedback Tracking of Faces," *Proc. Conf. Computer Vision and Pattern Recognition*, 1997.
- [18] Schodel, A., Haro, A. and Essa, I., "Head Tracking Using a Textured Polygonal Model," *Proc. 1998 Workshop Perceptual User Interfaces*, 1998.
- [19] Dellaert, F., Thorpe, C., and Thrun, S., "Jacobian Images of Super-Resolved Texture Maps for Model-Based Motion Estimation and Tracking," *Proc. IEEE Workshop Applications of Computer Vision*, 1998.
- [20] Hampapur, A., Senior, A., Brown, L., Tian, Y-L, Pankanti, S. "People Vision: A Multiscale Human Perception System," <http://arunh.userv.ibm.com/ARPresentation.doc>, 2001.
- [21] Toyama, K., "Prolegomena for Robust Face Tracking," Microsoft Research Technical Report MSR-TR-98-65, Nov. 13, 1998.
- [22] La Cascia, M., Sclaroff, S., and Athitsos, V., "Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models," *IEEE PAMI*, Vol 22, No. 4., April 2000.
- [23] <http://www.cs.bu.edu/groups/ivc/HeadTracking>
- [24] Cootes, T.F., Walker, K., Taylor, C.J., "View-Based Active Appearance Models," *Proc. 5<sup>th</sup> European Conf. On Computer Vision*, 1998.
- [25] Senior, A W., "Recognizing Faces in Broadcast Video," *Proc. Int'l Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems 1999*.
- [26] Turk, M. A. and Pentland, A. P., "Face Recognition Using Eigenfaces," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Hawaii, June, 1992.