Non-Intrusive Gaze Tracking for Human-Computer Interaction

Andrew Gee and Roberto Cipolla
University of Cambridge
Department of Engineering
Cambridge CB2 1PZ
England

Abstract

Current approaches to gaze tracking tend to be highly intrusive: the subject must either remain perfectly still, or wear cumbersome headgear to maintain a constant separation between the sensor and the eye. This paper describes a more flexible vision-based approach, which can estimate the direction of gaze from a single, monocular view of a face. The technique makes minimal assumptions about the structure of the face, requires very few image measurements, and produces a useful estimate of the facial orientation. The computational requirements are insignificant, so with automatic tracking of a few facial features it is possible to produce real-time gaze estimates. A robust, multiple hypothesis tracker is described, which utilises no expensive correlation operations and runs at video rate on standard hardware.

1. Introduction

Humans have little difficulty sensing where another person is looking, often using this information to redeploy their own visual attention. Even pre-renaissance artists were aware of this, using the gaze of characters *within* a painting to draw the viewer's eye to some significant part of the canvas. Yet this ability to determine a person's gaze, even from a single, monocular, uncalibrated view (as in paintings), is quite remarkable, especially considering the significant inter-subject variations in the facial features that provide the gaze cues.

Current approaches to gaze tracking use active sensing to measure the orientation of the subject's eyes. The eye is illuminated with infrared light, and the gaze direction inferred from the relative position of the *bright-eye* (the reflection off the retina) and the *glint* from the cornea [8]. The system's calibration is sensitive to movements of the subject's head, so the subject must either remain perfectly still, or wear cumbersome headgear to maintain a constant separation between the sensor and the eye. A passive, vision-based approach would ideally tolerate large head movements, and be able to follow a person's gaze at some distance, using little or no calibration, in much the same manner as humans do naturally. There are clearly applications for such a system in human-computer interfaces, especially in the field of virtual reality.

There are two major components to gaze direction: the orientation of the head, and the orientation of the eyes within their sockets. Here we concentrate on the first component, presenting a simple, efficient method to extract the facial normal from a single, monocular view of a face. This is very different to the approach taken in conventional gaze tracking systems, which work by measuring only the rotation of the eyes. Such systems produce very accurate gaze estimates (errors are typically less than 1 degree [2]) for a subject looking within the narrow field of view allowed by eye movements alone, but cannot cope with the much larger gaze shifts caused by head movements (unless the subject wears cumbersome headgear). By looking *solely* at head movements, we are trading accuracy for flexibility.

Gaze aside, the head orientation could also be used in virtual holography applications [1], or to guide a remote, synthesised "clone" face for low bandwidth video conferencing [9, 11]. In addition, a head tracker could provide a very useful computer interface for physically handicapped people, some of whom can only communicate using head gestures.

Earlier work along these lines can be found in [1], where 15 or so corner features are tracked on a moving face to estimate the facial orientation, assuming rigidity of the face. Compared with our approach, this technique is much slower, since expensive correlation operations are required to track the corner features. In addition, more a-priori modelling of the face is required (we require only one ratio of two facial dimensions), or else the model must be estimated on-line. However, the use of redundant features is effective in rejecting noise, and the technique in [1] is certainly

In *Proc. Mechatronics and Machine Vision in Practice, pages 112–117, Toowoomba, Australia 1994.*

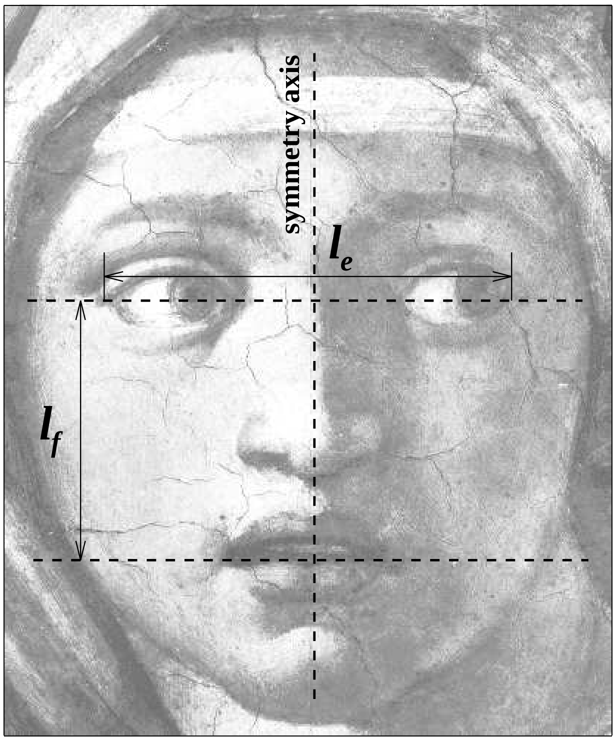

Figure 1. The facial model.

more accurate than the one presented here.

2. The facial model

There are many cues to facial orientation: up-down rotation is easily inferred from visibility of the underside of the chin or the crown of the head; left-right rotation can be estimated using ear visibility, or the position of the eyes relative to the occluding contour of the face. Yet all these cues, though undeniably strong, make use of features with a high variation across different subjects. In addition, these cues are rather vaguely defined, and will be difficult to detect in an image. What is required is a set of precise, geometric cues, easily extracted from an image and providing reliable estimates of facial pose across a wide variety of subjects. For this reason, the method presented here utilises measurements taken from only the eyes and mouth. It is, of course, necessary to assume some sort of underlying model for the 3-D geometry of faces, though the model should be as simple and generic as possible.

The facial model comprises a single ratio R_e of two world lengths: $R_e \equiv L_e/L_f$. The image quantities corresponding to L_e and L_f (denoted using lower case letters) are shown in Figure 1. The far corners of the eyes and mouth define a plane, which we term the *facial plane*. L_f is measured on the symmetry axis of this plane, while L_e is measured along the direction of symmetry correspondence. L_e and L_f span relatively *stable* features: we would not expect R_e to change very much for different facial expressions. In addition, R_e is fairly constant over a range of "normal" faces. Model calibration should only be necessary when dealing with an unusual face, or when high precision is important: in such cases R_e can be measured in a fronto-parallel view of the face, as in Figure 1.

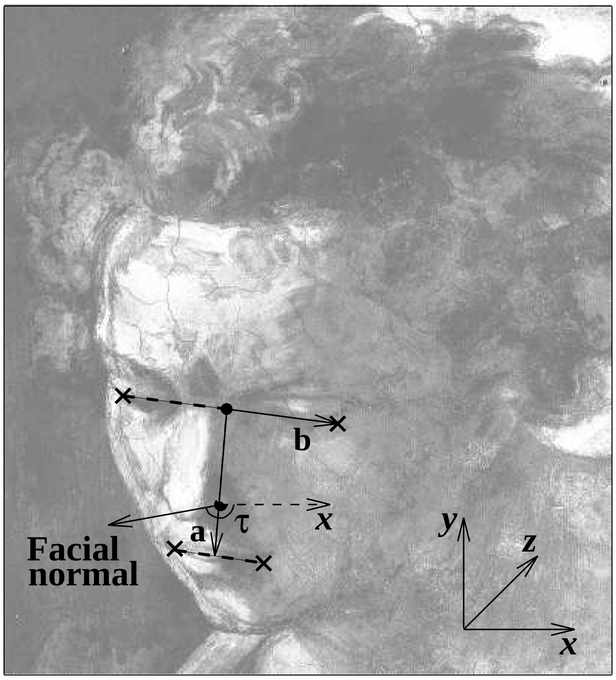

Figure 2. Estimating the facial normal.

3. Estimating the facial orientation

Throughout this work a *weak perspective* [14] imaging process is assumed, valid when depth changes on the face are small compared with the distance between the face and the camera. This is generally a good approximation, except when viewing the face from close range with a short focal length lens, which results in significant perspective distortion in the image. The renaissance artists were well aware of this, and consciously avoided short viewing distances, since the resulting images were displeasing to the eye [4]. Indeed, Leonardo's own rule of thumb was to depict figures as viewed from at least ten times the depth change across the figure [4]: the same ratio is often used in more recent vision research to justify a weak perspective imaging assumption [15].

Consider a single image of a face in general pose, as in Figure 2. Assume a camera-centered coordinate system, with x and y axes aligned along the horizontal and vertical directions in the image, and z-axis along the normal to the image plane. Assume also that the far corners of the eyes and mouth have been located in the image: these points are marked with crosses. Weak perspective preserves length ratios along parallel lines, and particularly midpoints [10]. So it is possible to locate (in the image) the symmetry axis of the facial plane, by finding the midpoints of the eye and mouth points, and joining them up. This provides the image vectors **a** and **b**, along the symmetry axis and eye-line. Assuming an affine imaging process (a generalisation of weak perspective with an uncalibrated camera [13]), **a** and **b** can be mapped onto world coordinates using a 2×2 affine transformation matrix **U** [12]:

$$\begin{aligned} \mathbf{U}[\,\mathbf{a}\;\;\mathbf{b}\,] &= \begin{bmatrix} 0 & \frac{1}{2}R_e \\ -1 & 0 \end{bmatrix} \\ \Leftrightarrow \mathbf{U} &= \begin{bmatrix} 0 & \frac{1}{2}R_e \\ -1 & 0 \end{bmatrix} [\,\mathbf{a}\;\;\mathbf{b}\,]^{-1} \end{aligned}$$

Given **U**, we show in the appendix how to recover the slant σ and tilt τ of the facial plane, up to a plus/minus 180^o ambiguity in the tilt. The slant is the angle between the optical axis and the facial normal in 3-D space. The tilt is the orientation *in the image* of the facial normal, quantified using the angle τ between the imaged normal and the x-axis. In camera-centered coordinates, the facial normal $\hat{\mathbf{n}}$ is given by

$$\hat{\mathbf{n}} = [\sin\sigma\cos\tau, \sin\sigma\sin\tau, -\cos\sigma] \tag{1}$$

The tilt ambiguity cannot be resolved using the positions of the eyes and mouth alone — see Figure 3. When the face is being tracked in a continuous image sequence, the ambiguity can usually be resolved by imposing frame-to-frame continuity of the facial normal $\hat{\mathbf{n}}$. This strategy breaks down when the slant of the face is small (ie. when the view is nearly fronto-parallel), since then the two estimates of $\hat{\mathbf{n}}$, corresponding to the two possible tilts, are very similar. In such cases we predict the two possible positions of the eyebrows in the image (which is straightforward once $\hat{\mathbf{n}}$ is known) , and look for photometric evidence to support each hypothesis — the eyebrows are easily distinguished as a dark-light-dark pattern going from the left eyebrow to the bridge of the nose and on to the right eyebrow. The tilt hypothesis which agrees with the detected eyebrow position is selected.

Once the facial normal is known, it is possible to calculate the directions of the eye-line and symmetry axis relative to the camera-centered coordinate system: the necessary geometry is presented in [7]. In [7] we also investigate the accuracy of the technique, and its sensitivity to noise in the image and errors in the facial model. Under typical noisy imaging conditions, we find that the facial normal $\hat{\mathbf{n}}$ is reliably estimated to within a few degrees of its true value, except for near frontal views of the face, when the error can be as large as 15^o. This is to be expected, since the image measurement $l_e : l_f$, implicitly used to obtain $\hat{\mathbf{n}}$, is fairly insensitive to changes in pose for near-frontal views of the face. In [7] we present an alternative method for estimating $\hat{\mathbf{n}}$, which utilises measurements taken from the nose, as well as from the eyes and mouth. This technique produces accurate gaze estimates for near-frontal views of the face. However, since the nose is difficult to track in an image sequence without expensive correlation operations, we shall concentrate here on the simple eye-mouth technique.

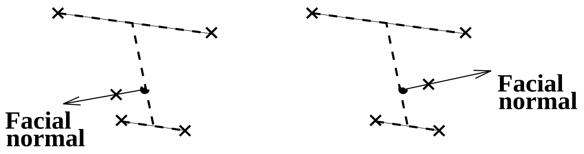

Figure 3. Eye and mouth points cannot resolve a tilt ambiguity.

For near-profile views of the face it is likely that some of the eye and mouth features are occluded in the image. However, these features *are* visible for a wide range of poses, including cases where the face is heavily slanted (see Figure 7). When the points are obscured, their positions could be estimated using the locations of other facial features.

4. Simple feature tracking

To deliver continuous gaze estimates we need to track the eye and mouth corners in an image sequence. In developing a tracker, we have sought to keep the computational requirements to a minimum, so the tracker can run as fast as possible. Eyes are tracked by simply looking for the darkest pixel near the previous eye position: this locks on to the pupil, which is not as stable a feature as the eye corner (since the eyeball is free to rotate relative to the rest of the face), but suffices to produce an approximate gaze estimate. The division between the lips appears as a dark line in the image, whose end points locate the mouth corners. The line and its end points are easily tracked from frame to frame. The details of the tracker are as follows.

Initialization

In this simple implementation, the eye and mouth features are located by hand in the first frame. However,

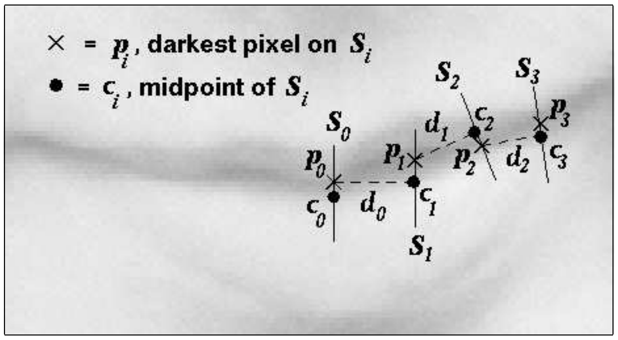

Figure 4. Tracing the mouth line.

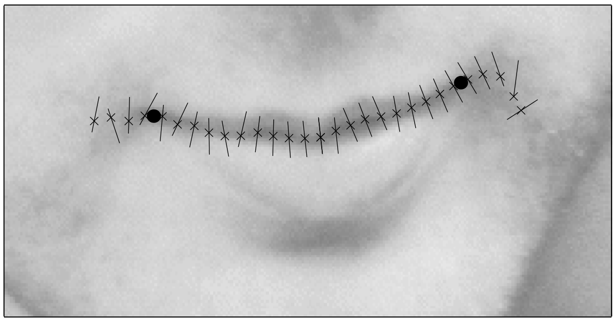

Figure 5. Locating the mouth corners.

techniques do exist for the automatic location of facial features, using either parameterized models (eg. [6]) or grey-level templates (eg. [3]).

Eye tracker

Prediction: Predict the image location of the eye using linear extrapolation over the previous two frames.

Detection: Search a $w \times w$ pixel window around the expected eye position for the darkest pixel.

Filter: Smooth the eye track using a first order low-pass filter (ie. update the eye position to half way between the darkest pixel and the eye position in the previous frame).

The tracker is not sensitive to the parameter w, so long as w is large enough to cope with typical accelerations of the eyes in the image. We use $w = 10$ pixels.

Mouth tracker

Prediction: Predict the image location of the mouth corners using linear extrapolation over the previous two frames. Obtain an estimate $\mathbf{c}_0$ of the centre of the mouth, midway between the two corners. Estimate the local orientation $\mathbf{d}_0$ of the mouth from the relative position of the corners.

Trace the mouth line (Figure 4): Read a line $\mathbf{S}_0$ of n pixels around $\mathbf{c}_0$, at right angles to $\mathbf{d}_0$. Locate the darkest pixel, $\mathbf{P}_0$, along this line. Move to $\mathbf{c}_1$, a fixed distance r along $\mathbf{d}_0$. Read a new line $\mathbf{S}_1$ of n pixels around $\mathbf{c}_1$, at right angles to $\mathbf{d}_0$. Locate the darkest pixel $\mathbf{P}_1$, and move to $\mathbf{c}_2$, a distance r along a new estimate of the local mouth orientation $\mathbf{d}_1$. From now on we can use $\mathbf{P}_i - \mathbf{P}_{i-1}$ to obtain $\mathbf{d}_i$. Continue reading lines $\mathbf{S}_i$ orthogonal to $\mathbf{d}_{i-1}$ until m lines of pixels have been read. Repeat travelling the other way along the mouth.

Locate the corners: Calculate the contrast (the difference between the lightest and darkest pixels) along each line $\mathbf{S}_i$. While $\mathbf{S}_i$ straddles the mouth the contrast will be fairly large, but falls off abruptly when the scan hits areas of smooth skin by the side of the mouth. Look within a small $u \times u$ pixel window around each predicted mouth corner, and locate the corner at the dark pixel $\mathbf{P}_i$ where the contrast drops most sharply.

Filter: Smooth the mouth corner tracks using a first order low-pass filter.

The technique is not sensitive to the parameters n, m, u and r, so long as they remain within suitable, broad bounds. n must be large enough to span the lips, m must ensure that the scan shoots off the ends of the mouth, and r must be small enough to locate the mouth corners with sufficient accuracy. The $u \times u$ validation window simply rejects spurious measurements from beyond the corners of the mouth. We use $n = 20$ pixels, $u = 15$ pixels, $r = 5$ pixels, and $m = 15$.

A typical mouth scan is shown in Figure 5, with the located corners shown as filled blobs. The corners are reliably tracked even when the mouth is opened, so long as a dark region remains between the lips.

5. Multiple hypothesis tracking

The simple tracker described in the previous section works well but eventually, and inevitably, a spurious measurement causes the tracker to lose lock on the facial features. The tracker can be made more robust by exploiting constraints inherent in the facial geometry and imaging process. For instance, we know from our weak perspective assumption that the lines joining the eye and mouth corners should be parallel. We also know that the angular acceleration of the face should not be too large. Finally, so long as the subject does not overly contort his or her mouth, we know that any fractional change in the imaged inter-eye distance should be mirrored as an identical change in the imaged mouth length. All these constraints can be exploited to improve the reliability of the tracker.

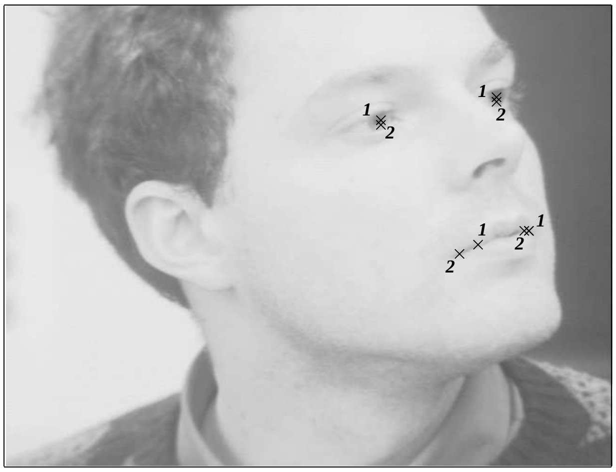

Figure 6. Ranked feature hypotheses.

The key is to allow the individual feature trackers to return ranked hypotheses for each feature position. So the eye trackers return not the single darkest pixel, but the p darkest pixels. Likewise, the mouth trackers return the p sharpest drops in contrast within the $u \times u$ validation windows. This produces p^4 combined hypotheses, which are sorted into order of decreasing photometric evidence. The list is descended, and the first combined hypothesis which satisfies the constraints to within some tolerance is accepted. If no combined hypothesis satisfies all the constraints, then we revert to the hypothesis with the greatest photometric evidence at the top of the list. We have found significantly improved performance for p as small as 2 or 3. The process is illustrated in Figure 6, where an erroneous measurement from the left mouth tracker can be rejected (in favour of the second choice hypothesis) on the grounds that it causes the mouth length to contract more than the inter-eye distance.

The multiple hypothesis tracker is fast and robust. The combined tracking and gaze estimation process runs at 100 Hz (four times frame rate) on a Sun SparcStation 10 (the head tracker in [1] runs at 10 Hz). Several frames taken from a typical image sequence, along with a drawing pin representation of the estimated facial orientation, are shown in Figure 7.

Even better performance should be possible by maintaining multiple tracks *over time*. By deferring hypothesis selection for several frames, it soon becomes clear which hypothesis to accept. This entails maintaining a tree of active hypotheses, and separate feature trackers for each, which greatly increases the computational burden. Such trackers have been developed by the radar community over the years, and are reviewed for computer vision purposes in [5].

Figure 7. Real-time gaze tracking.

6. Conclusions

The facial normal can be extracted from a single, monocular view of a face, making minimal assumptions about the underlying facial structure. A gaze estimate based on the facial orientation alone is useful, and, compared with conventional eye-tracking systems, very large shifts of gaze can be accommodated. By tracking the eyes and mouth in the image, it is possible to produce gaze estimates at video rate on standard hardware, with many applications in the field of human-computer interaction. The reliability of the tracker is improved by adopting a multiple hypothesis approach: in this way constraints inherent in the facial geometry and imaging process can be fruitfully exploited.

Acknowledgements

The authors would like to thank Jonathan Lawn and Mark Wright for many helpful discussions relating to this work, and Nick Hollinghurst for his invaluable assistance with the coding of the real-time demonstration. Andrew Gee gratefully acknowledges the financial support of Queens' College, Cambridge, where he is a Research Fellow.

A Calculating the slant and tilt

In this appendix we show how to obtain the slant σ and the tilt τ of the facial plane from the affine transformation matrix $\mathbf{U}$. See [12] for a full derivation of these results.

$$\text{Let } \mathbf{U}^T\mathbf{U} = \begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix} \text{ and } \mu = \frac{(\alpha+\gamma)^2}{4(\alpha\gamma-\beta^2)}$$

$$\text{Then } \cos\sigma = \frac{1}{\sqrt{\mu}+\sqrt{\mu-1}} \text{ and } \tan 2\tau = \frac{2\beta}{\alpha-\gamma}$$

To find the slant τ, the inverse tangent should be chosen so that $\mathrm{sign}(\beta) = \mathrm{sign}(\sin 2\tau)$. This fixes τ up to a plus/minus 180° ambiguity.

References

[1] A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland. Visually controlled graphics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):602–605, 1993.

[2] S. Baluja and D. Pomerleau. Non-intrusive gaze tracking using artificial neural networks. Technical Report CMU-CS-94-102, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, January 1994.

[3] R. Brunelli and T. Poggio. Face recognition: features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.

[4] A. Costall. Beyond linear perspective: A cubist manifesto for visual science. *Image and Vision Computing*, 11(6):334–341, 1993.

[5] I. J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10(1):53–66, 1993.

[6] I. Craw and P. Cameron. Finding face features. In *Proceedings of the 2nd European Conference on Computer Vision*, pages 92–96, 1992.

[7] A. Gee and R. Cipolla. Determining the gaze of faces in images. Technical Report CUED/F-INFENG/TR 174, Cambridge University Department of Engineering, March 1994.

[8] T. E. Hutchinson, K. P. White, W. N. Martin, K. C. Reichert, and L. Frey. Human-computer interaction using eye-gaze input. *IEEE Transactions on System, Man and Cybernetics*, 19(6):1527–1533, November/December 1989.

[9] R. Koch. Dynamic 3-D scene analysis through synthesis feedback control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):556–568, 1993.

[10] J.J. Koenderink and A.J. van Doorn. Affine structure from motion. *Journal of the Optical Society of America A — Optics and Image Science*, 8(2):377–385, 1991.

[11] H. Li, P. Roivainen, and R. Forchheimer. 3-D motion estimation in model-based facial image coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):545–555, 1993.

[12] D. P. Mukherjee, A. Zisserman, and M. Brady. Shape from symmetry — detecting and exploiting symmetry in affine images. Technical Report OUEL 1988/93, Oxford University Department of Engineering Science, June 1993.

[13] J. L. Mundy and A. Zisserman, editors. *Geometric Invariance in Computer Vision*. MIT Press, Cambridge MA, 1992.

[14] L.G. Roberts. Machine perception of three-dimensional solids. In J.T. Tippet, editor, *Optical and Electro-Optical Information Processing*. MIT Press, 1965.

[15] D. W. Thompson and J. L. Mundy. Three-dimensional model matching from an unconstrained viewpoint. In *Proceedings of IEEE Conference on Robotics and Automation*, pages 208–220, April 1987.