A Stabilized Adaptive Appearance Changes Model for 3D Head Tracking

Zoran Zivkovic, Ferdinand van der Heijden

Laboratory for Measurement and Instrumentation
Faculty of Electrical Engineering, University of Twente
P.O. box 217, 7500 AE Enschede, The Netherlands
E-mail: {Z.Zivkovic,F.vanderHeijden}@el.utwente.nl

Abstract

A simple method is presented for 3D head pose estimation and tracking in monocular image sequences. A generic geometric model is used. The initialization consists of aligning the perspective projection of the geometric model with the subjects head in the initial image. After the initialization, the gray levels from the initial image are mapped onto the visible side of the head model to form a textured object. Only a limited number of points on the object is used allowing real-time performance even on low-end computers. The appearance changes caused by movement in the complex light conditions of a real scene present a big problem for fitting the textured model to the data from new images. Having in mind real human-computer interfaces we propose a simple adaptive appearance changes model that is updated by the measurements from the new images. To stabilize the model we constrain it to some neighborhood of the initial gray values. The neighborhood is defined using some simple heuristics.

1. Introduction

The reconstruction of 3D position and orientation of objects in monocular image sequences is an important task in the computer vision society. This paper concentrates on 3D human head tracking. The applications we have in mind are: model-based coding for video conferencing, view stabilization for face expression recognition and various possible human-computer interface applications. Anyway, the approach proposed here can be applied in general for rigid object tracking in 3D.

In the initialization procedure we align our generic geometric head model with the observed subject's head. This can be done manually, or automatically by using some other algorithm. For new images in the sequence, tracking consists of estimating the human head pose with respect to this initial pose. Because of the perspective projection of standard cameras it is possible to estimate the 3D pose from the 2D image data. We use an initially aligned generic geometric 3D head model. Therefore, as described later, the 3D pose is estimated only up to a scaling factor. However, this is of no importance for the applications we are considering.

The paper is organized as follows. Related work is presented in the next section. Then the geometric part of our model based approach is described. The adaptive radiometric model is presented in section 6. Finally, the whole algorithm is described and some experimental results are discussed.

2. Related work

One of the big problems in tracking algorithms is the object appearance change caused by movement under realistic light conditions. These effects are usually very hard to model. In almost all realistic situations light conditions are complex and unknown.

Many 3D head tracking methods start from tracking some distinctive feature points on the head (for example eyes, nose, mouth corners etc.) in the 2D image plane[11]. The appearance changes caused by movement in realistic light conditions are addressed by choosing appropriate similarity norms for tracking the selected feature points. A generic 3D model is then fitted to these 2D measurement to estimate the 3D head pose. The biggest drawback is that features can be lost because of occlusions or some other not modeled effects. Knowledge about the 3D object geometry can be used to predict feature point occlusion and to recover it if it appears again. An attempt is reported in [10] where they also used the 2D feature trajectories in a structure from motion algorithm to update the generic 3D model geometry.

Another way is to use the generic 3D model geometry directly. This is usually done by forming a textured 3D head model using the initial image [1]. This textured model is then fitted to the data from the new images. We also use the textured 3D model in this paper. The novelty is that only a limited number of points on the object is used to allow fast implementation.

In practice, because of complex light conditions, head movements introduce large changes in the texture of the previously described textured 3D head model. An approach is to form a large image database of the object under various light conditions. Then a model should be constructed from the data. This is usually done by finding a representative orthogonal linear subspace using *principal component analysis* (PCA) [12] [9]. This subspace is used to represent the whole database (all possible appearances of the object when it moves in realistic light conditions). How much is a new image ”face like” is calculated by measuring the distance from this subspace. This ”brute force” method needs a long and hard to perform preparation procedure, which is highly unpractical for real user interface applications. A textured cylindrical model with PCA appearance changes modeling is presented in [7]. We search here for other simpler and more appropriate solutions. In a typical situation we have only one image of the object - the initial image. Then, using some heuristics we define some neighborhood around the initial gray values to constrain possible object appearance changes.

No big appearance changes are expected for small movements between two consecutive images. We then try to use the gray levels from the new images to update the 3D objects texture. This is somewhat similar to the methods that use *optical flow* (movement of gray level patterns in the images) as their input [3]. Because of error accumulation these methods were not able to deal with longer image sequences. Some solutions were proposed trying to prevent this drift away [8] [13]. Our method constrains the texture appearance to the neighborhood of the initial values and in this way prevents the drift away.

3. Model based approach

We use a model-based approach where we try to find the best fit of the model to the images in the sequence. The parameters we want to estimate are contained in the vector:

$$\vec{q} = [\begin{array}{cccccc} x & y & z & \alpha & \beta & \gamma \end{array}]^T \qquad (1)$$

where x, y, z describe the position and α, β, γ are the Euler angles describing the head orientation in the camera coordinate system.

If we don’t take into account the previous history of the head movement and we consider all the image pixel measurements equally important, the problem can be formulated as follows:

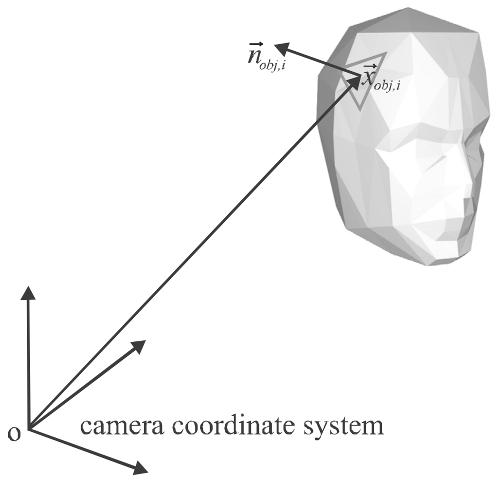

Figure 1. The Geometric Model

$$\widehat{\vec{q}} = \underset{\vec{q}}{argmin}\left(\int_{image} (model(\vec{q}) - currentimage)^2\right) \qquad (2)$$

where integration is done over the whole image area and $\widehat{\vec{q}}$ presents the estimated pose parameters. Here, $model(\vec{q})$ presents the model generated image and $currentimage$ is the current image from the camera.

In practice, it is not feasible to have the complete model of the imaging process. Therefore, we are bound to use a number of approximations. We divide the model into two parts: a geometric part and a radiometric part. These two parts are described in the next sections.

4. The geometric model

There are various ways to describe the geometry of 3D objects [2]. We use a triangular mesh, the common representation supported by fast graphics hardware. The mesh we use (Figure 1) is generated as an attempt to represent the 3D geometry of a human head.

Let $\vec{x}_{obj,i} = [\begin{array}{ccc} x_{obj,i} & y_{obj,i} & z_{obj,i} \end{array}]^T$ present the position of a fixed point i on the object’s surface in the camera coordinate system. This position, of course, depends on the head pose $\vec{x}_{obj,i} = \vec{x}_{obj,i}(\vec{q})$. For simplicity of notation we will further often omit $\vec{q}$.

We assume that the camera is calibrated. Therefore we know the perspective projection function $\vec{x}_{im,i} = p(\vec{x}_{obj,i})$ of the camera lens system that projects the 3D point $\vec{x}_{obj,i}$ to the 2D image plane point $\vec{x}_{im,i}$. If the camera doesn’t introduce any distortions we have:

$$\vec{x}_{im,i} = \begin{bmatrix} x_{im,i} \\ y_{im,i} \end{bmatrix} = p(\vec{x}_{obj,i}) = f \cdot \begin{bmatrix} \frac{x_{obj,i}}{z_{obj,i}} \\ \frac{y_{obj,i}}{z_{obj,i}} \end{bmatrix} \quad (3)$$

where f presents the focal length of the camera lens.

A generic geometric human head model is used. The size of the subject's head is unknown and we don't want to complicate the initialization procedure. Therefore, the initial position (contained in $\vec{q_0}$) is known only up to a scaling factor. As a consequence 3D head position is estimated only up to the scaling factor. As mentioned before, this doesn't present a problem for the applications we are considering.

5. Problem definition

We define a set of test points on the object $\vec{x}_{obj,i}$. For our triangular mesh model we choose the centers of the triangles in the mesh as shown in Figure 1. Our proposal is to check the fit of the model only at these object defined points and not to try to reconstruct the image. This can heavily reduce the amount of data to be processed and speed up the tracking algorithm. Therefore, our problem (2) for the k-th image can be redefined as:

$$\begin{aligned} \widehat{\vec{q}_k} &= \arg\min_{\vec{q}_k} [\frac{1}{\sum\limits_i w(i, \vec{q}_k)} \\ &\quad \sum_i w(i, \vec{q}_k) \rho(M_k(i) - I_k(p(\vec{x}_{obj,i}(\vec{q}_k))))] \quad (4) \end{aligned}$$

where the summing is done over all test points. Here, $M_k(i)$ presents the model-predicted gray value to be observed when $\vec{x}_{obj,i}$ is projected to the image plane and $I_k(p(\vec{x}_{obj,i}))$ is the actual observed value at that position in the current image. We search for the pose parameters $\widehat{\vec{q}_k}$ that give the best fit.

The measurements are weighted by:

$$w(i, \vec{q}_k) = \begin{cases} A(i) \cdot \vec{x}_{obj,i} \cdot \vec{n}_{obj,i}, & \text{for } \vec{x}_{obj,i} \cdot \vec{n}_{obj,i} < 0 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

A point $\vec{x}_{obj,i}$ corresponds to a triangular patch i of the object surface as defined by the generic geometric model. The size of the patch is denoted by $A(i)$. The current normal of the patch is described by $\vec{n}_{obj,i} = \vec{n}_{obj,i}(\vec{q}_k)$. In total, the weight $w(i, \vec{q}_k)$ presents the visible size of the triangular patch i. Note that occlusion by other triangular patches is not included in this model. However we don't expect such situations to occur often.

Because of many not modeled effects some measurements can contain unexpectedly high errors. Therefore, instead of the standard quadratic norm we use the less sensitive Geman & McClure robust error norm:

$$\rho(x) = \frac{x^2}{1 + x^2/\sigma^2} \quad (6)$$

Here, σ controls the difference beyond which a measurement is considered as an outlier [4].

6. The radiometric model

The radiometric model describes which gray value $M_k(i)$ is expected to be observed in the k-th image when object point $x_{obj,i}$ is projected onto the image plane. In general this depends on the local surface radiometric properties, local surface orientation and light conditions. Approximate radiometric models exist [6] and theoretically $M_k(i)$ should then be written as: $M_k(i, \vec{x}_{obj,i}(\vec{q}_k), \vec{q}_k)$. However, the local surface properties are unknown. Also the lighting conditions in real scenes are very complex in general and we are forced to use a number of approximations.

6.1. Approximate radiometric models

After initial alignment of the 3D object with the first image ($k = 0$) in the sequence we can obtain the values $M_0(i)$ for the test points $\vec{x}_{obj,i}$ visible for that head model pose:

$$M_0(i) = I_0(f(\vec{x}_{obj,i}(\vec{q}_0))) \quad (7)$$

where the $\vec{q}_0$ presents the parameters selected to align the generic model with the subjects head in the initial image.

The simplest approximate model is the so called *constant brightness assumption* that predicts the gray value in the k-th image as:

$$M_k^{cb}(i) = M_0(i) \quad (8)$$

This model is correct for Lambertian surfaces and with only ambient light present, which is far from realistic. A simple relaxation is to allow global brightness changes by adding a constant a to all points gray values. Further approximation is to include linear brightness changes in the image plane over the object.[5]. This crude model can be written as:

$$M_k^{lin}(i) = M_0(i) + a + \begin{bmatrix} b & c \end{bmatrix} \cdot \vec{x}_{im,i} \quad (9)$$

where we have a dot product of the vector $\begin{bmatrix} b & c \end{bmatrix}$ and image projection of the i-th object point, vector $\vec{x}_{im,i}$. The parameters a, b, c should be estimated for each new image k.

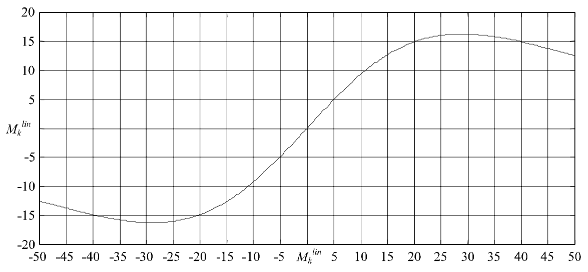

Figure 2. Function $\eta(I_k)$ for $\beta = 0.0004$

We use this additive model to describe some global illumination changes. Although this model does not need any preparation procedure, the changes in the appearance of the human face are to complex to be well approximated in this way. We introduce an adaptive model in the next section which allows floating around this model.

6.2. Adaptive radiometric model

For small object movements between two consecutive images we don't expect large changes in appearance and the *constant brightness* model can still be used. Then, an adaptive model can be formed. After model fitting on the new image using the *constant brightness assumption* between two images, the measurements from the new image can be used to update the model. The predicted value for the next $k+1$-th image becomes:

$$M_{k+1}^{adaptive}(i) = M_k^{adaptive}(i) + \alpha \cdot (innovation) \quad (10)$$

$$innovation = I_k(p(\vec{x}_{obj,i}(\widehat{\vec{q}_k}))) - M_k^{adaptive}(i) \quad (11)$$

here constant α encodes our assumption that the gray value is not supposed to change rapidly by taking into account the previous values with exponentially decreasing weights (1st order AR filtering). With this kind of $innovation$ we have the error accumulation problem but now low-pass filtered. For $\alpha = 1$ this is similar to some optical flow approaches, and for $\alpha = 0$ we get the *constant brightness assumption model*.

The initially obtained values $M_0(i)$ contain the gray values for certain head pose and illumination. We can try to use this measurements too to form the $innovation$. A crude approximation of the appearance changes from this initial values is the linear model described by (9). Our assumption is that the gray values are not going far away from this model. We incorporate this in the $innovation$ by using the following combination of the current measurement $I_k(p(\vec{x}_{obj,i}))$ and linear model $M^{lin}(i)$ which is based on the initial measurements $M_0(i)$:

$$\eta(I_k(p(\vec{x}_{obj,i}))) = \frac{I_k(p(\vec{x}_{obj,i})) - M_k^{lin}(i)}{(1+\beta\cdot(I_k(p(\vec{x}_{obj,i})) - M_k^{lin}(i))^2)^2} + M_k^{lin}(i) \quad (12)$$

The function η compresses the measured values $I_k(p(\vec{x}_{obj,i}))$ to some neighborhood of the simple model $M_k^{lin}(i)$ as presented in Figure 2. This is controlled by the constant β. Note that this function has actually the form of the derivative (*influence function*) of the robust norm introduced in (6).

Finally we define our adaptive model with:

$$innovation = \eta(I_k(p(\vec{x}_{obj,i}))) - M_k(i) \quad (13)$$

This simple model encodes our two assumptions. First, the gray values are not changing rapidly, controlled by the parameter α. Second, we approximate changes from the initial values $M_0(i)$ by a linear model $M_k^{lin}(i)$ and assume that the gray values remain in the neighborhood of this simple model , controlled by the parameter β. Only initial alignment of the 3D model is needed to form the model.

7. Algorithm

For each new image we have to find the optimal head pose vector $\widehat{\vec{q}}$ according to (4). We already need the initial alignment of the 3D model. Afterwards, we assume that there are no large changes in head pose between two successive images and for each new image we use the previous head pose as the starting position. Than we search for the nearest local minimum using Gauss-Newton iterative procedure. For determining image derivatives we use Gaussian kernels. Our measurements are also done with Gaussian blurred image at the same scale.

Note that (4) has also the weights $w(i, \vec{q}_k)$ described by (5) that depend on the current pose $\vec{q}_k$. For simplicity, we don't include this in derivatives for the Gauss-Newton iterative procedure. Anyway, this is included in the line search part after we determine the search direction. Also, the robust norm is included only as a weight factor in every iteration forming in total an iteratively reweighted least square (IRLS) minimization procedure.

Further, the parameters a, b, c for the linear approximate radiometric model should also be estimated. This could be done together with $\widehat{\vec{q}}$. For simplicity we do this separately. Since (9) is linear with respect to its parameters this is done in a single iteration. The same weights $w(i, \vec{q}_k)$ described by (5) and the same robust norm (6) are used.

Finally the whole algorithm can be described as follows:

1. initialization

   input: initial image I_0 and pose $\vec{q}_0$

   output: initial texture $M_0(i)$

- obtain $M_0(i)$ for the visible points according to the initial pose $\vec{q}_0$

2. tracking -

   input: current image I_k, current texture $M_k(i)$ and predicted pose $\overline{\vec{q}}_k(=\vec{q}_{k-1}$ in our case, we don't use any temporal model for head movement in this paper)

   output: $M_{k+1}(i), \vec{q}_k, \overline{\vec{q}}_{k+1}=\vec{q}_k$

   - constant brightness assumption, find optimal $\vec{q}_k$ according to (4)
   - fit the approximate linear model, find a, b, c
   - update model, according to (10) and (12)

8. Experiments

Various experiments were conducted. Our unoptimized test version of the algorithm was able to work at standard PAL 25 frames/second even on a low-end Pentium Celeron 500MHz. The time needed for an IRLS algorithm iteration was less than 10ms. We used three iterations per image and the rest of the time was used for visualization. A cheap web camera is used that gives 320x240 pixel images with a large amount of noise. Using few images of known objects we approximately determined the camera focal length in a simple experiment. The camera pixels are assumed to be squares. Smoothing and differentiation is done with Gaussian kernels (with standard deviation =2). For the robust norm $\sigma=100$ is used.

A demonstrational version of the algorithm can be downloaded at: `http: //www.mi.el.utwente.nl /zzz /HeadTracking /index.htm`

8.1. Experiment 1

To illustrate the operation of the adaptive model we constructed an experiment where the subject has rotated his head parallel the image plane (roll rotation) and than remained in that position. We wanted to investigate only the influence of the appearance changes. This kind of movement is chosen because it doesn't suffer from the geometric model errors. The light conditions were chosen to be not too difficult (no specular reflections and only small global brightness changes). For better comparison instead of the linear approximate model $M^{lin}(i)$ in (12) we used only the *constant brightness* model $M^{cb}(i)$. The adaptive algorithm (here only around $M^{cb}(i)$) for $\alpha=0.3$ and $\beta=0.0004$ could handle the changes but they were to big for the pure constant brightness approach ($\alpha=0$) which diverged after some time. It was also quite instable before it diverged (see Figure 3). For parameters $\alpha=1$ and $\beta=0$ the adaptive model can drift away similar to some optical flow approaches. As it can be observed in Figure 3, after this short movement the model was already not fitting the target properly.

8.2. Experiment 2

We conducted a series of experiments in typical office conditions at various locations. Some captures from the tests are presented in Figure 4. Difficult light conditions caused large appearance changes. The movements were of normal speed. Rapid movements can also be handled except for large out the plane rotations (pitch and jaw rotations). Out the plane rotations of up to approximately 35 degrees can be handled. This, however, depends on the camera focal length and object -camera distance. Web cameras have usually very small focal length and for this angle we could almost see only one half of the head (see the figures). For the parameters α and β we always used $\alpha=0.3$ and $\beta=0.0004$ and that appeared to work good for various situations. In future we plan to obtain ground truth data in order to investigate the precision of the algorithm and the influence of the parameters α and β. For the moment the results were checked only visually by backprojecting the 3D mesh head model over the images. For bigger α the tracker relied too much on the new measurements and tended to float away sooner. The parameter β describes how much the appearance can change. Too small β (big changes possible) allows the model to float away with time. At least for the initial pose (initial image) we would like to have the neighborhood defined by β small enough that the model can not float away. This can then be used as a criterion for choosing an appropriate β.

9. Conclusions and further work

A real-time 3D head tracking algorithm is presented. A simple heuristic model is used to describe the appearance changes caused by movement in realistic light conditions. The algorithm was able to operate in various realistic conditions using cheap low-end equipment. Together with an automatic initialization procedure and reinitialization when the target is lost, the algorithm seems to be a promising solution for a number of applications. The algorithm heavily relies on the initial image. Therefore, small movements around the initial head pose were handled the best. However, for many human-computer interaction applications this would be exactly the way the system would be used.

Evaluating the system using the "ground truth" data is our next step. Automatic initialization and reinitialization should be realized. The algorithm relies on the generic geometric head model that is not appropriate for all the faces. Automatically adapting the geometry to new faces would greatly improve the algorithm.

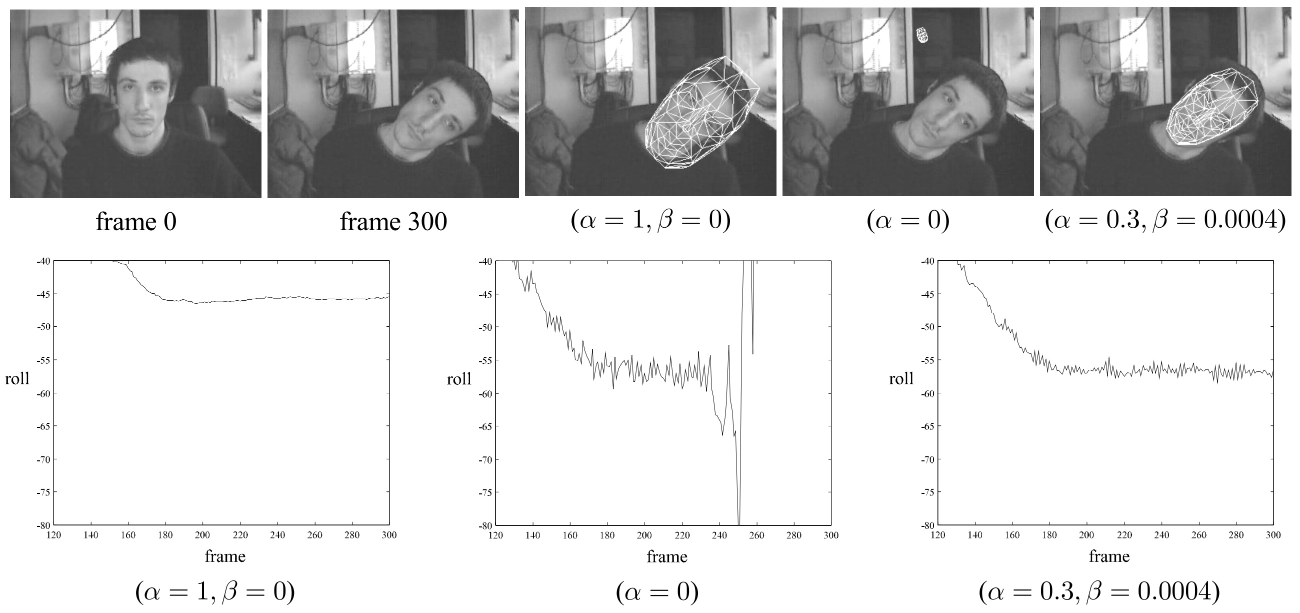

Figure 3. Adaptive model and estimated angle γ (roll rotation)

References

[1] I. E. Arno Schodl, Antonio Haro. Head tracking using a textured polygonal model. *In Proceedings of Workshop on Perceptual User Interfaces*, November 1998.

[2] N. Badler, C. Phillips, and B. Webber. *Simulating Humans: Computer Graphics Animation and Control*. Oxford University Press, 1993.

[3] S. Basu, I. Essa, and A. Pentland. Motion regularization for model-based head tracking. *In Proceedings of Intl. Conf. on Pattern Recognition*, 1996.

[4] M. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Comput. Vis. Image Understanding*, 63(1):75–104, 1996.

[5] M. Black, D. J. Fleet, and Y.Yacoob. Robustly estimating changes in image appearance. *Comput. Vis. Image Understanding*, 78(1):8–31, 2000.

[6] P. Bui-Thong. Illumination for computergenerated pictures. *Communications of the ACM*, 18(6), 1975.

[7] M. L. Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models. *IEEE Transactions on PAMI*, 22(4):322–336, 2000.

[8] D. DeCarlo and D. Metaxas. Deformable modes-based shape and motion analysis from images using motion residual error. *In Proceedings of ICCV*, pages 113–119, 1998.

[9] P. Hallinan. A low-dimensional lighting representation of human faces for arbitrary lighting conditions. *In Proceedings of Computer Vision and Pattern Recognition*, pages 995–999, 1994.

[10] T. Jebara and A. Pentland. Parametrized structure from motion for 3d adaptive feedback tracking of faces. *In Proceedings of Computer Vision and Pattern Recognition*, 1997.

[11] C. Tomasi and J. Shi. Good features to track. *In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[12] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1), 1991.

[13] Y. Zhang and C. Kambhamettu. Robust 3d head tracking under partial occlusion. *In Proceedings of Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 2000.

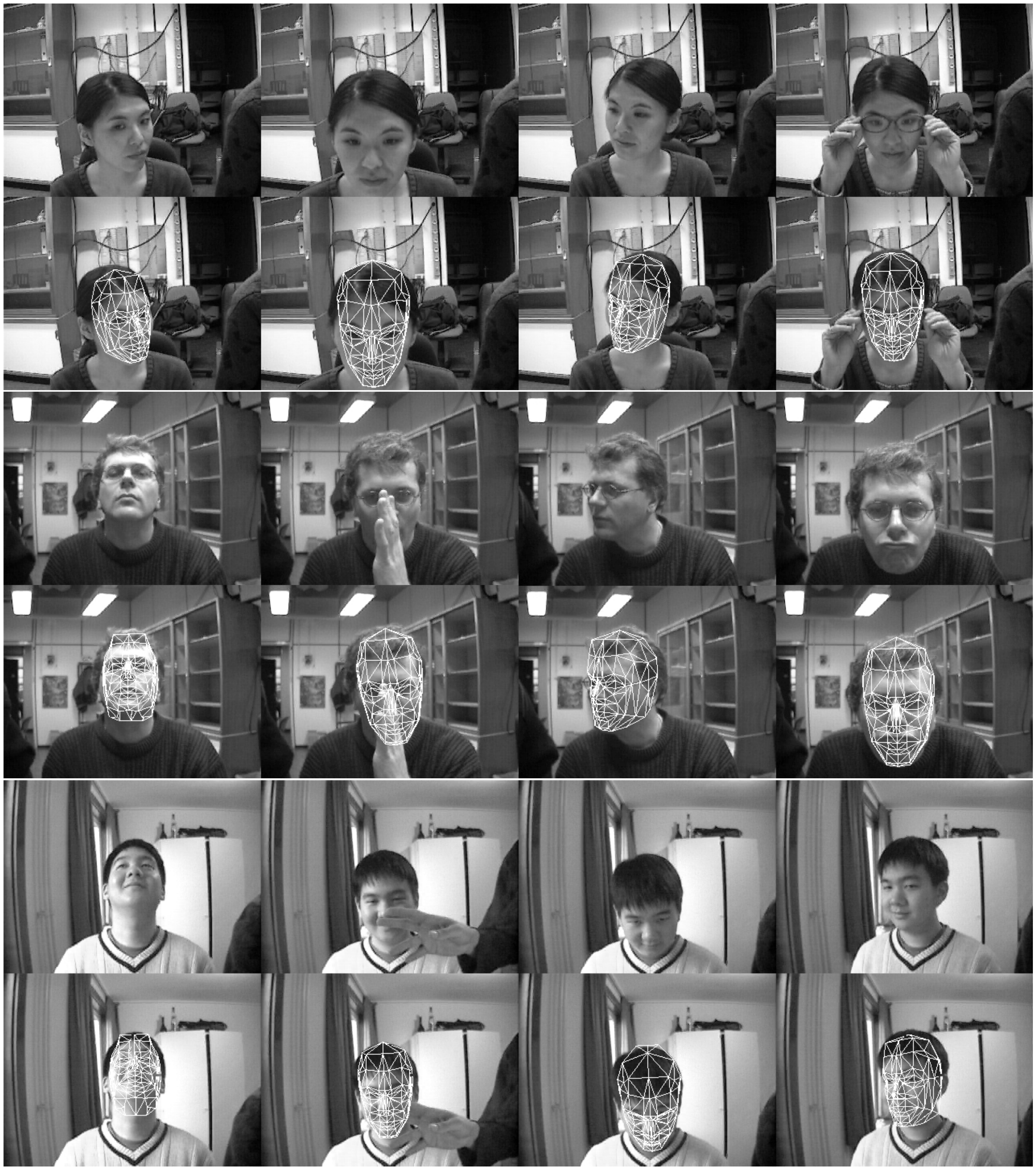

Figure 4. Real time tracking