MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Orientation Histograms for Hand Gesture Recognition

William T. Freeman, Michal Roth

TR94-03   December 1994

Abstract

We present a method to recognize hand gestures, based on a pattern recognition technique developed by McConnell [16] employing histograms of local orientation. We use the orientation histogram as a feature vector for gesture classification and interpolation. For moving or dynamic gestures, the histogram of the spatio-temporal gradients of image intensity form the analogous feature vector and may be useful for dynamic gesture recognition.

IEEE Intl. Wkshp. on Automatic Face and Gesture Recognition, Zurich, June, 1995

1. First printing, TR94-03, May 1994
2. Second printing, TR94-03a, December 1994

Orientation Histograms for Hand Gesture Recognition

William T. Freeman and Michal Roth
Mitsubishi Electric Research Labs
201 Broadway
Cambridge, MA 02139 USA
e-mail: {freeman, roth}@merl.com

From: IEEE Intl. Wkshp. on Automatic Face and Gesture Recognition, Zurich, June, 1995.

Abstract

We present a method to recognize hand gestures, based on a pattern recognition technique developed by McConnell [16] employing histograms of local orientation. We use the orientation histogram as a feature vector for gesture classfication and interpolation.

This method is simple and fast to compute, and offers some robustness to scene illumination changes. We have implemented a real-time version, which can distinguish a small vocabulary of about 10 different hand gestures. All the computation occurs on a workstation; special hardware is used only to digitize the image. A user can operate a computer graphic crane under hand gesture control, or play a game. We discuss limitations of this method.

For moving or "dynamic gestures", the histogram of the spatio-temporal gradients of image intensity form the analogous feature vector and may be useful for dynamic gesture recognition.

1 Introduction

Computer recognition of hand gestures may provide a more natural human-computer interface, allowing people to point, or rotate a CAD model by rotating their hands. Interactive computer games would be enhanced if the computer could understand players' hand gestures. Gesture recognition may even be useful to control household appliances.

We distinguish two categories of gestures: static and dynamic. A static gesture is a particular hand configuration and pose, represented by a single image. A dynamic gesture is a moving gesture, represented by a sequence of images. We focus on the recognition of static gestures, although our method generalizes in a natural way to dynamic gestures.

For the broadest possible application, a gesture recognition algorithm should be fast to compute. Here, we apply a simple pattern recognition method to hand gesture recogntion, resulting in a fast, useable hand gesture recognition system.

1.1 Related Work

The trackball, the joystick, and the mouse are extemely successful devices for hand-based computer input. Yet all require that the user hold some hardware, which can be awkward. Furthermore, none accomodates the richness of expression of a hand gesture.

Devices such as the Dataglove [1] can be worn which sense hand and finger positions [9]. While this captures the richness of a hand's gesture, it requires the special glove. We seek a visually based method which will be free of gloves and wires.

Relying on visual markings on the hands, previous researchers have recognized sign language and pointing gestures [24, 5, 8]. However, these methods require the placement of markers on the hands.

The marking-free systems of [12, 21] can recognize specific finger or pointing events, but not general gestures. Employing special hardware or off–line learning, several researchers have developed successful systems to recognize general hand gestures [22, 14, 6, 7, 20]. Blake and Isard [4] have developed a fast contour-based tracker which they applied to hands, but the discrimination of different hand poses is limited. The real-time hand gesture recognition systems we are aware of require special hardware or lengthy training analysis.

2 Our Approach

We seek a simple and fast algorithm, which works in real-time on a workstation. We want the recognition to be relatively robust to changes in lighting.

A high level approach might employ models of the hand, fingers, joints, and perhaps fit such a model to the visual data. This approach offers power and robustness, but at the expense of speed.

A low-level approach, such as was taken by [6], would process data at a level not much higher than that of pixel intensities. This approach would not have the power to make inferences about occluded data. However, it could be simple and fast. We chose this approach.

We use a pattern recognition system of the form of Fig. (1). Some transformation, T, converts an image or image sequence into a feature vector, which we then compare with the feature vectors of a training set of gestures. We will use a Euclidean distance metric.

We seek the simplest possible transformation T which allows recognition of hand gestures. To motivate the algorithm, let us first examine a transformation which is too simple, and then fix it.

Suppose we did nothing as our transformation, T and used the image itself as our feature vector. We would sum the squares of image pixel differences to measure distances between gestures. Figure 2 illustrates a problem with this scheme, and suggests a solution. (a) and (b) show the same hand gesture under two different lighting conditions, illustrating that pixel intensities can be sensitive to changes of scene lighting. A pixel-by-pixel difference of the images (a) and (b) would show a large distance between these identical gestures. However, others have observed [3] that local orientation measurments are less sensitive

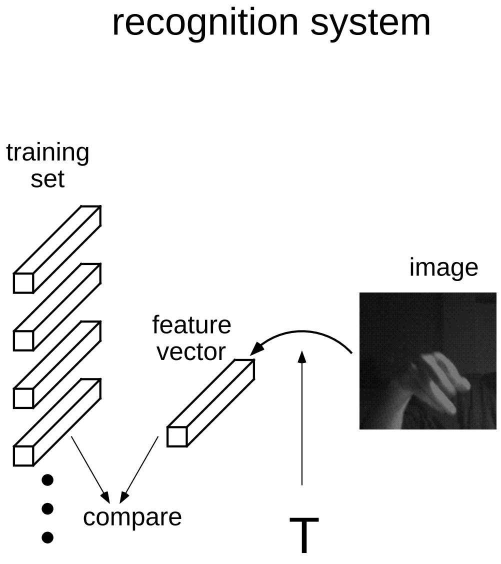

Figure 1: Outline of the recognition system. We apply some transformation T to the image data to form a feature vector which represents that particular gesture. To classify the gesture, we compare the feature fector with the feature vectors from a previously generated training set. For dynamic gesture recognition, the input would be a sequence of images.

to lighting changes. (d) and (e) show the local direction of dominant orientation (as computed in [10]) for the images (a) and (b). In this representation, the two gestures look quite similar.

Next, we need to enforce translation invariance. The same hand at different positions in the image should produce the same feature vector. A simple way to enforce this is to form a histogram of the local orientations. This treats each orientation element the same, independent of location.

With one minor modification, this orientation histogram will be our feature vector to represent hand gestures. The orientation analysis gives robustness to lighting changes; the histogramming gives translational invariance. As we will see, such a histogram can be calculated quickly. This simple method will work if examples of the same gesture map to similar orientation histograms, and different gestures map to substantially different histograms. (We show some cases where this doesn't hold in Fig. 6). McConnell [16] proposed this pattern recognition method, although he used a more complicated histogram comparison scheme than the squared error measure we use here.

Figure 3 shows an example histogram and our additional modification. When analyzing local orientation, we set a contrast threshold below which we assume the orientation measurement is inaccurate. For the image, (a), all points would be below that threshold except for the region near the horizontal line, where the pixels all have the same orientation (b). We then blur the histogram in the angular domain, (c). This allows for a gradual fall-off in the distance between orientation histograms of image features of gradually differing orientation. (d) shows the same data plotted in polar coordinates, which allows convenient comparison with the original image.

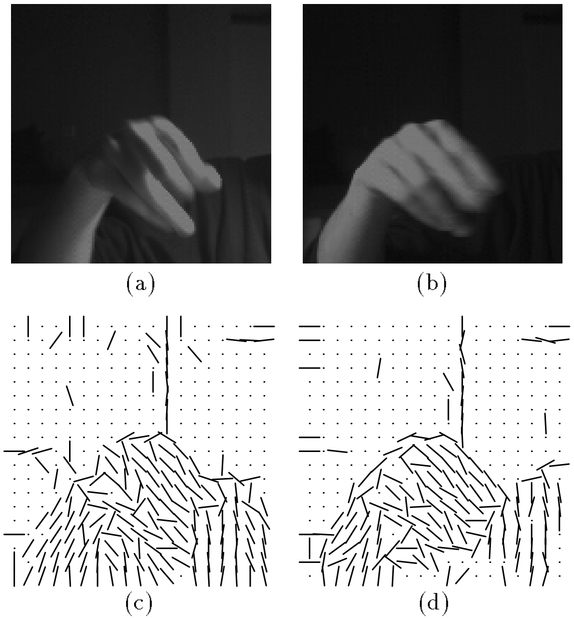

Figure 2: Showing the robustness of local orientation to lighting changes. Pixel intensities are sensitive to lighting change. (a) and (b) show the same hand gesture illuminated under two different lighting conditions. The pixel intensities change significantly as the lighting changes. Maps of local orientation, (c) and (d), are more stable. (The orientation maps were computed using steerable filters [10]. Orientation bars below a contrast threshold are suppressed.)

One has a choice of representing orientation in the angle or twice angle representations [13]. A representation by angle would treat a given edge and a contrast reversed version as having opposite orientation. A double angle representation maps these into the same orientation. A double angle representation would map a hand on dark and light backgrounds to approximately the same feature vector. A (single) angle representation may allow more gestures to be distinguished. For our work, we used an angle representation, to allow more differentiation between gestures.

The resulting algorithm is simple and fast to compute. The representation for each gesture is small, and comparisons between feature vectors can be very fast.

Our approach relates to other examples of image analysis through orientation analysis. In [3], Bichsel analyzed faces, using local orientation to achieve some lighting invariance. Gorkani and Picard [18] used orientation histograms to compute dominant texture orientations. Nelson [17] used orientation patterns for visual homing. This work is also in the same spirit as texture analysis schemes which analyze the outputs of ensembles of oriented filters at differing orientations [2, 15].

We have implemented this algorithm with an HP-735 workstation, and a Raster Ops digitizing board. We took various steps to achieve real-time speed. The 640 x 480 digitized image is averaged and subsampled to a resolution of 106 x 80. We use black and white video. We measure the gradient direction and local contrast using simple two 2-tap x and y derivative filters. With the above steps, the total processing time is 100 msec per frame.

If dx and dy are the outputs of the x and y derivative operators, then the gradient direction is $\arctan(dx, dy)$, and the contrast is $\sqrt{dx^2 + dy^2}$. We divide orientation into 36 bins and use a 1 4 6 4 1 filter to blur the orientation histogram. We set the contrast threshold as some amount, k, times the mean image contrast. Values of k between 1.2 and 2.7 worked well.

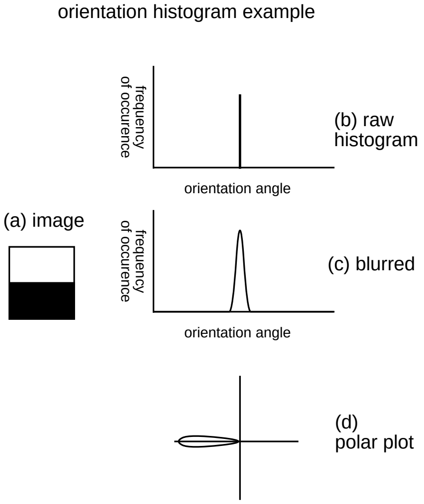

Figure 3: Simple illustration of orientation histogram. An image of a horizontal edge (a) has only one orientation at a sufficiently high contrast. Thus the raw orientation histogram (b) has counts at only one orientation value. To allow neighboring orientations to sense each other, we blur the raw histogram, giving (c). (d) shows the same information, plotted in polar coordinates. We define the orientation to be the direction of the intensity gradient, plus 90°.

3 Operation

Figure 4 illustrates operation. There is first a training phase. The user first indicates the hand positions for the desired vocabulary of gestures, such as the commands for “up”, “down”, “right”, “left” and “stop” in this example, (a). We show only 3 commands in the figure, but typically more are used. The user may show several gesture examples corresponding to a single command. The computer stores the orientation histograms corresponding to each image, (b).

In the run phase, the user repeats the gesture for a desired command (c). The computer forms the orientation histogram of the new image (d), and compares it with each of the orientation histograms from the training phase. In our implementation described here, we selected the command corresponding to the closest feature vector. With this system we have made a computer graphic crane, (g), which the user may control in real-time. (see [23] for a related system, not visually controlled). Rows (e) and (f) show the same gestures made under a different lighting condition. The system can still identify the gestures properly.

Figure 5 shows a measure of performance for the gestures shown in Fig. 4. Each matrix indicates the distance between for the feature vectors from each gesture of the test set and each gesture of the training set. Darker intensities correspond to closer distances. The gestures of the test set made under the same lighting conditions are unambiguously classified by the orientation histogram feature vectors. Even under the different lighting condition of Fig. 4 (e), each gesture is still properly classified, although the discrimination is less clear-cut.

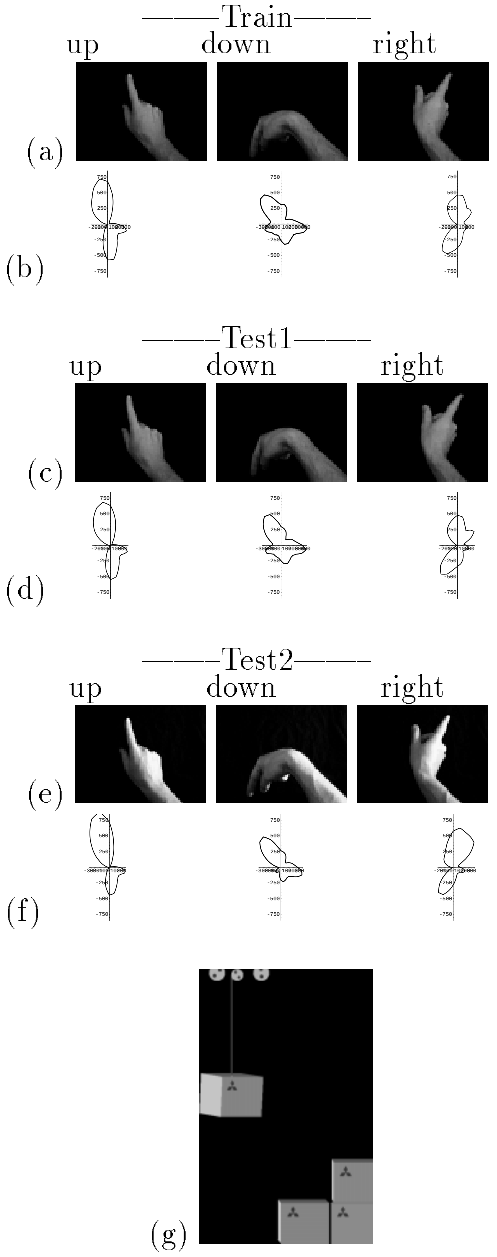

Figure 4: Subset of vocabulary of gestures used to control computer graphic crane. (a) shows the training set of gestures for the commands up, down and right. (c) shows a test set of the same gestures, under the same lighting conditions. (e) is a test set, made under different lighting conditions. (b), (d), and (f) are the corresponding orientation histograms. Note that the shapes look approximately the same as for the same hand positions made under different lighting conditions, (b). An extension of this vocabulary of commands can control in real-time a computer graphic crane, (g).

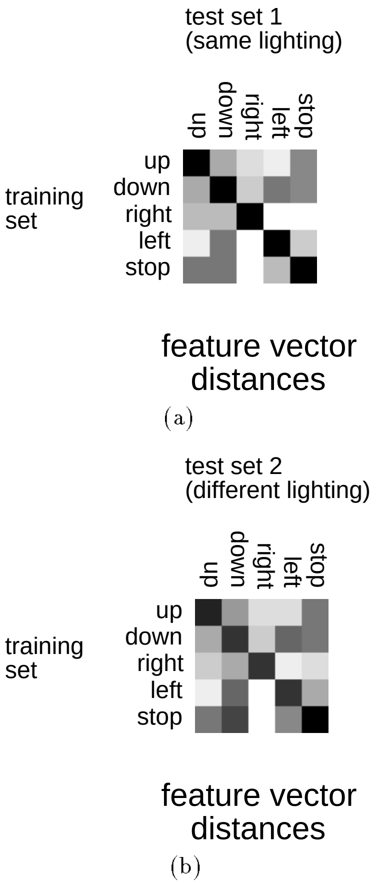

Figure 5: The two matrices display, in grey level, the distances between training and test sets such as those shown in Fig. 4 (black means close). The test gestures, made under the same lighting conditions, are well classified according to the training gestures, as indicated by the diagonal dark line in the matrix of feature vector distances, (a). For the case of test gestures made under different lighting conditions, (b), a nearest neighbor classifier still gives the correct answer, indicating some degree of lighting invariance. Some of the test feature vectors have moved closer to training feature vectors of different categories—the lighting invariance is not perfect.

Our system has two real-time demonstrations of gesture classification: control of the computer graphic crane of Fig. 4, and the game of "scissors/paper/stone", where the computer analyzes the user's hand gesture to decide the winner of each round.

3.1 Other uses

One can interpolate between orientation histogram values. For example, one can train the system at several different hand orientations. Then interpolation, for example, by radial basis functions [19], allows the machine to interpolate arbitrary angles from the user's hand input. We have demonstrated a simple one parameter version of such orientation interpolation [11]

The ideas above can be extended in a straightforward manner to temporal gestures [11]. A natural extension of our 2-dimensional approach is to measure the gradient orientation in space–time caused by image intensities changing over time and space. As before, we can calculate a histogram of the orientation measurments. The resulting two-dimensional vector of orientation frequency is the feature vector for the dynamic gesture.

4 Problem images

From our experience watching many people use the system, we have observed several conditions where the user is not satisfied with the gesture classification. These are illustrated in Fig. 6.

(a) and (b) show two images which many users feel should represent the same gesture. However, their orientation histograms are very different, (c). In the present system, this problem can only be addressed by providing multiple training images for the same gesture.

Some different gestures have very similar orientation histograms. (d) and (e) show an example of this, with the histograms overlaid in (f). One must choose a vocabulary of gestures that avoids such confusable pairs.

The hand must dominate the image for this simple statistical technique to work. (g) and (h) show images where the hand is a small part of the image. Even though user has very different hand positions, the orientation histograms of the two images are very similar, (i). This orientation histogram method is most appropriate for close-ups of the hand.

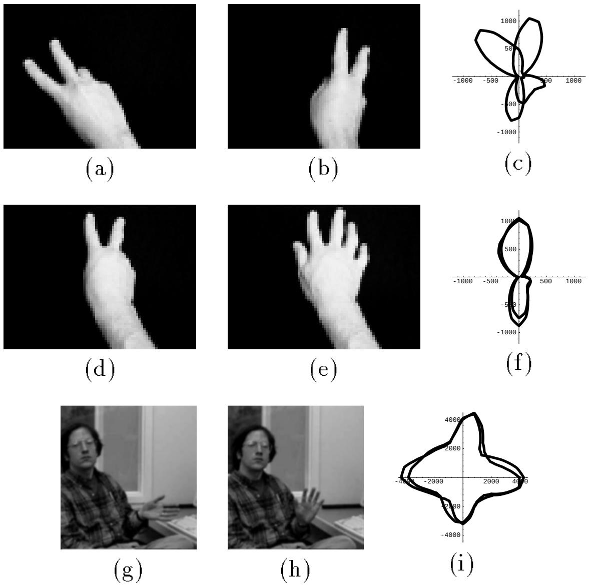

Figure 6: Problem images for the orientation histogram based gesture classifier. Users typically feel that (a) and (b) represent the same gesture, yet their orientation histograms are very different, shown overlaid in (c). A remedy for this problem is to provide training images of the gesture at various orientations. (Mathematical rotation of the feature vector is not sufficient; the corresponding orientation histograms are typically not simple rotations of each other.)
Sometimes small changes in the image can cause large semantic differences, while changing the orientation histograms little. Users classify (d) and (e) as different gestures, yet their orientation histograms are nearly identical, (f). One has to construct a gesture vocabulary which avoids such gestures with similar orientation histograms.
Finally, for this simple statistical technique to work, the hand must dominate the image. If it does not, then even large changes in the hand pose can cause negligible changes to the orientation histogram (g) – (i).

5 Conclusions

We have applied a simple pattern recognition technique to the problem of hand gesture recognition. For static hand gestures, we use the histogram of local orientations as a feature vector for recognition. This method has a training phase and a run phase. In the training phase, the user shows 5 to 15 example hand gesture commands. The computer stores one or more feature vectors, blurred orientation histograms, for each command. In the run phase, the computer compares the feature vector for the present image with those in the training set, and picks the category of the nearest vector, or interpolates between vectors.

The methods are image-based, simple, and fast. We have implemented a real-time version, using an ordinary workstation with no special hardware beyond a video digitizer. The technique works well to identify hand gestures from a training vocabulary of gestures for close-up images of the hand. The real-time system lets the user control a computer graphic crane by hand gestures, monitor hand orientation, and play games such as scissors/paper/stone.

References

[1] Dataglove model 2 operating manual. VPL Research Inc., 1989.

[2] J. R. Bergen and M. S. Landy. Computational modeling of visual texture segregation. In M. S. Landy and J. A. Movshon, editors, *Computational Models of Visual Processing*, chapter 17. MIT Press, Cambridge, MA, 1991.

[3] M. Bichsel. *Strategies of robust object recognition for the automatic identification of human faces*. PhD thesis, ETH Zurich, 1991. #9467.

[4] A. Blake and M. Isard. 3D position, attitude and shape input using video tracking of hands and lips. In *Proceedings of SIGGRAPH 94*, pages 185–192, 1994. In *Computer Graphics*, Annual Conference Series.

[5] R. Cipolla, Y. Okamoto, and Y. Kuno. Robust structure from motion using motion parallax. In *Proc. 4th Intl. Conf. Computer Vision*, pages 374 – 382, Berlin, Germany, 1993. IEEE.

[6] T. J. Darrell and A. P. Pentland. Space-time gestures. In *Proc. IEEE CVPR*, pages 335–340, 1993.

[7] J. Davis and M. Shah. Gesture recognition. Technical Report CS-TR-93-11, University of Central Florida, Orlando, FL 32816, 1993.

[8] B. Dorner. Hand shape identification and tracking for sign language interpretation. In *Looking at people workshop*, Chambery, France, 1993. IJCAI.

[9] S. S. Fels and G. E. Hinton. Glove-talk: a neural network interface between a data-glove and a speech synthesizer. *IEEE Trans. Neural Networks*, 4(1):2–7, 1992.

[10] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Pat. Anal. Mach. Intell.*, 13(9):891–906, September 1991.

[11] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. Technical Report 94-03, Mitsubishi Electric Research Labs., 201 Broadway, Cambridge, MA 02139, 1994.

[12] M. Fukumoto, K. Mase, and Y. Suenaga. Real-time detection of pointing actions for a glove-free interface. In *Workshop on Machine Vision Applications*, Tokyo, 1992. IAPR.

[13] G. H. Granlund. In search of a general picture processing operator. *Comp. Graphics, Image Proc.*, 8:155–173, 1978.

[14] K. Ishibuchi, H. Takemura, and F. Kishino. Real time hand shape recognition using pipeline image processor. In *IEEE Intl. Workshop on robot and human communication*, pages 111–116. IEEE, 1992.

[15] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *J. Opt. Soc. Am. A*, 7:923–931, 1990.

[16] R. K. McConnell. Method of and apparatus for pattern recognition. U. S. Patent No. 4,567,610, Jan. 1986.

[17] R. C. Nelson. Visual homing using an associative memory. In *Proceedings of the DARPA Image Understanding Workshop*, pages 245–262, 1989.

[18] R. W. Picard and M. Gorkani. Finding perceptually dominant orientations in natural textures. To appear in Spatial Vision, special Julesz birthday issue. also available as Percep. Comp. TR #229, M.I.T. Media Laboratory, 1993.

[19] T. Poggio and F. Girosi. Networks for approximation and learning. *Proc. IEEE*, 78(9):1481–1497, 1990.

[20] J. M. Rehg and T. Kanade. Digiteyes: vision-based human hand tracking. Technical Report CMU-CS-93-220, Carnegie Mellon School of Computer Science, Pittsburgh, PA 15213, 1993.

[21] D. Rubine and P. McAvinney. Programmable finger-tracking instrument controllers. *Computer Music Journal*, 14(1):26–41, 1990.

[22] J. Segen. Gest: a learning computer vision system that recognizes gestures. In *Machine Learning IV*. Morgan Kauffman, 1992. edited by Michalski et. al.

[23] D. J. Sturman. *Whole hand input*. PhD thesis, Massachusetts Institute of Technology, 1992. MIT Media Lab.

[24] V. C. Tartter and K. C. Knowlton. Perception of sign language from an array of 27 moving spots. *Nature*, (239):676–678, Feb. 19, 1981.