Finger Edge Contour Perimeter as a Biometric Based Identification System

by Kusworo Adi

Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Finger Edge Contour Perimeter as a Biometric Based Identification System

To cite this article: C E Widodo and K Adi 2019 *J. Phys.: Conf. Ser.* **1217** 012033

Finger Edge Contour Perimeter as a Biometric Based Identification System

C E Widodo[1] and K Adi [2]
Department of Physics, Faculty of Science and Mathematics, Diponegoro University, Semarang, Indonesia
E-mail: catur.ediwidodo@gmail.com, kusworoadi@gmail.com

Abstract. Generally, personal identification system is still done by conventional techniques, such as using ID cards, passwords or PIN numbers. These systems is easily lost, damaged or forgotten. In this paper we present a new method for identifying a person with a biometric based on palm of the hand. This system does not require a peg like an identification system based on existing hand geometry. Phase of development are: first, the palm image scanned in the RGB format is converted into a gray level. Then the image is converted to binary by giving the correct threshold value. From the binary image, we carry out the edge detection process to get the contour perimeter of the palm. From the contour perimeter of the palm, we measure the length of 8 edges contour perimeter of the fingers i.e. the right edge of the thumb, the left edge of the index finger, the right edge of the index finger, the left edge of middle finger, the right edge of middle finger, the left edge of pinkie, the right edge of pinkie, and the left edge of ring finger. The identification process is performed by minimum distance, which is the smallest euclidean distance value associated with the test image and database. The experimental results giving good performance with a success of up to 90%.

1. Introduction

Some biometric techniques can be used as authentication systems [1,2], including fingerprint scanning, face verification, identification of retinal scans, and recognition based on hand geometry [3,4,5,6]. Although fingerprints are the most popular system, actually other biometric studies are very interesting, because we can explore different types of data. Each biometric system has its advantages and disadvantages, including data acquisition stage, preprocessing stage, and classification stage [7]. In this paper we propose a new technique that is an identification system using finger side contours. Feature features that are used are different from the geometry hand that was researched by previous researchers [7,8,9,10,11,12]. The advantage of this technique is that it does not require a peg, so its use is convenient and easy. The identification system using the side finger contour proposed here has the same typical as other biometric systems [7]. He works in two phases: enrollment phase and comparison phase. In the enrollment phase, the palm photos of the users are taken. The photo is then prepossessing, then feature extraction is performed to produce a feature vector from the user. Vector features of the user are then stored in the database. In the verification phase, photos of a user's palm are taken, then prepossessing, then feature extraction is performed to produce a feature vector from the user. The feature vector is then compared to the database to find out who the palm is.

In this work, an explanation of all stages of system development will be given. Section 2 will describe feature and classification extraction stages including image capture, preprocessing, contouring and measurement of finger side contour lengths, and classification based on the minimum distance from Euclid distance. In section 3, we present the results of the experiment. The conclusion of the observed results and the direction of future research will be presented in chapter 4.

IOP Conf. Series: Journal of Physics: Conf. Series **1217** (2019) 012033 doi:10.1088/1742-6596/1217/1/012033

2. Features axtractions and classification

The purpose of data extraction is the sample data in the form of hand photographs, the process, and produce feature vectors that present specific characteristics to a user among all members of the population. The purpose of classification is to enter user vectors into the comparative process to map users who you have found. In this section we will explain the method of doing palm photos, preprocessing, feature extraction, and classification.

2.1. Image Capturing

The signal from the sample is displayed with a color CCD camera with a resolution of 480 x 640 pixels, functioning on the platform designed to display the distance on the camera. The camera and platform are attached to a closed system to be stable and not related to outside. In this study, the distance of light with a sample is 30 centimeters and the light source is an LED lamp with a power of 3 watts. When the process of taking pictures, the position of the fingers must be tenuous. The prototype design and palm works can be used in Figure 1.

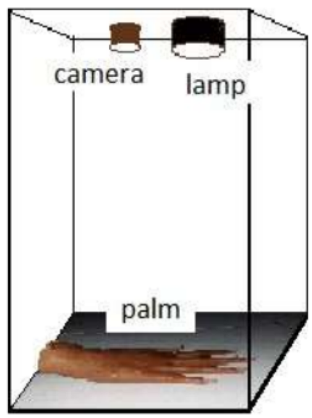

Figure 1: Prototype scheme

2.2. Preprocessing.

After the image is taken, preprocessing is done. Preprocessing is carried out in three phases, namely RGB to Gray scale transformation, segmentation, and contouring. The first step is to transform colored images into grayscale. To do this, the average operation is carried out on red, green and blue channels [13]. After being converted into grayscale image, then segmentation is done using treshold operation to separate objects with background. This threshold value is obtained by trial and error. From the results of the experiment or trial and error obtained that the appropriate threshold value for segmenting is 150. The part with an intensity value of less than 150 will be changed to number 0, and the intensity value above 150 will be changed to number 1, so that the binary image is separated between the palm segment and the background. Because in this study the input image used in the object's intensity value is less than 150, then it is converted to a number 0 or black. Whereas in the background which has an intensity above 150, it is changed to number 1 or white. After going through the segmentation process using treshold, the resulting image of the segmentation process is not perfect. The results of segmentation in the palm area is still have a small white hole, usually in the nail area. To overcome this problem, morphological operations are used to removing small island.

The next step is edge detection. This edge detection serves to identify the palm contour. The detection used is a second derivative operator (Laplacian of Gaussian) that can detect edges accurately especially on steep edges because it has zero crossing. The Laplacian of Gaussian is formed from a Gaussian process which functions to reduce noise and is followed by a Laplace operation which serves to minimize the possibility of edge databases. The Lapacian of Gussian operator works by finding the zero value in the second derivative of the image, because when the first derivative is at the maximum value, the second derivative will produce a zero value. The second derivative from the direction x and y is combined into one operator value. Figure 2 shown an example of the process of forming palm contour.

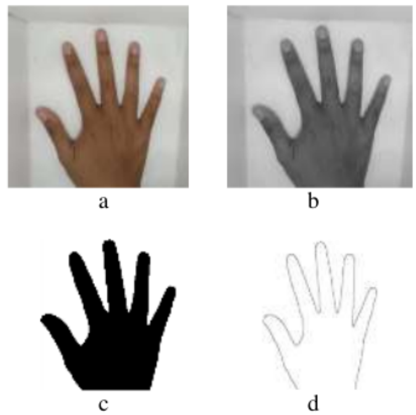

Figure 2: Example of the process of palm hand contour formation: (a) RGB image, (b) gray scale image, (c) binary segmentation image, and (d) contour image. For visualization clarity, image (b) is displayed in invert mode.

2.3. Feature Extraction

After we get the image contour of palm, we perform the length of 8 edges perimeter of the fingers i.e. the right edge of the index finger, the left edge of the finger, the right edge of the finger, the left edge of pinkie, the right edge of pinkie, and the left edge of ring finger. The eight segments are measured in the order shown in Figure 3a, where the initial measurement is at the middle finger. This is done to facilitate tracking. Tracking is done based on the chain code as shown in Figure 3b.

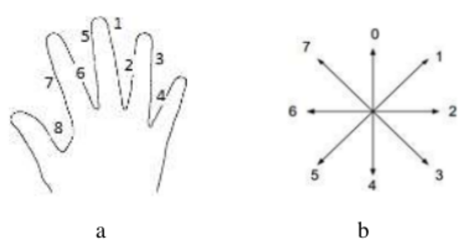

Figure 3: Eight segment (a) and chain code (b)

To measure segment length 1, tracking starts from the middle finger, moves right, down or left with a clockwise decision making sequence (with a chain code 1, 2, 3, 4 or 5) and will stop if the contour leads to above (chain code 6, 7 or 8). To measure the length of segment 2, tracking starts from the position of the final pixel in tracking segment 1, moves right, up or left (chain code 1 ,8, 7, 6 or 5), and stops when the contour points down (chain code 4 , 3 or 2). To measure the length of segment 3, a method such as the measurement of segment length 1 starts from the final pixel position of tracking segment 2. To measure the length of segment 4, a method such as segment 2 measurement starts from the final pixel position of tracking segment 3. To measure the length of segment 5, tracking starts from middle finger, move left, down or right (chain code 5 ,4, 3, 2, or 1) and will stop if the contour points up (code 8, 7 or 6). To measure the length of segment 6, tracking starts from the final pixel position on tracking 5, moves left, down or right (with a chain code of 5, 6, 7 ,8 or 1) and stops if the contour is directed down (chain code 2, 3 or 4). To measure the length of segment 7, a method such as measurement of segment 5 starts from the final pixel position of tracking segment 6. To measure the length of segment 8, a method like segment 6 starts from the final pixel position of segment tracking 7. To be clearer, you can see the following algorithm:

```
For i = 1 to 480
  For j = 1 to 640
        If im (i,j) == 1
              startX = i
              startY = j
segment1 = 0
posX = startX
posy = startY
while posX < 480
  if nextPos in [1, 2, 3, 4, 5]
           segmen1 = segmen1+1
           posX = newPosX
           posY = newPosY
  if nextPos in [6, 7, 8]
           break
segment2 = 0
while posX < 480
  if nextPos in [1, 8, 7, 6 ,5]
           segmen1 = segmen1+1
           posX = newPosX
           posY = newPosY
  if nextPos in [4, 3, 2]
           break
segment3 = 0
while posX < 480
  if nextPos in [1, 2, 3, 4, 5]
           segmen1 = segmen1+1
           posX = newPosX
           posY = newPosY
  if nextPos in [6, 7, 8]
           break
segment4 = 0
while posX < 480
  if nextPos in [1, 8, 7, 6 ,5]
           segmen1 = segmen1+1
```

```
                    posX = newPosX
                    posY = newPosY
        if nextPos in [4, 3, 2]
                    break
      segment5 = 0
      posX = startX
      posy = startY
      while posX < 480
        if nextPos in [5, 4, 3, 2, 1]
                    segmen1 = segmen1+1
                    posX = newPosX
                    posY = newPosY
        if nextPos in [8, 7, 6]
                    break
      segment6 = 0
      while posX < 480
        if nextPos in [5, 6, 7, 8, 1]
                    segmen1 = segmen1+1
                    posX = newPosX
                    posY = newPosY
        if nextPos in [2, 3, 4]
                    break
      segment7 = 0
      while posX < 480
        if nextPos in [5, 4, 3, 2, 1]
                    segmen1 = segmen1+1
                    posX = newPosX
                    posY = newPosY
        if nextPos in [8, 7, 6]
                    break

      segment8 = 0
      while posX < 480
        if nextPos in [5, 6, 7, 8, 1]
                    segmen1 = segmen1+1
                    posX = newPosX
                    posY = newPosY
        if nextPos in [2, 3, 4]
                    break
```

2.4. Classification

The resulting feature vector is entered into the comparison process to determine the user whose photograph of the palm of the hand has been taken. The method used is template matching [13] using the Euclidean distance. Euclidean distance is one of the most commonly used methods to calculate the similarity of 2 vectors. The Euclidean distance method calculates the square root of the difference in 2 vectors, namely a feature vector of the user that is entered with the characteristics in the existing database [8]. The Euclidean distance between two points is the length of the hypotenuse of a right triangle determined by the following equation (1)

$$d_{ij} = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \quad (1)$$

where d_{ij} is the euclidean distance, x_{ik} is the input feature vector, x_{jk} is the characteristic of the database vector and k is the number of feature vectors in this system is 8. Recognition is obtained by calculating the closest distance, which is the smallest Euclidean distance value.

3. Experimental results

Experiments carried out on 10 users. Of the ten users, one person was taken with 10 poses of palm, given the A1, A2, ... A10 code. Nine people took one photo of the palm with a code B, C, ..., J. With these two types of data will be determined the variation of Euclid distance of a user in various pose, and the variation of Euclid distance between users. The hypothesis we propose is: the variation of Euclid's distance in various pose is small, while the variation between users is large. The experimental results are shown in Tables 1, 2, 3, and 4, where Table 1 shows the measurement results of 8 finger segments of each pose in user A, Table 2 shows the competition results of Euclid distance calculation for each pose, Table 3 shows the results of measurements of each user's finger segment, and Table 4 shows the competition from the calculation of Euclid distance for each user.

Table 1: Results of measurement of 8 segments in user A

User Code	segment							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
A1	190	160	189	103	188	160	242	104
A2	193	162	188	108	189	159	241	98
A3	193	160	189	109	191	163	241	105
A4	195	161	187	107	187	162	235	99
A5	195	163	187	108	188	159	244	97
A6	192	161	190	109	189	160	238	102
A7	195	160	192	97	189	156	240	98
A8	191	159	189	100	190	158	237	101
A9	188	158	189	100	191	150	234	99
A10	193	159	191	107	191	148	235	95

Table 2: Competition of calculating result of Euclidean distance for each pose for user A

User Code	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**	**A9**	**A10**
A1	0	-	-	-	-	-	-	-	-	-
A2	9	0	-	-	-	-	-	-	-	-
A3	8	9	0	-	-	-	-	-	-	-
A4	11	8	10	0	-	-	-	-	-	-
A5	11	4	11	10	0	-	-	-	-	-
A6	8	6	6	7	9	0	-	-	-	-
A7	11	12	16	14	13	14	0	-	-	-
A8	7	10	12	10	13	10	8	0	-	-
A9	15	16	19	16	18	15	12	9	0	-
A10	18	14	19	16	16	15	14	14	10	0

IOP Conf. Series: Journal of Physics: Conf. Series **1217** (2019) 012033 doi:10.1088/1742-6596/1217/1/012033

Table 3: Results of measurement of 8 finger segments per user

User Code	segment							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
A1	190	160	189	103	188	160	242	104
B	199	157	183	96	190	160	235	96
C	215	182	199	135	216	159	252	104
D	212	161	194	110	199	175	260	103
E	207	181	206	135	223	156	253	108
F	204	177	201	118	206	188	254	104
G	192	151	181	105	179	157	225	96
H	203	172	195	120	197	171	260	110
I	194	163	181	115	199	163	225	95
J	200	182	202	126	207	171	257	110

Table 4: Competition of calculating result of Euclidean distance for each user.

User Code	**A1**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**
A1	0	-	-	-	-	-	-	-	-	-
B	17	0	-	-	-	-	-	-	-	-
C	56	61	0	-	-	-	-	-	-	-
D	35	38	41	0	-	-	-	-	-	-
E	58	65	14	47	0	-	-	-	-	-
F	46	52	37	26	41	0	-	-	-	-
G	24	20	70	52	74	63	0	-	-	-
H	35	44	33	19	37	23	54	0	-	-
I	27	25	51	44	56	48	26	44	0	-
J	45	55	25	32	25	21	64	17	48	0

From Table 1 at a glance it can be seen that there are small variations between poses in each segment. If the Euclidean distance between poses competed (Table 2) shows that the Euclid distance between poses varies between 7 - 18. From Table 3 at a glance it can be seen that there is a large variation between users in each segment. If the Euclid distance between users is competed (Table 4) shows that the Euclid distance between users varies between 17 - 58. It turns out that there is an Euclid distance intersection between the various poses of a user and between users. This means that the hypothesis we propose is not entirely correct. An identifiable user can occur as someone else. For example: if the database is A10 .., B, C, ... J, and the user identified is A1, it will be identified as B. From this case (10 users with a user with 10 poses) the accuracy can be calculated as 90%.

4. Conclusions

Biometrics based on finger edge contour perimeter have been reported. The algorithm for determining finger edge contour perimeter has been provided and has proven to be able to do its work well. The experimental results have been shown, giving good performance with a success of up to 90%. The system as proposed is known to be a good alternative for identification systems, even though it requires further work that can be implemented easily. Future work must apply additional data bases to more general results.

Acknowledgement
Financial assistance was provided by the Government of the Republic Indonesia through the Faculty of Science and Mathematics of the Diponegoro University.

References

[1] Jain, A.K., Ross, A. and Prabhakar, S. 2004. *IEEE Transactions on Circuits and Systems for Video Technology.* **14** 1
[2] Spinella, E. 2003. *Information security reading room*, SANS Institute, U.S.
[3] Thai, L.H. and Tam, H.N. 2010. *International Journal of Computer Science Issues*. **7** 3
[4] Viola, P. and Jones, M.J. 2004. *International Journal of Computer Vision*. **57** (2) 137–154
[5] Farzin, H., Abrishami-Moghaddam, H. and Mohammad-Shahram, M. 2008. *EURASIP Journal on Advances in Signal Processing*. **2008**
[6] Sanchez-Reillo, R., Carmen Sanchez, A. and Gonzalez-Marcos, A. 2000. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **22** (10) 1168-1171
[7] Faúndez-Zanuy, M., Elizondo, D.A., Ferrer-Ballester, M.A., and Travieso-González, C.M. 2007. *Authentication of individuals using hand geometry biometrics: a neural network approach, Neural Processing Letters*. **26** (3) 201-216.
[8] Varchol, P., And Levický, D. 2007. *Radioengineering*. **16** 4
[9] Mathivanan, B., Palanisamy, V. and Selvarajan, S. 2012. *International Journal of Emerging Technology and Advanced Engineering*. **2** 7.
[10] Ayeni,J.K., Sadiq, K.A. and Adeyinka, A. 2013. *International Journal of Scientific Research Engineering and Technology.* **2** (6) 352-357
[11] Rathi, P. and Dubey, S., 2013. *International Journal of Advanced Research in Computer Engineering and Technology.* **2** 6
[12] Park, G. and Kim, S., 2013. *Sensors.* **13** 2895-2910
[13] Gonzalez, R.C., Woods, R.E and Eddins, S.L. 2009. *Digital Image Processing Using MATLAB*, Gatesmark Publishing, U.S.A.
[14] Mahalakshmi, T., Muthaiah, R. and Swaminathan, P. 2012. *Research Journal of Applied Sciences, Engineering and Technology*. **4** (24) 5469-5473

Finger Edge Contour Perimeter as a Biometric Based Identification System

ORIGINALITY REPORT

6%	4%	7%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	R. Sanchez-Reillo, C. Sanchez-Avila, A. Gonzalez-Marcos. "Biometric identification through hand geometry measurements", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000 Publication	1%
2	"Computer Vision – ECCV 2008", Springer Science and Business Media LLC, 2008 Publication	1%
3	file.allitebooks.com Internet Source	1%
4	polen.itu.edu.tr Internet Source	1%
5	bioinfopublication.org Internet Source	1%
6	www.mitpressjournals.org Internet Source	1%
7	ijpediatrics.com Internet Source	

1%

Exclude quotes	Off	Exclude matches	< 1%
Exclude bibliography	Off		

Finger Edge Contour Perimeter as a Biometric Based Identification System

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9