Publihed in Proceedings of Engineering Human Computer Interaction, EHCI'95, Grand Targhee, USA, August 1995, Chapman&Hall Publ.

Vision for man machine interaction

J. L. Crowley
IMAG-GRAVIR
46 Ave Félix Viallet, 38031 Grenoble, France
email: jlc@imag.fr

J. Coutaz
IMAG-CLIPS
B.P 53, 38041 Grenoble CEDEX 9, France
email: coutaz@imag.fr

Abstract

Computer vision provides a powerful tool for the interaction between man and machine. The barrier between physical objects (paper, pencils, calculators) and their electronic counterparts limits both the integration of computing into human tasks, and the population willing to adapt to the required input devices. Computer vision, coupled with video projection using low cost devices, makes it possible for a human to use any convenient object, including fingers, as digital input devices. In such an "augmented reality", information is projected onto ordinary objects and acquired by watching the way objects are manipulated. In the first part of this paper we describe experiments with techniques for watching the hands and recognizing gestures.

Vision of the face is an important aspect of human-to-human communication. We have been experimenting with the use of computer vision to "watch the face". In the second part of this paper we describe techniques for detecting, tracking and recognizing faces. When combined with real time image processing and active control of camera parameters, these techniques can greatly reduce the communications bandwidth required for video-phone and video-conference communications.

Keywords

Computer Vision, Multi-modal Man Machine Interaction, Eigen-Faces, Tracking, Principal Components Analysis

Publihed in Proceedings of Engineering Human Computer Interaction, EHCI’95, Grand Targhee, USA, August 1995, Chapman&Hall Publ.

1 VISION AND MAN-MACHINE INTERACTION

One of the effects of the continued exponential growth in available computing power has been an exponential decrease in the cost of hardware for real time computer vision. This trend has been accelerated by the recent integration of image acquisition and processing hardware for multi-media applications in personal computers. Lowered cost has meant more wide-spread experimentation in real time computer vision, creating a rapid evolution in robustness and reliability and the development of architectures for integrated vision systems (Crowley 1984).

Man-machine interaction provides a fertile applications domain for this technological evolution. The barrier between physical objects (paper, pencils, calculators) and their electronic counterparts limits both the integration of computing into human tasks, and the population willing to adapt to the required input devices. Computer vision, coupled with video projection using low cost devices, makes it possible for a human to use any convenient object, including fingers, as digital input devices. Computer vision can permit a machine to track, identify and watch the face of a user. This offers the possibility of reducing bandwidth for video-telephone applications, for following the attention of a user by tracking his fixation point, and for exploiting facial expression as an additional information channel between man and machine.

Traditional computer vision techniques have been oriented toward using contrast contours (edges) to describe polyhedral objects. This approach has proved fragile even for man-made objects in a laboratory environment, and inappropriate for watching deformable non-polyhedric objects such as hands and faces. Thus man-machine interaction requires computer vision scientists to "go back to basics" to design techniques adapted to the problem. The following sections describe experiments with techniques for watching hands and faces.

2. WATCHING HANDS: GESTURE AS AN INPUT DEVICE

Human gesture serves three functional roles (Cadoz 1994): semiotic, ergotic, and epistemic.

- The *semiotic* function of gesture is to communicate meaningful information. The structure of a semiotic gesture is conventional and commonly results from shared cultural experience. The good-bye gesture, the American sign language, the operational gestures used to guide airplanes on the ground, and even the vulgar “finger”, each illustrates the semiotic function of gesture.

- The *ergotic* function of gesture is associated with the notion of work. It corresponds to the capacity of humans to manipulate the real world, to create artefacts, or to change the state of the environment by “direct manipulation”. Shaping pottery from clay, wiping dust, etc. result from ergotic gestures.

- The *epistemic* function of gesture allows humans to learn from the environment through tactile experience. By moving your hand over an object, you appreciate its structure, you may discover the material it is made of, as well as other properties.

All three functions may be augmented using an *instrument*t: Examples include a handkerchief for the semiotic good-bye gesture, a turn-table for the ergotic shape-up gesture of pottery, or a dedicated artefact to explore the world (for example, a retro-active system such as the pantograph [Ramstein 94] to sense the invisible).

Publihed in Proceedings of Engineering Human Computer Interaction, EHCI'95, Grand Targhee, USA, August 1995, Chapman&Hall Publ.

In Human Computer Interaction, gesture has been primarily exploited for its ergotic function: typing on a keyboard, moving a mouse, and clicking buttons. The epistemic role of gesture has emerged effectively from pen computing and virtual reality: ergotic gestures applied to an electronic pen, to a data-glove or to a body-suit are transformed into meaningful expressions for the computer system. Special purpose interaction languages have been defined, typically 2-D pen gestures as in the Apple Newton, or 3-D hand gestures to navigate in virtual spaces or to control objects remotely (Baudel 1993).

With the exception of the electronic pen and the keyboard which both have their non-computerized counterparts, mice, data-gloves, and body-suits are "artificial add-on's" that wire the user down to the computer. They are not real end-user instruments (as a hammer would be), but convenient tricks for computer scientists to sense human gesture.

We claim that computer vision can transform ordinary artefacts and even body parts into effective input devices. Krueger's seminal work on the videoplace (Krueger 1991), followed recently by Wellner's concept of digital desk (Wellner 1993) show that the camera can be used as a non-intrusive sensor for human gesture. However, to be effective the processing behind the camera must be fast and robust. The techniques used by Krueger and Wellner are simple concept demonstrations. They are fast but fragile and work only within highly constrained environments.

We are exploring advanced computer vision techniques to non-intrusively observe human gesture in a fast and robust manner. In the next section, we present FingerPaint, an experiment in the use of cross-correlation as a means of tracking *natural pointing devices* for a digital desk. By "natural pointing device", we mean a bare finger or any real world artefact such as a pen or an eraser.

2.1 Projecting the workspace

In the digital desk a computer screen is projected onto a physical desk using a video-projector, such as a liquid-crystal "data-show" working with standard overhead projector. A video-camera is set up to watch the workspace such that the surface of the projected image and the surface of the imaged area coincide. This coincidence can not match "pixel to pixel" unless the camera and projector occupy the same physical space and use the same optics. Since this is impossible, it is necessary to master the transformation between the real workspace, and the imaged area.This transformation is a mapping between two planes.

The projection of a plane to another plane is an affine transformation. Thus the video projector can be used to project a reference frame onto the physical desk in the form of a set of points. The camera image of these four points permits the calibration of 6 coefficients (A,B,C,D,E,F) which transform image coordinates (i, j) to workspace coordinates (x, y).

$$x = A\,i + B\,j + C \qquad\qquad y = D\,i + E\,j + F.$$

The visual processes required for the digital desk are relatively simple. The basic operation is tracking of a pointing device, such as a finger, a pencil or an eraser. Such tracking should be supported by methods to determine what device to track and to detect when tracking has failed. A method is also required to detect the equivalent of "mouse-down" and "mouse-up" events.

The tracking problem can be expressed as: "Given an observation of an object at time t, determine the most likely position of the same object at time t+ΔT". If different objects can be used as a pointing device, then the system must include some form of "trigger" which includes presentation of the pointing device to the system. The observation of the pointing device gives a small neighbourhood, w(*n,m*), of an image p(*i, j*). This neighbourhood will serve as a "reference template". The tracking problem can then be expressed as, given the position of the pointing device in the *k*th image, determine the most likely position of the pointing device in the *k+1*th image. The size of the tracked neighbourhood must be determined such that the neighbourhood includes a sufficiently large portion of the object to be tracked with a minimum of the background.

We have experimented with a number of different approaches to tracking pointing devices: these include color, correlation tracking, principal components and active contours (snakes) (Berard 1994) (Martin 1995). The active contour model (Kass 1987) presented problems which we believe can be resolved, but which will require additional experiments. Our current demonstration uses cross-correlation and principal components analysis.

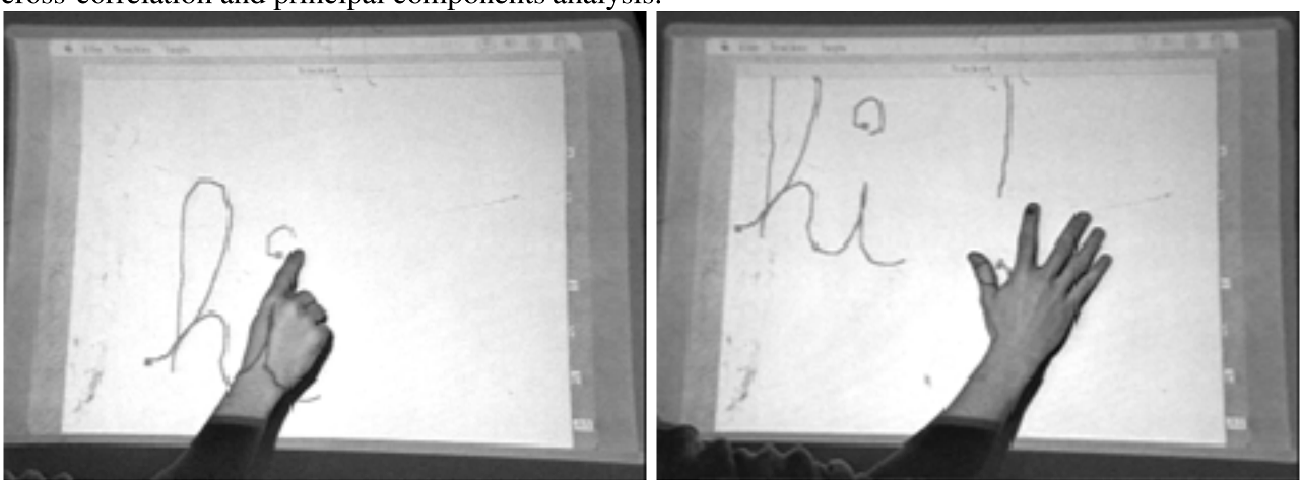


Figure 1 Drawing and placing with "Fingerpaint".

2.2 FingerPaint

As a simple demonstration, we constraucted a program called "FingerPaint"*. FingerPaint runs on an Apple Quadra AV/840 and uses a work-space projected with an overhead projector using a liquid-crystal display "data-show". A CCD camera with an 18mm lens observes this workspace and provides visual input. "Finger down" and "finger up" events are simulated using the space bar of the keyboard but they could be sensed using a microphone attached to the surface of the desk. As illustrated in Figure 1, any "natural pointing device" such as a finger can be used to draw pictures and letters, or to move a drawing.

The image at time $(k+1)\Delta T$ to be searched will be noted as $p_{k+1}(i, j)$. The search process can generally be accelerated by restricting the search to a region of this image, denoted $s(i,j)$, and called a "Region of Interest". Our system uses a square search region of size M by M centered on the location where the reference template was detected in the previous image.

The robustness of the tracking system is reasonable but, as discussed below, its performance is inadequate with respect to Fitt's law (Card 1983). Preliminary experiments with local users indicate however that the current performance is acceptable for investigation purposes. In addition, the

* The finger paint system has been implemented by Francois Berard.

widespread availability of image acquisition and processing hardware adequate for real time correlation should alleviate our current performance problem. Most importantly, this demonstration has permitted us to explore the problems involved in watching gesture.

Publihed in Proceedings of Engineering Human Computer Interaction, EHCI'95, Grand Targhee, USA, August 1995, Chapman&Hall Publ.

2.3 Correlation as a tracking technique for the digital desk

The basic operation for a digital desk application is tracking some pointing device, such as a finger, a pencil or an eraser. The tracking problem can be expressed as: "Given an observation of an object at time t, determine the most likely location of the same object at time t+ΔT". The pointing device can be modelled as a reference template. The reference template is a small neighbourhood, i.e., a window w(m, n) of a picture p(i, j) obtained at some prior time, t. The reference template is compared to an image neighborhood (i, j), by computing a sum of squared differences (SSD) between the N by N templace and the neighborhood of the image whose upper left corner is at (i, j).

$$SSD(i, j) = \sum_{m=0}^{N} \sum_{n=0}^{N} (p_k(i+m,j+n) - w(m, n))^2$$

A SSD value is computed for each pixel, (i, j), within the MxM search region. A perfect match between the template and the neighborhood gives a SSD value of 0. Such a perfect match is rarely obtained because of differences and appearance due to lighting and other effects. These effects can be minimized by normalising the energy in the template and the neighborhood.

Completing the squares of the SSD equation gives three terms, which can be written as:

$$SSD(i, j) = E^2_P(i, j) + E^2_W - 2 \langle p_k(i+m,j+n), w(m, n) \rangle$$

The term $\langle p_k(i+m,j+n), w(m, n)\rangle$ is the inner product of the template $w(m, n)$ with the neighborhood $p_k(i, j)$. The term $E^2_P(i, j)$ represents the energy in the image neighborhood and E^2_W is the energy in the reference window. The neighborhood and reference window may normalized by dividing by $E^2_P(i, j)$ and E^2_W, to give a normalized cross correlation, or NCC.

$$NCC(i, j) = \frac{\langle p_k(i+m,j+n)\, w(m, n)\rangle}{E^2_P(i, j)\; E^2_W}$$

Normalised cross correlation produces a peak with a value of 1.0 at a perfect match between window and neighborhood, and is relatively robust in the presence of noise, changes in scale and gray level, and image deformations (Martin and Crowley 1995). Hardware exists for computing a normalized cross correlation at video rates. However, in software, it is more efficient to use SSD.

Implementing cross-correlation by SSD requires solving practical problems such as determining the sizes of the reference template and of the search region, triggering and breaking tracking, and updating the reference template.

2.4 The size of the reference template

The size of the reference template must be determined such that it includes a sufficiently large portion of the device to be tracked and a minimum of the background. If the template window is too large, correlation may be corrupted by the background. On the other extreme, if the template is composed only of the interior of the pointing device then the reference template will be relatively uniform, and a high correlation peak will be obtained with any uniform region of the image, including other parts of the pointing device. For a reasonable correlation peak, the reference template size should be just large enough to include the boundary of the pointing device, which contains the information used for detection and localisation.

In fingerpaint, our workspace is of size 40 cm by 32 cm. This surface is mapped onto an image of 192 x 144 pixels, giving pixel sizes of 2 mm by 2.2 mm. At this resolution, a finger gives a correlation template of size 8 by 8 pixels or 16mm by 18mm, as shown in figure 2.

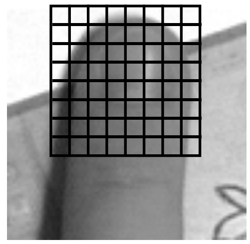


Figure 2 Reference template for a finger.

2.5 The size of the search region

The size M of the search region depends on the speed of the pointing device. Considerations based on Fitt's law (Card 1983) indicate a need for tracking speeds of up to 180 cm/sec. To verify this, we performed an experiment in which a finger was filmed making typical pointing movements in our workspace. The maximum speed observed in this experiment was V_m = 139 cm/sec. Expressed in pixels this gives $V_m = 695$ *pixels/sec*·

Given an image processing cycle time of ΔT seconds and a maximum pointer speed of V_m pixels/sec, it is possible to specify that the pointing device will be found within a radius of $M = \Delta T\ V_m$ pixels of its position in the previous frame. For images of 192 x 144 pixels, our built-in digitizer permits us to register images at a maximum frame rate of 24 frames per second, giving a cycle time of $\Delta T_{max} =$ 41.7 msec. This represents an upper limit on image acquisition speed which is attainable only if image tracking were to take no computation time.

The computational cost of cross-correlation is directly proportional to the number of pixels in the search region. Reducing the number of pixels will decrease the time needed for the inner loop of correlation by the same amount. This, in turn, increases the number of times that correlation can be operated within a unit time, further decreasing the region over which the search must be performed. Thus there is an inverse relation between the width of the search region, M, and the maximum tracking speed, V_m. The smaller the search region, the faster the finger movement that can be tracked, up to a limit set by the digitizing hardware.

The fastest tracking movement can be expected at a relatively small search region. This is confirmed by experiments. To verify the inverse relation between M and V_m, we systematically varied the size of the search region from M = 10 to 46 pixels and measured the cycle time that was obtained. The maximum speed of 126 pixels/sec is obtained with M=26. Although this is 5.5 times less than the maximum desirable speed (i.e., 695 pixels/sec), the system is quite usable to perform drawing and placements in a "natural" way.

2.6. Triggering and breaking tracking

When tracking is not active, the system monitors an N by N pixel "tracking trigger", $T_k(i,j)$, located in the lower right corner of the workspace. As each image is acquired at time k, the contents of this tracking trigger are subtracted from the from the contents at the previous image *k-1*. This creates a difference image as shown in figure 3. The energy of the difference image is computed as

$$E_k = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \left(T_k(m,n) - T_{k-1}(m,n)\right)^2$$

When a pointing device enters the tracking trigger, the energy rises above a threshold. In order to assure that the tracking device is adequately positioned, the system waits until the difference energy drops back below the threshold before acquiring the reference template. At that point, the contents of the tracking trigger, $T_k(m, n)$ are saved as a reference image, and the tracking process is initiated.

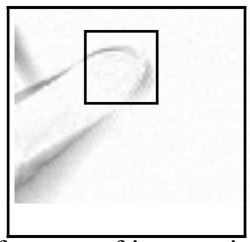


Figure 3 Temporal difference of images in the reference square.

Tracking continues as long as the minimum value of SSD remains below a relatively high threshold. However, it can happen that the tracker locks on to a pattern on the digital desk (for example a photo of the pointing device!). To cover this eventuality, if the tracked location of the pointer stops moving for more than a few seconds (say 10), the system begins again to observe the difference energy in the tracking trigger. If the trigger energy rises above threshold, the tracker will break the previous track and re-initialise the reference pattern with the new contents of the tracking trigger.

2.7. Updating the reference mask

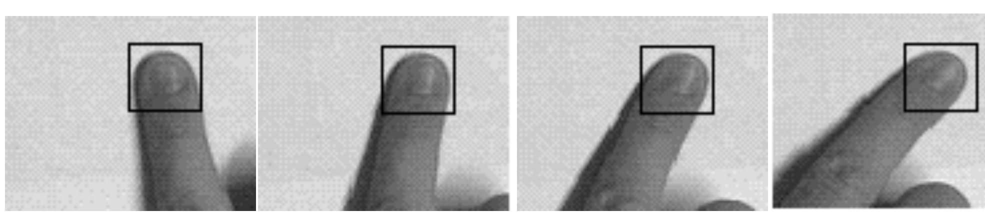


Figure 4 : Change in reference template as a function of finger orientation.

As the user moves the pointing device around the workspace, there is a natural tendency for the device to rotate, as shown in figure 4. This, in turn, may cause loss of tracking. In order to avoid loss of tracking, the smallest SSD value from each search is compared to a threshold. If the smallest SSD rises above this threshold, then the reference template is updated using the contents of the image at time k-1 at the detected position.

Tracking fingertips is an example of a simple fast visual process which can be used to change the nature of the interaction between man and machine. Vision can also be used to make the machine aware of the user by detecting, tracking and watching his face.

Publihed in Proceedings of Engineering Human Computer Interaction, EHCI'95, Grand Targhee, USA, August 1995, Chapman&Hall Publ.

3 FACES: DETECTING, TRACKING, AND WATCHING THE USER

Face to face communication plays an important role in human to human communication. Thus it is natural to assume that an important quantity of non-verbal information can be obtained for man-machine interaction by watching faces. However, even more than hands, face interpretation poses difficult problems for established machine vision techniques. In this section we briefly report on experiments with simple techniques for detecting, tracking and interpreting images of faces. The key to robustness in such tracking and interpretation is the integration of complementary techniques.

3.1 Why watch a face?

Detection and interpretation of a face image can have a number of applications in machine vision. The most obvious use is to know whether a person is present in front of a computer screen. This makes a cute, but very expensive, screen saver. It is also possible to use face recognition as a substitute for a login password, presenting a person with his preferred workspace as soon as he appears in front of the computer system. Slightly more practical is the use of computer vision to watch the eyes and lips of a user. Eye tracking can be used to determine whether the user is looking at the computer screen and to which part his fixation is posed. This could conceivably be used to activate the currently active window in an interface. Observing the mouth to detect lip movements can be used to detect speech acts in order to trigger a speech recognition system.

None of the above uses would appear to be compelling enough to justify the cost of a camera and digitizer. However, there is an application for which people are ready to pay the costs of such hardware: video communications. Recognizing and tracking faces can have several important uses for the applications of video telephones and video conferencing. We are currently experimenting with combining face interpretation with a rudimentary sound processing system to determine the origin of spoken words and associate speech with faces. Each of the applications which we envisage require active computer control of the direction (pan and tilt) zoom (focal length), focus and aperture of a camera. Fortunately, such cameras are appearing on the market.

In the video-telephone application, we use an active camera to regulate zoom, pan, tilt, focus and aperture so as to keep the face of the user centered in the image at the proper size and focus, and with an appropriate light level. Such active camera control is not simply for esthetics. Keeping the face at the same place, same scale and same intensity level can dramatically reduce the information to be transmitted. One possible such coding is to define (on-line) a face space using principle components analysis (defined below) of the sample images from the last few minutes. Once the face basis vectors are transmitted, subsequent images can be transmitted as a short vector of face space coefficients. Effective use of this technique is only possible with active camera control. Other image codings can also be accelerated if the face image is normalized in position, size, gray level and held in focus.

In the video-conference scenario, a passive camera with a wide angle lens provides a global view of the persons seated around a conference table. An active camera provides a closeup of which ever person is speaking. When no one is speaking, and during transitions of the close-up camera, the wide-angle camera view can be transmitted. Face detection, operating on the wide-angle images, can be used to estimate the location of conference participants around the table. When a participant speaks, the high resolution camera can zoom onto the face of the speaker and hold the face in the center of the image.

What are the technologies required for the above applications? Both scenarios require an ability to detect and track faces, and an ability to servo control pan, tilt, zoom, focus and aperture so as to maintain a selected face in the desired position, scale and contrast level. From a hardware standpoint, such an application requires a camera for which these axes can be controlled. Such camera heads are increasingly appearing on the market. For example, we have purchased a small RS232 controllable camera from a Japanese manufacturer which produces excellent color images for little more than the price of a normal color camera.

A second hardware requirement is the ability to digitize and process images at something close to video-rates. The classic bottle-neck here is communication of the image between the frame-grabber and the processor. Fortunately, the rush to multi-media applications has pushed a number of vendors to produce workstations in which a framegrabber is linked to the processor by such a high speed bus. Typical hardware available for a reasonable cost permits acquisition of up to 20 frames per second at full image size and full video rates for reduced resolution images. Adding simple image processing can reduce frame rates to 2 to 10 Hz (depending on image resolution). Such workstations are suitable for concept demonstrations and for experiments needed to define performance specifications. An additional factor of 2 (18 months) in band-width and processing power will bring us to full video-rates.

The questions we ask in the laboratory are: What software algorithms can be used for face detection, tracking and recognition, and what are the systems concepts needed to tie these processes together. Systems concepts have been the subject of our ESPRIT Basic Research Project "Vision as Process", described in the book (Crowley and Christensen 1994) or the paper (Crowley 94) for more details.

3.2 Detection: Finding a face with color

One of the simplest methods for detecting pixels which might be part of a face is to look for skin color. Human skin has a particular hue and saturation. The intensity, however, will vary as a function of the relative direction to the illumination. Of course, the perceived hue and saturation will be the product of the color of the ambient light and the skin color (Schiele and Weibul 1995).

We have found that candidate pixels for faces and hands can be detected very rapidly using a normalized color histogram. Histogram color can be normalised for changes in intensity by dividing the color vector by the luminance. This permits us to convert a color vector [R, G, B] having three dimensions into a vector [r, g] of normalised color having two dimensions. The normalized color histogram H(r, g) provides a fast means of skin detection. The histogram is initialised by observing a patch of skin and counting the number of times each normalized color value occurs. The histogram contains the number of pixels which exhibit a particular color vector [r, g]. Dividing by the total number of pixels in the histogram, N, gives the probability of obtaining a particular vector given that the pixel observes skin.

$$P(color \mid skin) = \frac{1}{N} H(r, g)$$

Bayes rule can be used to determine the probability of skin given the color has values [r, g].

$$P(skin \mid color) = \frac{P(skin)}{P(color)} P(color \mid skin)$$

P(skin) can be taken as constant. P(color) is the global statistic for the vector [r, g]. In practice this ratio is often approximated by a constant. The result at each pixel is the estimate of the probability of skin. An example (unfortunately printed here in black and white) is shown in figure 5a through 5d.

Figure 5 a. Black and white rendition of color image of Y.H. Berne

Figure 5 b. Probability of skin in color image of Y.H. Berne.

Figure 5 c. Thresholded skin probability with bounding box of connected components

Figure 5d Bounding box for the face.

Histogram matching can provide a very fast indicator that a face is present at a part of an image. However, reliability requires that this information be confirmed by another detection means. Such a means can be provided by a blink detector.

3.3 Finding a face by blink detection

A human must periodically blink to keep his eyes moist. Blinking is involuntary and fast. Most people do not notice when they blink. However, detecting a blinking pattern in an image sequence is an easy and reliable means to detect the presence of a face. Blinking provides a space-time signal which is easily detected and unique to faces. The fact that both eyes blink together provides a rendundance which permits blinking to be discriminated from other motions in the scene. The fact that the eyes are symmetrically positioned with a fixed separation provides a means to normalize the size and orientation of the head.

We have built a simple blink detector which works as follows: As each image is acquired, the previous image is subtracted. The resulting difference image generally contains a small boundary region around the outside of the head. If the eyes happened to be closed in one of the two images, there are also two small roundish regions over the eyes where the difference is significant.

The difference image is thresholded, and a connected components algorithm is run on the thresholded image. A bounding box is computed for each connected components. A candidate for an eye must have a bounding box within a particular horizontal and vertical size. Two such candidates must be detected with a horizontal separation of a certain range of sizes, and little vertical difference in the vertical separation. When this configuration of two small bounding boxes is detected, a pair of blinking eyes is hypothesized. The position in the image is determined from the center of the line between the bounding boxes. The distance to the face is measured from the separation. This permits to determine the size of a window which is used to extract the face from the image. This simple technique has proven quite reliable for determining the position and size of faces.

3.4 Tracking the face by correlation (SSD)

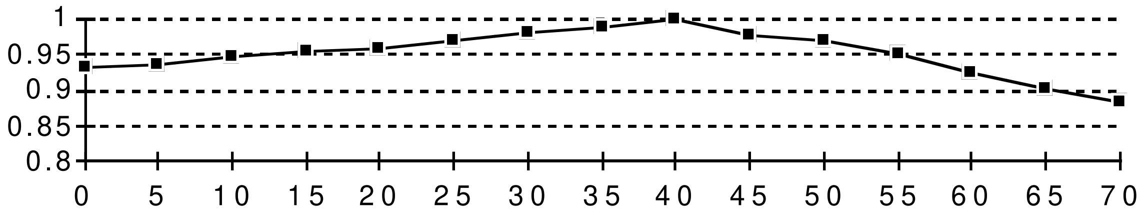
Detection of a face by color is fast and reliable, but not always precise. Detection by blinking is precise, but requires capturing an image pair during a blink. Correlation can be used to complete these two techniques and to hold the face centered in the image as the head moves. The crucial question is how large a window to correlate (N) and over how large a search region to search.

We have found that a 7 by 7 search region extracted from the center of a face is sufficient to determine the precise location of the face in the next image. The size of the search region is determined by the speed with which a face can move between two frames. Calling on the finger tracking results above, we optimised the search region to maximize the frequency of image acquisition. With our current hardware this is provided by a search region to 26 by 26 pixels, but must be adjusted when several image processes are running in parallel.

Correlation can also be used in isolation to track a face. As a test of the robustness of correlation, we acquired a sequence of images of a head turning (shown in figure 6 below). We then correlated the center image from the sequence to produce the correlation graph shown in figure 7. The graph shows the results of zero-mean normalized correlation for the 40th face image with the other images in this set.

Figure 6 Every 5th image from a sequence of 70 images of Jerome Martin turning.

Publihed in Proceedings of Engineering Human Computer Interaction, EHCI'95, Grand Targhee, USA, August 1995, Chapman&Hall Publ.



F
igure 7 Correlation values for correlation of face number 40 with the face images from figure 6.

3.5 Identifying the face by EigenSpace decomposition

The pixels which compose an image can be considered as a very long vector. Thus, an image of a prototypical face can be considered as a basis vector for describing other images by inner product. A set of such images can form a space for describing a shape. For example, images number 0, 30 and 60 from figure 6 can be used to define a space of a face turning. A particular face image can be represented by the vector of 3 inner products obtained with these three images. It is necessary to compute this inner product at an appropriate location, but since correlation is a sequence of inner products, it is possible to find the peak correlation, and then describe the image by the vector of inner products obtained at that position.

The problem with this approach is that it can rapidly become expensive as the number of images increases. However, the image set can be reduced to minimal orthogonal basis set, and the correlation with this basis set used to describe the contents. This is the idea behind the EigenSpace coding made popular by Turk and Pentland (1991). Correlation with a set of Eigen images is commonly thought of as a technique for recognition. However, it is also possible to use such a coding as a compact image transmission code provided that the position, scale and contrast are suitably normalised.

To construct an eigenspace, we begin with a database of images, as for example shown in figure 8. We then compute an average image as shown in figure 9. Finally, an fast algorithm is used to compute the covariance of the of the image data based. For a 128 by 128 = 65,536 pixel image, this covariance matrix has, in theory, 65,5362 coefficients. Fortunately, this covariance is highly diagonal, and a fast algorithm can be used to compute and diagonalise this covariance. The principal components of the covariance matrix form an orthogonal basis set, which are the axis of the EigenSpace as shown in figure 9.

One of the simplest applications of the eigenfaces method is the recognition of a subject We have prepared a simple demo which works as follows. At the beginning of a session, the system classifies the subjects face in order to determine if the subject is known. Classification is performed by multiplying the normalized face by each of the principle component images in order to obtain a vector. The vector positions the image in the "face space" defined by the current Eigenfaces. If the face is near a position of this space which corresponds to a known subject, then the subject's image from the face-space database is displayed. If the vector is not near a known subject, the subject is classified as unknown and no face is displayed. Using the Eigenface technique, our Quadra 700 with no additional hardware can digitize and classify a face within a 108 by 120 image for a database of 12 images at about 1 frame per second.

It is possible to use the Eigenface technique to measure parameters. One example of this is for eye-tracking. We train a set of images of the subject looking in different directions and use these images to form an Eigen-Space. During execution of a task, a high-resolution window is placed over the subjects eyes, and the position in the Eigen-Space is computed. The nearest principle components are used to interpolate the current horizontal and vertical direction.

We are experimenting with this technique to determine the trade-off between resolution of the windows on the eyes, the number of eigen-images needed, and the precision which we can obtain in eye tracking. The goal is to be able to drive a pointing device, such as a mouse with such eye tracking.

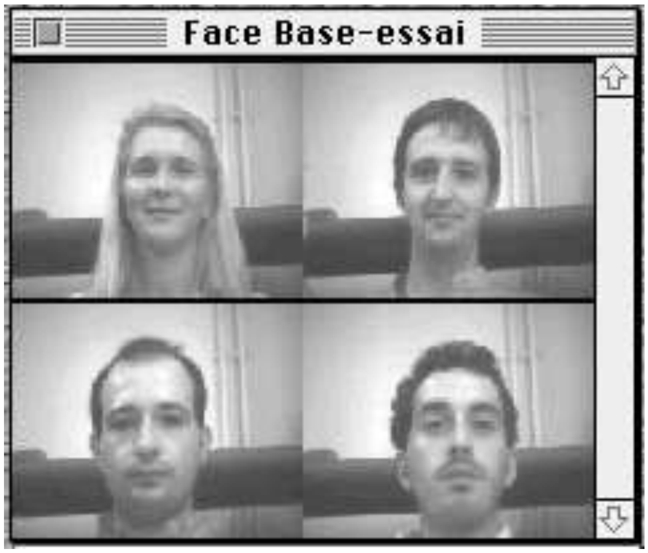


Figure 8 A small face data base composed of 4 images.

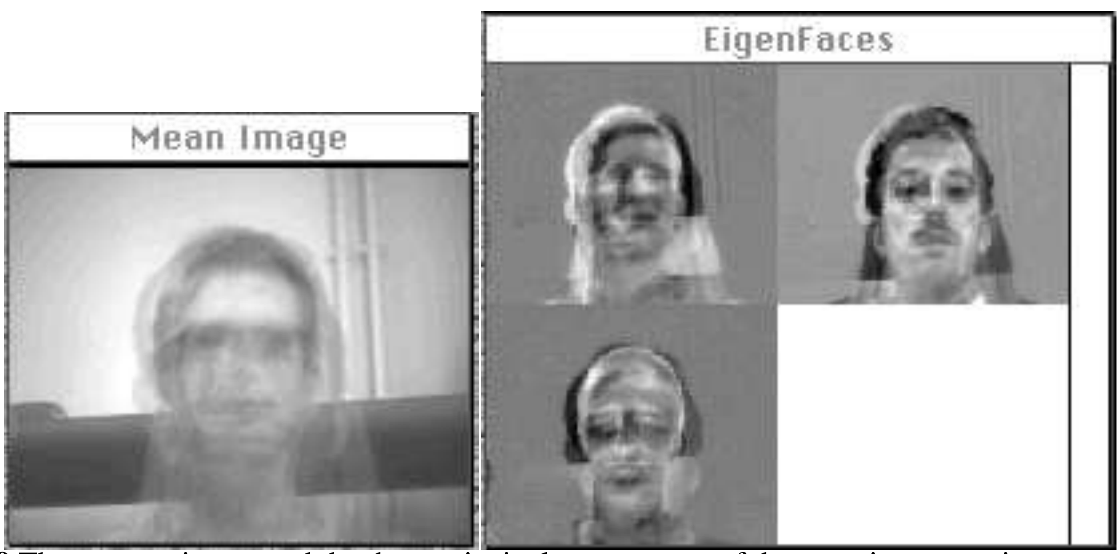


Figure 9 The average image and the three principal components of the covariance matrix.

Facial expression contains useful informations about the user's state of mind. The Eigenfaces principle can be easily extended to classifying the users facial expression. A set of facial expressions are obtained as the subject performs his task. These facial expressions are then used to form an Eigenspace. At each instant, the system determines the face expression class which most closely corresponds to the users current expression. In this way, we can experiment with anticipating the users "mood" based on facial expression.

4. CONCLUSION

In this paper we have presented some preliminary results with detection and tracking of fingers and faces. We have also experimented with eyetracking using snakes and other techniques based on signal processing, as well as face interpretation using EigenFaces. It is increasingly apparent that computer vision has the potential to provide a rich new source of interaction modes for computer human interaction. Vision can make the machine "aware" of the user, his movements and environment, in ways that are yet to be invented.

BIBLIOGRAPHY

Anandan, P. (1987) *"Measuring Visual Motion From Image Sequence"*, Phd dissertation and COINS Technical Report 87-21, University of Massachusetts, Amherst.

Balbo, S. Coutaz J. and Salber, D. (1993)*"Towards Automatic Evaluation of Multimodal User Interfaces"*, International Workshop on Intelligent User Interfaces, Orlando, USA.

Baudel, T. Beaudouin-Lafon, M., (1993) "Charade: Remote Control of Objects Using Free-Hand Gestures", Communications of the ACM, Vol.36 No.7, pp. 28-35.

Berard, F. (1994) "*Vision par Ordinateur pour la Réalité Augmentée: Application au Bureau Numérique*", Mémoire du D.E.A. en Informatique, Univeristé Joseph Fourier.

Blake A. and Isard, M. (1994) "3D Position, attitude and shape input using video tracking of hands and lips", ACM - SIGGRAPH Annual Conference on Computer Graphics.

Cadoz, C. (1994), *Les réalités virtuelles*, Dominos, Flammarion.

Coutaz, J. (1994), *Interfaces hommes-ordinateur Conception et réalisation*, Dunod Informatique.

Crowley, J. L. and Christensen, H. (1994) *Vision as Process*, Springer Verlag, Heidelberg.

Crowley , J. L. and Bedrune, J. M. (1994), "Integration and Control of Reactive Visual Processes", 1994 European Conference on Computer Vision, (ECCV-'94), Stockholm, may 94.

Harris, C. (1992), "Tracking with Rigid Models" in "*Active Vision*", The MIT Press.

H. Inoue, H. Tashikawa T. and Inaba, M. I (1992), "*Robot vision sytem with a correlation chip for real time tracking, optical flow, and depth map generation*", The 1992 IEEE Conference on Robotics and Automation, Nice.

Kass, M. Witkin, A. and Terzopoulos, D. (1987), "Snakes : Active Contour Models". Proc. 1st International Conf. on Computer Vision, pp. 259-268.

Krueger, M. (1991), "*Artifical Reality II*", Addison Wesley.

Maes, P. Darrel, T., Blumberg B. and Pentland, A. (1994), *"The ALIVE system : Full-Body Interaction with Animated Autonomous Agent"*. M.I.T. Media Laboratory Perceptual Computing Technical Report No. 257.

Martin, J. (1995) *"Suivi et Interprétation de Geste : Application de la Vision par Ordinateur à l'Interaction Homme-Machine*", Rapport DEA Informatique, IMAG - INPG.

Martin J. and Crowley, J. L. (1995), "Experimental Comparison of Correlation Techniques", IAS-4, International Conference on Intelligent Autonomous Systems, Karlsruhe.

Newman, W. Wellner. P (1992), *"A Desk Supporting Computer-based Interaction with Paper Documents"*. Proceedings CHI'92, pp. 587-592.

Ramstein, C. (1994), "The Pantograph: A Large Workspace Haptic Device for Multimodal Human Computer Interaction", CHI'94 Interactive Experience, Adjunct Proceedings, pp. 57-58.

Rehg J. M. and Kanade, T. (1993)*"DigitEyes : Vision-Based Human Hand Tracking"*. Carnegie Mellon University Technical Report CMU-CS-93-220.

Scheile B. and Weibul, A. (1995) "Gaze Tracking Based on Face Color", International Workshop on Face and Gesture Recognition, Zurich.

Turk, M. and Pentland A. (1991), "*Eigenfaces for Recognition*" Journal of Cognitive Neuroscience, 3(1):71-86, 1991.

Wellner, P. Mackay, W. and Gold, R. (1993),*"Computer-augmented environments : back to the real world"*. Special Issue of Communications of the ACM, Vol.36 No.7.

Wellner, P. (1993), *"Interacting with paper on the DigitalDesk"*, Communications of the ACM, Vol.36 No.7.

Wozencraft, J. M. and Jacobs, I. M. (1965), *Principles of Communication Engineering*, John Wiley and Sons.

Publihed in Proceedings of Engineering Human Computer Interaction, EHCI'95, Grand Targhee, USA, August 1995, Chapman&Hall Publ.

Biography of James L. Crowley

James L. Crowley completed a Ph.D. at Carnegie Mellon University, in Pittsburgh, USA in 1981. From 1980 until 1985 he was a Research Scientist in the CMU Robotics Institute, where he directed the "Laboratory for Household Robotics". Since 1985 he has held the position of Professeur at the Institut National Polytechnique de Grenoble (I.N.P.G.), where he teaches courses in Artificial Intelligence, Machine Vision, and Robotics at l'Ecole Nationale Supérieure d'Informatique et de Mathématiques Appliquées (ENSIMAG).

Professor Crowley performs his research within the Institut Mathématique Appliqué de Grenoble (IMAG), where he is responsable for the project: "PRIMA - Integration of Mobile Robotics and Active Perception". In the area of machine vision, he is the technical coordinator of ESPRIT project 7108, "Vision as Process-II", which has as its goal to develop an integrated active vision system. He is organizer and coordinator of the European Network of Excellence in Computer Vision (CVNET) and is also coordinator of the European HCM Network: SMART (Mobile Autonomous Surveillance Robots).

Ten doctoral dissertations have been completed under the direction of Professor Crowley, and he is currently directing five other doctoral students. He is on the editorial board of "Robotics and Autonomous Systems" and has served on numerous program committees (ECCV, IAS, SCIA). He has organised seven workshops and symposiums on computer vision and/or robotics. He is the author of over 100 articles on AI, vision and mobile robotics and has recently published books on active computer vision and on integration environments for computer vision and image processing.

Biography of Joëlle Coutaz

Joëlle Coutaz is professor at the University Joseph Fourier of Grenoble where she has been teaching the foundations for operating systems since 1973 and Man-Machine Interaction since 1986. She is the founder and director of the User Interface group at LGI-IMAG (Laboratoire de Génie Informatique of IMAG). This group is primarily concerned with the software engineering aspects of Computer Human Interaction and aims at transferring HCI design principles and theories into software design modelling.

In particular, the group has developed PAC, a conceptual architecture model that captures software engineering issues such as modularity and modifiability, as well as user-centred concerns such as multithreading and multimodal interaction. The group has developed demonstrators and software tools based on the PAC model: a fusion engine has been implemented as a reusable mechanism for supporting multimodal input interaction; PAC-Expert, an expert system based on heuristic rules, demonstrates the automatic generation of a PAC global architecture from the description of the system behaviour as perceived by the user; an environment based on PAC, has been developed for Renault, the French car company, to specify and generate embedded user interfaces; NEIMO, a usability lab based on PAC, is under development in collaboration with CNET, the French telephone company.

Joëlle Coutaz is the author of two books : "Developing Software for the User Interface (co-author Len Bass), Addison Wesley, 1991; and "Interfaces Homme-Ordinateur : Conception et Réalisation", Dunod, 1990. She is a member of the ESPRIT BRA AMODEUS and member of the Working Group 2.7 of IFIP. In France, she is the founder of the research group "Multimodal man-

Publihed in Proceedings of Engineering Human Computer Interaction, EHCI'95, Grand Targhee, USA, August 1995, Chapman&Hall Publ.

machine communication" of the national programme PRC Man-Machine Communication. She also co-chairs the CSCW working group of PRC Man-Machine Communication and is member of the CNRS research group Cogniscience.

Publihed in Proceedings of Engineering Human Computer Interaction, EHCI'95, Grand Targhee, USA, August 1995, Chapman&Hall Publ.

Biography of James L. Crowley

James L. Crowley completed a Ph.D. at Carnegie Mellon University, in Pittsburgh, USA in 1981. From 1980 until 1985 he was a Research Scientist in the CMU Robotics Institute, where he directed the "Laboratory for Household Robotics". Since 1985 he has held the position of Professeur at the Institut National Polytechnique de Grenoble (I.N.P.G.), where he teaches courses in Artificial Intelligence, Machine Vision, and Robotics at l'Ecole Nationale Supérieure d'Informatique et de Mathématiques Appliquées (ENSIMAG).

Professor Crowley performs his research within the Institut Mathématique Appliqué de Grenoble (IMAG), where he is responsable for the project: "PRIMA - Integration of Mobile Robotics and Active Perception". In the area of machine vision, he is the technical coordinator of ESPRIT project 7108, "Vision as Process-II", which has as its goal to develop an integrated active vision system. He is organizer and coordinator of the European Network of Excellence in Computer Vision (CVNET) and is also coordinator of the European HCM Network: SMART (Mobile Autonomous Surveillance Robots).

Ten doctoral dissertations have been completed under the direction of Professor Crowley, and he is currently directing five other doctoral students. He is on the editorial board of "Robotics and Autonomous Systems" and has served on numerous program committees (ECCV, IAS, SCIA). He has organised seven workshops and symposiums on computer vision and/or robotics. He is the author of over 100 articles on AI, vision and mobile robotics and has recently published books on active computer vision and on integration environments for computer vision and image processing.

Biography of Joëlle Coutaz

Joëlle Coutaz is professor at the University Joseph Fourier of Grenoble where she has been teaching the foundations for operating systems since 1973 and Man-Machine Interaction since 1986. She is the founder and director of the User Interface group at LGI-IMAG (Laboratoire de Génie Informatique of IMAG). This group is primarily concerned with the software engineering aspects of Computer Human Interaction and aims at transferring HCI design principles and theories into software design modelling.

In particular, the group has developed PAC, a conceptual architecture model that captures software engineering issues such as modularity and modifiability, as well as user-centred concerns such as multithreading and multimodal interaction. The group has developed demonstrators and software tools based on the PAC model: a fusion engine has been implemented as a reusable mechanism for supporting multimodal input interaction; PAC-Expert, an expert system based on heuristic rules, demonstrates the automatic generation of a PAC global architecture from the description of the system behaviour as perceived by the user; an environment based on PAC, has been developed for Renault, the French car company, to specify and generate embedded user interfaces; NEIMO, a usability lab based on PAC, is under development in collaboration with CNET, the French telephone company.

Joëlle Coutaz is the author of two books : “Developing Software for the User Interface (co-author Len Bass), Addison Wesley, 1991; and “Interfaces Homme-Ordinateur : Conception et Réalisation”, Dunod, 1990. She is a member of the ESPRIT BRA AMODEUS and member of the Working Group 2.7 of IFIP. In France, she is the founder of the research group “Multimodal man-machine communication” of the national programme PRC Man-Machine Communication. She also co-chairs the CSCW working group of PRC Man-Machine Communication and is member of the CNRS research group Cogniscience.

Publihed in Proceedings of Engineering Human Computer Interaction, EHCI'95, Grand Targhee, USA, August 1995, Chapman&Hall Publ.