Visual Processes for Tracking and Recognition of Hand Gestures

James L. Crowley and Jerome Martin
Institut National Polytechnique de Grenoble
46 Ave Félix Viallet
38031 Grenoble, France
email: jlc@imag.fr

Abstract

This paper describes experiments with techniques for tracking hands and recognizing gestures. Complementary techniques are presented for detecting and tracking hands and tools. These techniques are integrated within a system which uses multiple image processing techniques to estimate the position and orientation of a hand.

Images of the tracked hand are normalized in orientation and position and then projected into a principal components space. Hand configurations are represented using a probabilistic classification. Gestures are recognized in this space as sequences of hand configurations using finite state machines.

1 Direct Manipulation of Objects as an Interaction Modality

Human gesture serves three functional roles [4]: 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 artifacts, 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* or tool. Examples include a handkerchief for the semiotic good-bye gesture, a turn-table for the ergotic shape-up gesture of pottery, or a dedicated artifact to explore the world (for example, a retro-active system such as the pantograph [13] to sense the invisible).

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 [1].

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.

Computer vision can transform ordinary artefacts and even body parts into effective input devices. Krueger’s work on the video-place [11], followed recently by Wellner and Mackay's concept of "digital desk" [18] 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 the integration of appearance-based computer vision techniques to non-intrusively observe human gesture in a fast and robust manner.

The techniques described in this paper are developed within the context of a digital desk. In this system, a computer screen is projected onto a physical desk using 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.

The workspace contains a number of physical and virtual objects which can be manipulated by a human hand. Both physical and virtual devices act as tools whose manipulation is a communication channel between the user and the computer. The identity of the object which is manipulated carries a strong semiotic message. The manner in which the object is manipulated provides both semiotic and ergotic information.

Tools may be virtual objects, or any convenient physical object which has been previously presented to the system. Virtual objects are generated internally by the system and projected

onto the workspace. Examples include cursors and other visual feedback symbols as well as shapes and words which may have meaning to the user. Virtual objects are easily created, but they lack the tactile (epistemic) feedback provided by physical objects.

The vision system must be able to detect, track and recognize the user's hands as well as the tools which he manipulates. The system must also be able to extract meaning from the way in which tools are manipulated.

The approach

Rather than attempt to reconstruct the 3D configuration of hands and tools, such as [14], we describe the space of possible appearances which the hands and tools can present. This space is reduced by registering and normalizing a window around the hand and tool using multiple visual processes for tracking. The space is further reduced by compressing the set of all possible appearances to a subspace defined by its principal components.

Registering and normalizing the images of the hand and objects requires tracking. We have experimented with a number of different approaches to tracking hands. These include color histogram matching, cross-correlation with images, projections to principle components space and active contours [2] [12]. These techniques are complementary, with different failure conditions and different operational requirement. By combining several techniques in such a way that they can be dynamically initiated and controlled, we arrive at a system which is both flexible and robust.

2. Integration of visual processes for tracking hands

This work builds on results with architectures for continuously operating vision systems [7]. Several systems have been constructed using this architecture for applications including appearance based navigation [Jones et al 97], for tracking people, and for tracking faces for video-communications [9]. We call this architecture synchronous ensemble of reactive processes (SERP) [5].

Within this architecture, reactive transformations are activated and disabled in response to scene events and to explicit system goals. Processes are executed in a synchronous manner with an explicit limit placed on the computing time which each process may use in each cycle. When executed in a standard Unix environment, such a system provides only soft real-time response. Hard real time response can be obtained when such an architecture is

used with a real time kernel. In either case, the supervisor must manage the time used by each phase so as to assure a fixed cycle time.

The SERP model is illustrated in figure 1. The system is built around an interpreter, to which have been linked procedures for image acquisition, visual processes, and device controllers. The system supervisor is expressed as a set of rules which react to events and commands, as well as a set of objects which represent the current state of visual processes. The supervisor manages a scheduler which drives the system as a sequence of phases.

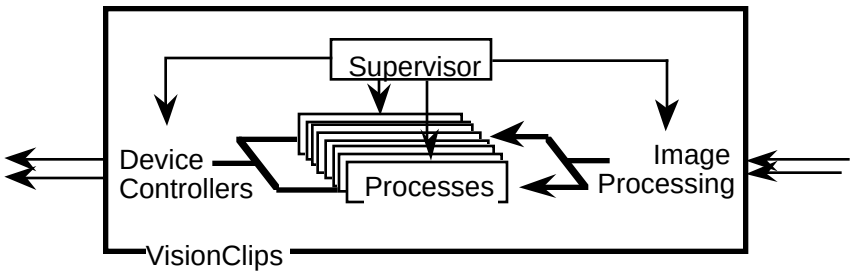

Figure 1. Synchronous Ensemble of Reactive Processes"

The supervisor receives messages from the visual processes concerning commands and visual events. In reaction to these messages, visual processes are activated or disactivated. At the beginning of each cycle, the supervisor selects individual processes and executes the image acquisition procedure to obtain a new image. The supervisor then calls the selected visual process with its parameters and time allocation. Processes generate symbolic messages to the supervisor which can change the state of subsequent processes.

The tracking process can be separated into separate parts for detection and estimation. The results of such tracking process can be easily combined in a tracking process, provided that the output of tracking is accompanied by a covariance matrix which indicates precision of detection, and a confidence factor which indicates the reliability of the result.

3. The tracking process.

Visual processes pass information to a tracking process which maintains an estimate of the center point and size of the hand and tool. This tracking process is a form of recursive estimator programmed using a zeroth Kalman filter. The result is a normalized window around the hand.

Tracking must be accompanied by methods to determine when to initiate tracking as well as

when tracking has failed. Tracking is initiated by detecting regions of movement using space time filters. The reliability of each detection is estimated using a confidence factor, based on the likelihood of the detected parameters.

The tracking process is formulated as a recursive estimation process [3]. Such a process maintains an estimate of a state vector, and its uncertainty. By adding a confidence factor to the tracking process, the system can determine the reliability of tracking results, and initiate corrective action when necessary [6].

Our system samples and treats images at approximately 10 images per second. Because a manipulating hand will often have accelerations which are very fast with respect to this sample rate, we do not attempt to estimate hand velocity. Thus our tracking process is a zeroth order estimation system. Uncertainty due to motion and acceleration by modeled as an error term to the uncertainty of the state vector during each cycle (a process noise term).

The state maintained by the tracking process is the estimated center position of the hand and tool (x, y), and the bounding box of its spatial extent, (Δx, Δy). Normally, the covariance of these four parameters is a 4x4 covariance matrix. Because position can be considered to be independent of size, this covariance can be separated into two independent 2x2 covariance matrices, yielding an important gain in the estimation process.

$$\hat{X} \equiv \begin{bmatrix} \hat{X}_{pos} \\ --- \\ \hat{X}_{box} \end{bmatrix} \equiv \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{\Delta x} \\ \hat{\Delta y} \end{bmatrix}$$

$$\hat{\mathbf{C}}_X \equiv \left[\begin{array}{cc|cc} \multicolumn{2}{c|}{\multirow{2}{*}{$\hat{\mathbf{C}}_{pos}$}} & \multicolumn{2}{c}{\multirow{2}{*}{0}} \\ & & & \\ \hline \multicolumn{2}{c|}{\multirow{2}{*}{0}} & \multicolumn{2}{c}{\multirow{2}{*}{$\hat{\mathbf{C}}_{box}$}} \\ & & & \end{array}\right] \equiv \begin{bmatrix} \hat{\sigma}_x^2 & \hat{\sigma}_{xy} & 0 & 0 \\ \hat{\sigma}_{xy} & \hat{\sigma}_y^2 & 0 & 0 \\ 0 & 0 & \hat{\sigma}_{\Delta x}^2 & \hat{\sigma}_{\Delta x \Delta y} \\ 0 & 0 & \hat{\sigma}_{\Delta x \Delta y} & \hat{\sigma}_{\Delta y}^2 \end{bmatrix}$$

where

$$\hat{X}_{pos} = \begin{bmatrix} x \\ y \end{bmatrix} \qquad \hat{\mathbf{C}}_{pos} = \begin{bmatrix} \hat{\sigma}_x^2 & \hat{\sigma}_{xy} \\ \hat{\sigma}_{xy} & \hat{\sigma}_y^2 \end{bmatrix}$$

$$\hat{X}_{box} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \qquad \hat{\mathbf{C}}_{box} = \begin{bmatrix} \hat{\sigma}_{\Delta x}^2 & \hat{\sigma}_{\Delta x \Delta y} \\ \hat{\sigma}_{\Delta x \Delta y} & \hat{\sigma}_{\Delta y}^2 \end{bmatrix}$$

Confidence $CF \in [0, 1]$

Prediction

In each cycle, a process noise term is added to the covariance parameters. This process noise depends on the elapsed time since the last processing cycle, ΔT. Thus if all visual processes fail, a default value, with a steadily falling precision, is maintained. At the same time the confident factor is degraded by an exponential decay rate, α which is a value less than 1.

$$X^* := \hat{X}$$

$$\mathbf{C}_X^* := \hat{\mathbf{C}}_X + N_X \Delta T^2$$

$$CF := CF^{\alpha}$$

Validation

The results provided from detection are verified by computing the Mahalanobis distance from the predicted hand, X. The distance must be less than a threshold for the detection to be used in updating the estimation.

$$d = \frac{1}{2} (Y - X^*)^T \mathbf{C}_X^{*-1} (Y - X^*)$$

Estimation

The visual processes produce estimates and covariances which are in the same parameters as the tracking process. In this case, the equations for recursive estimation are quite simple. It is well known that probabilities combine as masses (more precisely, mass is a probabilistic quantity). Thus, parameter covariance are updated as the inverse of the sum of the inverses. For the k^{th} tracking processes, each producing estimates Y_k with covariance $\hat{\mathbf{C}}_k$:

$$\hat{\mathbf{C}}_X := \left(\mathbf{C}_X^{*-1} + \sum_{k=1}^{K} \mathbf{C}_k^{-1} \right)^{-1}$$

The updated covariance and can then be used to update the state vector, by providing a weighting factor for each estimate.

$$\hat{X}_k = \hat{C} \left(X^* C_X^{*-1} + \sum_{k=1}^{N} Y_k C_k^{-1} \right)$$

4 Visual Processes Detecting and Tracking Hands

The detection process operates using a variety of detection techniques. These include cross-correlation between frames, color histogram matching, active contours and correlation with an orthogonal basis space. These processes are complementary. They fail in different circumstances. Fusing the results of several tracking processes provides improvements in both robustness and precision.

Initiating tracking

Tracking is initiated by motion. When the system is inactive, a motion detection process monitors the energy in the difference between. The process subtracts a low resolution version of each image, $P_k(i,j)$, of the workspace from the previously acquired image, $P_{k-1}(i,j)$. The energy of the difference image indicates how much activity has occurred

$$E_k = \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} (P_k(i,j) - P_{k-1}(i,j))^2$$

When the energy rises above a threshold, the difference image is threshold and a bounding box is determined for the region above threshold. A message indicating motion detection, as well as the bounding box, is communicated to the supervisor who then disactivates the motion detection and initiates processes to track and interpret the movement. Tracking continues as long as the confidence remains above a threshold. An alternative detection technique, which has been used by a number of investigators, is to subtract each image from empty background image. This can be made to work when the camera is static.

Tracking using correlation.

Energy normalized cross-correlation tracking can be shown to be optimum in the presence of additive Gaussian noise [8]. The dominant noise in the case of digital desk is changes in viewing position and hand configuration, which is neither Gaussian nor additive. However, when assisted by other detection processes, correlation tracking provides a technique which is inexpensive, relatively reliable, and formally analyzable.

The hand and tool can be modeled as a reference template, acquired from the bounding box when tracking is initiated. The search region can be estimated from the expected speed of the users gestures. The reference template is a small neighborhood, i.e., a window W(m, n) of size Δx, Δy, 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 the energy normalized cross-correlation between the N by N template and the neighborhood of the image whose upper left corner is at (i, j).

$$NCC(i, j) = \sum_{m=0}^{N} \sum_{n=0}^{N} \frac{(P_k(i+m,j+n)\ W(m, n))}{E_P^2(i, j)\ \ E_W^2}$$

Our system allows us to switch between NCC and Sum of Squared Distance (SSD). NCC is more robust to changes in light color, but SSD seems to be more precise. In either case, the estimated position of the template is determined by finding the position at which this measure is a maximum. The peak value of the NCC measure provides a confidence measure of the detection. When this confidence measure drops below a threshold, it is necessary to re-initialize the template. The covariance of the detection can be estimated from the second moment of the correlation value.

Color Histogram for Skin Detection

Simple image processing operations can be used to determine the position of a hand. One such technique is to determine the probability that a pixel is skin based on a luminance-normalized color vector [15]. Such a probability can be determined by a table look-up operation using a multi-dimensional color histogram. Color histogram matching requires only a few instructions per pixel and thus can be applied to an entire image in order to determine where to apply further processing. This technique can be used to find regions which correspond to any distinctive color.

A normalized color histogram h(r, g) based on a sample of N pixels, gives the conditional probability of observing a color vector $\vec{C}$ = (r, g), given that the pixel is an image of skin. The vector (r, g) is obtained by dividing the red and green components by the luminance.

$$p(\vec{C} \mid skin) = \frac{1}{N} h(r, g)$$

What we need to find skin is the conditional probability of skin given the color vector,

$p(skin \mid \vec{C})$. These two probabilities are related by Bayes rule:

$$p(skin \mid \vec{C}) = p(\vec{C} \mid skin) \frac{p(skin)}{p(\vec{C})}$$

The probability of skin, p(skin), is the proportion of all possible images covered by skin. The probability $p(\vec{C})$ is the global probability for all color vectors. These values may be estimated by computing the histogram over the entire image and dividing by the number of pixels. We have found that satisfactory results are obtained by simply approximating this ratio by a constant, K. The constant K is chosen so that the maximum peak in the histogram has a value of 255. This allows us to construct a probability image in which each pixel is replaced by the probability that it is the projection of skin.

An example of hand detection using the same method is presented in figure 2. In this example, two individuals standing in the background result in additional regions of non-zero probability. These regions are in fact the hands of the first background individual and the face of the second. None the less, the bounding box and center of gravity of the largest hand are determined without problem.

Figure 2. A photo of a hand with two people in the background (left), the probability of skin (right), the thresholded probability of skin (bottom).

The confidence in factor is computed using a maximum likelihood estimate based on the parameters of the bounding box. The covariance matrix is determined a priori and assigned as a constant.

5 Recognizing Hand Configurations

Hand configurations can be classification in a space defined by a principal component analysis (PCA) of the distribution of hand images. Introduced by Sirovich and Kirby [16] for the characterization of human faces, the PCA has been successfully employed by Turk and Pentland [17] for face recognition.

Principal components analysis of a population of vectors provides an ordered orthogonal set

of basis vectors for describing the population. The basis vectors are ordered based on the degree of scatter of the population set. Similarity between members in the population can result in a small number of basis vectors for describing the scatter within the population. In this case, the population of vectors can be described in a much smaller linear subspace. Such a space is increasingly used in computer vision for recognition.

This property is commonly used to define recognition algorithms for images by noting that similar images project to similar locations in principal components space. However, a projection to a linear subspace also preserves structure. For example, a sequence of images of a deformable object project to a contour in principle components space. Thus such methods can also be used to define techniques for recognizing temporal sequences.

The calculation of the principal components of the distribution of a sample of hand configurations defines a linear subspace of images (an eigenspace). Each dimension of the eigenspace, or eigenvector, codes variations between hand images from sample set. Images from the sample set can be represented exactly in terms of a linear combination of the eigenvectors. A hand configuration can be represented as a vector of coefficients obtained by inner product of the region containing the hand, $\tilde{H}_n$, with the images which make up the principal components space, φ_n.

$$\vec{\alpha}_n = <\varphi_n, \tilde{H}_n>$$

In order to reduce the dimensionality of the space, the first N principal components images are used. The number of dimensions, N, is chosen such that the sum of the first N principal values attains 95% of the sum of all principal values.

Training Hand Configuration Classes

Using this technique, we create a feature space for recognizing hand configurations. A set of hand configurations are predefined. A number of sample images are made for each hand configuration. These images are run through the detection and tracking process so as to obtain normalized images of a standard size and zero mean, $\tilde{H}_n$. The normalized hand images for each class, k, are projected into principal components space, but inner product with the N principal component vectors $\vec{\alpha}_{n,k}$. The mean, μ_k, and covariance, $\mathbf{C}_k$, is computed from the training images for each class, k.

Given an observed, normalized image of a hand, the configurations is determined as the class, m, for which the projection to principal components space, $\vec{\alpha}$, has the highest probability, as computed using a Gaussian Density Function.

$$p(\text{Class m} \mid \vec{\alpha}) = \frac{1}{\sqrt{2\pi}\det(C)} e^{-\frac{1}{2}(\vec{\alpha}-\mu_m)^T \mathbf{C}_m^{-1}(\vec{\alpha}-\mu_m)}$$

Recognizing Gestures as sequences of configurations

A gesture can be defined as a sequence of configuration classes. The gesture recognition process consists in testing if each normalized hand image is part of the pre-stored classes. The processes is modeled as a set of finite state machines. Each known gesture is represented as a finite state machine. States of the machine correspond to hand configurations or "transition states". Transition states represent unexpected configurations obtained during dynamic gesture. Unexpected postures for example includes postures with half--opened fingers. Transitions between states are done by new posture classes. In order to deal with slow gestures or pauses during gestures, self-transitions are allowed.

In order to recognize several gestures at the same time, all finite state machines are updated with each new posture. A gesture recognized as soon as its finite sate machine arrives the final state.

6 Conclusions

Images of hands manipulating tools can be interpreted in real time using a purely appearance based approach. We have described a system in which a hand is detected and tracked using multiple simple detection processes and a zeroth order recursive estimator. A hand configuration is recognized by projecting the normalized hand image to a principal components space. Each class of configuration is represented by a mean and covariance of the vectors in this space. A gesture is recognized as a sequence of recognized configurations.

References

[1] Baudel, T. Beaudouin-Lafon, M., “Charade: Remote Control of Objects Using Free-Hand Gestures”, Communications of the ACM, Vol.36 No.7, pp. 28-35.

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

[3] K. Brammer, G. Siffling, Kalman Bucy Filters, Artech House Inc., Norwood MA, USA, 1989.

[4] Cadoz, C., Les réalités virtuelles, Dominos, Flammarion, 1994.

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

[6] Crowley J. L., P. Stelmaszyk, T. Skordas et P. Puget, "Measurement and Integration of 3-D Structures By Tracking Edge Lines", International Journal of Computer Vision, July 1992.

[7] Crowley, J. L. and Christensen, H. Vision as Process, Springer Verlag, Heidelberg, 1994.

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

[9] J. L. Crowley and F. Berard, "Multi-Modal Tracking of Faces for Video Communications", IEEE Conference on Computer Vision and Pattern Recognition, CVPR '97, St. Juan, Puerto Rico, June 1997.

[10] S. D. Jones, Claus Anderssen et J. L. Crowley, "Appearance Based Processes for Visual Navigation", IROS '97, IEEE International Conference on Intelligent Robots and Systems, Grenoble, Sept. 1997.

[11] Krueger, M. , “Artifical Reality II”, Addison Wesley, 1991.

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

[13] Ramstein, C., “The Pantograph: A Large Workspace Haptic Device for Multimodal Human Computer Interaction”, CHI’94 Interactive Experience, Adjunct Proceedings, pp. 57-58, 1994.

[14] Rehg J. M. and Kanade, T. “DigitEyes : Vision-Based Human Hand Tracking”. Carnegie Mellon University Technical Report CMU-CS-93-220, 1993.

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

[16] I. Sirovich and M. Krby, "Low dimensional procedure for the characterization of human faces", Journal of the Optical Society of America, Vol 4, No. 3, pp 519-524, March 1987.

[17] Turk, M. and Pentland A. , “Eigenfaces for Recognition” Journal of Cognitive

Neuroscience, 3(1):71-86, 1991.

[18] Wellner, P. Mackay, W. and Gold, R. ,“Computer-augmented environments : back to the real world”. Special Issue of Communications of the ACM, Vol.36 No.7., 1993.