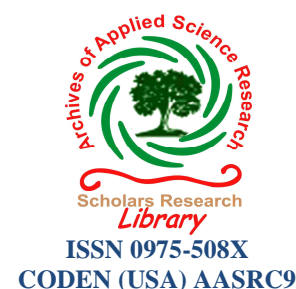




Scholars Research Library

Archives of Applied Science Research, 2013, 5 (3):261-269
(<http://scholarsresearchlibrary.com/archive.html>)



Hand shape based gesture recognition in hardware

Prince Nagar, Ashwani Sengar and Mayank Sharma

Department of Electronics and Communication, School of Engineering & Technology, Sharda University, Greater Noida, India

ABSTRACT

It is possible to recognize and classify ten hand gestures based solely on their shapes. This paper discusses a simple recognition algorithm that uses three shape-based features of a hand to identify what gesture it is conveying. The overall algorithm has three main steps: segmentation, feature calculation, and classification. The algorithm takes an input image of a hand gesture and calculates three features of the image, two based on compactness, and one based on radial distance. The parameters found in the classification step were obtained empirically using 200 hand images. The algorithm was tested on another 200 hand images, and was able to successfully classify 182 images, or with an overall success rate of 91 percent.

Key words: Human hand gesture, pattern recognition, compactness, Euclidean distance, radial distance

INTRODUCTION

The human hand has remained a popular choice for expressing simple ideas and actions. Hand gesture recognition is therefore a popular subject in the computer vision and pattern recognition field. A hand gesture, which is the representation of ideas using unique hand shapes or finger orientation, has the potential to interface with a computer system. Many recognition algorithms have been developed for recognizing various hand gestures, but most algorithms focus either on one complex feature, as in [3],[4], or a combination of numerous features, such as [5], [6], for classification. Alternatively, some algorithms employ the use of neural network and learning schemes that require significant amount of training and sample data present, as in [3], [5], [6], [7]. Advancement in computer technology has also allowed for more and more complex algorithms without the sacrifice in speed and accuracy. Amidst this trend in pattern recognition, this paper aims to demonstrate that three simple shape-based features could be used to classify ten hand gestures with minimal computational cost and complexity.

The proposed method in this paper is closely related to the approaches described in [1] and [2], but the proposed method employs different features and is tested on more images. No training is required, and no complex feature calculations are necessary. The parameters to the recognition algorithm are obtained empirically from analyzing a set of sample images. Additionally, the algorithm introduced here treats the thumb as one of the three features. This feature has not been employed in other recognition algorithms.

The algorithm consists of three main steps: image segmentation and enhancement, feature calculation, and classification. The hand gestures, referred to as patterns here henceforth, are shown on Fig. 1. In the first step, an input image is prepared for use by undergoing various standard image processing techniques. In the second step, the

algorithm calculates three features of the image: compactness of the entire image, compactness of the left half of the hand, and the number of fingers by using radial distance profile. Finally, the algorithm classifies the pattern type based on these three features. An explanation of each step is found in its respective sections.

MATERIALS AND METHODS

A. Segmentation

The focus of this paper is on the recognition algorithm, so traditional segmentation methods are used. Image preparation steps include RGB to binary conversion and morphological operations, as described in our previous work [8], but instead of using a fixed threshold value for segmentation, this time we implemented the Otsu's method, as described in [9]. Color information is not used in the segmentation process. This allows the method to be color-invariant, essentially making this algorithm more robust to varying light conditions. Arm removal is achieved as outlined in [8]. This procedure calculates the distance between the upper and bottom edges of the arm, and cut the arm exactly at the wrist, where the distance between the two edges increases rapidly. The procedure accounts for varying hand and arm sizes by using proportional changes in distances between the upper and lower edges of the arm instead of fixed distance changes. A similar method is also found in [1], and this emphasizes the importance of extracting only the hand area and not the rest of the images. All input images are captured using a low-cost web camera to simulate practical application settings.

B. Feature Calculations

The focus of this paper is on using shape-based features to recognize ten hand patterns. Shape-based features have not been widely used in hand gesture recognition algorithms because the hand can assume many shapes. To overcome this problem, three shape-based features are used in combination. These features are described in the following sections.

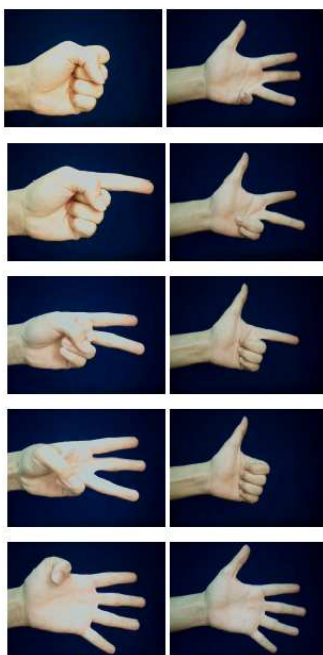


Fig. 1. A sample set of hand gesture patterns used. From top left going counter-clockwise, pattern Zero to pattern Nine

1) *Compactness I* (C_A): Compactness is a shape-based descriptor and the first feature of the image to be computed. The compactness of a shape is found by using the following Eq. 1.

$$\text{Compactness} = \frac{\text{Perimeter}^2}{4\pi \times \text{Area}} \quad (1)$$

From the equation it is clear that compactness is the ratio of the squared perimeter of the shape to its area. This also means that two patterns with almost the same squared perimeter to area ratio would exhibit the same compactness value. It is likely that some of the patterns would have overlapping compactness values. This deficiency is overcome by the next feature.

2) *Compactness II (C_L)*: The second hand feature we are interested in is the thumb. In other hand gesture recognition algorithms, the algorithms treat the hand as one whole area. In our algorithm, we treat the hand, and the image, as consisting

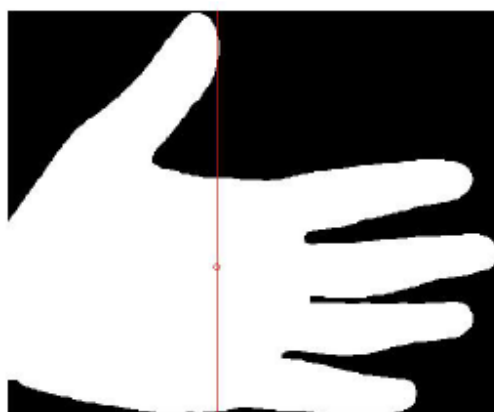


Fig. 2. Example of hand partitioning. The circle represents the centroid and the image is partitioned along the vertical line

of two halves: the half that contains the thumb (hereon referred to as the left half), and the half that contains the four fingers (hereon referred to as the right half). The hand is partitioned into two halves at its geometric center, or centroid, by a vertical line parallel to the image edge. The centroid of a digital image can be found by calculating the image moment using

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y), \quad (2)$$

where $I(x, y)$ is the intensity at coordinate (x, y) . The coordinate of the centroid (\bar{x}, \bar{y}) is found by using

$$\bar{x} = \frac{M_{10}}{M_{00}}, \quad (3)$$

$$\bar{y} = \frac{M_{01}}{M_{00}}. \quad (4)$$

The centroid of the hand lies within the hand, and therefore suitable for separating the thumb from the other fingers. An example of image partitioning is shown in Fig 2.

Compactness is used to detect the presence of the thumb in the left half of the image. The thumb shape significantly increases the compactness value because of its peninsula-like shape. Hand patterns that include the thumb pointing away from the palm would have a noticeably higher compactness value in the left half than hand patterns that do not.

Compactness is rotation, scaling, and translation invariant (RST-invariant), but is heavily shape dependent. Thus it is possible for one hand pattern to produce varying compactness values because human hands inherently vary. To improve successful recognition it is necessary to use a feature that produces discrete values. The next feature demonstrates this idea.

3) *Radial Distance*: A radial distance profile of a hand is the plot of the Euclidean distance between all the boundary

points of the hand and a reference point within the hand. The term radial distance is coined in [10] and also used in [1] and [2] to find the number of fingers and angles associated

TABLE I Expected number of peaks for each pattern

Pattern	Number of Peak(s)
0	N/A
1	1
2	2
3	3
4	4
5	4
6	1
7	1
8	2
9	3

with those fingers. The Euclidean distance is calculated using Eq. 5

$$ED(p, q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}, \quad (5)$$

where p includes all the boundary points and q is the reference point. Previously the reference point was taken at the center of the wrist, described in [8]. In this version, the reference point is taken at the centroid, and any boundary points left of the centroid are not included in the profile. The focus is now on the right half of the hand, where we want to extract the number of fingers found in the image. In this step we also make sure that any discontinuous regions are discarded.

To extract the number of fingers, we define a threshold line at seventy five percent of the maximum distance away from the centroid of the hand. We count the number of points that this line intersects with the radial distance function. The number of intersections determine the number of fingers. Table I shows the expected result of this feature calculation method.

RESULTS

200 test images were used to test the effectiveness of the three features mentioned in the previous section. The purpose of testing these features is to define three parameters that can be used in a recognition scheme to classify the patterns.

A. Compactness I (C_A)

The result of plotting the compactness values of all images is shown in Fig. 3. The compactness values are plotted versus the actual pattern number for comparison and analysis purposes. We can see from Fig. 3 that compactness values of all hand patterns fall into three distinct groups. We also expected patterns with similar shapes to have overlapping compactness values, and this is proven correct by the plot.

We now define a parameter C_A , for compactness, and classify any image with C_A less than 1.65 to be identified as Pattern Zero. The value 1.65 was manually selected from the plot. Additionally, we now define any image with value of C_A between 1.65 and 2.53 be classified as either Pattern One or Six.

B. Compactness II (C_L)

The plot of the compactness values for the left half of the hand is shown in Fig. 4. Similar to Fig. 3, the values are plotted versus the actual pattern number for analysis purposes. Recall

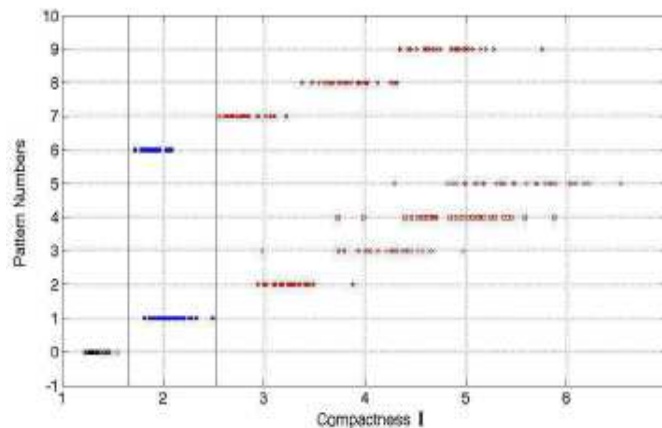


Fig. 3. The plot of compactness values C_A for all patterns

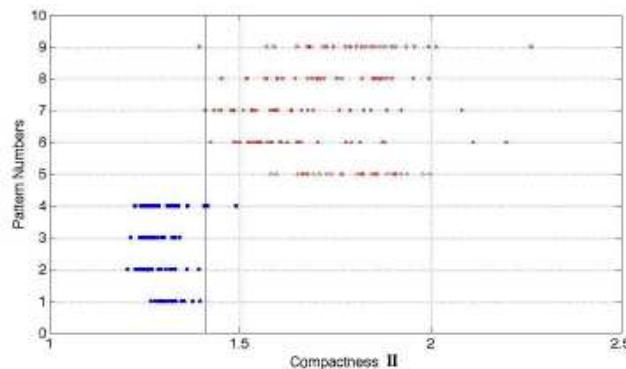


Fig. 4. The plot of compactness values of the left half, C_L , for pattern one to pattern nine

that the left half of the hand is analyzed to detect the presence of the thumb in the image. As expected, patterns without a thumb produce a smaller compactness value than patterns with a thumb.

We define a second parameter C_L , for compactness of the left half, and classify any image with C_L less than 1.47 to be Pattern One, Two, Three, or Four. Again, the value 1.47 was obtained manually from the plot. An image with C_L greater than 1.47 is classified as either Pattern Five, Six, Seven, Eight, or Nine.

C. Radial Distance

Sample radial distance plots are shown in Fig. 5, along with the threshold line at 75 percent of the maximum value. The third parameter P is now defined as the number of fingers identified from the radial distance plot. P is a positive integer between one and four.

A decision making flowchart using the three defined parameters is shown in Fig. 6. The threshold values used in each parameter of the flowchart is used for every pattern.

DISCUSSION

Using 200 additional input images, we tested the success rate of the recognition algorithm. The recognition algorithm has the same structure as shown in Fig. 6. Table II shows the result of running 200 new input images through the algorithm. Out of 200 input images, the algorithm correctly identified 182 cases, and falsely identified 18 cases, achieving a success rate of 91 percent. Table III shows the confusion table after the recognition algorithm has been tested with 200 sample images. It is apparent that most of the recognition error occurs from extracting the correct number of fingers from the radial distance function.

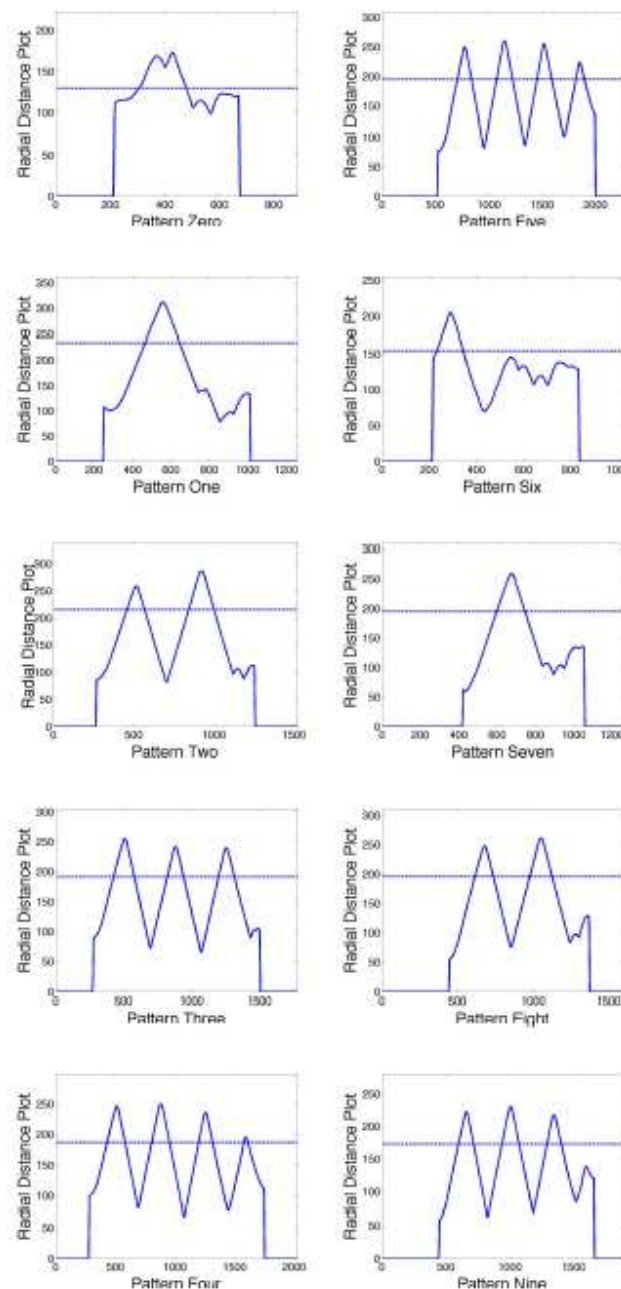


Fig. 5. Sample radial distance plot for one set of patterns

For example, Pattern Four was mistaken as Pattern Three quite often. This is also the case for Pattern Seven, which was mistaken for Pattern Eight. At this point we consider the effectiveness of the feature calculation methodologies used, and realize that flaws exist in the latter two. Hand partitioning, as described in Section 2.2, aims to separate the thumb from the rest of the fingers, but by using a line parallel to the image's edge, any image with varying hand orientation would cause part of the thumb to be included in the right half.

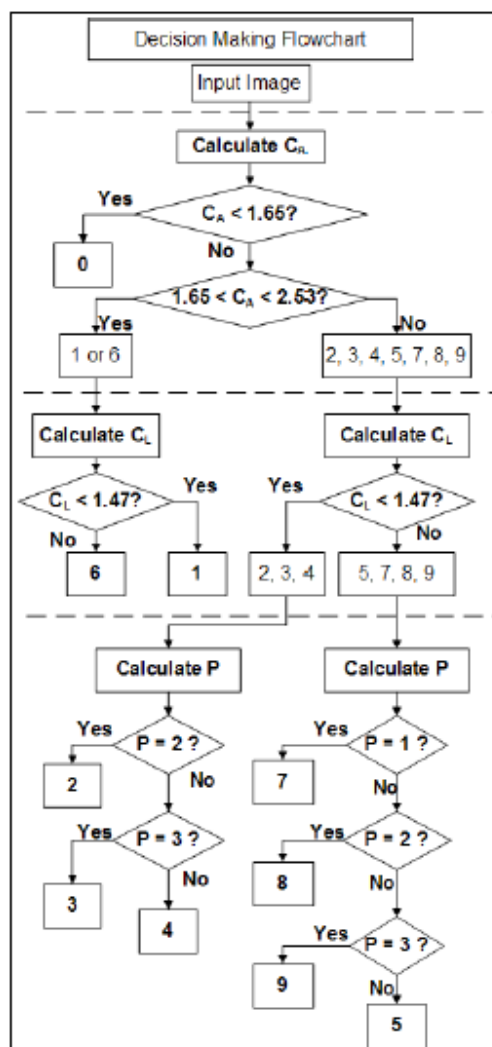


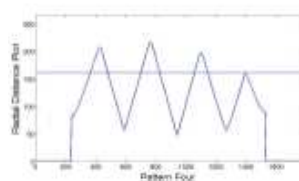
Fig. 6. The decision flowchart with threshold values obtained from testing 200 input images

TABLE II Hand pattern recognition results

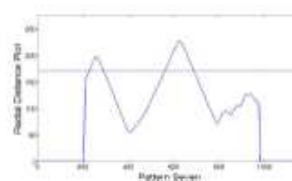
Pattern	# Input Images	Successful Cases	Recognition Ratio (%)
0	20	19	95
1	20	19	95
2	20	19	95
3	20	19	95
4	20	14	70
5	20	20	100
6	20	20	100
7	20	15	75
8	20	20	100
9	20	17	85
All	200	182	91

TABLE III Confusion table

	Recognized									
	0	1	2	3	4	5	6	7	8	9
Actual	0	19	0	0	0	0	1	0	0	0
	1	0	19	0	0	0	0	1	0	0
	2	0	0	19	0	0	0	0	1	0
	3	0	0	0	19	0	0	0	0	1
	4	0	0	0	4	14	0	2	0	0
	5	0	0	0	0	0	20	0	0	0
	6	0	0	0	0	0	0	20	0	0
	7	0	0	0	0	0	0	0	15	4
	8	0	0	0	0	0	0	0	20	0
	9	0	0	1	0	0	1	0	1	17



(a)



(b)

This would in turn affect the radial distance plot and create a “false” finger. Fig 7(b) illustrates this error.

Finger extraction method, as described in Section 2.3, also uses a horizontal line to determine the number of fingers, and that causes a lot of problem for Pattern Four, because the fifth finger (the pinky) is inherently shorter than the other three fingers, and would not show up on the line. Fig 7(a) illustrates this error.

The errors described above is likely caused by varying hand orientation, which is present because the hand is hovering in front of the camera and not fixed to any stationary objects. This problem should either be fixed first before any useful features are extracted, or prevented using a more robust partitioning method. A method that analyzes the direction of the thumb using slope or line information of the fingers could be considered, and would make the entire algorithm more orientation invariant.

CONCLUSION

There are many approaches to hand gesture recognition, and each approach has its strengths and weaknesses. The strength of the proposed method in this paper is the combination of three very useful features. Other methods, which use only one feature, become reliant on that one feature, and any error there would jeopardize the entire algorithm. The same can be said with a bloated algorithm that relies on many inter-dependent features. The proposed method uses three features, and these three features are independent of each other, but also covers each other's weaknesses.

The weakness of this method is the lack of a systematic approach to defining certain parameters. The threshold values for the three parameters were obtained empirically in this paper. Using neural network techniques might be able to find more appropriate values that would minimize the chances of error, and improve the success rate. However, this would go against the theme of this algorithm, which is recognition with minimal computational cost.

Compared to the previous method found in [8], this current method produces better results overall. The number of test images have been increased from 100 to 200 images, and the success rate improved from 90 to 91 percent. The current algorithm is also more robust to varying hand orientation because the radial distances are computed from the centroid of the hand, instead of the center of the wrist, as used in [8]. Another notable improvement is the use of a varying threshold value for determining the number of fingers. Because the threshold line is not fixed and varies according to the maximum radial distance, the overall algorithm shows a better performance than the previous version of this algorithm.

Several parts of the algorithm can be improved for a more robust recognition. Aligning the hand properly before any

feature calculations are made, or partitioning the hand using orientation information, are two possible directions. Aligning the hand properly would require spatial manipulation of the hand image, so proper partitioning method would be the better choice. Normal human behavior dictates that hand orientation is an uncommon problem, because it is unlikely for someone to show the pattern six upside down, with the thumb pointing down. In the future, we may increase the number of sample images, patterns, and may expand our project to cover image from video sequence or implement real time recognition.

REFERENCES

- [1] N. Yasukochi, A. Mitome and R. Ishii, "A Recognition Method of Restricted Hand Shapes in Still Image and Moving Image as a Man-Machine Interface," *IEEE Conference on Human System Interactions*, page 306-310, **2008**.
- [2] T. Niidome and R. Ishii, "A GUI Support System for a Sight Hand-icapped Person by Using Hand Shape Recognition," The 27th Annual Conference of the IEEE Industrial Electronics Society (EICON), Vol. 1, page 535-538, **2001**.
- [3] S. Zhao, W. Tan, C. Wu, C. Liu, and S. Wen, "A Novel Interactive Method of Virtual Reality System Based on Hand Gesture Recognition," Chinese Control and Decision Conference, page 5879-5882, **2009**.
- [4] P. Premaratne and Q. Nguyen, "Consumer Electronics Control System based on Hand Gesture Moment Invariants," *Computer Vision IET*, Vol. 1, page 35-41, March 2007.
- [5] H. Tauseef, M.A. Fahiem, and S. Farhan, "Recognition and Translation of Hand Gestures to Urdu Alphabets Using a Geometrical Classification," Second International Conference in Visualization, page 213-217, July **2009**.
- [6] H. Xu, X. Hou, R. Su, and Q. Ni, "Real-time Hand Gesture Recognition System based on Associative Processors," The 2nd IEEE International Conference on Computer Science and Information Technology, page 14-18, August **2009**.
- [7] Y.-T. Chen and K.-T. Tseng, "Developing a Multiple-angle Hand Gesture Recognition System for Human Machine Interactions," The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON), page 489-492, **2007**.
- [8] A. Jinda-Apiraksa, W. Pongstiensak, and T. Kondo, "Shape-Based Finger Pattern Recognition using Compactness and Radial Distance," The 3rd International Conference on Embedded Systems and Intelligent Technology (ICESIT 2010), Chiang Mai, Thailand, February **2010**.
- [9] Q. Zhang, F. Chen and X. Liu, "Hand Gesture Detection and Segmentation Based on Difference Background Image with Complex Background," The 2008 International Conference on Embedded Software and Systems (ICCESS2008), Page 338-343, July **2008**.
- [10] E. Yrk, E. Konukoglu, B. Sankur, and J. Darbon, "Shape-Based Hand Recognition," *IEEE Transaction On Image Processing*, Vol. 15, No. 7, page 1803-1815, July **2006**.