A Visual System for Hand Gesture Recognition in Human-Computer Interaction

Matti-Antero Okkonen, Vili Kellokumpu, Matti Pietikäinen,
and Janne Heikkilä

Department of Electrical and Information Engineering, P.O. Box 4500, FI-90014,
University of Oulu, Finland
`okm@ee.oulu.fi`
<http://www.ee.oulu.fi/mvg>

Abstract. Visual hand gestures offer an interesting modality for Human-Computer-Interaction (HCI) applications. Gesture recognition and hand tracking, however, are not trivial tasks and real environments set a lot of challenges to algorithms performing such activities. In this paper, a novel combination of techniques is presented for tracking and recognition of hand gestures in real, cluttered environments. In addition to combining existing techniques, a method for locating a hand and segmenting it from an arm in binary silhouettes and a foreground model for color segmentation is proposed. A single hand is tracked with a single camera and the trajectory information is extracted along with recognition of five different gestures. This information is exploited for replacing the operations of a normal computer mouse. The silhouette of the hand is extracted as a combination of different segmentation methods: An adaptive colour model based segmentation is combined with intensity and chromaticity based background subtraction techniques to achieve robust performance in cluttered scenes. An affine-invariant Fourier-descriptor is derived from the silhouette, which is then classified to a hand shape class with support vector machines (SVM). Gestures are recognized as changes in the hand shape with a finite state machine (FSM).

1 Introduction

Hand tracking and gesture recognition have evolved during the past two decades from cumbersome data gloves and tracking aids to purely visual-based recognition systems. Visual interpretation of hand gestures offers a non-contact HCI-modality, but it has a lot of challenges to be tackled. The human hand is a complex object to be modeled and tracked: It consist of 27 bones and is capable of performing variegated and non-rigid motion. An accurate representation of a hand's 3D orientation requires high dimensional models, which often leads to problems with the real-time requirements of HCI-applications. In addition, complex models often contain irrelevant information from the application point of view. Besides the human hand, the real environments also contain factors that complicates the recognition process, such as background clutter and unideal lighting conditions.

Skin color has a unique chromaticity value, which makes it a popular cue for hand area detection. Parametric approaches usually present the skin colour distribution with Gaussian functions [1,2], whereas non-parametric methods use simple look-up-tables or histograms [3]. The last-mentioned have the ability to present arbitrary distributions, albeit they do not share the compactness of the parametric techniques. Another discriminative characteristic among colour models is their adaptivity. Some methods, like the system of Pantrigo et al., use pre-defined models that remain fixed throughout the processing [4]. These models are prone to fail under environments where lighting conditions differ from the samples that the models are built on. More advanced models adapt themselves to the current conditions and thus specify the skin area more robustly [1]. Another problem with skin area segmentation is the specification of the palm and the fingers from other skin coloured areas. For example, if a user wears a t-shirt, the arm should be somehow separated from the palm. In many studies the users wear long sleeved shirts, as in the work of Zhai et al., when this problem is not addressed at all [3].

In vision-based HCI applications, there are usually two kinds of information extracted from the user: gestural and spatial signals. To replace a mouse, for example, both of these are needed. Some of the earlier approaches extract only pointing information[5,3,1], whereas other approaches analyze only the static hand poses [6]. Also methods that exploit both spatial and gestural information have been developed [4,7,8,9,2]. Fingertips are the most typical choices to be tracked when spatial information is needed. Hardenberg and Bérard also used the number of fingertips to recognize different hand poses [9]. In addition, some techniques track the area of the hand and extract the trajectory of the hand from its bounding box [3,4]. More detailed information about previous work in hand gesture recognition can be found in the references [10,11].

In this paper, a new approach for hand gesture recognition for HCI purposes is presented. The proposed method combines background subtraction techniques and colour segmentation for a robust skin area segmentation. In addition, it distinguishes palm and fingers from the arm and recognizes five different hand gestures. The rest of the paper is organized as follows: An overview of the system is given in section 2, section 3 explains the techniques for hand segmentation and tracking and section 4 presents the gesture recognition process in detail. Section 5 reports the experimental results and finally, discussion and conclusions are given in section 6.

2 Overview

At the top level, the system is controlled by a state machine. In the idle state, the system is using background subtraction to detect an appearing object. The state changes from idle to initialization, when an object bigger than a predefined threshold appears in the input stream. The colour-model and it's constraints are defined in the initialization state. After that, the system starts the actual algorithm. A flowchart of the system architecture is given in Fig. 1(a). In the

Fig. 1. (a) Top level architecture of the system. (b) Overview of the actual algorithm.

processing stage, each frame is analyzed and a possible gesture is recognized. In addition, spatial coordinates are extracted from the hand silhouette. An overview of the process is illustrated in Fig. 1(b).

3 Hand Segmentation and Tracking

Robust hand segmentation in silhouette-based systems is essential, because the rest of the processing is highly dependent on it. In the method presented here, the silhouette of the hand is extracted by combining background subtraction in gray-level space and in Normalized Color Coordinates (NCC) with adaptive color model based segmentation.

3.1 Background Subtraction and Color Segmentation

In the background subtraction method used here, the background image B is composed as an average of the first N images in a sequence, containing only the background. A binary silhouette S_G is obtained simply by subtracting each frame from the background image in gray-level values and thresholding the result. The second silhouette S_{NCC} is calculated on normalized R -channel in the same way as S_G . Now both gray-scale and a chromatic difference between the foreground object and the background have been exploited. A combined silhouette $S_G \cup S_{NCC}$ is used in object detection and colour model initialization in the *idle* and *initialization* stages of the system.

The color-based segmentation is achieved by *histogram backprojection* [12]. For a frame I_t , an r,g -histogram H_t of the foreground object is created using the segmentation result as a mask. For the following frame I_{t+1} , a skin likelihood

image S_{t+1} is built using the H_t : For each pixel in S_{t+1} , a value $H_t(r', g')$ is assigned, where r' and g' are the quantized r, g -values of the corresponding pixel in I_{t+1} . The likelihood image S_{t+1} is then thresholded to a binary image S_{HBP} . To provide smooth transition between frames, the colour histogram H_t is calculated as an average of five previous object histograms. The specificity of the histogram is enhanced by dividing it with an r, g -histogram of the background image. With this operation, the r, g -values that appear often in the background are diminished in the resulting histogram and background pixels are less likely to be classified as foreground.

Soriano et al. used a predefined mask in the r, g -space for selecting the pixels for object histogram [13]. In this paper, another type of constraints are proposed: To ensure that pixels that are used to build the object histogram do not belong to the background, a mask in a $(R, R - G, G - B)$ -colour space is defined with a minimum and maximum value for each channel. Minimum values α_{min}^j are constructed from normalized R -, $R - G$ - and $G - B$ -histograms H_j as

$$\alpha_{min}^j = \beta_{max}^j - C \quad (1)$$

by finding the maximum values β_{max}^j that satisfy the constraints

$$\sum_{k=0}^{\beta_{max}^j} H_j(k) < \gamma, \quad (2)$$

where γ is a coefficient that defines what portion of the histogram is located at bins bigger than β_{max}^j . Coefficient C is an offset value and together with γ it can be used for adjusting the looseness of the limits. The maximum values

$$\alpha_{max}^j = \beta_{min}^j + C \quad (3)$$

are defined in a similar way by finding the minimum values β_{min}^j that satisfy the constraints

$$\sum_{k=0}^{\beta_{min}^j} H_j(k) > 1 - \gamma. \quad (4)$$

In our experiments, we used the values $\gamma = 0.05$ and $C = 7$. Instead of predefined samples, the limits are established during the initialization stage of the algorithm. This assures that the limits correspond particularly to the current environment.

The three binary images, S_G , S_{NCC} and S_{HBP} , are combined as one with a bitwise logical OR operation to produce a robust segmentation of the hand.

3.2 Hand and Arm Segmentation

So far the segmentation method offers a binary silhouette of the foreground object, i.e. the user's hand. Since the interest lays on the palm and the fingers, they have to be segmented from the arm. In this approach the hand area is segmented based on its geometrical features.

It is assumed that there is only the user's hand in the image. In this case, the biggest blob on the foreground object is the palm of the hand. The binary image is convolved with a Gaussian element, when the palm center is positioned at the maximum of the convolution. The operation is illustrated in Fig. 2.

Fig. 2. Example of palm center positioning. (a) A binary hand silhouette (240x320), (b) Gaussian convolution element (100x100) and (c) Convolution of (a) and (b). The crosses in (a) and (c) mark the maximum of the convolution.

The value of the convolution maximum is dependent of the number of pixels valued one in the binary image that fall in the range of the convolution mask. Thus, if the palm is the biggest object in the binary image and the palm is considered to be disk shaped, it's radius can be estimated from the maximum of the convolution. In our approach, we used an empirically inferred polynomial function $R(\hat{I}_C)$ to model the radius as a function of the convolution maximum. Now, when the radius is known, the palm and the fingers can be segmented from hand with a mask that consist of the bounding box of the hand and an ellipse

$$\frac{(y - y_{max})^2}{\epsilon_1 R} + \frac{(x - x_{max})^2}{\epsilon_2 R} = 1, \quad (5)$$

where y_{max} and x_{max} are the coordinates of the convolution maximum and ϵ 's are constants adjusting the axis lengths. In our implementation we used the values $\epsilon_1 = 1.0$ and $\epsilon_2 = 1.1$. The segmentation is illustrated in Fig. 3. The result contains the palm and fingers of the hand, and it is used for both gesture recognition and as a mask for the color model adaptation.

In addition to the segmentation, the convolution maximum also offers a stable key point for controlling a pointer in a HCI-application. Other key points, like fingertips and silhouette's center of gravity, are greatly dependent on the segmentation result and in unideal scenes the trajectories of them may involve excessive noise. The convolution, however, is insensitive to small errors in the silhouette and does not react, for example, to finger movements in the silhouette.

4 Gesture Recognition

The gesture recognition is based on the classified hand shapes. In this approach, the hand shapes are classified with SVM's and the gestures are recognized as changes from a hand shape to another.

Fig. 3. The hand is segmented from the arm with a mask that consists of an ellipse positioned at the center of the palm and the bounding box of the whole object. The dark gray values are excluded from the binary silhouette of the hand.

4.1 Hand Contour Description and Posture Classification

The support vector machine is a learning machine that constructs a hyperplane based on a subset of the training data. These data points are called support vectors. A single SVM classifies data to two classes, but connecting them with some topology a multi-class classifier can be constituted. A more profound introduction is given in [14].

In our method, the extracted hand contour is re-sampled to an evenly spaced sequence with a constant length. An affine-invariant Fourier-descriptor is calculated from the resampled contour, and the pose is classified with SVM's using only the lowest components of the Fourier-descriptor. Along with the affine-invariance, another advantage of the descriptor is that small disturbances in the contour alter only the higher frequency components of the Fourier descriptor and thus do not affect the hand shape recognition. A detailed explanation of the descriptor can be found in [15]. similar approach for human body pose estimation has been used in [16]. In our implementation, we used 200 samples for the contour and the first fourteen components of the Fourier-descriptor (not counting the first one which always scales to unity). The classifier was trained altogether with 233 manually labeled samples from five different hand shape classes to provide a proper amount of within-class variance. All the hand shapes were gathered from a single person. As a kernel function we used the linear type.

4.2 Gesture Recognition with a Finite State Machine

The most common approach for dynamic gesture recognition are the Hidden Markov Models (HMM). They are well suited for time series analysis, such as recognizing hand written letters based on trajectory information. however, in our approach a gesture is simply a change from a hand pose to another with no complicated temporal behavior. Therefore there is no need for advanced time series analysis.

Here, a change of the hand's posture from one to another is interpreted as a gesture. To recognize the changes, a finite state machine (FSM) was used, which consisted of states corresponding to the shape classes and all possible transitions between the states. A transition from a state i to j happens when

the last k poses are classified as j 's. Through this condition, the recognition is invariant to transient shape misclassifications of duration $\tau < k$. To control the mouse in our implementation, each of the states in the FSM defines a state of the mouse, as explained in Fig. 4.

Fig. 4. (a) The mouse cursor is controlled by the palm center and none of the mouse buttons are down. (b) Left button is down. (c) Right button is down. (d) The roll of the mouse is used by extracting only the vertical coordinate of the palm center. (e) Mouse control is disabled.

5 Experiments

The system was implemented on a normal PC with 1 GB of memory and a 3.0 GHz Pentium 4 processor. The implementation processed about 18 frames per second, and the speed could be increased further with optimization. The performance of the system was tested against different backgrounds: Three test sequences were made by different persons, where a hand, positioned like in Fig. 3, moved and changed its pose continuously. The backgrounds of the sequences are given in Fig. 5. The person used for the training of the system did not participate in the test sequences. The lengths of the sequences were approximately 55, 53 and 62 seconds long, containing altogether 131 gestures (i.e. hand shape changes). To the authors knowledge, there is no databases which could be used to evaluate the presented system.

To evaluate the system's performance, false positive and false negative ratios of the segmentation were calculated: For each sequence five randomly selected frames were manually segmented and compared to the result of the algorithm. The result are given in Table 1.

Fig. 5. Backgrounds of the test sequences one, two and three containing different levels of clutter and skin coloured objects

Table 1. False positive and false negative ratios of the segmentation

Measure	Test sequence		
	1	2	3
False positives	2.1%	2.6%	2.3%
False negatives	4.8%	8.7%	11.2%

The results of the gesture recognition are given in the form of a confusion matrix in Table 2, where all the gestures ending in the same hand shape are treated as one to avoid a sparse representation. From the results we can deduce the overall recognition rate of 94.6%, defined as the ratio of the correct detections to all the gestures. In addition, the results gives us a recognition accuracy of 89.2%, defined as the ratio of the correct detections to all the detections.

Table 2. Confusion matrix of the gesture recognition. The rows and columns represent the detected gestures and the ground truths, respectively. The names of the gestures are adopted from Fig. 4, where (x) means any shape.

	(a) ↑ (x)	(b) ↑ (x)	(c) ↑ (x)	(d) ↑ (x)	(e) ↑ (x)	No gesture
(x) → (a)	27					1
(x) → (b)		25				1
(x) → (c)			30	1	1	2
(x) → (d)				24		4
(x) → (e)					18	5
No detection					5

Fig. 6. Example frames and hand segmentations from tests sequence number three. The crosshairs mark the estimated palm centers.

As most of the gesture recognition systems, this approach contains some restrictions concerning the user and the environment. The camera view should be static and comprise only of a single hand with no occlusions. In addition, the hand should not rotate too much about the image coordinate axis and it should be in somewhat parallel with the image plane. As an example, some frames from test sequence number three are given in Fig. 6

6 Conclusions

In this paper, a novel method for real-time hand tracking and gesture recognition in cluttered environments is presented. This approach uses a new way to achieve robust segmentation of a users hand from video sequences. Unlike many algorithms, this method uses a colour-model that auto-initializes and adapts itself during processing. Further, the method uses the hand's geometric properties and affine-invariant features to recognize the hand poses. In our experiments, the system achieved a recognition rate of 95% and an accuracy of 89%, even when the background was cluttered and contained skin coloured objects. Altogether, the method operates both in real-time and real environments, and hence offers potential techniques for HCI-applications. For future research, the system could be possibly enhanced by using an adaptive background model instead of a static one.

References

1. Kurata, T., Okuma, T., Kourogi, M., Sakaue, K.: The hand mouse: Gmm hand-color classification and mean shift tracking. In: Proc. of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS'01), Vancouver, Canada, pp.119–124 (July 2001)
2. Wagner, S., Alefs, B., Picus, C.: Framework for a portable gesture interface. In: FGR '06: 7th International Conference on Automatic Face and Gesture Recognition, pp. 275–280 (2006)
3. Zhai, H., Wu, X., Han, H.: Research of a real-time hand tracking algorithm. In: Proc. of the International Conference on Neural Networks and Brain (ICNN&B'05), Beijing, China, pp. 1233–1235 (October 2005)
4. Pantrigo, J., Montemayor, A., Sanchez, A.: Local search particle filter applied to human-computer interaction. In: Pan, Y., Chen, D.-x., Guo, M., Cao, J., Dongarra, J.J. (eds.) ISPA 2005. LNCS, vol. 3758, pp. 279–284. Springer, Heidelberg (2005)
5. Quek, F., Mysliwiec, T., Zhao, M.: Fingermouse: A freehand computer pointing interface. In: Proc. of Int'l Conf. on Automatic Face and Gesture Recognition, pp. 372–377 (1995)
6. Horimoto, S., Arita, D., Taniguchi, R.-i.: Real-time hand shape recognition for human interface. In: Proc. of the 12th International Conference on Image Analysis and Processing (ICIAP '03), Mantova, Italy, pp. 20–25 (September 2003)
7. Liu, Y., Jia, Y.: A robust hand tracking for gesture-based interaction of wearable computers. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 22–29. Springer, Heidelberg (2004)

8. Malik, S., Laszlo, J.: Visual touchpad: a two-handed gestural input device. In: Proc. of the 6th international conference on Multimodal interfaces (ICMI'04), State College, PA, USA, pp. 289–296 (October 2004)
9. von Hardenberg, C., Bérard, F.: Bare-hand human-computer interaction. In: Proceedings of the 2001 workshop on Perceptive user interfaces (PUI'01), Orlando, FL, USA, pp. 1–8 (November 2001)
10. Pavlovic, V.I., Sharma, R., Huang, T.S.: Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Trans. Pattern Anal. Mach. Intell.* 19(7), 677–695 (1997)
11. Ong, S.C.W., Ranganath, S.: Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Trans. Pattern Anal. Mach. Intell.* 27(6), 873–891 (2005)
12. Swain, M.J., Ballard, D.H.: Color indexing. *Int. J. Comput. Vision* 7(1), 11–32 (1991)
13. Soriano, M., Martinkauppi, B., Huovinen, S., Laaksonen, M.: Adaptive skin color modeling using the skin locus for selecting training pixels. *Pattern Recognition* 36(3), 681–690 (2003)
14. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2(2), 121–167 (1998)
15. Arbter, K., Snyder, W.E., Burhardt, H., Hirzinger, G.: Application of affine-invariant fourier descriptors to recognition of 3-d objects. *IEEE Trans. Pattern Anal. Mach. Intell.* 12(7), 640–647 (1990)
16. Kellokumpu, V., Pietikäinen, M., Heikkilä, J.: Human activity recognition using sequences of postures. In: Proceedings of the IAPR Conference on Machine Vision Applications (MVA 2005), pp. 570–573 (2005)