An Appearance–Based Approach to Gesture–Recognition

Jérôme Martin and James L. Crowley

GRAVIR - IMAG
46 av. Félix Viallet
38031 Grenoble, FRANCE
email: Jerome.Martin@imag.fr

Abstract. This paper describes techniques for the design of a system able to interact with the user by visual recognition of hand gestures. The system is composed of three modules including tracking, posture classification and gesture recognition. A description of each module is given. In order to increase the robustness and the precision of the tracking, several complementary tracking processes are coupled. A classification process is presented for recognizing hand posture using distance in an eigenspace. The classification of hand posture leads to the gesture recognition by a set of finite state machines.

1 Introduction

Classical computer input devices are limited to keyboard and mouse. There is a growing interest in developing more intuitive and natural [10] interaction modes between user and computer.

Hand gesture is a potentially rich communication channel between user and computer [2]. Symbolic information can be communicated using a gesture language [1]. Gesture is often included in Multi-Modale interfaces, which combine several channels of communication [8]. Virtual reality environments [11] employ gesture in the manipulation of virtual objects in virtual worlds. Augmented reality systems [21] merges the physical world and a virtual world including documents and functions (e.g. spell checkers). Several operative gestural language recognizers have been constructed to recognize sign language [4,9].

The potential power of gesture has been demonstrated in systems which use data–gloves. For example, the system of Takahashi and Kishimo provides reliable recognition of Japanese Sign Language [19]. Unfortunately, data–gloves restrict movements by tying a user to the compute via a cable. Such systems have also proven to be fragile and bulky. Our goal is to replace the data–glove with "hands-free" gesture recognition using computer vision.

The continued exponential growth in available computing power has brought has reached the point where it is now possible to build real time systems for acquisition and interpretation of images at reasonable costs. In addition, the multi–media "revolution" has made image acquisition equipment a standard

component in personal computers. A powerful processor with adequate memory and a fast image acquisition board provides exactly the support required for real time computer vision. The required hardware support is now commonly available. Our task is to provide the appropriate techniques.

The topic of this paper is to describe initial work on a system for hand gesture recognition based on computer vision. Our system is inspired by the *Xerox DigitalDesk* [21] and *Charade* [1]. The goals of our system is to provide methods to use visual processes to interpret human actions.

2 Appearance–Based Approach

Classic computer vision suggests constructing a 3D model of the human hand and then matching the resulting structure to an a priori model. This is the approach used in the *DigitEyes* system [15].

Maintaining a 3D model of hand configuration can involves modeling as many as 27 degrees of freedom [15,4]. In addition, visual aspects of hands from a 2D image have to be compared with the 3D model. The comparison can be computed by extracting features (such as lines) from the image and by matching the features with the model [15]. A second approach generates candidates from the model and evaluates its matching confidence with the image [17]. In both approaches, the matching between 2D image and 3D model is computationally expensive, is often unreliable. We propose an alternative approach based on the appearance of the hand, without estimating geometric or kinetic features.

3 System Description

The following provides an overview of our system, followed by a more detailed description of the components.

3.1 System Overview

The input of the system consist of a 2D color image of the workspace such as in Wellner's *DigitalDesk* [21] (Fig. 1).

The system is divided in three modules: a hand tracker, a hand posture classifier and a dynamic gesture recognizer.

1. The Hand Tracker determines hand position in images. Several vision processes are combined in order to increase precision and robustness.
2. The Hand Posture Classifier determines the class of the posture represented in the hand image. Hand Posture includes hand point of view and geometric fingers configuration. This process uses a feature space and a distance to such space.
3. The Dynamic Gesture Recognition system starts from posture classification to determine what was the gesture during time. This process is based on a set finite state machines.

The following describes these three modules with more details.

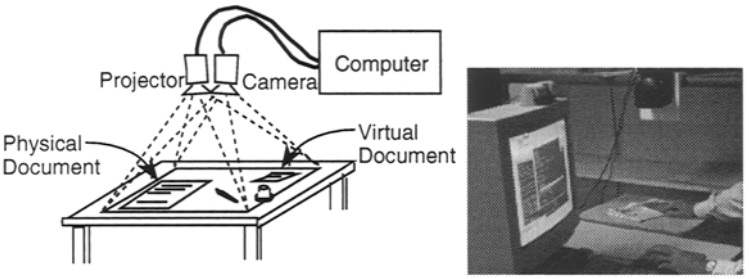

Fig. 1. Workspace of our system.

3.2 Hand Tracker

Tracking is a recursive process of hand detection employing a priori knowledge of the environment. The result of tracking is the position of a hand. Several techniques can be used to detect position: image differencing, skin color detection, cross–correlation, active contours [3,13] and point distribution models [5]. These techniques are complementary, with different failure/success conditions and different requirements. By combining several techniques where each process initiates and/or controls others, the system becomes more flexible, precise and robust [6].

A short description and a discussion of our techniques follows:

Image Differencing Image differencing is a simple method to determine moving object in front of a static background. The difference in luminance of pixels from two successive images is close to zero for pixels of the background. By choosing and maintaining an appropriate threshold, moving objects can be detected within a static scene. The search for connected regions of pixels above threshold provides a bounding box enclosing the moving hand.

In a simplified variation of this technique, an image of the background–scene without the hand can be stored. However, this simplification requires to update the background image each time an object has been added to the scene.

Skin Detection Skin color provides a relatively simple method to determine the probability that a pixel is an image of a hand. The technique is based on a luminance–normalized color vector [16]. A 2–D histogram $h(r, g)$ of luminance normalized color values of skin is constructed based on a sample of N pixels. The histogram provides the conditional probability of observing a color vector $C = (r, g)$ given that the pixel correspond to skin:

$$p(\boldsymbol{C}|skin) = \frac{1}{N}.h(r, g)$$

Using Bayes rule, the conditional probability of skin given the color vector becomes:

$$p(skin|\boldsymbol{C}) = p(\boldsymbol{C}|skin)\frac{p(skin)}{p(\boldsymbol{C})}$$

The probability of skin color, $p(skin)$, can be establish by the proportion of skin color in all images. The probability $p(\boldsymbol{C})$ is the global probability for all color vectors. It is estimated by computing the histogram over the entire image and by normalizing by the number of pixels.

Cross-Correlation The appearance of a hand can be modeled as a reference template. The reference template is compared to a neighborhood in the current image using normalized cross–correlation [14]. Correlation requires the definition of a reference template. In order to define the reference template, an appropriate hand detection method, e.g. color detection or image differencing, may be used. During tracking, the template is updated each time the correlation peak decreases to a minimum value [3].

Point Distribution Model As a last method, the appearance of a hand can be model as a contour. In the "Point Distribution Model" [5], the contour X of a hand posture is defined by:

$$X = (x_0, y_0, \ldots, x_{n-1}, y_{n-1})^T$$

where (x_k, y_k) is the position of a point in the image. Each point is updated to fit the new hand position in the new image. This update is searching the best possible match of pixels using the technique described in [12]. Because the model was trained, X_t must fit a covariance matrix describing the model.

Discussion The techniques presented above have different requirements and failure/success conditions. Image differencing is the fastest techniques but presents the disadvantages of incorporating shadows in the bounding box. Cross-correlation with a reference template can be quite fast but the template must be updated as the hand turns or deforms. Bérard shows in [3] that frequent updates can cause tracking to drift off target and on to the background. The use of color histograms to detect skin requires initialisation of skin color model. The initialization to a specific user typically provides reliable detection. However, the color model must be updated if the ambient illumination changes in spectrum or if another user is to be tracked. A point distribution model prevents the incorporation of shadows, does not depend on skin color but is computationally expensive due to the update of the model. Advantages and disadvantages of the techniques leads to a coordination of visual processes as described by Coutaz, Bérard and Crowley in [6] and by Crowley [7].

3.3 Hand Posture Classification

Hand posture classification can be reliably performed in a space defined by a principal component analysis (PCA) of the distribution of hand images. Introduced by Sirovich and Kirby [18] for the characterization of human faces, the PCA has been successfully employed by Turk and Pentland [20] for face recognition. Before presenting the classification process, a short introduction of the PCA (or eigenspace) method is given.

Eigenspaces of configurations 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, referred to as an "eigenspace" [20] is increasingly used in computer vision for recognition.

An image can be considered as a vector of pixels. Principal components analysis of a set of images determines a small orthogonal set of images which describe the set. Each basis image provides a dimension in the eigenspace.

Similar images tend to have similar projections into an eigenspace. This property is commonly used to define recognition algorithms by noting that similar vectors project to similar locations in eigenspace. Our innovation is to note that projection to a linear subspace also preserves structure. For example, a sequence of images of a deformable object project to a contour in eigenspace. Thus eigenspace methods can also be used to define techniques for recognizing human actions including gestures.

The calculation of the principal components of the distribution of a large sample of hand configurations (fig. 2a) defines a linear subspace of images (an eigenspace). Each dimension of the eigenspace, or eigenvector, codes variations between hand images from initial set. Eigenvectors can be represented as images and resemble "ghosts" of hands (fig. 2b). In mathematical terms, 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 combinations i.e. as a point in the eigenspace. In order to reduce the dimensionality of the eigenspace, a hand may be approximated using only the "best eigenvectors". The best eigenvectors, in the least square error sense, are those associated with the largest degree of scatter of the sample population, as determined by the eigenvalues.

Using this technique, we can create an eigenspace for each hand posture. From figure 2a, six eigenspaces are created: one per line.

Classification using eigenspaces The classification problem consists in finding the appropriate class given a hand image. In our approach, classes are defined

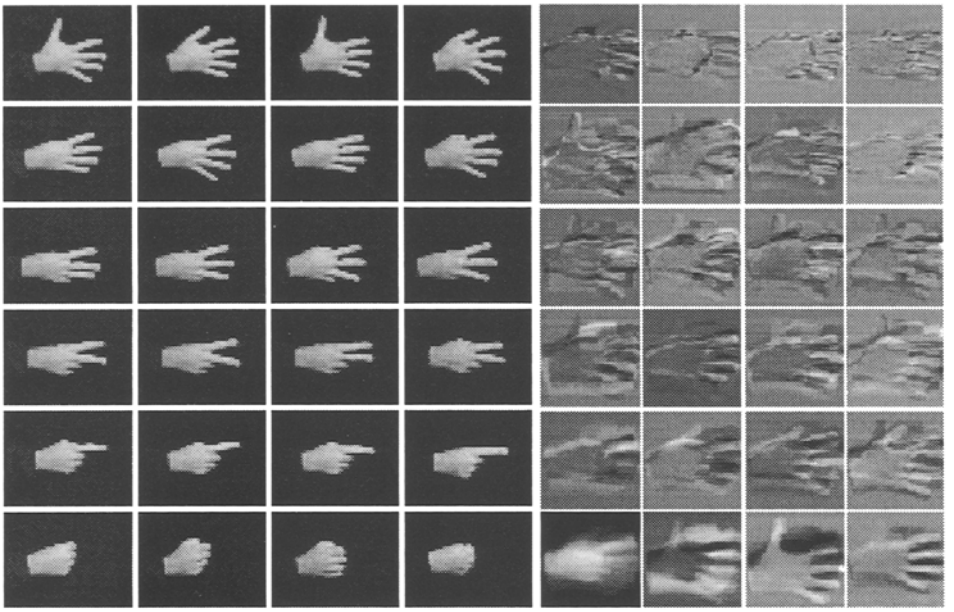

Fig. 2. (a) A sample set of hands configurations, (b) Mean and principal components of the complete sample set

by eigenspaces. In order to classify a hand posture, the "distance from features space" ($dffs$) [20] is used. The distance from features space is defined as the Euclidean distance ϵ_k between an image $\mathcal{I}$ and the image reconstructed after projection in the eigenspace Ω_k. The class $\mathcal{C}_\mathcal{I}$ of an image $\mathcal{I}$ is the class minimizing distance ϵ_k:

$$\mathcal{C}_\mathcal{I} = \mathcal{C}_j \mid \epsilon_j = \min_k \epsilon_k$$

Having a classification of hand posture, the dynamic gesture recognition process can be described.

3.4 Dynamic Gesture recognition

A gesture $\boldsymbol{g}$ can be defined as a sequence of posture classes: $\boldsymbol{g} = \{\mathcal{C}_i\}$ where $\mathcal{C}_i$ is a configuration (or class) as defined in section 3.3. The gesture recognition process consists in testing if $\boldsymbol{g}$ is part of the pre–stored gestures. 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 postures or "transition states". "Transition states" represent unexpected postures 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.

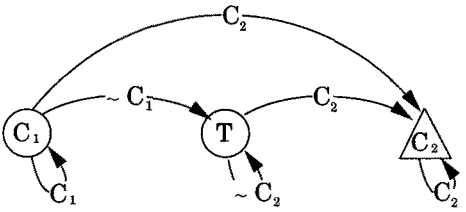

Fig. 3. Finite State Machine

Fig. 4. Images describing the gesture: counting from 1 to 2.

Figure 3 gives an example of a finite state machine recognizing a gesture with two different configuration (such as in fig. 4). $\mathcal{C}_1$ represents posture with one finger and $\mathcal{C}_2$ with two fingers.

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 (triangle in fig. 3).

4 Conclusion

The paper has described a system for the visual recognition of hand gestures. The system is divided in three modules and techniques of each module are presented. In order to make tracking more precise and robust, a combination of complementary techniques is proposed. Hand posture can be classified using distance from eigenspace of hand posture. The recognition of dynamic gestures is enabled by using a set of finite state machines. A finite state machine represents a known gesture.

Major portions of the system have been implemented and show promising results. The tracking processes are currently studied in more detail in order to define a flexible cooperation between processes. Although the complete system has not been tested in real–time, the recognition with stored images is promising. The main problems encountered during recognition are due to hand rotation. We believe that incorporating hand postures with different rotation can overcome this problem

References

1. T. Baudel and M. Beaudouin-Lafon. Charade : Remote control of objects using free-hand gestures. *Communication of the ACM*, 36(7):29–35, July 1993.

2. T. Baudel and A. Braffort. Reconnaissance de gestes de la main en environnement réel. In *Actes de "Informatique'93", Interface des mondes réels et virtuels*, pages 207–216, Montpellier, 1993.
3. F. Bérard. Vision par ordinateur pour la réalité augmentée : Application au bureau numérique. Master's thesis, Université Joseph Fourier - INP Grenoble, June 1994.
4. A. Braffort. *Reconnaissance et Compréhension de gestes, application à la langue des signes*. PhD thesis, Université Paris-XI Orsay, Juin 1996.
5. T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Training models of shape from sets of examples. In David Hogg and Roger Boyle, editors, *British Machine Vision Conference*, pages 9–18. Springer-Verlag, 1992.
6. J. Coutaz, F. Bérard, and J.L. Crowley. Coordination of perceptual process for computer mediated communication. In *Second International Conference on Automatic Face- and Gesture-Recognition*, Killington, Vermont, USA, October 1996.
7. J.L. Crowley and J.M. Bedrune. Integration and control of reactive visual processes. In *European Conference on Computer Vision, (ECCV'94)*, Stockholm, May 1994.
8. P. Dauchy, C. Mignot, and C. Valot. Joint speech and gesture analysis – some experimental results on multimodal interface. Technical Report CRIN 93–R–121, Centre de Recherche en Informatique de Nancy, September 1993.
9. B. Dorner. Hand shape identification and tracking for sign language interpretation. In *'Looking At People' : Recognition and Interpretation of Human Action, 13th International Joint Conference on Artificial Intelligence*, Chambery, France, 1993.
10. J.D. Foley. Interfaces for advanced computing. *Scientific American*, pages 83–90, October 1987.
11. M.W. Krueger. *Artificial Reality II*. Addison-Wesley, 1991.
12. J. Martin. Interprétation de gestes par modèle de distribution de points. Master's thesis, Magistère Université Joseph Fourier, September 1995.
13. J. Martin. Suivi et interprétation de geste : Application de la vision par ordinateur à l'interaction homme–machine. Master's thesis, DEA Université Joseph Fourier – Institut National Polytechnique de Grenoble, 1995.
14. J. Martin and J.L. Crowley. Comparison of correlation techniques. In U. Rembold et al., editor, *Intelligent Autonomous Systems – IAS–4*, pages 86–93, Karlsruhe, Germany, March 27–30 1995.
15. J.M. Rehg and T. Kanade. Digiteyes:vision-based human hand tracking. Technical Report CMU-CS-93-220, Carnegi Mellon University, 1993.
16. B. Schiele and A. Waibel. Estimation of the head orientation based on a face–color–intensifier. In *3rd International Symposium on Intelligent Robotic Systems '95*, 10–14 July 1995.
17. N. Shimada, Y. Shirai, and Y. Kuno. Hand gestures recognition using computer vision based on model mathcing method. In *Symbiosis of Human and Artifact*, 1995.
18. I. Sirovich and M. Kirby. Low-dimensional procedure for the caracterization of human faces. *Journal of Optical Society of America A*, 4(3):519–524, March 1987.
19. T. Takahashi and F. Kishino. Hand gesture coding on experiments using a hand gesture interface device. *SIGCHI Bulletin*, 23(2):67–73, April 1991.
20. M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Neuroscience*, 3(1):71–86, 1991.
21. P. Wellner. The digitaldesk calculator : Tactile manipulation on a desktop. In *ACM Symposium on User Interface Software and Terchnology*, pages 27–33, November 1991.