Low-Cost Real-Time Gesture Recognition

Brian C. Lovell
Intelligent Real-Time Imaging and Sensing (IRIS) Group
School of Information Technology and Electrical Engineering,
The University of Queensland, Brisbane, Australia, 4072
lovell@csee.uq.edu.au

Daniel Heckenberg
c/ School of Information Technology and Electrical Engineering,
The University of Queensland, Brisbane, Australia, 4072
D.Heckenberg@lake.com.au

Abstract

A major impediment to developing real-time computer vision systems has been the computational power and level of skill required to process video streams in real-time. This has meant that many researchers have either analysed video streams off-line or used expensive dedicated hardware acceleration techniques. Recent software and hardware developments have greatly eased the development burden of real-time image analysis leading to the development of portable systems using cheap PC hardware and software exploiting the Multimedia Extension (MMX) instruction set of the Intel Pentium chip. This paper describes the implementation of a computationally efficient computer vision system for recognizing hand gestures using efficient coding and MMX-acceleration to achieve real-time performance on low cost hardware.

1. Introduction

One of the major impediments to developing real-time computer vision systems has been the computational power and level of skill required to process video streams in real-time. This has meant that many researchers have either analysed video streams off-line or used expensive dedicated hardware acceleration techniques. While valuable, off-line demonstrations tend to be unconvincing as they are typically performed on just a few sequences. Hardware accelerated real-time systems suffer from high cost, lack of portability, and high maintenance due to the enormously rapid development in image processing hardware platforms. Recent software and hardware developments have greatly eased the development burden of real-time image analysis leading to the development of portable systems using cheap PC hardware and software exploiting the Multimedia Extension (MMX) instruction set of the Intel Pentium chip.

Notable software packages to accelerate the development of MMX-accelerated computer vision applications have been released by Intel (`www.intel.com`) and Microsoft (`www.microsoft.com`). Intel has released the MMX-accelerated Image Processing and Open Computer Vision Libraries to the public domain. Microsoft has released the DirectX Software Development Kit which provides the necessary drivers for a large number of video capture devices including universal serial bus and Firewire (IEEE-1394) connected cameras and streaming video formats.

This paper describes the implementation of a system using efficient C coding and MMX-acceleration to achieve real-time performance on low cost hardware. The system is a computationally efficient computer vision system for recognizing hand gestures. The system is intended to replace the mouse interface on a standard personal computer to control application software in a more intuitive manner. The system tracks hand motion at 30 fps on a standard PC.

2. Background

Humans are highly literate in gestural communication. Every interaction with the physical world involves some form of physical manipulation which may be considered as a gesture. MIME (MIME is Manual Expression) is intended to harness the intuitiveness and flexibility of gesture through passive video acquisition to produce a natural user interface.

In general, the language of gesture can be very complicated as it is conveyed through highly articulated human

hands. To make the language decoding process feasible, the input language may be simplified to provide a less detailed representation of gesture. MIME is intended to create a gesture driven interface with a level of sophistication commensurate with the current processing power and hardware of a personal computer to provide an extremely low-cost computer-vision gestural interface.

The enormous potential for sophisticated and natural human computer interaction using gesture has motivated work as long ago as 1980 with systems such as Bolt's seminal "Put-That-There" [1]. Whilst "Put-That-There" used a dataglove as input, video has been used in more recent systems. Starner and Pentland's video-based American Sign Language recognition system is a worthwhile and impressive achievement [9]. One of the most sophisticated gesture recognition system to date is Rehg and Kanade's DigitEyes [8] which captures the complete three-dimensional configuration of the hand. Until recently, gesture systems have been limited by the expense and computational demands of video acquisition and processing systems. Dataglove approaches are limited to specific users and constrained environments by the physical requirements of the glove, and have therefore had little potential for applications beyond research. Proliferation of low-cost digital video technology has allowed the first applications of video based gesture recognition to reach the public [2].

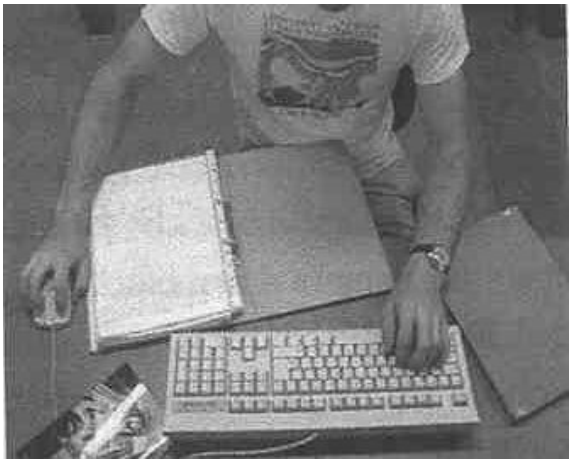

Figure 1. Camera view of cluttered desktop

2.1. Design Principles and Methodology

The guiding principles of the MIME design are computational efficiency and interface usability. Efficiency is demanded so the gesture recognition system can run concurrently with other applications on the host computer. This allows MIME to be used to interact with software performing useful and perhaps sophisticated tasks. Such a goal is not at all trivial as most gesture recognition systems consume all available processing time on fast (and often specially accelerated) computer hardware [6]. The value of MIME as an interface is that it can be used to perform meaningful tasks on the current generation of personal computers using popular low-cost video capture cards (costing about US$150) or even universal serial bus video-conferencing cameras attached to laptop computers.

2.2. A model-based gesture system

Model-based gesture recognition systems employ a mathematical model of the hand and match the parameters of the model to live video images of a hand. A model that can capture the complete range of possible movements is extraordinarily complicated as there are over 30 joints in this region. Moreover, the extraction of all of these parameters is extremely difficult [6]. The output of the MIME system is a feature vector which includes the parameters of a simple hand model (see Figure 2). MIME is essentially a two-dimensional system as it uses a single, uncalibrated camera — once again to reduce cost. The three-dimensional hand is modelled through its two-dimensional appearance in the input video images. A number of assumptions are made about the hand pose in order to unambiguously and efficiently extract the model parameters from each hand image.

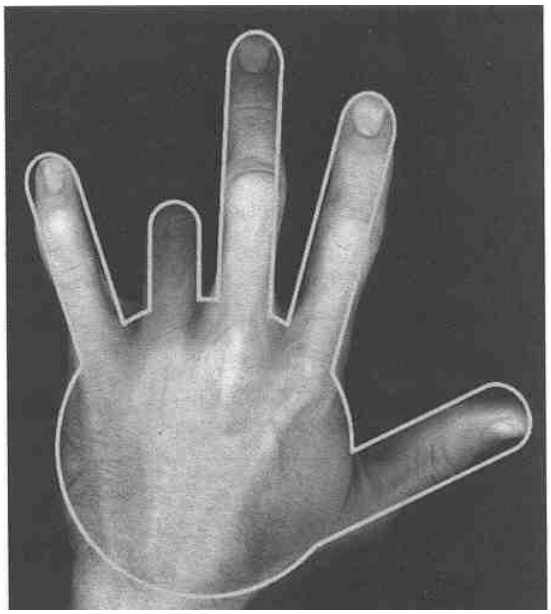

Figure 2. Appearance model

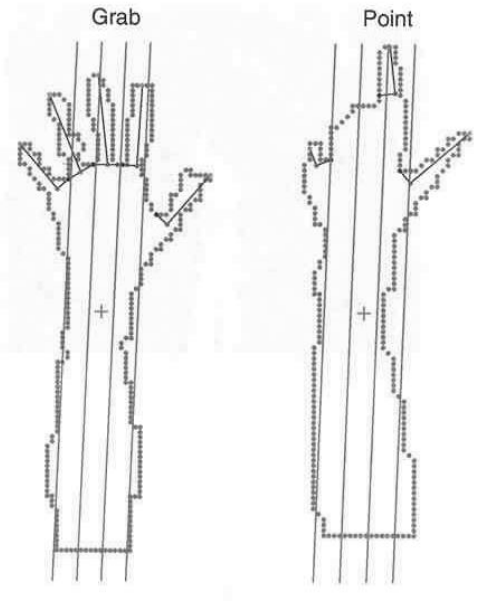

Figure 3. Gestures

2.3. Color histogram based segmentation

The hand region is identified in input images using a discrete probability model of human skin colour and typical image backgrounds. Skin segmentation is made invariant to illumination changes and differing skin tones by using an illumination normalized colour space and a skin colour probability distribution generated from a range of different ethnic groups. MIME uses a technique for skin classification developed by Jones and Rehg [4] which achieves over 90% correct detection of skin.

2.4. Normalized chromatic space

Colour segmentation is performed in a chromatic colour space where RGB components are normalised for variations in intensity as per (1) and (2) using the technique of Jones and Rehg [4].

$$r = \frac{R}{R+B+G} \tag{1}$$

$$g = \frac{G}{R+B+G} \tag{2}$$

Note that the blue component effectively becomes redundant in this representation. For the two-dimensional chromatic colour space with 32 bins per component (the optimal number of bins found in Jones and Rehg[4]) an extremely compact and elegant representation may be used for the precalculated colour classification table which indicates the presence of either skin or non-skin at each pixel. Using just a single bit for each bin result (*ie.*, skin/non-skin), the table can be stored in an array of 32 double words (each of which is 32 bits in length). The double words index one of the colour components and the bits index the other. This efficient representation is exploited in MIME.

After pixels are classified as skin or non-skin, connected regions must be established as hand/arm candidates. This is performed using contour following techniques which test far fewer pixels than scan-line based techniques for objects which are basically convex as discussed in Section 2.6. The only environmental constraint is that the hand does not move in front of another skin coloured area. This constraint precludes hands or arms crossing each other, but does not greatly impede natural gestures.

2.5. Contour based feature extraction

Hand model features (see Figure 2) are extracted from the segmented hand region represented by its boundary contour. All image processing operations are implemented with efficient contour methods. The key features of the contour for feature extraction are fingertips and the valleys between the fingers. Fingertips and valleys are identified as extrema in the curvature of the contour and used to measure the length and angle of each visible finger. Particular fingers are distinguished by the estimated position of the knuckle with respect to the centre of the palm.

2.6. Fast moment calculation through Green's Theorem

Green's theorem states the duality between a double integral over a region and a single contour integral along the boundary of that region. For a piecewise smooth, closed curve C which bounds a region R, Green's theorem may be given by (3) [7], where the integral is taken in the clockwise direction.

$$\oint_{(x,y)\in C} f(x,y)dy = \int\int_{(x,y)\in R} \frac{\delta}{\delta x} f(x,y)dxdy \tag{3}$$

An approximation for discrete regions can be obtained by converting the integrations in (3) to summations as given in (4). An obvious advantage of the contour form is that it reduces computational complexity from $O(N^2)$ to $O(N)$ [5].

$$\sum_{(x,y)\in C} f(x,y)\Delta y = \sum\sum_{(x,y)\in R} \frac{\delta}{\delta x} f(x,y)dxdy \tag{4}$$

Li and Shen [5] give a systolic structure to optimize moment calculations up to an arbitrary order. As MIME uses only low-order moments, the concept is easily implemented

by an ordered set of update equations. The optimized moment calculation scheme simultaneously computes all moments up to the highest order moment required as it traverses the boundary contour.

2.7. Kalman filter tracking

To accelerate hand segmentation and choose the correct skin region when multiple image regions are skin coloured, a two-dimensional Kalman filter (stochastic estimator) is used to track the hand region centroid. Using a model of constant acceleration motion the filter provides an estimate for hand location which guides the image search for the hand.

The Kalman filter tracks the movement of the hand from frame to frame to provide an accurate starting point for a search for a skin colour region. As the hand region may be assumed to have a certain minimum area, a grid of pixel points tested in order of increasing distance from the estimated centroid should find the best matching region. The spacing of the grid is determined by the minimum allowable hand size. Upon finding a skin coloured pixel, the contour following routine is started to trace the connected skin region around the pixel. If the area of the region is below the hand area threshold then the region is discarded and the search is continued with that region excluded from the search grid.

2.8. Fitting the appearance model

The key points in the appearance model are the knuckle and fingertips which define the length and angle of the fingers. Examination of the general shape of the appearance model suggests that the notion of discrete contour curvature may be used to isolate these points. Fingertip and knuckle candidate points can then be used to extract finger lengths and angles which define the appearance model. The tracked hand is specified as either right or left in the system initialization process.

Fingertips appear in the appearance model as extrema in local convex curvature of the boundary contour. The notion of scale allows features specifically of feature size to be identified. This excludes the more rounded shapes of the rest of the hand and arm, and the smaller noisy features in the hand contour. The extent of curvature of convex local extrema is compared to a threshold to determine whether the point is considered a fingertip.

The appearance model defines knuckles as the midpoint on the line that joins the points of intersection of the finger and palm. These points of intersection may be characterized as local extrema of concave curvature, so they are called *valleys*. Again curvature is used to filter features of appropriate scale and the extent of curvature is compared to an empirically determined threshold. Knuckle position may then easily be extracted from the midpoint property.

In some hand poses, the valleys may not be sufficiently convex on both sides of each finger to be correctly identified. This is always the case for the index finger and little finger. In this case, some further knowledge from the appearance model must be employed. Using a geometrical construction shown in Figure 4 where r is set to half the model finger width, one of the two tangents to the circle of radius r may be chosen as the finger line. The tangent is unambiguously chosen by the relative position of the fingertip F and valley V in the contour. In this fashion, the position of the knuckle K can be estimated and hence the length of the finger, l.

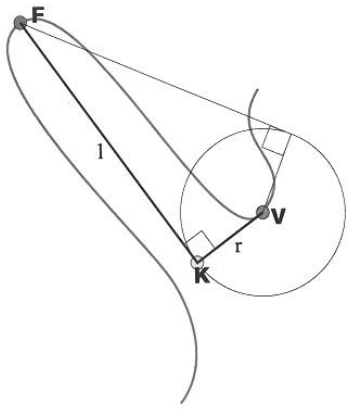

Figure 4. Estimating Knuckle Position

2.9. Generating the pose parameters

Fingertips and valleys must be matched into finger groups before the finger features can be measured. Grouping is done using knowledge of maximum finger lengths and widths, as well as the sequence of points in the contour. The procedure may be summarized as follows:

1. Each fingertip is grouped with valley points on either side of the contour; if such a valley can be found before another fingertip.
2. The finger length is measured for each valley.
3. If the finger length for a valley is greater than the model limit, then discard the valley.
4. If two valleys remain, calculate the finger width as the distance between the valleys.
5. If the width test fails, then the valley further from the fingertip is discarded.

6. if no valleys remain, then discard the fingertip.

This effects of parameter selection on this process are illustrated in Figure 5 and its use for pose recognition is shown in 6.

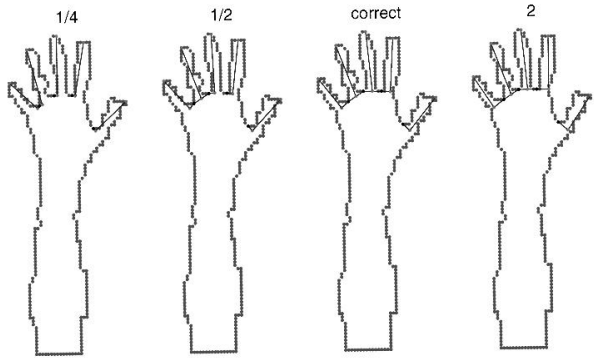

Figure 5. Effects of variation of finger widths value.

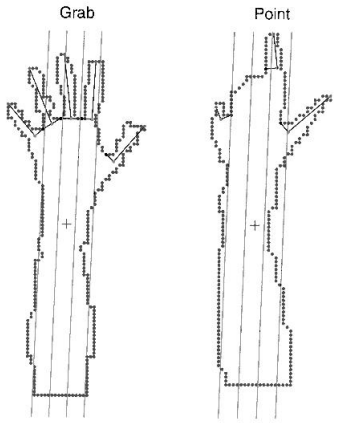

Figure 6. Recognizing grabbing and pointing poses.

2.10. Positioning the camera

MIME is not only intended to operate in the normal settings of a personal computer, but in less constrained environments made possible by replacing physically-based interfaces with the gesture interface. An obvious choice for camera location is the top of the monitor which is often used for video conferencing purposes. A better choice for visibility of the hands, and the position used for MIME, is to mount the camera directly over the desktop.

Taking the desktop setting as the model environment, MIME should operate from a top-down view of the hands against a background that may frequently change or be cluttered as shown in Figure 1. This choice of viewpoint conveniently matches the well-known *desktop metaphor* which is the basis of most modern GUI's.

3. The Full System

In operation, the system reliably tracks a hand with the fingers slightly spread to replace the motion of a computer mouse. Small finger gestures operate the mouse buttons and a closed fist allows the hand to be repositioned in the field of view without moving the mouse pointer. Tracking is smooth and reliable but some standard "mouse acceleration" techniques could be used to improve positional accuracy and to provide a more intuitive interface to the user.

The MIME system was implemented in a real-time computer-vision MATLAB environment and then ported to executable C code, both running under the Windows 98 Operating System. MIME performs full feature extraction in real time on a Pentium III 450MHz personal computer at 30 fps. This rate is achieved using the lowest available input resolution of 160 x 120 pixels which is sufficient for robust tracking.

A video of the MIME system in operation is available from `http://www.csee.uq.edu.au/~iris/ComputerVision/UQ/mime.mpg`.

3.1. Current and Future Development

The MIME system is currently being reimplemented to take advantage of the MMX instruction set of the PC and to be able to transparently support a much larger range of cameras via the Direct X interface. The final system is intended to to handle two-handed gestures in a variety of poses. A drawback of the first implementation of MIME is an overdependence on skin colour. The new system will use several features to detect the hand position including edge and motion cues. To this end an MMX-accelerated object tracking module based on Hausdorff transform image matching has already been completed as shown in figure 7.

The key feature of this tracking technique is an efficient technique for comparing binary feature maps (such as intensity edges from, say, a Canny edge detector). The method uses the generalized Hausdorff measure to locate the transformed model of the tracked object within the present frame. The model is transformed by some affine distortion (translation, rotation, shear and scaling) from one frame to the next. The Hausdorff measure determines the resemblance of one point set to another, by examining the fraction of points in one set that are near points in the other (and perhaps vice versa). Thus there are two parameters in

deciding whether or not two point sets resemble one another — what distance apart must points be to be considered close together, and what fraction of the points are (at most) this close distance away from points of the other set. The Hausdorff distance determines how good a match is required between the model and the image and the Hausdorff fraction determines how much occlusion of the tracked object is permitted. This distance measure differs from correspondence-based techniques such as point matching methods and binary correlation, in that there is no pairing of points in the two sets being compared. The technique is based on the work of Huttenlocher and Rucklidge (1992). [3].

Figure 7. MMX-accelerated Hausdorff tracking showing region of interest and edge maps.

4. Conclusions

MIME achieves robust performance under its set of defining constraints. The segmentation module achieves excellent tracking of a skin region under very few constraints. Feature extraction is effective at low computational cost but imposes a more restrictive set of constraints. Throughout the gesture recognition system, computational efficiency is provided by the compact contour representation and fast contour processing algorithms.

Potential applications of MIME could range from gesture driven presentation software to deliver lecture material to hands-free operation of web-browsers in shopping malls and industrial environments as a replacement for touch-screens.

References

[1] R. A. Bolt. Put-that-there: Voice and gesture at the graphics interface. *Proc. SIGGRAPH80*, 1980.

[2] W. T. Freeman. Hand gesture machine control system. *Proc. SIGGRAPH80*, 1980.

[3] D. P. Huttenlocher and W. J. Rucklidge. Multi-resolution technique for comparing images using the hausdorff distance. Technical Report Technical Report TR92-1321, Cornell University, December 1992.

[4] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. Technical Report Tech. Rep. CRL 98/11, Cambridge Research Laboratory, December 1998.

[5] B.-C. Lin and J. Shen. Fast computation of moment invariants. *Pattern Recognition*, 24(8):807–813, 1991.

[6] V. Pavlovic. Visual interpretation of hand gestures for human computer interaction. *Proc. Human Interaction with Complex Systems*, September 1995.

[7] W. Philips. A new fast algorithm for moment calculation. *Pattern Recognition*, 26(11):1619–1621, 1993.

[8] J. Rehg and T. Kanade. Digiteyes: Vision-based human hand tracking. Technical Report Tech. Rep. CMU-CS93220, CMU, December 1993.

[9] T. Starner and A. Pentland. Real-time american sign language recognition. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 20:1371–1375, December 1998.