On Importance of Nose for Face Tracking

Gorodnichy, Dimitry

National Research Council Canada
Institute for Information Technology

Conseil national de recherches Canada
Institut de technologie de l'information

NRC · CNRC

*On Importance of Nose for Face Tracking **

Gorodnichy, D.
May 2002

* published in Proceedings of the International Conference on Automatic Face and Gesture Recognition (FG2002). Washington, DC, USA. May 20-21, 2002. NRC 45854.

On Importance of Nose for Face Tracking *

Dmitry O. Gorodnichy
Computational Video Group, IIT, National Research Council, Ottawa, Canada K1A 0R6
http://www.cv.iit.nrc.ca/∼dmitry

Abstract

Human nose, while being in many cases the only facial feature clearly visible during the head motion, seems to be very undervalued in face tracking technology. This paper shows theoretically and by experiments conducted with ordinary USB cameras that, by properly defining nose – as an extremum of the 3D curvature of the nose surface, nose becomes the most robust feature which can be seen for almost any position of the head and which can be tracked very precisely even with low resolution cameras.

1 Introduction

1.1 Perceptual user interfaces

We consider a problem of tracking faces using a video camera and focus our attention on the design of vision-based perceptual user interfaces. These are the systems which use a videocamera to track user's face position in 3D in order to convert it to a position of a cursor or another virtual object on a 2D screen. They are aimed to provide a hands-free alternative to mouse, joystick and keyboard.

There are a few popular applications of perceptual user interfaces. First, they can be used to control commercial computer games, immersive 3D world *etc.* [3]. This technology has also applications in the industry for disabled. Users who have difficulty using a standard mouse could manipulate an on-screen cursor by moving their heads [18].

Face-tracking-based control can also be seen as an additional way of interfacing with a computer, which can be used, for example, to switch a focus of attention in windows environment. In addition to being hands-free, perceptual user interfaces provide a way for multiple-user interaction – several users can be tracked at the same time with several cameras.

1.2 The problems and solutions

All mentioned above applications require face tracking to be fast, affordable and, most importantly, precise and robust. Ideally, the precision of face tracking should allow a user to pinpoint to any pixel on the screen with the head, while the robustness should allow a user the convenience and the flexibility of head motion.

A few hardware companies have developed hands-free mouse replacements. In particular, in accessibility community, several companies developed products which can track a head both accurately and reliably. These products however either use dedicated software or use structured environment (e.g. markings on the user's face) to simplify the tracking process.

At the same time, recent advances in hardware, decrease of camera cost, and increase of computer power brought a lot of attention to the real-time face tracking problem from the computer vision community. The obtained vision-based solutions though still do not exhibit required precision and robustness. Let us review these solutions.

The approaches to vision-based face tracking can be divided into two classes: global and local approaches [11, 22]. Global approaches use global facial cues such as skin colour, head geometry and motion. They are robust to head rotation and scale and can be used to recover the approximate 3D position of the head. They do not require high quality images and can be used with such low-resolution cameras as USB webcams. These approaches however lack precision and therefore can not be used for the precise control of the programs.

In order to achieve precise and smooth face tracking, local approaches are used [2, 17, 5, 13, 18, 4, 16, 20, 21, 23]. These approaches are based on tracking individual facial features. These features can be tracked with pixel accuracy, which allows one to convert their positions to the cursor position. The disadvantage of these approaches however is that they usually require expensive high-resolution cameras. They are also not robust to the head motion, especially to head rotation and scale. Because of this it became a common practice in using feature-based face tracking to restrict the user motion.

Another problem of feature-based face tracking concerns the selection of facial features to be tracked. Commonly used features such as eye pupils, nostrils, corners of the brows and corners of the mouth are not always clearly visible for some face positions and expressions. Some of these

*In Proc. IEEE Intern. Conf. on Automatic Face and Gesture Recognition (FG'02), Washington DC, May 20-21, 2002.

features also present difficulty for tracking those users who wear eyeglasses, mustaches or beards.

In order to overcome this difficulty, some authors suggest tracking several facial features at the same time. This however, while indeed makes the system more robust, causes unwanted jitter of the pose determination due to the fact that facial features move independently from each other.

Another way to track both robustly and precisely is to combine local feature tracking with global cues tracking. Still the achieved results leave much to be desired. For example, the vision-based human interface system designed in [18], which uses a Sony single-CCD colour camera and a combination of local and global tracking techniques for a hands-free control of the cursor on the computer screen, while claimed to be very precise and robust, is also mentioned to require additional work prior to practical implementation.

In this paper, we argue in favour of using only one feature for the vision-based cursor control problem. First, this resolves the jitter problem and gives a user a very intuitive way of controlling the cursor by simply visualizing the feature as a tip of a joystick or a mouse. Second, if properly designed, it can be tracked both robustly and precisely. We show that this feature is nose. Instead of associating a nose feature with an actual physical point on a nose, we define it as a point on the nose surface that is the closest to the camera. This makes the nose feature the most robust feature of the face which can be seen for almost any position of the head and which can also be tracked very precisely even with low resolution cameras.

The organization of the paper is the following. The paper consists of two parts. The first part presents the theoretical basis for defining the nose feature. The second part presents experimental data which show the allowable range of the tracked user motion and demonstrates the results obtained by several programs which use the desined nose feature. The discussion concludes the paper.

2 Defining the nose

2.1 The setup

The following setup is used throughout the paper in studying and testing the approach. A user sits comfortably in front of the computer monitor, on top of which is mounted a video camera (see Figure 1). The distance from the camera to the user is such that user's hands are on the keyboard at all times. During the tracking, the user is also expected to look at the computer screen. The camera is not assumed to be calibrated or be high-resolution. Futhur more, we will assume that the camera is a generic USB camera, one of those which are widely available on the market.

Tracking of the facial feature is accomplished by means of local template matching tracking technique such as the one described in [7], which consists of scanning a window of interest with the peephole mask and comparing each thus obtained feature vector with the template vector.

The window of interest in our case, is set to half of the face size and is centered at the previous position of the feature. When the previous position of the feature is not known, the size and the location of the window of interest is obtained using colour and motion cues.

The peephole mask and the values of the template vector are learnt in advance and do not change during the tracking procedure. This brings us to the following proposition, which we make use of in selecting a robust facial feature for tracking.

Proposition: *The problem of robust tracking is the problem of constructing such a feature template vector which stays invariant during the motion a user may exhibit during the tracking.*

Below we show that the best feature to be tracked, according to the proposition, is the tip of the nose which we define in the next section.

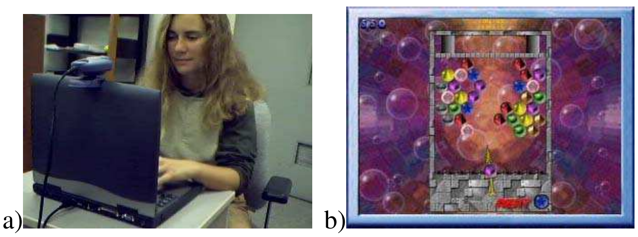
a) b)


Figure 1: A user mounts a USB web-camera (a) to play a computer game (b) hands-free; in the game, the turret fires in the direction pointed by the nose.

2.2 Feature vectors and their properties

In image processing, a feature is often thought of as a pixel $u = (i, j)$ which has large change of intensity gradient $\|\nabla I(u)\|$. This explains the choice of facial features most commonly used in tracking, such as corners of the mouth, brows, nostrils, eye pupils etc. As mentioned above, this choice of features is not always the best choice for tracking.

In order to select a robust facial feature, we will use a pattern recognition paradigm of treating features. According to this paradigm, a feature is associated with a vector made of feature attributes. Feature attributes can be pixel intensities or they can be the parameters of geometric primitives. In the case of template-based feature tracking, feature attributes are the intensity values obtained by centering a specially designed peephole mask on the position of the feature.

In the following we will denote a pixel where a facial feature is projected on the image plane as f and the associated feature vector as V_f. The distance between two vectors V_1 and V_2, which measures the error of matching of the two vectors, is denoted as $\rho(V_1, V_2)$.

For a facial feature to be easily tracked it should possess the uniqueness and robustness properties.

The *uniqueness property* states that a feature vector should lie as far as possible from other vectors in multidimensional space of feature attributes. That is

$$\rho(V_f, V_u) \to max \tag{1}$$

The *robustness property*, on the other hand, states that a feature vector should not change much during the tracking process. That is

$$\rho(V_f^{t=0}, V_f^t) \to min \tag{2}$$

These two properties define what is called the *attraction radius* of the feature, which is the largest distance from within which a feature is guaranteed to be correctly recognized [10]. The larger the attraction radius of a feature, the more robust the feature.

In addition to these properties we require a feature to have the *continuity property* which is the following. The closer pixel u in an image is to the pixel corresponding to feature f, the smaller the distance between vector V_u and feature vector V_f should be.

The continuity property appeared to be very important for precise and smooth tracking. It allows one to robustly obtain the position of the feature in an image with sub-pixel accuracy. In order to do this, the evidence theory is employed [19], and the position of the feature is calculated using the evidence-based filter as

$$\hat{u}_f = \sum_k e(u_f^k) u_f^k, \tag{3}$$

where function $e(u_f^k)$ calculates the evidence that point u_f^k in the neighborhood of the candidate feature point belongs to this feature. It is a nonlinear function, such as the Tuckey byweight [6, 14], of the correlation (or distance $\rho(V_f, V_u)$) between vector V_u and feature vector V_f.

Imposing the continuity property and using the evidence-based filter is similar to the approach taken in [7] where the adaptive logic network (ALN) [1] is used to detect eye pupils. In that work the continuity is achieved by selecting the output scheme for learning of the eye features and it is the ALN that does the filtering.

2.3 Nose feature

With the considerations of the previous section we define the nose feature as follows.

Definition: *The nose feature is defined as the point on the nose surface that is the closest to the camera. This point is termed the tip of the nose.*

Due to the symmetry and the convex shape of the nose, the nose feature is always visible in the camera, and it stays almost the same during the rotations of the head. It also does not change much with head moving towards and from the camera. Thus, the nose tip defined above can always be located.

This is very a important property of the nose which does not hold for any other facial feature. It gives a user the flexibility and convenience of motions and as will be shown below it also allows one to design the feature template vector that exhibits the continuity property.

It should be mentioned that the nose feature, defined as above, is not associated with a particular point on a nose. In fact, as will be seen from our experimental data, the located nose feature *moves* on the nose surface. The smoothness of the nose feature motion on the image plane is guaranted by the continuity property of this feature, which, as shown in the next section, it possesses.

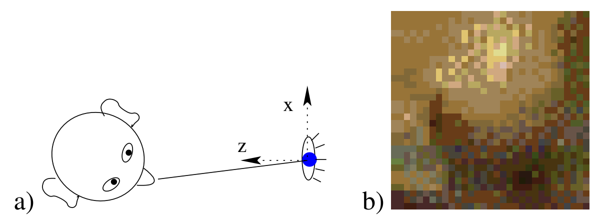


Figure 2: The nose seen by a camera: system of coordinates (a), the intensity values around the nose tip (b).

2.4 The intensity profile of the nose

In the camera centered coordinate system with Oz axis perpendicular to the image plane going from the camera (see Figure 2), the nose tip corresponds to the extremum (x_0, y_0, z_0) of the nose surface $z = z(x, y)$. Using the Taylor series, this surface can be approximated in the area around the tip of the nose as a sphere with quadratic convergence.

Using a shape from shading techniques [12], we estimate the intensity pattern around the nose tip in the projected image of the nose. For a surface of constant albedo, the intensity of pixel u where a surface point (x, y, z) is projected is related to the gradients of the surface (p, q) by the following expression

$$I(u) = aR(p, q) \tag{4}$$

where R is the reflectance map, which depends on the position of the light source, the observer, and the type of surface material, and a is a constant that depends on the surface albedo and the gain of the imaging system.

For Lambertian surfaces reflectance R is proportional to the cosine of the angle ϕ between the surface normal and the incident ray, and thus

$$I(u) \sim acos(\phi). \quad (5)$$

It can be seen from here that for a spherical surface of constant albedo, which the nose surface around the nose tip approximately is, $I(u)$ is a continuous function of u:

$$I(u + du) = I(u) + dI, \text{ with } \lim_{du \to 0} dI = 0. \quad (6)$$

There are two results following from here. First: while the head rotates, the intesity profile of around the tip of the nose stays the same. Second: the template vector made of intensity values around the tip of the nose exhibits the continuity property described in Section 2.2.

2.5 Template vector

In can be seen that under approximately the same lighting conditions, the intensity pattern around the tip of the nose is not affected much by the orientation of the nose and the distance from the nose to the camera. This helps in selecting the feature attributes to be used in designing the nose feature template. In particular, it follows from the above that the nose template vector can be made by using the gray intensities around the nose tip. If the direction of the light changes, the shape from shading technique allows one to re-compute the nose feature template so that it stays robust to the changing lighting conditions [24].

The size of the template vector is chosen so that it preserves the information about the spherical surface around the tip of the nose. It has been found that even for low resolution images such as those obtained by USB cameras, this information can be extracted.

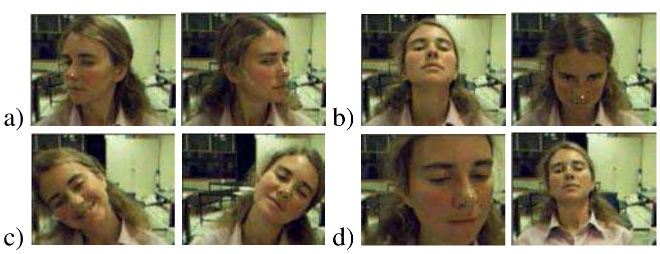


Figure 3: The figure shows the range of motion within which the nose feature is tracked.

3 Using the nose

When a new technique is proposed, it must be accompanied by adequate performance evaluation. In computer vision however and feature tracking, in particular, this is very difficult to achieve sometimes. One reason for this is the difficulty in reproducing the exactly the same video setup. The other reason is that it is often impossible to recreate all possible environments and conditions, where the technique can be applied.

In our work we strive to provide such an evaluation. This is done in three ways. First, we collect experimental data. While the paper shows only a few snapshops of our experiments, full *mpeg* videos of real-time performance are made available on our website [15]. Second, we apply the proposed technique to specific applications where its advantages can be clearly seen. Third, we promote public evaluation of the technique by making the binary code of the used programs available on the web. Thanks to the affordability of USB webcams, anybody can try it.

3.1 Collecting experimental data

The following experiments have been conducted with different USB cameras (Intel, 3Com, Creative), different computers (laptops and desktops), different people (more than twenty) and different lighting conditions. No special framegrabers or image processing hardware are used.

The experiments are aimed at evaluating the robustness, precision and also the convenience of the nose-based tracking. Due to the minimalistic nature of tracking only one feature and because fast USB cameras are used, the speed of tracking, which is a common problem of many tracking approaches, is not a problem in our experiments.

Robustness test. The first set of experiments deals with the robustness of tracking and finds the range of the head motion within which the position of the nose tip can be recovered.

A user is asked to move the head in all possible rotations while still looking at the computer screen: a) "Yes" motion, b) "No" motion, c) "Don't know" (roll of head) motion, and d) "Scale" (body) motion. The position of the located nose tip is measured. Figure 3 shows a typical result. Within the shown range the nose tip is tracked successfully. The nose is normally lost at positions beyond those shown in the figure. When this happens, which is usually accompanied by the discontinuity of the feature motion and a drastic increase of the match error, global tracking is engaged in order to rediscover the nose. This involves using the motion and colour cues and scanning the whole image.

The figure also clearly shows the main advantage of the convex-shape nose feature over the commonly used corner-based features. – The nose feature does not stay on the same spot on a face. Instead, its position relocates according to the position of the head.

Precision test. For the precision test, *NousePaint* paint program has been written which plots the position of the located nose on top of the image. Starting from the normal head position, a user is asked to draw horizontal and vertical lines by rotating the head only ("Yes" and "No" motions) and then to make a circular motion with the head. This can

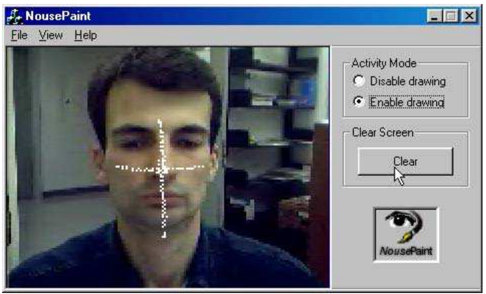


Figure 4: *NousePaint* program allows a user to draw with the nose.

be seen as a set exercises commonly used for stretching the neck. The result is shown in Figure 4 and Figure5.a. The robustness of tracking to head rotation as well as the subpixel accuracy and the continuity of tracking can be seen. An interesting application arose from this experiment, when it has been observed that not all users were able to make the exact circular motion with their heads: it can be used for testing the user's coordination and for possible neck injury.

Testing the convenience. By making it possible to use cheap and easy to install USB cameras, the proposed approach makes robust tracking very affordable. The main convenience of the approach however is seen in a very intuitive way it provides for communicating with your head.

A user may think of a nose as of a chalk and simply write with it. Due to the robustness and precision of the nose feature tracking, a user can move the nose the way it makes him/her comfortable. This makes it possible to write quite complex words and patterns hands-free. It should be noted that the speed of writing can be as fast as user is able to do it. For example (see Figure 5.b), it took 20 seconds for the author to write the title of this paper. The videos showing nose writing in real-time are available at our web-site [15].

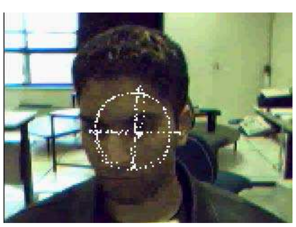
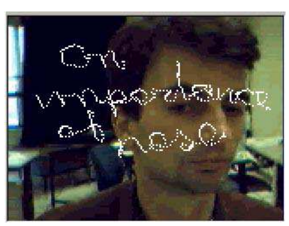
a)  b) 

Figure 5: You can write with your nose as with a chalk.

3.2 Appying the technique

The nose feature tracking presented in this paper is used in *Nouse "Use your nose as a mouse"* technology [15], which allows a user to control a computer application hands-free with the motion of his/her nose instead of using a joystick or a mouse. In order to run *Nouse*, a user adjusts a USB camera mounted on the top of the computer monitor so that his/her nose is seen in the center of the image. This initial position of the nose, referred to as the *Nouse zero* position, is equivallent to the zero position of a joystick and is used in calculation of the offsets of the nose.

Nouse software is used to initialize all nose tracking applications, such as the already mentioned *NousePaint* hands-free painting program and an arcanoid-like *BubbleFrenzy* game[1] shown in Figure 1.b. The goal of *BubbleFrenzy* is to aim a "bubble turret" (seen at the bottom of the image) in the desired direction to match the colour of the bubbles. Traditionally played with left/right mouse movements or key presses, this game can now be also played hands-free by pointing with the nose. This game was displayed at various open houses, with more than fifty people trying it. All participated players agreed that aiming with the nose is not only more fun, but is also much less tiring than doing it with mouse or keyboard. Some users experienced severe wrist fatigue when they played with mouse for longer than 15 minutes. This does not happen with *Nouse*. Using nose to aim the turret was found very natural, while the precision of aiming with nose was as good as with mouse.

Nouse face tracking technology has been applied to a few interactive games: a terrain navigation game (which uses both vertical and horizontal motions of the nose to view the terrain), *NousePong* game (which uses two cameras to allow two users to play a ping-pong game by bouncing a virtual ball with their heads), *etc.* The performance characteristics of these games are presented in [9]. All of the mentioned programs are also made available for public evaluation at our website [15].

4 Discussion

Because of the low prices and the ease of installation, USB cameras have become very popular nowadays. However, up till now these cameras are barely used for anything more intelligent than web-casting and video-surveillence. Automatic motion detection is considered almost the top high-level vision task which can be performed using these cameras.

This paper demonstrates that USB cameras can also be used for designing vision-based perceptual user interfaces. We show theoretically and by extensive experiments that a human face possesses a very unique feature – the nose; because of its prominence and convex shape, the nose feature can be seen at all times during the interaction with the computer screen, regardless of the orientation of the head and the camera. This makes it possible to track faces both robustly and precisely even with low quality cameras such as USB webcams.

[1] *BubbleFrenzy* game is available at http://www.extendedreality.com.

4.1 On importance of using two cameras

While the proposed technique allows one to use a USB camera for tracking the nose, tracking of other facial (or non-facial) features with USB cameras remains the problem, which is due to the poor quality of video captured by these cameras. In [8] we show however that this problem can be overcome by using two USB cameras.

While the idea of using stereo for tracking is not new, only recently, with the advances in the projective vision theory, it has become possible to do stereo tracking with such uncalibrated stereo setups as those made by two arbitrarily positioned USB cameras. As we show in [8], the so called fundamental matrix, which relates two images of the stereo to one another and which can be computed automatically by viewing the same scene with both cameras, not only makes it possible to track faces in 3D, but also makes tracking more precise and robust, regardless of the quality of the cameras.

It should be noted that in stereo-tracking the nose feature also plays an important role; being the most robust to the motion of the head feature, it helps detecting other facial features.

Acknowledgments

Nouse software used in taking the experimental data for the paper is the trademarks of the Computational Video Group of NRC. *BubbleFrenzy* game is supplied by Extended Reality Ltd. The author expresses special thanks to Shahzad Malik for his help in developing the code and also to Gerhard Roth and Chang Shu for inspiring discussions.

References

[1] W. Armstrong and M. Thomas. Adaptive logic networks. *Handbook of Neural Computation, Section C1.8, IOP Publishing and Oxford U. Press, ISBN 0 7503 0312 3*, 1996.

[2] S. Baluja and D. Pomerleau. Non-intrusive gaze tracking using artificial neural networks. Technical Report CMU//CS-94-102, CMU, January 1994.

[3] G. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, (2), 1998.

[4] A. Colmenarez, B. Frey, and T. Huang. Detection and tracking of faces and facial features. In *ICIP proceedings*, 1999.

[5] A. Gee and R. Cipolla. Fast visual tracking by temporal consensus. *Image and Vision Computing*, 14(2):105–114, 1996.

[6] D. Gorodnichy and W. Armstrong. Single camera stereo for mobile robots. In *Proc. Intern. Conf. on Vision Interface (VI'99), pp. 73-80, May 18-21*, 1999.

[7] D. Gorodnichy, W. Armstrong, and X. Li. Adaptive logic networks for facial feature detection. In *Proc. Intern. Conf. on Image Analysis and Processing (ICIAP'97), Vol. II (LNCS, Vol. 1311), pp. 332-339, Springer*, 1997.

[8] D. Gorodnichy, S. Malik, and G. Roth. Affordable 3D face tracking using projective vision. In *Proc. Intern. Conf. on Vision Interface (VI'2002)*, pages 383–390, Calgary, May 2002.

[9] D. Gorodnichy, S. Malik, and G. Roth. Nouse 'Use your nose as a mouse' - a new technology for hands-free games and interfaces. In *Proc. Intern. Conf. on Vision Interface (VI'2002)*, pages 354–361, Calgary, May 2002.

[10] D. Gorodnichy and A. Reznik. Static and dynamic attractors of autoassociative neural networks. In *Proc. Intern. Conf. on Image Analysis and Processing (ICIAP'97), Vol. II (LNCS, Vol. 1311), pp. 238-245, Springer*, 1997.

[11] E. Hjelmas and B. K. Low. Large receptive fields for optic flow detectors in humans. *Computer Vision and Image Understanding*, 83:236 – 274, 2001.

[12] B. K. P. Horn. Understanding image intensities. *Artificial Intelligence, Vol. 8, pp 201-231*, 1977.

[13] G. Loy, E. Holden, and R. Owens. 3D head tracker for an automatic lipreading system. In *Proc. Australian Conf. on Robotics and Automation (ACRA2000)*, Melbourne, Australia, August 2000.

[14] P. Meer, D. Mintz, A. Rosenfeld, and D. Kim. Robust regression methods for computer vision: A review. *International journal of computer vision*, 6(1):59–70, 1991.

[15] NRC. Nouse 'Use your nose as a mouse' technology website. http://www.cv.iit.nrc.ca/research/Nouse, 2001.

[16] R.Newman, Y. Matsumoto, S. Rougeaux, and A. Zelinsky. Real-time stereo tracking for head pose and gaze estimation. In *Proc. IEEE Intern. Conf. on Automatic Face and Gesture Recognition Gesture Recognition (FG2000)*, 2000.

[17] L. D. Silva, K. Aizawa, and M. Hatori. Detection and tracking of facial features. In *SPIE Visual Communications and Image Processing'95 (VCIP'95), Vol. 2501, pp. 2501/1161–2501/1172*, 1995.

[18] K. Toyama. Look, Ma – no hands! Hands-free cursor control with real-time 3D face tracking. In *Proc. Workshop on Perceptual User Interfaces (PUI'98)*, San Fransisco, November 1998.

[19] F. Voorbraak. Reasoning with uncertainty in AI. In *Reasoning with Uncertainty in Robotics (RUR'95) Intern. Workshop proceedings*, pages 52–90, 1995.

[20] M. Xu and T. Akatsuka. Detecting head pose from stereo image sequence for active face recognition. In *Proc. IEEE Intern. Conf. on Automatic Face and Gesture Recognition (FG'98)*, 1998.

[21] J. Yang, R. Stiefelhagen, U. Meier, and A. Waibel. Real-time face and facial feature tracking and applications. In *Proc. AVSP'98, pages 79–84, Terrigal, Australia*, 1998.

[22] M. Yang, N. Ahuja, and D. Kriegman. Detecting faces in images: A survey. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.

[23] K. Yow and R. Cipolla. Finding initial estimates of human face location. In *In Proc. 2nd Asian Conf. on Comp. Vision, volume 3, pages 514–518, Singapore*, 1995.

[24] W. Zhao and R. Chellappa. Robust face recognition using symmetric shape-from-shading. Technical Report CARTR -919, Center for Automation Research, University of Maryland, College Park, 1999.