Robust Full-Motion Recovery of Head

by Dynamic Templates and Re-registration Techniques

Jing Xiao, Tsuyoshi Moriyama, Takeo Kanade
Robotics Institute, Carnegie Mellon University
{jxiao, tmoriyam, tk}@cs.cmu.edu

Jeffrey F. Cohn
University of Pittsburgh
jeffcohn@pitt.edu

Abstract

This paper presents a method to recover the full-motion (3 rotations and 3 translations) of the head from an input video using a cylindrical head model. Given an initial reference template of the head image and the corresponding head pose, the head model is created and full head motion is recovered automatically. The robustness of the approach is achieved by a combination of three techniques. First, we use the iteratively re-weighted least squares (IRLS) technique in conjunction with the image gradient to accommodate non-rigid motion and occlusion. Second, while tracking, the templates are dynamically updated to diminish the effects of self-occlusion and gradual lighting changes and to maintain accurate tracking even when the face moves out of view of the camera. Third, to minimize error accumulation inherent in the use of dynamic templates, we re-register images to a reference template whenever head pose is close to that in the template. The performance of the method, which runs in real time, was evaluated in three separate experiments using image sequences (both synthetic and real) for which ground truth head motion was known. The real sequences included pitch and yaw as large as 40° and 75°, respectively. The average recovery accuracy of the 3D rotations was about 3°. In a further test, the method was used as part of a facial expression analysis system intended for use with spontaneous facial behavior in which moderate head motion is common. Image data consisted of 1-minute of video from each of 10 subjects while engaged in a 2-person interview. The method successfully stabilized face and eye images allowing for 98% accuracy in automatic blink recognition.

1. Introduction

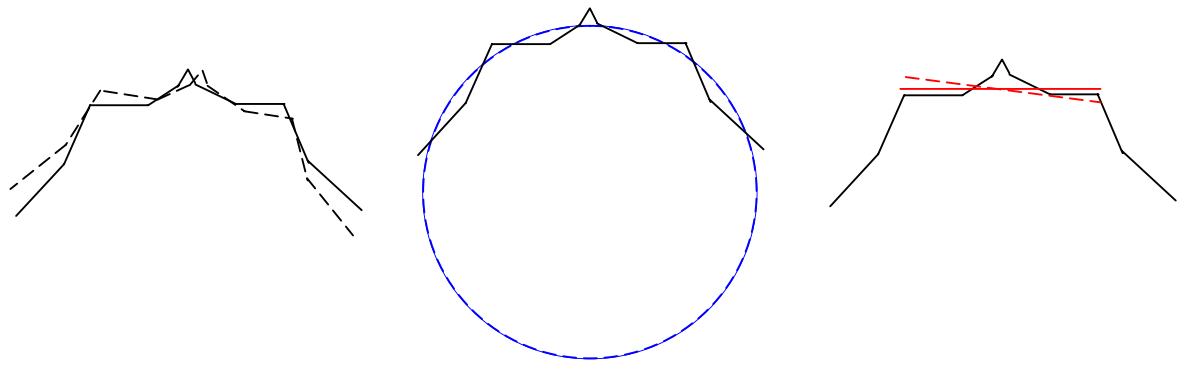
Three-dimensional head motion recovery is an important task for many applications, such as human-computer interaction and visual surveillance. An aligned image according to the recovered head motion would facilitate facial expression analysis and face recognition.

Many approaches have been proposed to recover 3D head motion. One is to use distinct image features [Liu, 2000; Lowe, 1992; Gennery, 1992; Jebara, 1997], which work well when the features may be reliably tracked over the image sequence. When good feature correspondences are not

possible, tracking the entire head region using a 3D head model is more reliable. Both generic and user-specific 3D geometric models have been used for head motion recovery [DeCarlo, 1996; Essa, 1997]. With precise initialization, such models introduce minimal error and perform well. When initialization is not perfect (i.e., the initial estimate of head orientation deviates from ground truth), model error will increase substantially and degrade motion recovery (Figure 1a).

Use of a much simpler geometric head model often is effective and robust against initialization errors. Various planar model-based methods have been presented [Black, 1992; Hager, 1998]. They model the face (not the entire head) as a plane and use a single face texture (static template) to recover head motion. The approximation of a planar face model introduces small model error, which is insensitive to small initialization errors (Figure 1c). When the head orientation is not far from the frontal view, (i.e., the static face template can be fit to the visible head image), planar models work well. To represent the geometry of the entire head, a more complete 3D model is necessary. In [Bregler, 1998; Basu, 1996], an ellipsoidal model was used with good results on 3D head tracking. Cascia et al. [Cascia, 1999] developed a fast 3D head tracker that models a head as a texture-mapped cylinder. The head image is treated as a linear combination of a set of bases that is generated by changing the pose of a single head image (template). The head pose of the input image then is estimated by computing coefficients of the linear combination. While simple and effective, use of a single, static template appears unable to accommodate cases in which large out-of-plane rotation turns the face away from the camera.

The relative error between the cylindrical model and the real geometry of a head is small and is invariant to the initialization error on head orientation (Figure 1b). In practice, precise initialization is usually not available. Therefore, in this paper, we utilize the cylindrical head model and present a robust method to recover full motion of the head under perspective projection. Given an initial reference template of the head image and the corresponding head pose, the cylindrical head model is created and the full head motion is recovered from the input video automatically. Three main techniques contribute to the robustness of the approach. First, to accommodate non-rigid motion and occlusion, we use the iteratively re-weighted least squares (IRLS) technique [Black, 1992]. The unintended side effect of IRLS, however, is to discount some useful information, such as edges. We compensate for this effect with use of image gradients. Second, we update the templates dynamically in order to accommodate gradual changes in lighting and self-occlusion. This enables recovery of head motion even when most of the face is invisible. As the templates are updated as the motions are recovered, the errors of motion recovery accumulate over time. The third technique, re-registration, is used to rectify the accumulated errors. We prepare images of certain reference poses, and re-register the head image with a reference image when the estimated head pose is close to that in the reference. Based on this approach, we built a real-time 3D head tracking system. As part of a facial expression analysis system intended for use with spontaneous facial behavior in which moderate head motion is common, it successfully stabilized face and eye images allowing for 98% accuracy in automatic blink recognition.



(a) Precise head model (b) Cylindrical head model (c) Planar head model

Figure 1: The cross section of a head (simplified for clarity) and the corresponding head models: precise model (black), cylindrical model (blue), and planar model (red). The solid curves show the correct initial fit of the models to the head and the dashed curves show the cases when initialization is not prefect.

2. Motion Recovery Using a Template

Suppose we observe an image $I(\boldsymbol{u}, t)$ at time t, where $\boldsymbol{u}=(u, v)$ is a pixel in the image. At $t+1$, $\boldsymbol{u}$ moves to $\boldsymbol{u}'=\boldsymbol{F}(\boldsymbol{u},\boldsymbol{\mu})$, where $\boldsymbol{\mu}$ is the motion parameter vector and $\boldsymbol{F}(\boldsymbol{u},\boldsymbol{\mu})$ is the parametric motion model (such as the affine motion), which maps $\boldsymbol{u}$ to the new location $\boldsymbol{u}'$. If we assume that the illumination condition does not change, then,

$$I\left(\boldsymbol{F}\left(\boldsymbol{u}, \boldsymbol{\mu}\right), t+1\right) = I\left(\boldsymbol{u}, t\right) \qquad (1)$$

One of the standard ways to obtain the motion vector $\boldsymbol{\mu}$ is by minimization of the following objective function,

$$min \quad E(\boldsymbol{\mu}) = \sum_{\boldsymbol{u}\in\Omega} \left(I\left(\boldsymbol{F}(\boldsymbol{u},\boldsymbol{\mu}), t+1\right) - I(\boldsymbol{u}, t)\right)^2 \qquad (2)$$

where Ω is the region of the template at t, i.e., only the pixels within Ω are taken into account for motion recovery. For simplicity of notation, we omit $\boldsymbol{u}$ and t in some of the following equations.

In general, this class of problems can be solved by the Lucas-Kanade method [Lucas, 1981],

$$\boldsymbol{\mu} = -\left(\sum_{\Omega}\left(I_{\boldsymbol{u}}\,\boldsymbol{F}_{\boldsymbol{\mu}}\right)^{\mathrm{T}}\left(I_{\boldsymbol{u}}\,\boldsymbol{F}_{\boldsymbol{\mu}}\right)\right)^{-1}\sum_{\Omega}\left(I_t\left(I_{\boldsymbol{u}}\,\boldsymbol{F}_{\boldsymbol{\mu}}\right)^{\mathrm{T}}\right) \qquad (3)$$

where I_t and $I_{\boldsymbol{u}}$ respectively are the temporal and spatial image gradient. $\boldsymbol{F}_{\boldsymbol{\mu}}$ means the partial differential of $\boldsymbol{F}$ with respect to $\boldsymbol{\mu}$, which depends on the motion model and is computed at $\boldsymbol{\mu} = 0$.

Since (3) comes from the linear approximation of (2) by the first-order Taylor expansion, this process has to be iterated. At each iteration, the incremental motion parameters are computed. Then the template is warped using the incremental transformation and the warped template is used for the next iteration. When the process converges, the motion is recovered from the composition of the incremental transformations instead of adding up the incremental parameters directly.

If we want to assign different weights to pixels in the template due to outliers and non-uniform density, (3) can be modified as:

$$\boldsymbol{\mu} = -\left(\sum_{\Omega} \left(w \left(I_{\boldsymbol{u}} \boldsymbol{F}_{\boldsymbol{\mu}} \right)^{\mathrm{T}} \left(I_{\boldsymbol{u}} \boldsymbol{F}_{\boldsymbol{\mu}} \right) \right) \right)^{-1} \sum_{\Omega} \left(w \left(I_t \left(I_{\boldsymbol{u}} \boldsymbol{F}_{\boldsymbol{\mu}} \right)^{\mathrm{T}} \right) \right) \tag{4}$$

We describe how to determine the weights $w(\boldsymbol{u}) \in [0,1]$ in Section 4.

3. Full-Motion Recovery under Perspective Projection

The rigid motion of a head point $\boldsymbol{X} = [x,y,z,1]^{\mathrm{T}}$ between time t and $t+1$ is:

$$\boldsymbol{X}(t+1) = \mathrm{M} \cdot \boldsymbol{X}(t) = \begin{bmatrix} \mathrm{R} & \mathrm{T} \\ 0 & 1 \end{bmatrix} \cdot \boldsymbol{X}(t) \tag{5}$$

$\mathrm{R}_{3\times3}$ is the rotation matrix with 3 degrees of freedom and $\mathrm{T}_{3\times1}$ is the 3D translation vector. The full head motion has 6 degrees of freedom.

We follow [Bregler, 1998] and use the twist representation [Murray, 1994]. The transformation M can be represented as [Bregler, 1998; Murray, 1994]:

$$M = \begin{bmatrix} 1 & -\omega_z & \omega_y & t_x \\ \omega_z & 1 & -\omega_x & t_y \\ -\omega_y & \omega_x & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

where $[\omega_x, \omega_y, \omega_z]$ represents the rotations relative to the three axes, and $[t_x, t_y, t_z]$ the 3D translation T.

Under perspective projection (assuming the camera projection matrix depends only on the focal length.), the image projection $\boldsymbol{u}$ of $\boldsymbol{X}$ $(=[x,y,z,1]^{\mathrm{T}})$ at $t+1$ is:

$$\boldsymbol{u}(\mathrm{t}+1) = \begin{bmatrix} \mathrm{x} - \mathrm{y}\omega_{\mathrm{z}} + \mathrm{z}\omega_{\mathrm{y}} + \mathrm{t}_{\mathrm{x}} \\ \mathrm{x}\omega_{\mathrm{z}} + \mathrm{y} - \mathrm{z}\omega_{\mathrm{x}} + \mathrm{t}_{\mathrm{y}} \end{bmatrix} \cdot \frac{\mathrm{f}_{\mathrm{L}}}{-\mathrm{x}\omega_{\mathrm{y}} + \mathrm{y}\omega_{\mathrm{x}} + \mathrm{z} + \mathrm{t}_{\mathrm{z}}}(\mathrm{t}) \tag{7}$$

where f_L is the focal length. (7) is the parametric motion model $\boldsymbol{F}(\bullet)$ in (1) with the 6D full-motion parameter vector $\boldsymbol{\mu} = [\omega_x, \omega_y, \omega_z, t_x, t_y, t_z]$. Note that t_z is included in (7), so the translation in the depths can be recovered.

If we compute $\boldsymbol{F}_{\boldsymbol{\mu}}$ at $\boldsymbol{\mu} = 0$,

$$\boldsymbol{F}_{\boldsymbol{\mu}}\Big|_{\boldsymbol{\mu}=0} = \begin{bmatrix} -xy & x^2+z^2 & -yz & z & 0 & -x \\ -\left(y^2+z^2\right) & xy & xz & 0 & z & -y \end{bmatrix} . \frac{f_L}{z^2}(t) \tag{8}$$

After each iteration, we compute the incremental transformation using $\boldsymbol{\mu}$ and compose all the incremental transformations to get the final transformation matrix. The full head motion is recovered from this matrix [Meriam, 1987]. The new head pose is also computed from the composition of the previous pose and the current transformation.

4. Weights of Pixel Contribution

4.1 Compensated IRLS Technique

Because of the presence of noise, non-rigid motion, and occlusion, some pixels in the template may disappear or may have been changed in the processed image. Those pixels should contribute less to motion estimation than others.

To take this factor into account, we apply a robust technique, called iteratively re-weighted least squares (IRLS) [Black, 1992]. Recall at each iteration of using (4), we warp the template by the incremental transformation and use the warped template to compute the new incremental parameters. The warped template is also used for computing the weights. For a pixel $\boldsymbol{u}$ in the template, its IRLS weight w_I is :

$$w_I = c_I exp\left(-\left(I(\boldsymbol{u},t+1)-\hat{I}(\boldsymbol{u},t)\right)^2 / 2\sigma_I^2\right) \tag{9}$$

where $\hat{I}$ is the warped template and σ_I is:

$$\sigma_I = 1.4826 \cdot \underset{\boldsymbol{u}\in\Omega}{median} \left|I(\boldsymbol{u},t+1)-\hat{I}(\boldsymbol{u},t)\right| \tag{10}$$

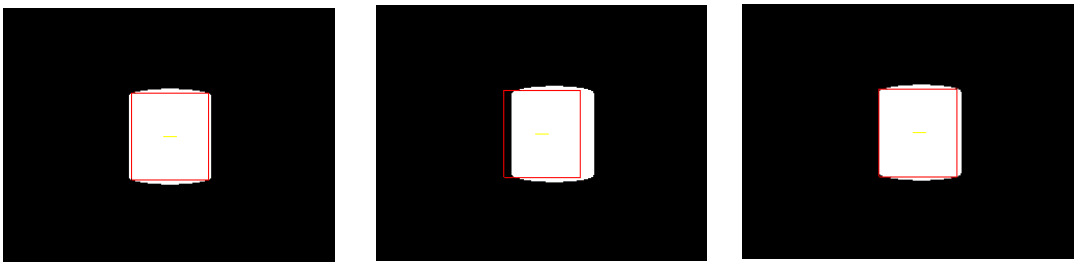
where the factor 1.4826 is a correction term that makes the median equal to the standard deviation of a normal distribution [Rousseeuw, 1987]. c_I is a scalar. Ω is the region of the warped template.

For the pixels with large residuals, w_I is small so that those pixels only have little contribution to motion recovery. However, large residuals don't necessarily mean outliers (with little contribution). Sometimes those pixels may give us useful information, such as the edges in Figure 2(a). To compensate for this side effect, we apply another weight w_G by using the gradient of the processed image:

$$w_G = c_G\left(1 - exp\left(-\left(I_{\boldsymbol{u}}(\boldsymbol{u},t+1)\right)^2 / 2\sigma_G^2\right)\right) \tag{11}$$

where σ_G is set as 128 and c_G is a scalar. c_G decreases at each iteration so that w_G has less influence while the recovered motion is getting more accurate. This weight prefers to large gradients and only affects the pixels on strong edges for several iterations. So the side effect of w_I will be reduced and its good effect is still preserved since the weights of most pixels within the outlier areas are very

small. Figure 2(b) and (c) respectively show the result of tracking a white cylinder with the IRLS and compensated IRLS. The cylinder translates horizontally in the black background. The compensated IRLS can recover the motion pretty well but the pure IRLS almost loses the object.



(a) (b) (c)

Figure 2: A white cylinder translates horizontally in the black background: (a) the template (within the red square); (b) the tracked region (red square) with IRLS; (c) the tracked region with compensated IRLS.

4.2 Non-Uniform Density of Template Pixels

The template pixels are projected from the 3D object. According to the surface geometry, they will not have a uniform density in the image. This will also affect their contribution and should be represented in the weights. A pixel with high density should have small weight, since it is projected from the side of the object surface.

Suppose $\boldsymbol{u}$ is the projection of a head point $\boldsymbol{X}$. θ is the angle between the surface normal at $\boldsymbol{X}$ and the direction from the head center to the camera center, as shown in Figure 3. We compute the pixel density weight by a quadratic function (because we use a quadratic surface (cylinder) as the model):

$$w_D = c_D \left(1 - min\left(\left|\theta\left(\boldsymbol{u}\right)\right|, \pi/2\right) \cdot 2/\pi\right)^2 \qquad (12)$$

where c_D is a scalar. When $\theta \geq \pi/2$, w_D is 0, which represents $\boldsymbol{u}$ is not visible. Smaller θ means $\boldsymbol{u}$ is closer to the template center and has lower density, so w_D is larger accordingly.

Finally we get the total weight w for each pixel as:

$$w = \left(w_I + w_G\right) \cdot w_D \qquad (13)$$

From (4), it is not necessary to normalize the weights since the normalization term will be same to each pixel.

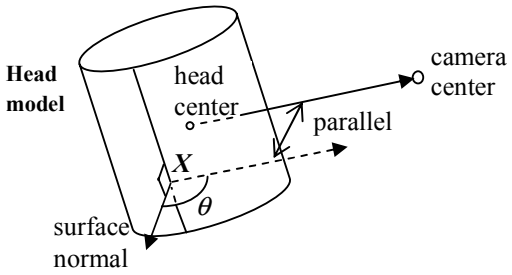


Figure 3: Angle θ between the surface normal and the direction from the head center to the camera center, which is used to determine the pixel density weight w_D.

5. Dynamic Templates & Re-registration

To achieve long-term robustness, it is not good to use a single template through the entire image sequence, because a template from one image cannot cover the entire head, but only part of the head. When most of that part is not visible, the approach using this template only may fail. In addition, it is difficult for a single template to deal with the problems like gradual lighting changes and self-occlusion. Therefore, we dynamically update the template while tracking.

At each frame (except the initial one), once the head pose is recovered, the head region facing the camera is extracted as the template for the following frame. When occlusion occurs, there might be some outliers in the template region. They should be removed from the template before the next tracking. Robust statistics are used again for this purpose. We detect the outliers by comparing the common region of the current template and the warped image of the last template, using the estimated motion between them. A pixel $\boldsymbol{u}$ in the common region will be removed from the new template as an outlier if,

$$\left| I(\boldsymbol{u},t) - \hat{I}(\boldsymbol{u},t-1) \right| > c\sigma_I \tag{14}$$

where $c \in [2.5,3.5]$ is a scalar that represents the strictness of judgment on outliers. σ_I is computed using (10) with Ω as the common region.

Because of the usage of dynamic templates, errors might be accumulated through the sequence. To prevent this from occurring, certain frames and associated head poses (usually including the initial frame and pose) are stored as references. Whenever the estimated head pose at a frame is close to that of one reference frame, we re-register this frame to the reference so that the accumulated error can be rectified.

After recovering the head pose in the current frame, the system calculates the tracking error by warping the template into the estimated pose and computing the difference between the current head image and the warped one. If the error is larger than a pre-set threshold, the system will re-register it to the reference frame with the closest pose. If the error after this step is still large, we re-register it to the initial reference frame and use the initial reference as the template to track the following

frames. This initial template will not be updated until the tracking error is smaller than the threshold. This process enables the approach to recover from errors, especially when the head is momentarily lost, such as occurs when the head moves temporarily out of the camera's view.

6. Regularization

The aperture problem will cause the singularity of the Hessian matrix ($\sum_{\Omega}\left(w\left(I_{\boldsymbol{u}}\boldsymbol{F}_{\boldsymbol{\mu}}\right)^{\mathrm{T}}\left(I_{\boldsymbol{u}}\boldsymbol{F}_{\boldsymbol{\mu}}\right)\right)$). This will make the approach ill-conditioned (with high condition number). To reduce the condition number and improve the robustness, the regularization technique is applied and a regularization term is incorporated into the objective function (2):

$$\min\ \mathrm{E}(\boldsymbol{\mu})=\sum_{\boldsymbol{u}\in\Omega}\mathrm{w}(\boldsymbol{u})\left(\mathrm{I}(\boldsymbol{F}(\boldsymbol{u},\boldsymbol{\mu}),\mathrm{t}+1)-\mathrm{I}(\boldsymbol{u},\mathrm{t})\right)^2+\lambda\sum_{\boldsymbol{u}\in\Omega}\mathrm{w}(\boldsymbol{u})\left\|\boldsymbol{F}(\boldsymbol{u},\boldsymbol{\mu})-\boldsymbol{u}\right\|^2 \qquad (15)$$

where $\lambda>0$ is a scalar that controls how strong the regularization term is. The larger λ means the stronger regularization. This term tends to limit the amount of the optic flows so that in the cases of ill-conditioning, the estimated motion parameters will not be exploded and can be possibly recovered in the following iterations. It thus improves the robustness of the approach. We decrease λ after each iteration so that the regularization has less influence while the motion recovery is getting better.

The solution of (15) is:

$$\boldsymbol{\mu}=-\left(\sum_{\Omega}w\left(\left(I_{\boldsymbol{u}}\boldsymbol{F}_{\boldsymbol{\mu}}\right)^{\mathrm{T}}\left(I_{\boldsymbol{u}}\boldsymbol{F}_{\boldsymbol{\mu}}\right)+\lambda\boldsymbol{F}_{\boldsymbol{\mu}}^{\mathrm{T}}\boldsymbol{F}_{\boldsymbol{\mu}}\right)\right)^{-1}\sum_{\Omega}\left(wI_t\left(I_{\boldsymbol{u}}\boldsymbol{F}_{\boldsymbol{\mu}}\right)^{\mathrm{T}}\right) \qquad (16)$$

where $\sum_{\Omega}w\left(\left(I_{\boldsymbol{u}}\boldsymbol{F}_{\boldsymbol{\mu}}\right)^{\mathrm{T}}\left(I_{\boldsymbol{u}}\boldsymbol{F}_{\boldsymbol{\mu}}\right)+\lambda\boldsymbol{F}_{\boldsymbol{\mu}}^{\mathrm{T}}\boldsymbol{F}_{\boldsymbol{\mu}}\right)$ is the new Hessian matrix. In the experiments, when the amount of the condition number of the previous Hessian has the order of $O(10^6)$, that of the new Hessian has the order of $O(10^4)$.

7. A Real-Time System

Based on the above formulations, we built a real-time full head motion recovery system on a desktop (PIII-500). A low quality CCD camera positioned atop the computer monitor captures images in a typically illuminated office with a desk lamp also atop the monitor. The pixel resolution of the captured images is 320×240 with 24-bit color resolution. In each frame, the head occupies roughly between 5 and 30 percent of the total area. Tracking speed averages about 15 frames per second.

We applied the progressive Gaussian pyramid (three levels) to speed up the system and

accommodate large motions. At the top level of the pyramid, we simplify the motion to only consist of three translations and roll (in-plane rotation). Then at the other two levels, the full motion is considered. To avoid the loss of information when filtering, according to the Sampling Theorem, we set the standard deviation of the Gaussian filter at each level as the corresponding scalar (2^i for the ith level from the bottom in our system.)

To delimit the reference frame and achieve the cylindrical head model, the user presents a frontal face to the camera in the initial frame or identifies an appropriate frame if using a pre-recorded image sequence. The face image then is extracted as the reference either manually or automatically using a face detector [Rowley, 1998]. Using the position and size of the face image, the head model is generated automatically. Unless the physical face size or the distance between the face and camera are known, the head model and its initial location will be up to a scale. In the experiments, the approach appears insensitive to small variations in the initial fit.

In further pre-processing, histogram matching reduces the effect of global lighting changes, and a 2D color-blob face tracker roughly estimates the 2D head translations as an initial guess for the recovery of the full head motion.

8. Performance Evaluation

We evaluate the system in three experiments. In the first, we use synthetic image sequences with known ground truth and specified error source. In the second, we use real image sequences from Boston University (*www.cs.bu.edu/groups/ivc/HeadTracking*) whose ground truth head motion had been measure by "Flock of Birds" 3D tracker, and those from our own university whose ground truth was measured by Optotrak. In the third, we use real image sequences that contain large pitch and yaw motions (up to 50° and 90°, respectively) and occlusion.

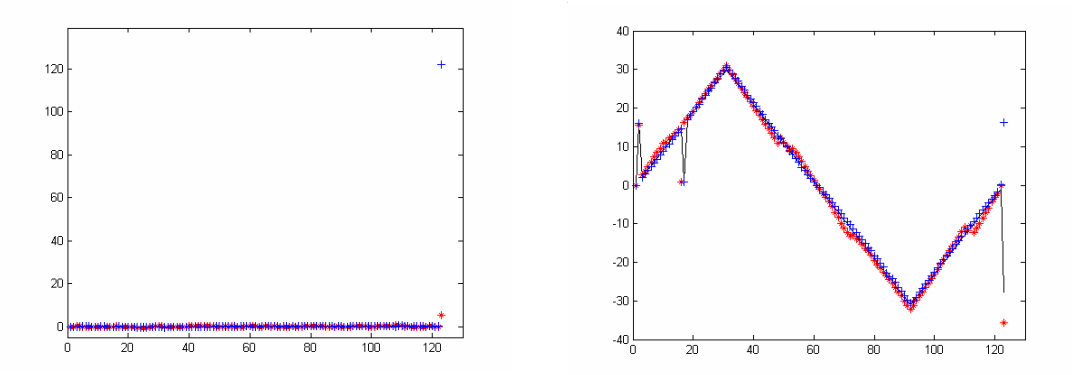
8.1 Synthetic case

Figure 4 and 5 show an example of a synthetic image sequence. A texture-mapped cylinder with Gaussian noise moves (mostly with rolls and yaws) on the black background. The meshes represent the estimated positions of the frontal area (the initial template) of the cylinder in the images. The region covered with red meshes is visible and that with yellow meshes is invisible (self-occluded) in that frame. The first row consists of the original images, the second row shows the tracking results with the pure IRLS and without regularization, and the third row shows the results using the compensated IRLS and regularization. In most cases, the system works well in both situations. When the motion between two contiguous frames is very large, however, as shown in Figure 4(c) and 4(d), compensated IRLS with regularization works much better.

Figure 5 shows the estimated pitches and yaws by the system, with and without the compensated IRLS and regularization, compared with the ground truth. Their colors are red, blue and black respectively. The horizontal axis means the frame numbers and the vertical axis means the pitch or yaw angles (degrees). Note that in most cases the compensated IRLS and regularization don't improve the recovery, but it helps when the motion is very large.

(a) (b) (c) (d)

Figure 4: A synthetic sequence with Gaussian noise: (a) Frame 1; (b) Frame 31; (c) Frame 122; (d) Frame 123. Row 1: the original images; Row 2: the results with IRLS and without regularization; Row 3: the results with compensated IRLS and regularization.



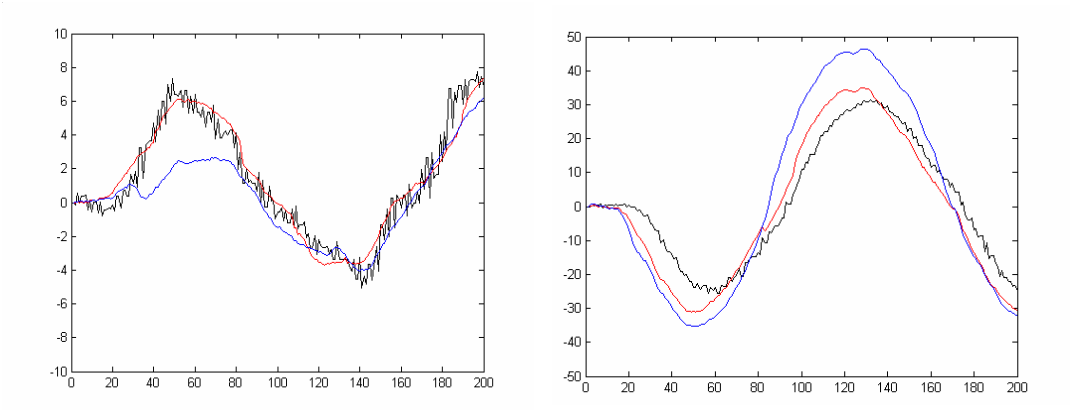
Pitch Yaw

Figure 5: Comparison between estimated poses and the ground truth. Red Star: The estimates with compensated IRLS and regularization; Blue Crossing: the estimates with pure IRLS and without regularization; Black Curve: the ground truth.

8.2 Real sequences with ground truth

We used for evaluation over 45 image sequences with associated ground truth from Boston University. A typical example is shown in Figure 6 and 7. In this sequence, large yaws (up to 31°) are present. We compared our method with a planar model-based method in this experiment. The first row of Figure 6 shows the original images. The second row shows the results of the planar model-based method, where the quadrangles show the positions of the frontal face. The third row shows the results of our system, where the meshes show the frontal faces and the white arrows show the face orientations. In addition, the positions of several feature points are shown as the crossings. The user specifies these features in the initial frame and their positions in the following frames are computed according to the estimated motion. Whether they are well tracked shows the accuracy of the system. Figure 7 shows the estimated rolls and yaws using our system and the planar model-based method, compared with the ground truth. The colors are red, blue and black respectively, which demonstrate that better performance was achieved using the cylindrical model.

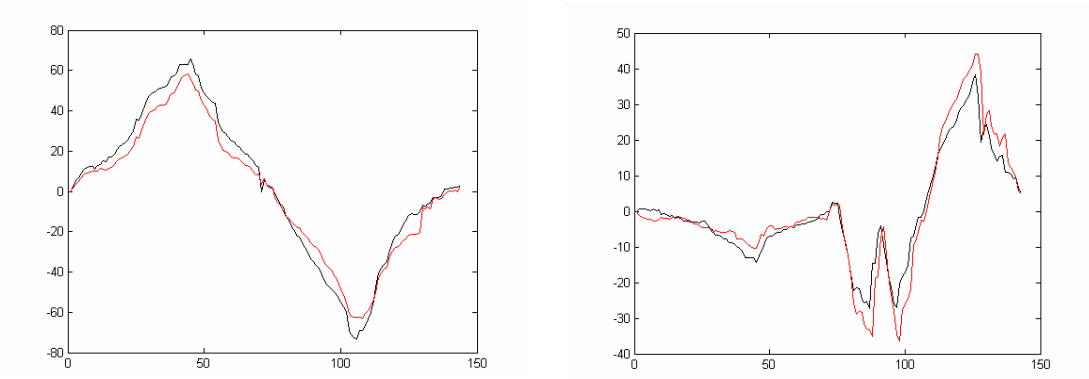
Figure 6: A real sequence: Column 1 to 4: Frame 1, 32, 100, 133. Row 1: the original images; Row 2: the results using the planar model-based method; Row 3: the results using our system.



Roll Yaw

Figure 7: Comparison among the estimated poses and the ground truth. Red: our system; Blue: the planar model-based method; Black: the ground truth.

Five sequences were obtained using an Optotrak. Lighting conditions were poor. There are apparent shadows, which are not invariant, on the faces. Figure 8 shows an example of two image sequences. One of the sequences involves large yaws (up to 75°) and another one includes large pitches (up to 40°). Estimated pitches and yaws, compared with the ground truth, are shown in Figure 8. The curve for estimated pose is highly consistent with that for ground truth, even after larger out-of-plane rotations.



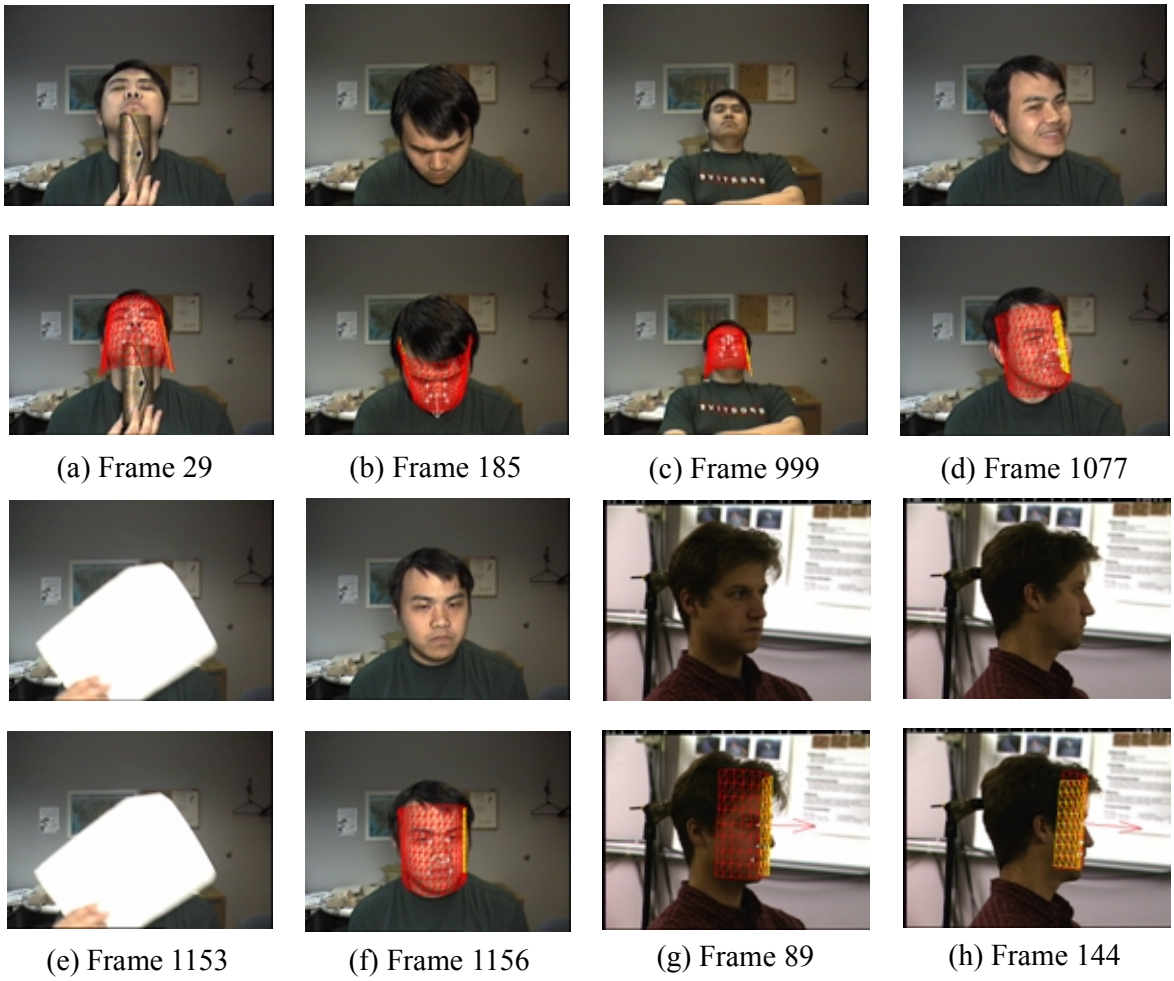
Yaw (sequence 1) Pitch (sequence 2)

Figure 8: Comparison between the estimated poses and the ground truth. Red: estimated poses using our system; Black: the ground truth.

For both databases, the system achieved high precision, e.g., in average, the recovery accuracy of rolls, pitches, and yaws are about 1.4°, 3.2°, and 3.8° respectively.

8.3 Real sequences with large motion and occlusion

We tested the system in hundreds of image sequences in an office environment. Two of them are shown in Figure 9. One sequence was taken and tracked online. It is over 20 minutes in duration and involves large rotations and translations, occlusion, and changes in facial expression. The other one includes large yaws (close to 90°). By visual inspection, estimated and actual pose were consistent (Figure 9). Even after the head is momentarily lost from view, the system can still recover 3D pose after re-registering to the initial reference frame, as shown in Figure 9(e) and (f). This result suggests that the system could work robustly for an indefinite period of time. The rough initial fits in all the experiments suggest the system is not sensitive to small initialization errors.



(a) Frame 29 (b) Frame 185 (c) Frame 999 (d) Frame 1077

(e) Frame 1153 (f) Frame 1156 (g) Frame 89 (h) Frame 144

Figure 9: More examples, including re-registration after losing the head. (a~f): Sequence 1; (g~h): Sequence 2.

9. Application: Automatic Eye-Blinking Recognition in Spontaneously Occurring Behavior

We tested the method as part of a facial expression recognition system for use with spontaneous facial actions. In spontaneous facial actions, moderate to large head motion is common. As examples, Kraut and Johnson [Kraut, 1979] found that smiling typically occurs while the head is turning toward another person. Camras, Lambrecht, and Michel [Camras, 1996] found that infant surprise expressions occur as the infant pitches her head back. With out-of-plane head motion, accurate registration between frames becomes a particular challenge. Previous literature has been limited to deliberate facial actions in which head motion is either absent [Bartlett, 1999; Donato, 1999] or predominantly parallel to the image plane of the camera [Tian, 2001]. By recovering 3D head motion, we were able to automatically align face images of spontaneous facial actions into a canonical view for facial expression analysis. The system recognized 98% of eye actions (blinking) in spontaneous behavior during a 2-person interview.

9.1 Eye image stabilization

Figure 10 shows the flow of eye image stabilization for blink recognition. The face region is delimited in the initial frame either manually or using a face detector [Rowley, 1998]. Head motion (6 *dof*) is recovered automatically as described above. Using the recovered motion parameters, the face region is stabilized; that is, warped to a standard or canonical view. Facial features are extracted in the image sequence, and eye blinking is recognized.

9.2 Blink recognition

For now we treat only the right eye (image left). The classification categories of eye actions are blink, multiple blink (eyelid 'flutter'), and non-blink. For this classification, the average intensity is calculated for the upper and for the lower half of the eye region. When mean intensities for the upper and lower halves are plotted over time (Figure 11), they cross as the eye changes between closed and open. When the eye is open, mean intensity in the upper half is smaller than that in the lower half, and reverses when closed. By computing the number and timing of crossings and peaks, we can detect eye actions (Figure 11).

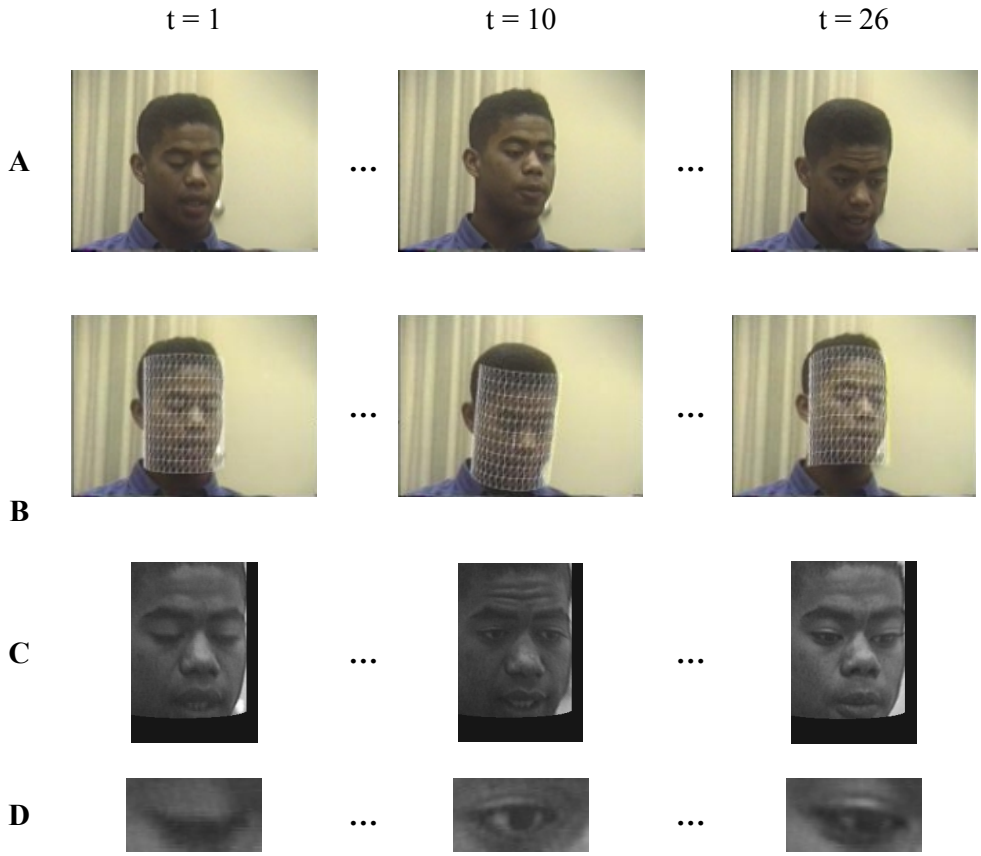
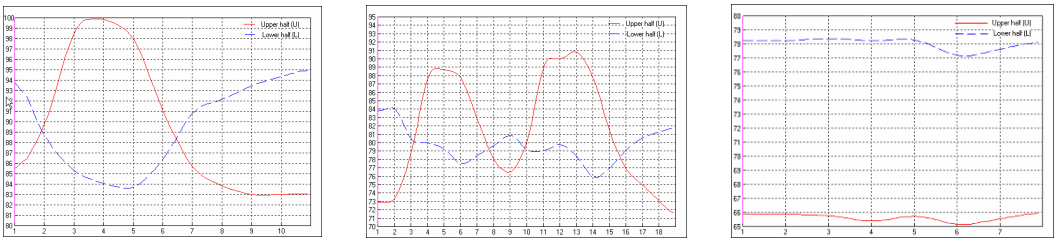


Figure 10: Automatic eye image stabilization: A) Frame 1, 10, and 26 from original image sequence; B) Tracked head in corresponding frames; C) Stabilized face images; D) Stabilized eye images.



(a) Blink **(b) Flutter** **(c) Non-blink**

Figure 11: Examples of intensity curves for blink, multiple blink (flutter), and non-blink.

9.3 Performance on eye-blinking recognition

We used video data from a study of deception by Frank and Ekman [Frank, 1997]. Subjects were 20 young adult men. Data from 10 were available for analysis. Seven of the 10 were Euro-American, 2 African-American, and 1 Asian. Two wore glasses. Subjects either lied or told the truth about whether they had stolen a large sum of money. They were video recorded using a single S-Video camera. Head orientation to the camera was oblique and out-of-plane head motion was common. The tapes were digitized into 640x480 pixel arrays with 16-bit color resolution. A certified FACS coder at Rutgers University under the supervision of Dr. Frank manually FACS-coded start and stop times for all action units [Ekman, 1978] in 1 minute of facial behavior in the first 10 subjects. Certified FACS coders from the University of Pittsburgh confirmed all coding.

Table 1: Comparison of Manual FACS Coding with Automatic Recognition

		Automatic Recognition		
		Blink	Flutter	Non-Blink
Manual FACS Coding	Blink	153	0	0
	Flutter	6	8	0
	Non-Blink	0	0	168

Overall agreement = 98% (kappa = .97). Combining blink and flutter, agreement = 100%.

Table 1 shows recognition results for blink detection in all image data in comparison with the manual FACS coding. The algorithm achieved an overall accuracy of 98% in analysis of 335 eye actions. Six of 14 multiple blinks were incorrectly recognized as single blinks. Rapid transitions from eye closure to partial eye closure to closure again, in which eye closure remains nearly complete, were occasionally recognized as a single blink. The measure we used (crossing of average intensities) was not consistently sensitive to the slight change between complete closure and partial closure. If blink and flutter are combined into a single category (which is common practice among FACS coders), classification accuracy of eye closure and opening was 100%. The 3D motion recovery was sufficiently accurate in this image database of spontaneous facial behavior to afford 98-100% accuracy for blink detection in 10 minutes of video from 10 subjects of diverse ethnic background.

10. Conclusions and Future Work

We developed a cylindrical model-based method for full head motion recovery in both pre-recorded video and real-time camera input. Three main components are compensated IRLS, dynamic templates, and re-registration techniques. The system is robust to full-head occlusion and video sequences as long as 20 minutes in duration. For pitch and yaw as large as 40° and 75°,

respectively, the system is accurate within 3° in average. We tested the method as part of a facial expression recognition system in spontaneous facial behavior with moderate head motion. The 3D motion recovery was sufficiently accurate in this image database of spontaneous facial behavior to afford 98-100% accuracy for blink recognition in 10 minutes of video from subjects of diverse ethnic backgrounds.

In current work, we are integrating the motion recovery method with feature extraction and facial action unit recognition [Lien, 2000;Tian, 2001]. With the recovered 3D head poses and the tracked features, more detailed geometry of the head can be reconstructed, including the non-rigid portions, such as mouth. To accommodate sudden and large changes of local lighting conditions, the illumination bases need to be incorporated efficiently.

Acknowledgements

This research was supported by grant number R01 MH51435 from the National Institute of Mental Health.

References

M.S. Bartlett, J.C. Hager, P. Ekman, and T.J. Sejnowski, Measuring facial expressions by computer image analysis, Psychophysiology 36, 253-263, 1999.

S. Basu, I. Essa, and A. Pentland, Motion regularization for model-based head tracking, ICPR96, 1996.

M. Black, Robust incremental optical flow, PhD thesis, Yale University, 1992.

M. Black and Y. Yacoob, Recognizing facial expressions in image sequences using local parameterized models of image motion, IJCV, vol. 25, no. 1, pp. 23-48, 1997.

C. Bregler and J. Malik, Tracking People with Twists and Exponential Maps, CVPR98, pp. 8-15, 1998.

L.A. Camras, L. Lambrecht, and G.F. Michel, Infant "surprise" expressions as coordinative motor structures, Infant Behavior and Development, 20, 183-195, 1996.

M.L. Cascia and S. Sclaroff, Fast, Reliable Head Tracking under Varying Illumination, CVPR99, pp. 604-610, 1999.

D. DeCarlo and D. Metaxas, The Integration of Optical Flow and Deformable Models with Applications to Human Face Shape and Motion Estimation, CVPR96, pp. 231-238, 1996.

G.L. Donato, M.S. Bartlett, J.C. Hager, P. Ekman, and T.J. Sejnowski, Classifying Facial Actions, IEEE Transactions on Pattern Analysis and Machine Intelligence 21, 974-989, 1999.

P. Ekman and W. Friesen, Facial Action Coding System, Consulting Psychologists Press, Palo Alto, CA, 1978.

I.A. Essa and A.P. Pentland, Coding analysis, interpretation, and recognition of facial expressions, PAMI, vol. 19, no. 7, pp. 757-763, 1997.

M. Frank and P. Ekman, The ability to detect deceit generalizes across different types of high-stake lies, Journal of Personality & Social Psychology, 72, 1429-1439, 1997.
D.B. Gennery, Visual tracking of known three-dimensional objects, IJCV, vol. 7, no. 3, pp. 243-270, 1992.
G.D. Hager and P.N. Belhumeur, Efficient region tracking with parametric models of geometry and illumination, PAMI, vol. 20, no. 10, pp. 1025-1039, 1998.
T. Jebara and A. Pentland, Parameterized Structure from Motion for 3D Adaptive Feedback Tracking of Faces, CVPR97, 1997.
R.E. Kraut and R. Johnson, Social and emotional messages of smiling: An ethological approach, Journal of Personality and Social Psychology, 37, 1539-1553, 1979.
J.J. Lien, T. Kanade, J. Cohn, and C. Li, Detection, tracking, and classification of subtle changes in facial expression, Journal of Robotics and Autonomous Systems, Vol. 31, 131 - 146, 2000.
Z. Liu and Z. Zhang, Robust Head Motion Computation by Taking Advantage of Physical Properties, HUMO2000, 2000.
D.G. Lowe, Robust model-based motion tracking through the integration of search and estimation, IJCV, vol. 8, no. 2, pp. 113-122, 1992.
B.D. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision, Proc. Int. Joint Conf. Artificial Intelligence, pp. 674-679, 1981.
J.L. Meriam and L.G. Kraige, Engineering Mechanics Vol. 2: Dynamics, John Wiley & Sons, New York, 1987.
R.M. Murray, Z. Li, and S.S. Sastry, A Mathematical Introduction to Robotic Manipulation, CRC Press, 1994.
P.J. Rousseeuw and A.M. Leroy, Robust Regression and Outlier Detection, John Wiley & Sons, New York, 1987.
H.A. Rowley, S. Baluja, and T. Kanade, Neural Network-Based Face Detection, PAMI, vol. 20, no. 1, pp. 23-38, Jan. 1998.
Y.L. Tian, T. Kanade, and J.F. Cohn, Recognizing action units for facial expression analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, 23, 97-116, 2001