Hand Shape Recognition for Human-Computer Interaction

Joanna Marnik

Abstract. The paper presents a novel method which allows to communicate with computers by means of hand postures. It is assumed that an input to the method is a binary image of a hand presenting a gesture. A curvature of a hand boundary is analysed in the proposed method. Boundary points which correspond to the boundary parts with specified curvature are used to create a feature vector describing a hand shape. Feature vectors corresponding to shapes which are to be recognised by a system are recorded in a model set. They serve as patterns in a recognition phase. In this phase an analysed shape is compared with all patterns included in the database. A similarity measure, proposed specifically for the method, is used here. One advantage of the method is that it allows to easily add a shape to the recognised shapes set. Moreover, the method can be applied to any shapes, not only hand shapes. The results of the tests carried out on the posture database, which includes 12 664 images of 8 hand shapes, are also presented in the paper.

Keywords: human-computer communication, shape recognition, similarity measure, binary images.

1 Introduction

In recent years, more natural ways of communicating with a computer than a keyboard and a mouse have become more and more important. Multitouch screens can serve this purpose, but they are not in a very wide use yet because of their cost [3].

Joanna Marnik
Rzeszów University of Technology,
Wincentego Pola 2, 35-235 Rzeszów, Poland
e-mail: jmarnik@prz-rzeszow.pl

K.A. Cyran et al. (Eds.): Man-Machine Interactions, AISC 59, pp. 95–102.
springerlink.com

Speech is the most natural way of communicating with others. A speech recognition technology is built in the Windows Vista operating system [9]. Such systems have some limitations concerning other voices and sounds appearing in the environment. Another means of communicating with machines are hand gestures. A technology allowing to control mobile phones or computers in this way has recently been patented by Sumsung [2]. Practical application of hand gestures recognition methods can be found on the Internet, to navigate a website, specifically prepared for this purpose [11].

There are a few approaches to resolving the problem of hand posture recognition. Wilkowski [14] uses a Linear Discriminant Analysis and a statistical maximum likelihood classifier to recognise 9 hand postures with 92.3% accuracy. Features used to build a feature vector in this method do not allow rotations and scale changes of the hand, and they cannot be applied to bigger sets of postures. Moreover, a different classifier must be learnt for each set of the postures under consideration.

Another approach uses morphological 'hit-or-miss' operation to analyse a hand shape and a neural network or a decision tree as a classifier [4, 8]. This approach was applied to recognise 8 hand postures which can be used in human-computer interaction [4] with a recognition ratio of 96.7% for the decision tree and 96.4% for the neural network. In [8], 20 hand shapes occurring in Polish finger alphabet (PFA) were recognised with about 87% accuracy by means of this method. The drawback of the method is that a neural network and a decision tree are also closely related to the set of the recognised postures.

Triesh and Malsburg use the elastic graph matching (EGM) technique to recognise 12 hand shapes [13] with a recognition ratio of 92.9% and 85.8% for the postures performed against simple and complex background, respectively. An advantage of the method is that the background of the hand can be complex. However, the method's computational time is too long for a real-time recognition.

This approach was applied to recognise hand shapes occurring in PFA [6]. Besides Gabor filters, edge information and depth information were also used for the graph nodes description. A recognition ratio of 80.9% was achieved in the experiment presented in the paper [6].

The paper proposes a novel method for recognising hand shape. The method allows to communicate with computers by means of hand postures. A set of recognised hand shapes can be defined by a user of the method. The method is also suitable for any shapes, not only hand shapes.

It is assumed that a binary image of a hand presenting a gesture is the input to the method in question. The binary image can be obtained, for example, by background subtraction [10] or using a skin colour model [7]. A curvature of a hand boundary is then analysed to build a feature vector. Feature vectors corresponding to shapes, which are to be recognised by the system, are recorded in a model set. They serve as the patterns in a recognition phase. In this phase, the analysed shape is compared with all patterns included in the model set. A similarity measure, proposed specifically for the method, is used here. A detailed description of the method is given in

Sect. 2. The tests carried out on the same gesture database as in [4] and their results are presented in Sect. 3. A discussion of the method is given in Sect. 4.

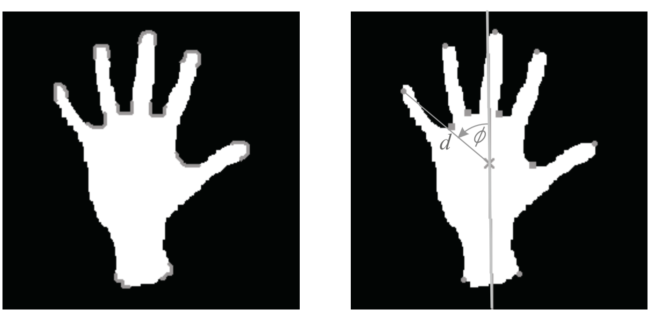
2 Method Description

The proposed method of a hand shape recognition consists of three stages. In the first stage, a set of a high curvature points is determined. On the basis of this set a feature vector is created in the next stage. Feature vectors obtained for sample images of the considered hand shapes are stored in a model set, which is used during a recognition stage.

2.1 High Curvature Points Detection

A binary image of a hand posture is the input to the method. At the beginning, a boundary points sequence is obtained with extended boundary tracing method [12]. The method is fast because it is based on a look-up table. The sequence of points serves as an input to the high curvature points detection algorithm [1]. It is a two-pass algorithm. In the first phase, candidate points are extracted. The points are depicted by the apex of the isosceles triangle of specified size and opening angle inscribed in a boundary. Such an approach allows to easily determine if the point lays on a concave or convex part of the boundary. A second pass is a post-processing step aiming at removing superfluous candidates. The remaining points are the desired high curvature points. There are 3 parameters of the algorithm: $L_{\min}$, $L_{\max}$ – minimal and maximal length of the triangle's both sides, respectively, and α – maximal acceptable opening angle between the sides of the triangle. An example binary image with candidate points obtained after the first pass of the algorithm is shown in Fig. 1a.

High curvature points can be seen in Fig. 1b. Circles correspond to the convex parts of the boundary, squares to the concave parts.



(a) (b)

Fig. 1 **(a)** An example image with candidate points. **(b)** High curvature points and feature vector construction

2.2 Structure of a Feature Vector

A binary image of a gesture and a set of high curvature points obtained for this image are used to build a feature vector. First, a centroid and a slope of the major axis of an object representing a given hand shape are calculated. Taking into account these two values allows to make the method insensitive to rotation and location of the object. The feature vector F consists of triples $f_i = (d, \phi, c)$, where $i = 1, 2, \ldots, n$ are indices of successive high curvature points, d is a normalised distance of the given point from the centroid, ϕ is the value of an angle created by the major axis and a ray coming out from the centroid and going through the point, and c is a convexity of the point. Elements d and ϕ can be treated as a vector $\mathbf{v}$ in a coordinate system originating at the centroid of the object and X axis determined by the major axis of the object. The normalisation of distance values consists in dividing the original value by the greatest of the distances obtained for the image's high curvature points. A feature vector construction is illustrated in the image Fig. 1b.

The feature vectors obtained for sample images of hand shapes, which are to be recognised with their labels are stored in the model database $\mathscr{M}$. This database is used in the recognition stage.

2.3 Recognition

At a recognition stage a similarity measure described below is used to calculate a similarity value of two feature vectors. Because the number of high curvature points can be different for different images, a special measure was proposed for the method.

2.3.1 Similarity Measure

The similarity $S(F_1, F_2)$ of two feature vectors F_1 and F_2 is an arithmetic average of similarities $S(F_1 \to F_2)$ and $S(F_2 \to F_1)$ of the model image to the examined image and the examined image to the model image, respectively:

$$S = \frac{S(F_1 \to F_2) + S(F_2 \to F_1)}{2} . \tag{1}$$

Similarity $S(F_1 \to F_2)$ is an average distance of all high curvature points included in F_1 to the nearest high curvature point included in F_2 with the same convexity and the angle value close to the angle corresponding to the given point from F_1:

$$S(F_1 \to F_2) = \frac{\sum_{i=1}^{n} DIST_{\min}(f_i, F_2)}{n} , \tag{2}$$

where

$$DIST_{\min}(f_i, F) = \begin{cases} |\mathbf{v}(f_i) - \mathbf{v}(f_{\mathrm{jmin}}(F))| & \text{if } |\phi(f_{\mathrm{jmin}}(F)) - \phi(f_i)| \leq \theta \\ d(f_i) & \text{otherwise} \end{cases} \tag{3}$$

and

$$\text{jmin} = \underset{j=1,\ldots,n_F}{\text{argmin}} \left(|\mathbf{v}(f_i) - \mathbf{v}(f_j(F))| : c(f_i) = c(f_j(F)) \right) \tag{4}$$

θ determines a range in which the angles of the vectors $\mathbf{v}(f_i)$ and $\mathbf{v}(f_j(F))$ are recognised as compatible. $d(f_i)$, $\phi(f_i)$ and $c(f_i)$ are the distance, the angle and the convexity of the point related to the triple f_i, respectively. $\mathbf{v}(f_i)$ is a vector determined by the values d and ϕ of the triple f_j, $f_j(F)$ is jth triple of the feature vector F, and n_F is a number of triples in F.

The proposed similarity measure gives the smallest values for more similar shapes.

2.3.2 Recognition stage

During the recognition stage, we calculate similarity values of the feature vector, obtained for the examined image, to each of the feature vectors stored in the model set $\mathscr{M}$. If the smallest of such obtained values is below specified threshold T, it is assumed that the result of the recognition is the shape related to the feature vector for which this value was achieved. Otherwise, the image remains unclassified.

3 Experiments

Experiments were performed on the image database compound of 14 360 binary images of 8 hand shapes presented in Fig. 2.

The resolution of the images was 256×256 pixels. This image database was earlier used in experiments concerning a hand shape recognition, described in [4]. A method using morphological ‘hit-or-miss’ operation was used to generate a feature

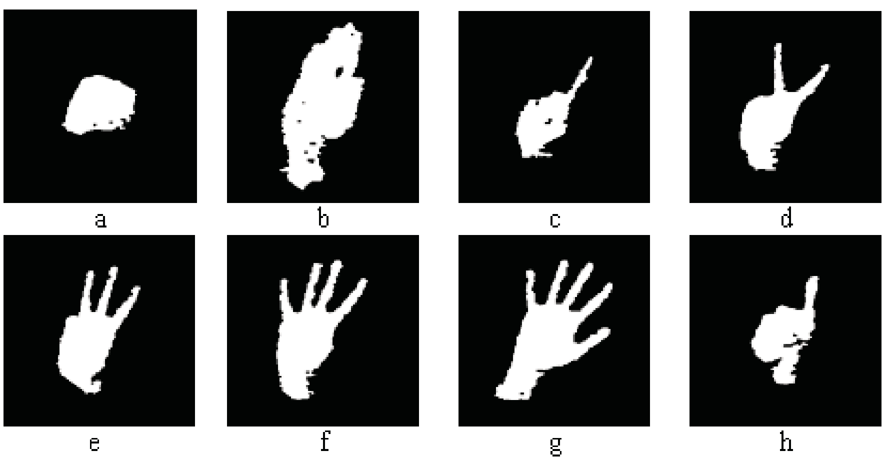


Fig. 2 Hand shapes used in the experiments: **a** wrist, **b** palm, **c**–**g** 1–5 fingers, **h** thumb

Table 1 The number of images of particular hand shapes in the test set

1 *finger*	2 *fingers*	3 *fingers*	4 *fingers*	5 *fingers*	*wrist*	*palm*	*thumb*
1751	1354	1561	1635	1410	1551	1290	2112

vector [8]. To classify a hand shape, an MLP neural network and a decision tree were used.

The image database contains two groups of images: a group of 1696 training images (212 for each of the 8 postures) and a group of 12 664 test images. The number of the test images of individual hand shapes is collected in Table 1.

The set of the model feature vectors was created on the basis of the training images. First, one image for each hand shape was chosen to create an 8-element model set. Next, the feature vectors obtained for each of the remaining training images were classified using the proposed method. The feature vectors for which the classification was faulty were added to the model set. As a result, the set of the model feature vectors contained 10, 7, 9, 11 and 5 elements for the 1–5 fingers respectively, 10 elements for *wrist*, 5 for *palm*, and 12 for *thumb*. All the images were earlier filtered by a morphological CO operation of size 7×7. The parameters of the method were set as follows: $L_{\text{min}} = 15$, $L_{\text{max}} = 40$, $\alpha = \frac{3}{4}\pi$, $\theta = \pi/9$ (see Sect. 2). L_{min} and L_{max} depended on the size of the objects representing the hand. α and θ had default values. The threshold T for the similarity value was set to 0.15.

The experiments were carried out on test images. Parameters values, the same as the ones used during creating a set of the model feature vectors, were used here. To make a comparison with the results presented in [4] possible, the threshold T was neglected. By considering this parameter, we obtain unrecognised gestures, which was not the case during the experiments to which the comparison is to be made. We expect that using this threshold will allow to depict moments of transition between the known postures during a continuous recognition in a video sequence.

Table 2 Results of the recognition obtained for the test images from the database

Correct [%]	1 *finger*	2 *fingers*	3 *fingers*	4 *fingers*	5 *fingers*	*wrist*	*palm*	*thumb*	*sum*
Our method	99.3	97.9	94.8	97.9	99.9	99.7	98.2	97.4	98.1
MLP	97.2	92.2	92.1	95.8	96.9	99.9	98.4	98.4	96.4
Decision tree	94.4	95.0	96.2	97.6	98.8	98.3	97.8	96.1	96.7

The results of our experiment are presented in Table 2. The rows present names of the methods and the percent of correctly classified images. MLP and Decision tree values are taken from [4]. The values obtained for all the test images are given in the last column. For almost all the postures the best results are obtained by means of the proposed method. Only for the postures *3* and *5 fingers* better results were achieved

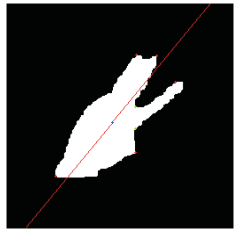


Fig. 3 A test image of *3 fingers* posture classified as *4 fingers*

for 'hit-or-miss' features and the decision tree classifier. Also postures *wrist* and *palm* were only slightly better recognised by the MLP classifier.

The most often misclassified shape was *3 fingers*. It was confused mainly with *2* and *4 fingers*, which are the postures most resembling *3 fingers* posture. This posture was also confused with *palm*. It occurred when the outstretched fingers are close to each other. An example image of *3 fingers* posture classified as *palm* posture is depicted in the Fig. 3.

4 Conclusions

The presented method allows for the recognition of hand shapes and can be used in a human–computer interaction. In the experiments performed on 8 hand postures, high recognition rates were achieved. However, the method can be applied to other sets of postures. Another advantage of the method is that it allows to easily add a posture to the model feature vectors set. Learning a classifier is not necessary here. Only a feature vector corresponding to the new gesture has to be added to that set. It can be done online while using a hand gesture recognition system. It was possible neither in case of the 'hit-or-miss' method [4, 8] nor with the method proposed by Wilkowski [14].

The method is suitable for a real–time recognition. In the experiments (Sect. 3 an average time of the classification of a single frame equalled 27 ms.

We are going to use this method to recognise hand shapes occurring in the Polish finger alphabet. Previous works [8] showed difficulty in recognising some postures from this set. We hope that the proposed method will provide better results, but more research is needed.

The drawback of the method is that it is based on binary images. They must be obtained on the basis of color camera images. Different methods can be used to do that, i.e. background subtraction [10] or a method using a skin colour model [7]. A quality of these images are crucial for the method. All irregularities of the boundary can lead to misclassification. A CO filter was applied to smooth the boundary in the above experiments, but more sophisticated methods can give better results, for example Latecki's Polygon Evolution by Vertex Deletion method [5].

Acknowledgements. The work is financed from Polish funds for science in years 2009–2011 under the research project No. NN516369736.

References

1. Chetverikov, D., Szabo, Z.: A simple and efficient algorithm for detection of high curvature points in planar curves. In: Proceedings of the 23rd Workshop of the Austrian Pattern Recognition Group, pp. 175–184 (1999)
2. Heon, K.S., Gumi-si, K.R., Sul, K.T., Imsil-gun, K.R.: Hand gesture recognition input system and method for a mobile phone, patent 20080089587 (2008), http://www.freepatentsonline.com/20080089587.html
3. Hodges, S., Izadi, S., Butler, A., Rrustemi, A., Buxton, B.: ThinSight: Versatile multi–touch sensing for thin form–factor displays. In: Proceedings of the ACM Symposium on User Interface Software and Technology, pp. 259–268 (2007)
4. Kapuściński, T., Marnik, J., Wysoki, M.: Rozpoznawanie gestów rąk w układzie wizyjnym. Pomiary Automatyka Kontrola 1, 56–59 (2005)
5. Latecki, L.J., Lakämper, R.: Polygon evolution by vertex deletion. In: Nielsen, M., Johansen, P., Olsen, O.F., Weickert, J. (eds.) Proceedings of the International Conference on Scale-Space in Computer Vision. Springer, London (1999)
6. Marnik, J.: The Polish finger alphabet hand postures recognition using elastic graph matching. In: Kurzyński, M., et al. (eds.) Computer Recognition Systems. Advances in Soft Computing, vol. 2, pp. 454–461. Springer, Heidelberg (2007)
7. Marnik, J., Kapuściński, T., Wysoki, M.: Rozpoznawanie skóry ludzkiej na obrazach cyfrowych. In: Materiały 5tej Krajowej Konferencji Naukowo-Technicznej Diagnostyka Procesów Przemysłowych, pp. 279–282. Łagów, Lubuski (2001)
8. Marnik, J., Wysoki, M.: Hand posture recognition using mathematical morphology. Archiwum Informatyki Teoretycznej i Stosowanej 16(4), 279–293 (2004)
9. Microsoft Corporation: Windows speech recognition, http://www.microsoft.com/enable/products/windowsvista/speech.aspx
10. Piccardi, M.: Background subtraction techniques: a review. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, The Hague, The Netherlands (2004)
11. Publics & Halriney, http://www.hrp.com
12. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision. Thomson Engineering, Toronto (2007)
13. Triesch, J., von der Malsburg, C.: A system for person-independent hand posture recognition against complex backgrounds. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(12), 1449–1453 (2002)
14. Wilkowski, A.: An efficient system for continuous hand-posture recognition in video sequences. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J. (eds.) Computational Intelligence: Methods and Applications, pp. 411–422. AOW EXIT, Warsaw (2008)