TLA Based Face Tracking

Matthew Turk, Changbo Hu, Rogerio Feris, Farshid Lashkari, Andy Beall‡
Computer Science Department ‡Psychology Department
University of California, Santa Barbara, CA 93106

Abstract

Human face tracking (HFT) is one of several technologies useful in vision-based interaction (VBI), which is one of several technologies useful in the broader area of perceptual user interfaces (PUI). In this paper we motivate our interests in PUI and VBI, and describe our recent efforts in various aspects of face tracking in the Interaction Lab at UCSB. The HFT methods (GWN, EHT, and CFD), in the context of VBI and PUI, are part of an overall "TLA approach" to face tracking.

> **TLA** /T-L-A/ n. [Three-Letter Acronym] 1. Self-describing abbreviation for a species with which computing terminology is infested. 2. Any confusing acronym…. (From the Jargon File v. 4.3.1)

1 Introduction: Perceptual interfaces

The interface between people and computers has progressed over the years from the early days of switches and LEDs to punched cards, interactive command-line interfaces, and the direct manipulation style of graphical user interfaces. The "desktop metaphor" of graphical user interfaces, a.k.a. WIMP interfaces (for Windows, Icons, Menus, and Pointing devices), has been the standard interface between people and computers for many years. Of course, software and technology for human-computer interaction (HCI) is not isolated from other aspects of computing. Computers have changed enormously over their short history, increasing their speed and capacity, and decreasing component size, at an astounding rate. The size of computers is shrinking, and there are now a plethora of computer devices of various sizes and functionality. In addition, there are many non-GUI (or "post-WIMP") technologies, such as virtual reality, speech recognition, computer vision, haptics, and spatial sound, that promise to change the status quo in computer-human interaction.

One can view human-computer interaction as a hierarchy of goals, tasks, semantics, and syntax, as shown in Figure 1. The goal level describes what a person wants to do, independent of the technology – talk with a friend, for example. Tasks are the particular actions that are required to attain the goal – e.g., locate a telephone, dial a number, talk into the headset. The semantics level maps the tasks onto achievable interactions with the technology, while the syntax level specifies the particular actions (such as double clicking an icon) that accomplish a subtask.

One may view user interfaces as a necessary evil, because they imply a separation between what one wants the computer to do and the act of doing it [11], i.e., a division between the goal level and the task, semantics and syntax levels. This separation imposes a cognitive load upon the user that is in direct proportion to the difficulty and awkwardness that the user experiences. Poor design, to be sure, exacerbates the problem, giving rise to the all-too-common experience of frustration when interacting with computers.

This frustrating user experience can certainly be improved upon in many ways, and there are many ideas, initiatives, and techniques intended to help – such as user-centered design, 3D user interfaces, conversational interfaces, intelligent agents, virtual environments, and so on.

One point of view is that direct manipulation interfaces, such as the GUI/WIMP model, where users manipulate visual representations of objects and actions, and "information appliances" [8], which are devices built to do one particular task well, will alleviate many of the problems and limitations of current computer interfaces. Although this is very likely true – and such devices may well be commercial successes – it is not clear that this interface style will scale with the changing landscape of form factors and uses of computers in the future.

To complicate things, it is no longer obvious just what "the computer" is; the largely stand-alone desktop PC is no longer the singly dominant device. Rapid changes in form factor, connectivity, and mobility, as well as the continuing effects of Moore's Law, are significantly altering the computing landscape. More and more, computers are embedded in objects and systems that people already know how to interact with (such as a telephone or a child's toy) apart from their experience with stand-alone computers.

So what might replace, or at least complement, the current HCI paradigm? In recent years, some have argued

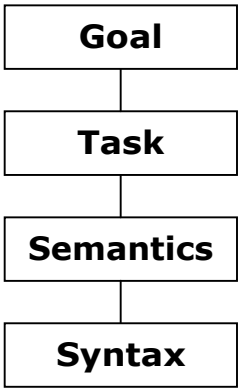

Figure 1: *Main levels of interaction*

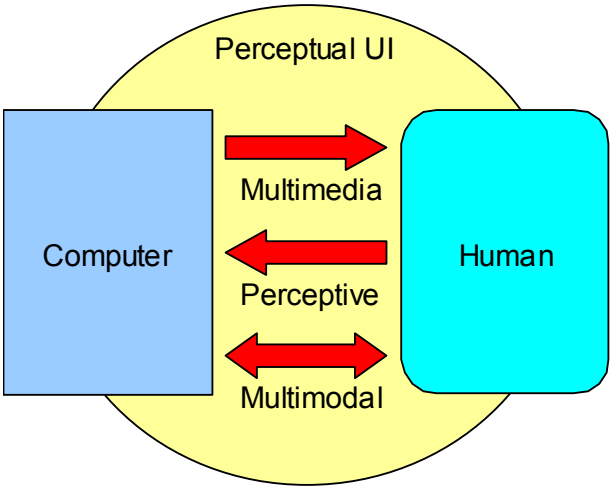

Figure 2: *Information flow in Perceptual User Interfaces (adapted from [10])*

that the primary abstraction between people and technology should be the model of human-human interaction. The most natural human interaction techniques are those which we use with other people and with the world around us – those that take advantage of our natural sensing and perception capabilities, along with social skills and conventions that we acquire at an early age. We would like to leverage these natural abilities, as well as our tendency to interact with technology in a social manner [9], to model human-computer interaction after human-human interaction. *Perceptual user interfaces (PUI)*, which seek to take advantage of both human and machine perceptual capabilities, may be defined as highly interactive, multimodal interfaces modeled after natural human-to-human interaction, with the goal of enabling people to interact with technology in a similar fashion to how they interact with each other and with the physical world [10]. Figure 2 depicts related terms and the flow of information in PUI.

Such interfaces must integrate in a meaningful way several relevant technologies, such as speech, vision, natural language, haptics, and reasoning, while seeking to understand more deeply the expectations, limitations, and possibilities of human perception and the semantic nature of human interactions.

2 Vision based interaction

Present-day computers are essentially deaf, dumb, and blind. Several people have pointed out that the bathrooms in most airports are smarter than any computer one can buy, since the bathroom "knows" when a person is using the sink or toilet. Computers, on the other hand, tend to ask us questions when we're not there (and wait 16 hours for an answer) and decide to do irrelevant (but CPU-intensive) work when we're frantically working on an overdue document.

Vision is clearly an important element of human-human communication. Although we can communicate without it, people still tend to spend endless hours traveling in order to meet face to face. Why? Because there is a richness of communication that cannot be matched using only voice or text. Body language such as facial expressions, silent nods and other gestures add personality, trust, and important information in human-to-human dialog. We expect it can do the same in human-computer interaction.

Vision based interfaces (VBI) is a subfield of perceptual interfaces which concentrates on developing visual awareness of people. VBI seeks to answer questions such as:

- Is anyone there? (Detection)
- Where are they? (Location, tracking)
- Who are they? (Identity recognition)
- What are the subject's movements? (Motion tracking and analysis)
- What are his facial expressions? (Expression analysis)
- Are his lips moving? (Lip modeling and tracking)
- What gestures is he making? (Gesture recognition)

These questions can be answered by implementing computer vision algorithms to locate and identify individuals, track human body motions, model the head and face, track facial features, interpret human motion and actions. (For a taxonomy and discussion of movement, action, and activity, see Bobick [1]).

VBI (and, in general, PUIs) can be categorized into two aspects: *control* and *awareness*. Control is explicit

communication to the system – e.g., put *that* object *there*. Awareness, picking up information about the subject without an explicit attempt to communicate, gives *context* to an application (or to a PUI). The system may or may not change its behavior based on this information. For example, a system may decide to stop all unnecessary background processes when it sees me enter the room – not because of an explicit command I issue, but because of a change in its context. Current computer interfaces have little or no concept of awareness. While many research efforts emphasize VBI for control, it is likely that VBI for awareness will be more useful in the long run.

3 Human face tracking

Of the various VBI technologies, human face tracking (HFT) is perhaps the most useful, as it can be used to support several other technologies as well as be used directly by various applications. Face tracking can serve as input to various other VBI modules, such as dynamic face recognition, facial expression analysis, audio-visual speech processing, body tracking and modeling, gesture recognition and activity analysis. The location, pose, and expression of the face is key to both extracting and interpreting human body information. Since faces are arguably the most stable and identifiable component of the body under various transformations, the face can serve as an anchor from which to relate other VBI tasks. Additionally, an understanding of face pose or expression can be vital to interpret high-level information such as facial identity or body gesture.

Face tracking may also be considered a prototype computer vision problem, since the difficulties encountered are quite similar to other difficult tracking problems in the field – there are both rigid and non-rigid components, there is both similarity and variation among members of the class of objects, there are time-varying changes that impact the problem, etc. There is an expectation that by focusing on face tracking we will contribute to the state-of-the-art in general visual object tracking as well.

In this section, we present approaches to three different problems in face tracking. These are the first steps in working toward a unified approach to tracking human faces for various applications.

3.1 Gabor Wavelet Networks for face tracking

In this section, we will show how to perform efficient face tracking using Gabor Wavelet Networks (GWN) [7]. We start by presenting the GWN approach, which is used to represent a face image as a weighted sum of specifically chosen Gabor wavelets. We then introduce the wavelet subspace tracking method [5] and discuss its main advantages and drawbacks over other related approaches.

3.1.1 Compression as Learning

To define a GWN, we start out by considering a family of N odd Gabor wavelets $\mathbf{\Psi} = \{\psi_{\mathbf{n}_1}, \psi_{\mathbf{n}_2}, ..., \psi_{\mathbf{n}_N}\}$ of the form

$$\psi_{\mathbf{n}_i}(\mathbf{x}) = \exp\left[-\frac{1}{2}(\mathbf{S}_i\mathbf{R}_i(\mathbf{x}-\mathbf{c}_i))^T(\mathbf{S}_i\mathbf{R}_i(\mathbf{x}-\mathbf{c}_i))\right] \cdot \sin\left[(\mathbf{S}_i\mathbf{R}_i(\mathbf{x}-\mathbf{c}_i))\begin{pmatrix}1\\0\end{pmatrix}\right], \quad (1)$$

where $\mathbf{x}$ represents image coordinates and $\mathbf{n}_i = (s_{x_i}, s_{y_i}, \theta_i, c_{x_i}, c_{y_i})$ are wavelet parameters which defines the wavelet scale (s_{x_i}, s_{y_i}), orientation θ_i and translation (c_{x_i}, c_{y_i}). These parameters are implicit in the equation and compose the dilation matrix $\mathbf{S}_i$, the rotation matrix $\mathbf{R}_i$ and the translation vector $\mathbf{c}_i$.

The choice of N is related to the degree of desired representation precision of the network. In order to learn the parameters of a GWN for a discrete gray level image I, the energy functional

$$E = \min_{\mathbf{n}_i, w_i} \| I - \sum_i w_i \psi_{\mathbf{n}_i} \|^2 \quad (2)$$

is minimized with respect to the weights w_i and the wavelet parameter vectors $\mathbf{n}_i$. The two vectors $\mathbf{\Psi} = (\psi_{\mathbf{n}_1}, \psi_{\mathbf{n}_2}, ..., \psi_{\mathbf{n}_N})^T$ and $\mathbf{w} = (w_1, w_2, ..., w_N)^T$ define then the Gabor wavelet network ($\mathbf{\Psi}$, $\mathbf{w}$) for image I. In other words, a Gabor wavelet network is defined through a N-dimensional vector of weights w_i and an N-dimensional vector of Gabor wavelets $\mathbf{\Psi}_{\mathbf{n}_i}$, where the weights w_i and the parameter vectors $\mathbf{n}_i$ are chosen such that the weighted sum of Gabor wavelets approximates the discrete image I optimally.

Clearly, the quality of the image representation and reconstruction depends on N, the number of wavelets, and can be varied to reach almost any desired precision.

We note here that the weights of a GWN can be computed directly using a family of dual functions. Gabor wavelet functions are not orthogonal, thus implying that, for a given family $\mathbf{\Psi}$ of Gabor wavelets, it is not possible to calculate a weight w_i by a simple projection of the Gabor wavelet $\mathbf{\Psi}_{\mathbf{n}_i}$ onto the image. In fact, a family of dual wavelets $\tilde{\mathbf{\Psi}} = \{\tilde{\psi}_{\mathbf{n}_1}, \tilde{\psi}_{\mathbf{n}_2}, ..., \tilde{\psi}_{\mathbf{n}_N}\}$ has to be considered. The wavelet $\tilde{\psi}_{\mathbf{n}_i}$ is the dual wavelet of the wavelet $\tilde{\psi}_{\mathbf{n}_j}$ iff $\langle \tilde{\psi}_{\mathbf{n}_i}, \tilde{\psi}_{\mathbf{n}_j} \rangle = \delta_{i,j}$. So, given a discrete image I, the optimal weights of the GWN that minimize

the energy in Eq. (2) are given by $w_i = \langle I, \tilde{\psi}_{\mathbf{n}_i} \rangle$. It can be shown that $\tilde{\psi}_{\mathbf{n}_i} = \sum_j (\mathbf{A}^{-1})_{i,j} \psi_{\mathbf{n}_j}$, where $\mathbf{A}_{i,j} = \langle \psi_{\mathbf{n}_i}, \psi_{\mathbf{n}_j} \rangle$.

3.1.2 Wavelet subspace tracking

As mentioned above, a discrete image I can be mapped into a vector $\mathbf{w} \in \Re^N$, using a family of dual functions $\tilde{\mathbf{\Psi}}$. This mapping corresponds to the orthogonal projection of the image I into the subspace <**Ψ**>. Similarly, the image reconstruction $\hat{I}$, i.e., the mapping of **w** into the image space, is obtained using the family of wavelets **Ψ**. Figure 3 better illustrates these mappings.

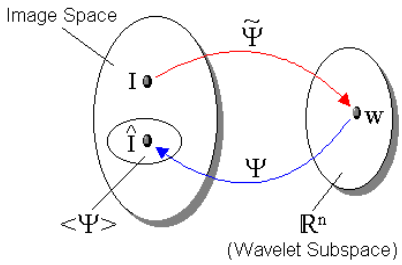

Figure 3: *Wavelet subspace mapping.*

The basic idea of the wavelet subspace tracking consists in orthogonally projecting the input video frames into the image subspace <**Ψ**>, while performing all further computations in the low-dimensional wavelet subspace $\Re^N$. Therefore, tracking is achieved based on wavelet weights, eliminating the time-consuming pixelwise difference computation in image-based approaches. It is interesting to note that the weights of the GWN are linearly related to the local Gabor filter responses and thus also reflect the underlying local image structure.

To discuss this method in more detail, let us consider a GWN (**Ψ**, **v**) that is optimized for a certain face image. As previously mentioned, the optimal weight vector **v** can be directly obtained by an orthogonal projection of the facial image into the closed linear span of **Ψ**. Hence, we say that the face template was mapped into the weights $\mathbf{v} \in \Re^N$ which we will call *reference weights*.

The tracking in wavelet subspace is performed by affinely deforming the subspace <**Ψ**>, until the weight vector $\mathbf{w} \in \Re^N$, obtained by the orthogonal mapping of the current frame into this subspace, is closest to the reference weight vector **v**. In order words, for each frame I, we need to minimize the following energy functional, with respect to the affine parameter vector $\mathbf{n} = (s_x, s_y, s_{xy}, \theta, c_x, c_y)$:

$$E = \min_{\mathbf{n}} \| \mathbf{v} - \mathbf{w} \|_{\psi} \quad \text{with}$$

$$\mathrm{w_i} = \sum_j \frac{1}{s_x s_y} (\mathbf{A}^{-1})_{i,j} \langle I, \psi_{\mathbf{n}_j} (\mathbf{SR}(\mathbf{x} - \mathbf{c})) \rangle \,, \quad (3)$$

The mapping of images into $\Re^N$ is carried out with low computational cost through a small number of local filtrations with the wavelets. As we can see in equation (3), the weights w_i are computed by a linear combination of filter responses, since matrix **A** is constant (except by a scale factor) and can be computed offline.

Figure 4 shows some video frames where the face is tracked by the wavelet subspace method. Facial features (eyes, nose and mouth) are marked at relative coordinates, just to show the face position and orientation. The image resolution is 160x120 pixels and the size of the inner face region in which the GWN was optimized is 50x65 pixels. Using only nine wavelets, the computing time for each Levenberg-Marquardt cycle (used to minimize Eq. 3) was 15ms on a 1GHz Linux-Athlon.

Figure 4: *Sample frames showing the wavelet subspace tracking.*

3.1.3 Discussion

Although the face is represented as a rigid object undergoing limited motion, different face expressions and small depth variations exhibited by facial features are well approximated by the affine wavelet model. Furthermore, since Gabor wavelets are DC free, the approach also shows robustness with respect to homogeneous illumination variations. On the other hand, the method fails under strong intensity changes and out-of-plane face deformations.

Increasing the number of wavelets in the representation leads to a more precise but slower tracking. For instance, using 51 wavelets, a computing time of 85ms per cycle was required in each frame. Clearly, the number of applied wavelets is task dependent and can be dynamically changed according to the available computer power.

GWNs invite the closest comparison with the well-known Gabor jet approach, which has also been used for real-time tracking. The advantage of GWNs is that they offer a sparser representation of image data. This is because the wavelet parameters are selectively chosen from the *continuous* space, in contrast with the Gabor jet

approach, which is based on the discrete wavelet transform. As an example, considering just 52 wavelets, GWNs provide a good representation for a face image, whereas the Gabor jet approach would require many more wavelets to get a comparable representation.

Finally, we should note the work of Hager and Belhumeur [6], which also uses an affine model to track the face. This approach has the advantage of being even faster than GWN tracking. On the other hand, the GWN approach avoids the discrete data interpolation required in the former, since, in this case, we deal with a continuous wavelet representation.

3.2 Integrating multiple cues in face tracking

Birchfield [2] proposed a robust head tracker by using boundary intensity gradients and skin color histograms. The tracker can deal with full 360-degree rotation and occlusion, but it demands a motion predicting model and an exhaustive search. It is hard to predict the head's motion by using a fixed motion model, and an exhaustive search is computationally costly.

Bradski [3] introduced a mean shift algorithm to find an optimal search path for tracking without a motion model. Mean shift is a robust nonparametric optimization technique based on probability distribution, in which the optimal solution is sought by climbing density gradients. Recently Comaniciu and Meer [4] used mean shift for non-rigid objects tracking. In their approach, the object's appearance varies little during tracking, so the tracker doesn't work well in the case of large rotation of the head or significant view variation.

We have developed a real-time robust head tracker based on a simple elliptical shape and an adaptive target color distribution similar to Birchfield's. Our Elliptical Head Tracker (EHT) method is composed of two parts. First, color (hue) is used to estimate the head's location in every frame, in which mean shift is adopted for optimal search. The hue is modified adaptively so that it can deal with the head's rotation or large view changes. After color mean shift tracking, a local search is performed to maximize the normalized gradient magnitude around the boundary of the elliptical head, in order that more accurate head location and scale can be obtained. Experimental results show that it is a real-time tracker and relatively robust to clutter, scale variation, brief full-occlusion, full 360-degree rotation and camera motion.

The head shape is modeled as an ellipse with a fixed aspect ratio of 1.2. From the tracked ellipse, the motion and pose parameters are extracted, including the position (x, y) and scale s (which is proportional to z, the distance from the camera) and in-plane rotation angle θ. That is, the state of an ellipse is described as state $\varphi=(x, y, s, \theta)$.

In the head tracking work of Birchfield [2], an ellipse is searched for in a window exhaustively. By using the mean shift color tracking, we already have the position and rotation parameter. We can also calculate the long and short axis of the ellipse by computing the eigenvalues and eigenvectors of the tracked region. But this value is in fact not the real shape of the head, especially when there is clutter. In order to produce an accurate tracking result, gradient information around the boundary of the elliptical head is used. Now given the elliptical head's state (x, y, θ), to obtain the best scale s*, a local search is performed to maximize the average of the gradient magnitude around the perimeter of the ellipse.

$$s^* = \arg\max_{s \in S} \left\{ \frac{1}{N_h} \sum_{i=1}^{N_h} |g_{si}| \right\} \quad (4)$$

where g_{si} is the intensity gradient at perimeter pixel *i* of the ellipse at scale s, and N_h is the number of pixels on the perimeter of the ellipse.

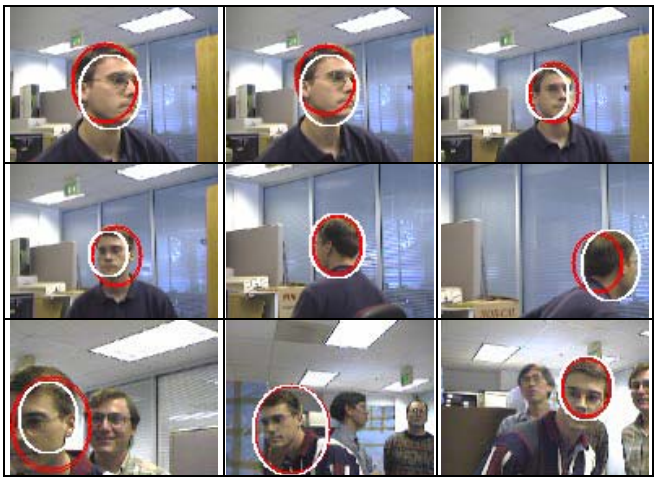

Figure 5: *Elliptical head tracking experiments*

3.2.1 Experiments

Two sets of experiments were performed on our tracking algorithm, comparing the performance of our head tracker with Birchfield's. Images from three different head motion tracking segments are shown in Figure 5. The dark (red) ellipse is the result of our implementation of Birchfield's tracker, while the white ellipse is the result of our tracker. The results indicate that our method is more stable in large motion and the head shape is more accurate. The tracker works in real-time and is relatively robust to occlusion, rotation, and scale variations.

3.3 Low-resolution face pose evaluation

The goal of this work is to coarsely track the head orientation of multiple people in a scene in real-time, with input from any number of cameras. Head location and

tracking are performed via color-based skin tracking and feature location based on non-skin areas within the face. A rough estimate of the subject's viewing direction is computed using statistics on the skin pixel positions within the head region.

Figure 6 shows the Coarse Face Direction (CFD) system locating and tracking two faces, while Figure 7 shows statistics computed over the face regions. We are currently working on fast mechanisms to estimate face orientation based on these simple statistics. Preliminary results are promising; as the graphs in Figure 6 show, there is a clear relationship between the four statistical measures and head orientation (pan angle). The x-coordinate corresponds to frame number of a sequence taken of a person panning from left to right.

This will be used to evaluate the general gaze directions of audiences, both large and small.

4 Discussion

Perceptual interfaces, modeled after human-to-human interaction, may enable people to interact with technology in ways that are natural, efficient, and easy to learn. A semantic understanding of application and user semantics, which is critical to achieving PUI, can enable a single specification of the interface to migrate among a diverse set of users, applications, and environments, transforming the way that interfaces are designed and built.

A perceptual interface does not necessarily imply an anthropomorphic interface, although the jury is still out as to the utility of interfaces that take on human-like characteristics. It is likely that, as computers are seen less as tools for specific tasks and more as part of our communication and information infrastructure, combining perceptual interfaces with anthropomorphic characteristics will become commonplace.

The research agenda for perceptual interfaces must include both (1) development of individual components, such as speech recognition and synthesis, visual recognition and tracking, and user modeling, along with (2) integration of these components. A deeper semantic understanding and representation of human-computer interaction will have to be developed, along with methods to map from the semantic representation to particular devices and environments. In short, there is much work to be done. But the possible benefits are immense.

Relevant vision-based interaction technologies include human tracking, analysis, and recognition with respect to heads, faces, hands, and whole bodies. We presented some of our recent work in different approaches to face tracking: precise feature-based tracking, general color-and-gradient based tracking, and fast coarse face orientation. These are ongoing research projects with several intended applications.

References

[1] A. Bobick, "Movement, activity, and action: the role of knowledge in the perception of motion," Royal Society Workshop on Knowledge-based Vision in Man and Machine, London, England, February 1997.

[2] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," IEEE Conference on Computer Vision and Pattern Recognition, pp. 232-237, 1998.

[3] G. R. Bradski, "Real-time face and object tracking as a component of a perceptual user interface," IEEE Workshop on Applications of Computer Vision, pp. 214-219, 1998.

[4] D. Comaniciu and P. Meer, "Real-time tracking of non-rigid objects using mean shift," IEEE Conference on Computer Vision and Pattern Recognition, pp. 142-149, 2000.

[5] R. Feris, V. Krueger, R. Cesar, "Efficient real-time face tracking in wavelet subspace," ICCV'2001 Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems, Vancouver, Canada, July 2001.

[6] G. Hager and P. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," IEEE Trans PAMI, 20(10):1025-1039, 1998.

[7] V. Krueger and G. Sommer, "Gabor wavelet networks for object representation," Technical Report CS-TR-4245, University of Maryland, May 2001.

[8] D. A. Norman, The Invisible Computer, MIT Press, Cambridge, MA, 1998.

[9] B. Reeves and C. Nass, The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places, Cambridge University Press, September 1996.

[10] M. Turk and G. Robertson, "Perceptual user interfaces," Communications of the ACM, March 2000.

[11] A. van Dam, "Post-WIMP user interfaces," Communications of the ACM, Vol. 40, No. 2, pp. 63-67, Feb. 1997.

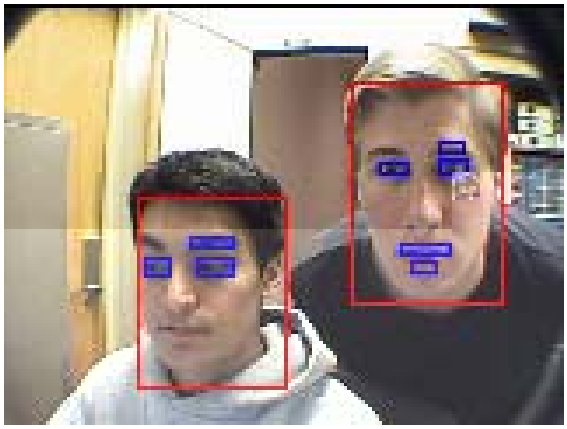

Figure 6: *Two faces located via skin color tracking; the estimated locations of facial features are displayed.*

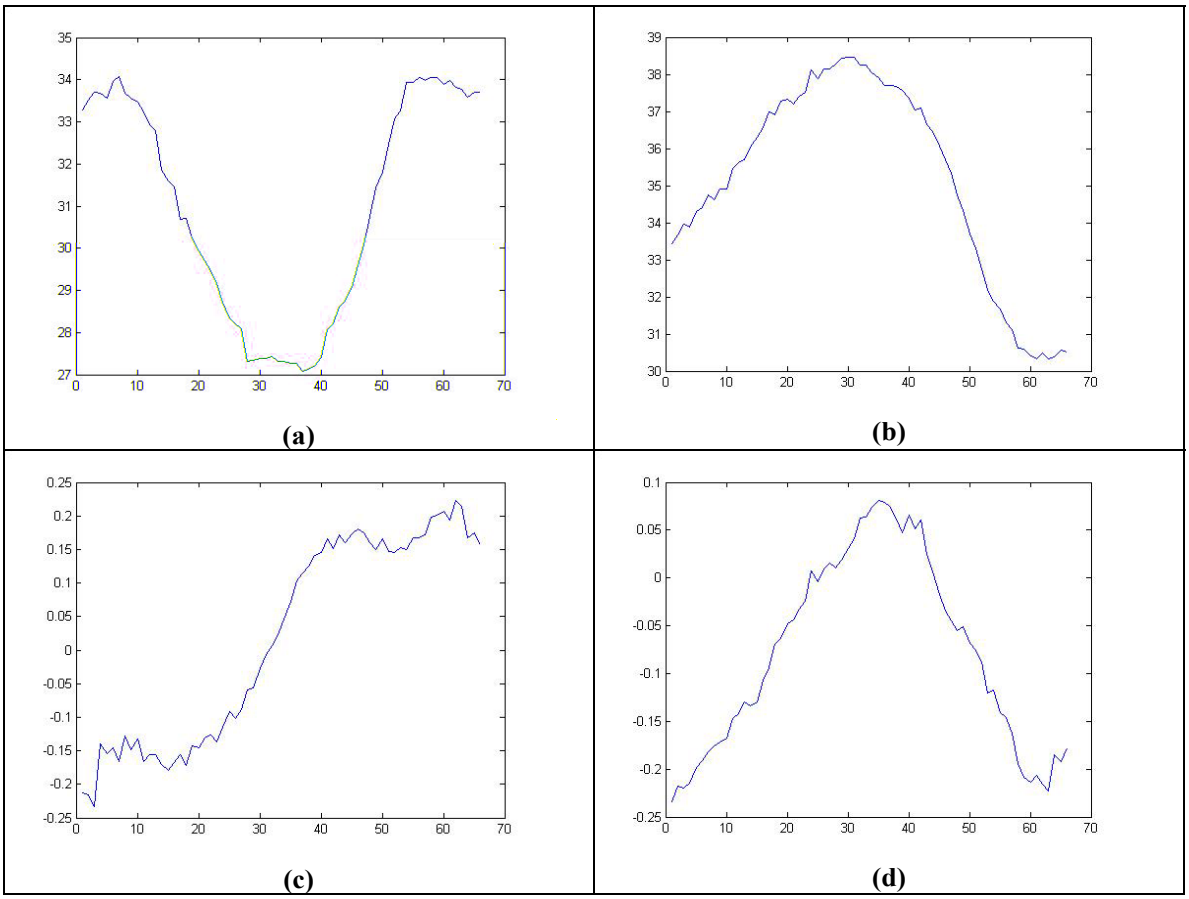

Figure 7: *Statistics on skin pixels within the face region, as an approximate function of head rotation (pan): (a) Standard deviation of skin pixels in x. (b) Standard deviation of skin pixels in y. (c) Skewness of skin pixels in x. (d) Skewness of skin pixels in y.*