Robust Full-Motion Recovery of Head by Dynamic Templates and Re-registration Techniques

Jing Xiao[1], Takeo Kanade[1], Jeffrey F. Cohn[2]
[1]Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213
[2]Department of Psychology, University of Pittsburgh, Pittsburgh, PA 15260
{jxiao, tk}@cs.cmu.edu, jeffcohn@pitt.edu

Abstract

This paper presents a method to recover the full-motion (3 rotations and 3 translations) of the head using a cylindrical model. 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 deal with 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 keep tracking the head when most of the face is not visible. Third, because the dynamic templates may cause error accumulation, we re-register images to a reference frame when head pose is close to a reference pose. The performance of the real-time tracking program was evaluated in three separate experiments using image sequences (both synthetic and real) for which ground truth head motion is known. The real sequences included pitch and yaw of as large as 40° and 75°, respectively. The average recovery accuracy of the 3D rotations was found to be about 3°.

Key Words: 3D head motion recovery, perspective projection, robust statistics, dynamic templates.

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 approach is to use distinct image features [2,4,7,18], which works well when the features are reliably tracked over the image sequence. When good feature correspondences over the entire sequence are not available, tracking the entire head region using a 3D head model is more reliable. One can use an anatomically-based model for head motion recovery. However, such a method tends to require precise or manual initialization for it to work well.

Use of a much simpler geometric model of a head is often effective. Various planar model-based methods have been presented [8,10]. They treat the face as a plane and use a single face texture (static template) to recover the head motion. They work well when the head orientation is not far from the frontal view. In [1,5], an ellipsoidal model was used with good results on 3D head or body tracking. Cascia et al. [3] developed a fast 3D head tracker that models a head as a texture-mapped cylinder. The head image is treated as the linear combination of a set of bases that are generated by changing the pose of a single head image (template). Then the head pose of the input image is estimated by computing coefficients of the linear combination. While simple and effective, the method does not seem to be able to deal with a case where a large out-of-plane rotation turns the face away from the camera because a single and static template of the head is used.

This paper presents a robust cylindrical model-based method to recover full motion of the head under perspective projection. Three main techniques contribute to the robustness of the approach. First, to deal with non-rigid motion and occlusion, we use the iteratively re-weighted least squares (IRLS) technique [14]. 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 deal with 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 head tracking system.

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,

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

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

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

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 [13],

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

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:

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

$\mathrm{R}_{3\times3}$ 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[1] and use the twist representation [15]. The transformation M can be represented as [1,15]:

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

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:

$$\boldsymbol{u}\left(t+1\right)=\begin{bmatrix}x-y\omega_z+z\omega_y+t_x\\ x\omega_z+y-z\omega_x+t_y\end{bmatrix}\cdot\frac{f_L}{-x\omega_y+y\omega_x+z+t_z}\left(t\right) \tag{7}$$

where f_L is the focal length. (7) is the parametric motion model $\boldsymbol{F}(\cdot)$ 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 $\boldsymbol{\mu}=0$,

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

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 [11]. 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) [14]. 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 :

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

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

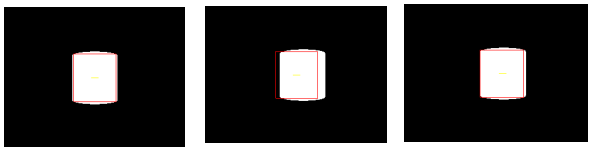
$$\sigma_I = 1.4826 \cdot \underset{\boldsymbol{u}\in\Omega}{median} \left|I(\boldsymbol{u},t+1)-\hat{I}(\boldsymbol{u},t)\right| \quad (10)$$

where the factor 1.4826 is a correction term that makes the median equal to the standard deviation of a normal distribution [12]. 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 Fig.1 (a). To compensate for this side effect, we apply another weight w_G by using the gradient of the processed image:

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

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. Fig.1 (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)

Fig. 1: 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 Fig. 2. We compute the pixel density weight by a quadratic function (because we use a quadratic surface (cylinder) as the model):

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

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.

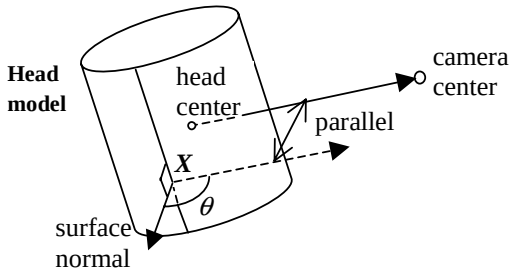


Fig. 2: Angle θ between the surface normal and the direction from the head center to the camera center, which is used to determine the pixel density weight w_D.

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

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

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

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 or error exceeds a threshold, we re-register this frame to the reference so that the accumulated error can be rectified. This process also enables the approach to recover 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):

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

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:

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

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 deal with 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 [17]. 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.

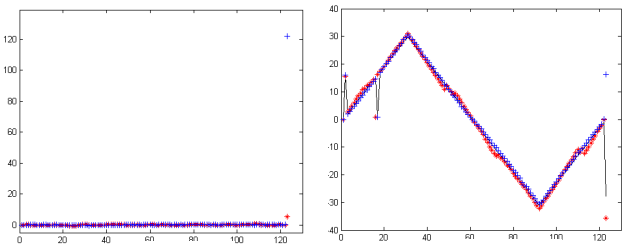
8.1 Synthetic case

Fig. 3 and 4 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 Fig. 3(c) and 3(d), compensated IRLS with regularization works much better.

Fig. 4 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)

Fig. 3: 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

Fig. 4: 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 Fig. 5 and 6. 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 Fig. 5 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. Fig. 6 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.

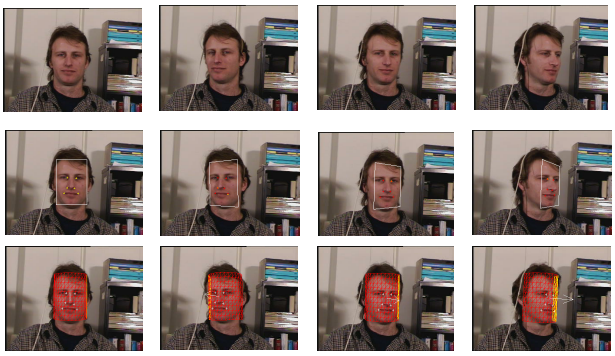
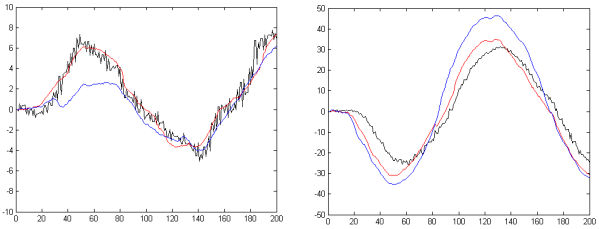


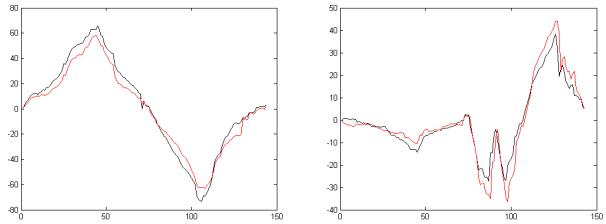
Fig. 5: A real sequence: Column 1 to 4: Frame 1, 32, 100, 133. Row 1: the original images; Row 2: the results using the planar model-based method; Row 3: the results using our system.



Roll Yaw

Fig. 6: 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. Fig. 7 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 Fig. 7. The curve for estimated pose is highly consistent with that for ground truth, even after larger out-of-plane rotations.



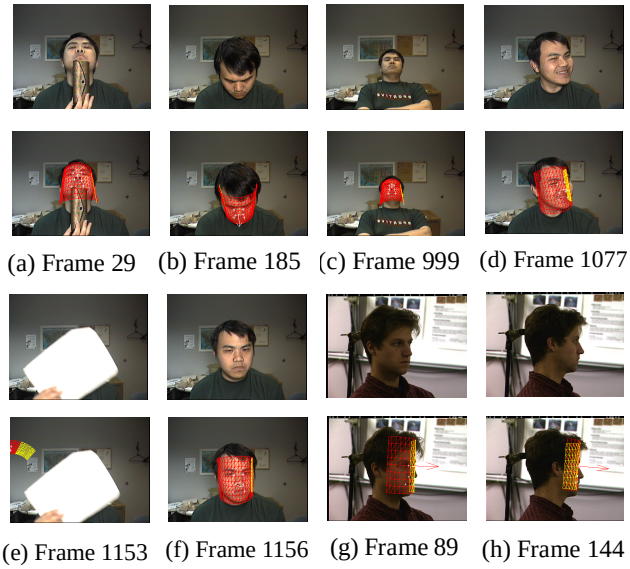
Yaw (sequence 1) Pitch (sequence 2)

Fig. 7: 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 Fig. 8. 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 [Fig. 8]. 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 Fig. 8(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

Fig. 8: More examples, including re-registration after losing the head. (a~f): Sequence 1; (g~h): Sequence 2.

While the system's primary function is to estimate 3D motion, it appears useful in separating non-rigid facial motion from rigid motion. The former manifests as outliers. After motion recovery, we can align the head images onto the head model and their residual errors provide a good measure of some expression changes, given the initial statuses. As shown in Fig. 9, eye-closure can be detected.

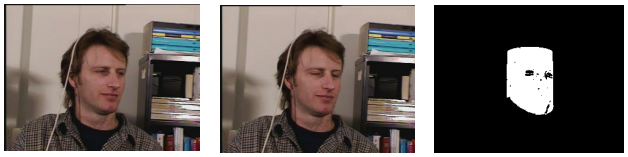


Fig. 9: Left: eyes open; Middle: eyes closed; Right: After alignment, the face mask in latter frame with the outliers taken out: eye-closure can be detected.

9. Conclusions and Future Work

In this paper, we developed a robust cylindrical model-based method for full head motion recovery. Three main techniques, compensated IRLS, dynamic templates,

and re-registration techniques, contribute to the robustness of the approach. A real-time system was built based on the method and its performance was evaluated.

In the future, first we plan to combine this method with the feature-based method for purpose of facial expression analysis. Because salient features should provide very important information about facial motion (non-rigid), a feature-based component may be necessary for facial expression analysis. Second, to deal with sudden and great changes of local lighting conditions, the illumination bases need to be incorporated efficiently. Third, although the cylindrical model is very rough, using enough head images with the different poses (estimated by the system), we can generate novel views of the head [16].

Acknowledgements

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

References

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

[2]Z. Liu and Z. Zhang. Robust Head Motion Computation by Taking Advantage of Physical Properties. In HUMO'2000, 2000.

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

[4]D.G. Lowe. Robust model-based motion tracking through the integration of search and estimation. In IJCV, vol. 8, no. 2, pp. 113-122, 1992.

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

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

[7]D.B. Gennery. Visual tracking of known three-dimensional objects. In IJCV, vol. 7, no. 3, pp. 243-270, 1992.

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

[9]D. DeCarlo and D. Metaxas. The Integration of Optical Flow and Deformable Models with Applications to Human Face Shape and Motion Estimation. In CVPR'96, pp. 231-238, 1996.

[10]G.D. Hager and P.N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. In PAMI, vol. 20, no. 10, pp. 1025-1039, 1998.

[11]J.L. Meriam and L.G. Kraige. Engineering Mechanics Vol. 2: Dynamics. John Wiley & Sons, New York, 1987.

[12]P.J. Rousseeuw and A.M. Leroy. Robust Regression and Outlier Detection. John Wiley & Sons, New York, 1987.

[13]B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In Proc. Int. Joint Conf. Artificial Intelligence, pp. 674-679, 1981.

[14]M. BLACK. Robust incremental optical flow. PhD thesis, Yale University, 1992.

[15]R.M. Murray, Z. Li, and S.S. Sastry. A Mathematical Introduction to Robotic Manipulation. CRC Press, 1994.

[16]C. Buehler, M. Bosse, L. McMillan, S. Gortler and M. Cohen, Unstructured Lumigraph Rendering, SIGGRAPH'01, LA, 2001.

[17]H.A. Rowley, S. Baluja, and T. Kanade, Neural Network-Based Face Detection, In PAMI, vol. 20, no. 1, pp. 23-38, Jan. 1998.

[18]T. Jebara and A. Pentland. Parameterized Structure from Motion for 3D Adaptive Feedback Tracking of Faces. In CVPR'97, 1997.