Computer Operation via Face Orientation

Philippe Ballard & George C. Stockman
Pattern Recognition and Image Processing Laboratory
Computer Science Department, Michigan State University
East Lansing, MI, 48824

Abstract

Computing via face orientation allows a means of communication for people with severe physical disability. We have developed a near-real-time computer system to track the eyes and the nose of a subject and compute the direction of the face. As presented by Ohmura et al *[OTK88], the direction can be computed from three natural feature points extracted from the face. The computational approach taken in this system is motivated by the practical requirements of near-real-time performances and accuracy. The approach is based on the reflection characteristic of the cornea. Any bright light source forms a twinkle in the eye that is fixed as long as the light source is fixed. Our face direction detection system is treated as a two-dimensional tracking problem. First, the eye twinkles are extracted from the image and then used to locate the nose. A final experiment shows that the face direction obtained from the three feature points is precise enough to control the mouse cursor to make selection from a menu with large light buttons.*

1 Introduction

The face is the primary focus of attention in human communication. During a conversation one may infer from face expressions such feelings as happiness, suspicion or attentiveness. We can also tell a lot from where the person is looking. Our ultimate goal is to be able to communicate with computers via facial language. In this paper we show how to compute face direction using natural features and then show how to make light button selections using the face position.

Developing a computational model of face communication via natural features is a difficult task. It requires processing of complex images under different lighting conditions, having a moving background and sometimes even partial occlusions of the face. While we have demonstrated the feasibility of our approach, a robust system would require much more engineering.

Light reflected off the cornea of the eye forms a bright twinkle whose location is determined by the location of the light source. Accordingly, we extract two feature points, one for each eye from the face. A third feature point, the nose, is located using the positions of the two eye twinkles. The direction of the face is the normal of the plane defined by the three points. The values of the yaw angle θ and the pitch angle ϕ give the magnitude of movement of the computer cursor/mouse. The system was tested on eight subjects. Calibration was necessary for the different skin complexions and facial geometry of the subjects.

1.1 Background and related work

Much of the work on faces in computer vision is not directly related to face direction detection, but some of the techniques used in face recognition, such as feature extraction, location of templates and matching algorithms can be applied to face communication systems. Near-real-time constraints rule out some approaches described in the computer vision literature [YCH89].

Ohmura, Tomono, and Kobayashi presented [OTK88] a method of non-contact detection of human face direction. They computed face direction from three points extracted from the face. In order to track face movement in real time, they pasted special blue spots on the face at the corner of each eye, and the center of the upper lip and developed special hardware to extract the centers of these spots from an image. Their method was able to compute 10 updates per second of face direction and therefore could operate in real time. In order to operate without cosmetic feature points, Ballard and Stockman [BS90] first tried to detect the eyes and the mouth by fitting general elliptical templates to the image. Their conclusions were that the template matching was computationally too expensive for real time and could not reliably locate a point for the center of the mouth.

In related work, Schmandt, Ackerman and Hindus [SAH90] present a communication system via speech. They showed how window navigation tasks usually performed with a mouse can be controlled by voice. Pentland and Mase [PM89] describe a lip reading system that may be used to augment any speech recognition system. Several other papers describe useful algorithms to locate the face within a noisy background. Govindaraju, Srihari and Sher [GSS90] show how to develop procedures to locate human faces in newspaper photographs. Their approach is based on cost minimization of feature graphs. Turk and Pentland [TP91] find the face by analyzing frame differencing under the hypothesis that people are constantly moving. In an early paper, Baron [Bar81] shows how to locate the eyes in a face. Yuille, Cohen and Hallinan [YCH89] use deformable templates and an energy minimization function to find the eyes.

2 Computation of face position

Computing the direction of the face requires computation of the 3-D coordinates of the feature points from the 2-D projection in the image plane. Ohmura *et al* [OTK88] show that they can obtain the direction from three feature points, and the distances among them. Fishler and Bolles [FB81] gave a closed form solution and showed that there can be up to 4 solutions in general, but 2 solutions are typical. By tracking over time, one solution can usually be selected. Our experiments showed that the algorithm always converged to the right position when the initial values were close enough to the real values.

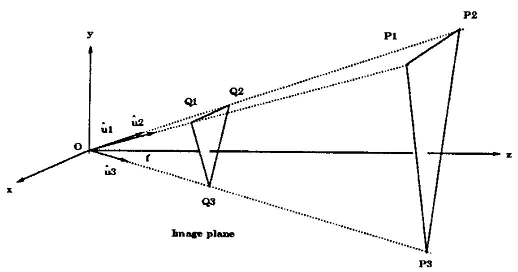

Figure 1: **Coordinate system and perspective transformation.**

a_1, a_2, a_3 are the distances from the origin (O) or projection point to the three 3-D points. They are computed from the following equations:

$$\begin{cases} a_1^2 + a_2^2 - 2(u_1, u_2)a_1a_2 - l_{12}^2 = 0 \\ a_2^2 + a_3^2 - 2(u_2, u_3)a_2a_3 - l_{23}^2 = 0 \\ a_3^2 + a_1^2 - 2(u_3, u_1)a_3a_1 - l_{31}^2 = 0 \end{cases} \quad (2.1)$$

where $\hat{u}_i$ are unit vectors along a_i and l_{ij} are the real distances among the 3-D points. This system of quadratic equations is solved using the method of successive approximations. Once a is found, the 3-D points $\hat{P}_i$ are determined and the face direction can be computed as the vector product of the difference of the vectors:

$$\left(\hat{P}_2 - \hat{P}_1\right) \times \left(\hat{P}_3 - \hat{P}_1\right) \quad (2.2)$$

3 Extracting natural feature points

Three feature points must be extracted from the face. These points should be fairly rigid on the face so that head movements and facial gesturing will induce only minimal changes to these features. Our first work [BS90] attempted to find the eyes and the mouth by chamfering the image [BTBW77] and then fitting optimal ellipses to the ridges. However, the computation time was too large and the mouth was too often badly fit.

Our second and faster approach involves the following operations:

1. Acquire a face image.
2. Locate the twinkles. Also find the center of the iris as a confirming point in each eye.
3. Compute the position of the nose from the position of the twinkles, and locate the feature point of interest.
4. Compute face direction from the three feature points.
5. Move the computer cursor accordingly.

3.1 The twinkle approach

The human eye is very nearly spherical. The twinkle, a bright specular reflection from the cornea, may be assumed to be fixed. We studied the eye motion and found that the twinkle moved on the image with a maximum magnitude of five pixels resulting in a direction error of less than one degree. The twinkles are the brightest points on the gray-level image and also form the highest contrast neighborhood. Thus, they can be extracted by thresholding either the gray scale histogram or the edge magnitude histogram.

A divide and conquer algorithm is used to quickly locate the position of the twinkles. The search window is recursively divided by half, first according to the x axis and then according to the y axis. An histogram analysis is performed on each of the window. If the window is empty, it is rejected. When the window is small enough (about 30 pixels by 30 pixels) an exhaustive search of the window is started and a connected component algorithm is run to find the 8-connected region of the twinkle.

Sometimes twinkle candidates are caused by tears in the subject's eyes and shape features of pairs of these small bright regions have to be used to detect the appropriate regions. Twinkles are fairly small, usually less than 20 pixels and have almost a circular shape. We defined this shape by three features: the area, or twinkle size that should be less than 20 pixels, the elongation which should be close to one and the compactness which should be as close as 4π as possible. All the features are scaled to fall within the interval [0-100]. A similarity index s_{ij} is computed for each pair (T_L^i, T_R^j), where T_L^i is a twinkle from the set generated by the left eye and T_R^j a twinkle from the set generated by the right eye. Let $\overline{A}$, $\overline{E}$, $\overline{C}$ respectively denote the normalized area, normalized elongation and normalized compactness.

$$\begin{aligned} s_{ij} &= \frac{0.2(A_i + A_j)}{2} + 0.2\sqrt{\overline{A_i}^2 - \overline{A_j}^2} \\ &+ 0.3\sqrt{\overline{E_i}^2 - \overline{E_j}^2} + 0.3\sqrt{\overline{C_i}^2 - \overline{C_j}^2} \quad (3.3) \end{aligned}$$

The weights associated with each feature are system dependent. We looked for the twinkle pair that maximized the similarity index s_{ij}. The centroids m_x and m_y are the coordinates of our feature points.

3.2 Extracting the third feature point

We chose to detect the end of the nose as the third feature point because it is a fairly rigid point and should be present in all images where the eyes are detected. Also, our previous work in locating the mouth was relatively unsuccessful.

In static images, a nose twinkle is observed which is similar to, but weaker than the eye twinkles. After studying sample imagery, a procedure was developed to detect the nose twinkle. An important part of that procedure was a search constrained to lie along the bisector of the line between the detected eye twinkles. The procedure, which used shape features analysis, as described above, and knowledge of the human face proportions, was successful on the static test images but then failed on dynamic images. In dynamic images, the size and shape of the nose twinkle varied too much.

Extraction of the contour of the nose was found to work well in dynamic images, although there were three separate cases that had to be handled. The process used to search a constrained area for the contour feature was the same as described above. The three cases to be considered are shown in Figure 2. In Figure 2 a) the subject is facing the camera. We look for the nasals, that is two connected components with same size (about 250 pixels), same elongation (usually very low) and same compactness (usually high). The point in the middle of the two center points becomes our third feature point. In Figure 2 b) the subject is looking right. We look for the edge of the nose and the left nasal. The connected region describing the edge of the nose should have an elongation close to zero and a compactness as high as possible. If m_x^N and m_y^N are the coordinates of the centroid of the edge of the nose, $F3 = (m_x^N, max(m_y^L, m_y^R))$. The case for the subject looking left is similar (Figure 2 c)).

a) b) c)

Figure 2: **Nose extraction in real time.**

4 Mouse control

To test the capability of our system, we used the detected direction of the face to control the computer cursor and make light button selections. A grid was drawn on the video monitor and one box was selected at random and colored dark as a goal for the user. Starting from the top left of the grid, the user was to try to reach the designated box in a minimum time. Only a simple control algorithm is currently implemented. The values of the yaw angle θ and the pitch angle ϕ are used to respectively move the cursor horizontally and vertically. Movement is one box for every 10 degrees of θ and every 5 degrees of ϕ. This difference of increment is due to the difference of precision for the two angles. Moreover, the span of the pitch angle is smaller than the span of the yaw angle.

5 Results

We tested our system with eight different subjects under controlled lighting, scale and orientation. At least five different orientations were considered for each subject. Figure 3 illustrates the experimental system. The camera was arranged so that the triangle defined by the eyes and the nose was in the center of the image when the subject was looking straight at the camera. The amount of light falling on the face was carefully controlled. When there was not enough light, we got some shadows on the face and lost important features of the image. Too much light caused difficulty in detecting the nose. All the experiments were done with a white light. The processing was done on a 68000-based vision system (Innovision IDAS 150) at frame rate (30 frames/sec).

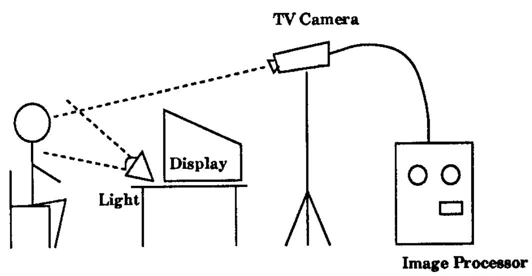

Figure 3: **Experimental system.**

a) b) c)

Figure 4: **Features extraction.**

5.1 Feature extraction and direction computation

Example image processing is shown in Figure 4 and example results of the face direction computation are given in Table 1. Two remarks must be made: first, an error of one or two pixels in the location of the features is insignificant. Secondly, the human face is not flat. Therefore, the pitch angle of the triangle, when the subject is facing the camera, is not null. To find this

Orient.	Left eye				Right eye			
	Rx	*Ry*	*Fx*	*Fy*	*Rx*	*Ry*	*Fx*	*Fy*
Straight	179	150	179	150	359	155	359	154
Right	156	150	156	150	326	159	327	159
left	266	174	267	173	436	175	436	175
Up	189	117	189	117	368	121	368	121
down	188	201	188	201	368	206	368	205

Orient.	Nose				Dir. (deg)	
	Rx	*Ry*	*Fx*	*Fy*	θ	ϕ
Straight	269	282	269	279	4.6	0.0
Right	206	279	205	277	11.3	1.4
left	383	308	381	306	20.3	3.1
Up	276	221	276	221	3.7	9.4
down	280	348	280	346	5.7	14.3

Table 1: **Location of the eyes and the nose for subject a). The lower threshold was set to 9 and the distances left-eye-right-eye, left-eye-nose and right-eye-nose are respectively equal to 6.0 (cm), 6.0 (cm), 6.0 (cm). R stands for real and F for found.**

angle, we asked our subjects to face the camera, ran our system and used the computed angle as a reference angle. As a result, if the subject is not perfectly facing the camera at calibration, all the pitch angles have an offset.

The system achieved 97.5% correct location of both the left eye and the right eye. In 25% of the cases there was error in the location of the nose. Only 7.5% were major errors, that is location of the wrong feature. As a direct consequence of the errors in the location of the features, we get 22.5% error on the yaw angle θ and 15% error on the pitch angle ϕ. This shows that θ is more sensitive than ϕ to the location errors of the nose.

5.2 Real time and motion

The system takes about three seconds per computation of the face direction. In order to measure the accuracy of the face direction on the cursor motion, we recorded the times necessary to find the random box (Table 2). These results show that the current system has too much inertia. It was difficult to move directly to the target. In most cases, we circled around the target before being able to reach it. Feature location error induced large numbers of iterations. However, the control would not break down with feature detection error - it would only slow down.

Grid size 7 × 7		Grid size 10 × 10		Grid size 15 × 15	
# of ite.	Dist.	# of ite.	Dist.	# of ite.	Dist.
5	2	46	3	10	9
11	2	3	5	4	1
2	2	2	1	4	5
4	1	5	7	14	1
2	1	4	7	28	5
10	3	13	8	22	3

Table 2: **Number of face direction computations necessary to select the random box. The distance Dist is expressed in number of boxes.**

6 Conclusion and future work

Previous attempts at computing the face direction in real time were limited by the use of artificial features [OTK88] and by the computation time [BS90]. We developed a system that extracts natural features from the face and computes face direction in near-real-time. Our approach is based on the reflective property of the cornea. We used the two eye twinkles and the nose to compute the face direction. We achieved 97.5% correct location of the twinkles and over 75% correct location of the nose. Each face direction computation is less than three seconds, with image processing taking the bulk of this time. The system was shown to work fairly well in a controlled environment. It is important to note that many applications such as menu driven systems or wheelchair orientation do not require a very accurate value of direction. Control can be done with the relative value of the yaw and pitch angles, rather than their absolute values. For the purpose of menu selection in our experiments, an accuracy of 5 to 10 degrees was shown to be reasonable. Future work will be directed toward achieving faster processing times and extracting more information from the face, such as glance direction and facial gestures.

References

[Bar81] Robert J. Baron. Mechanisms of Human Facial Recognition. *Int. J. Man-Machine Studies*, 15:137–178, 1981.

[BS90] Philippe Ballard and George Stockman. Face Direction Detection using Feature Extraction. *M.S.U. PRIP Lab. Technical Report*, September 1990.

[BTBW77] H.G. Barrow, J.M. Tenenbaum, R.C. Bolles, and H.C. Wolf. Parametric correspondence and Chamfer matching : two new techniques for image matching. In *IJCAI-77*, volume 2, 1977.

[FB81] M.A. Fischler and R.C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of ACM*, 24:381–395, June 1981.

[GSS90] Venu Govindaraju, Sargur. Srihari, and David Sher. A Computational Model For Face Location. In *ICCV, Osaka, Japan*, November 1990.

[OTK88] K. Ohmura, A. Tomono, and Y. Kobayashi. Method of Detecting Face Direction using Image Processing for Human Interface. *SPIE Visual Communication and Image Processing*, 1001:625–632, 1988.

[PM89] Alex Pentland and Kenji Mase. LIP READING: Automatic Visual Recognition of Spoken Words. *M.I.T. Media Lab Vision Science Technical Report 117*, January 1989.

[SAH90] Chris Schmandt, Mark Acckerman, and Debby Hindus. Augmenting a Window System with Speech Input. *Computer*, pages 50–56, August 1990.

[TP91] Matthew Turk and Alex Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, March 1991.

[YCH89] A. Yuille, D. Cohen, and P. Hallinan. Feature Extraction from Faces using Deformable Templates. In *CVPR, San Diego, CA*, June 1989.