Volume 4, No. 4, March-April 2013 ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Fingerprint Geometry matching by Divide and Conquer Strategy

Sanjukta Pal*
Student, M.Tech. (CST) Dr. B C Roy Engineering College, Durgapur, West Bengal, India
sanjukta_2008pal@rediffmail.com

Sucharita pal
[2]Electrical Engeneering, Asansol engineering College, Asansol, West Bengal, India

Prof. (Dr) Pranam Paul
[3]Computer Application, Narula Institute of Technology, Agarpara, West Bengal, India

Abstract: In recent day for security purpose biometric authentication is used. The fingerprint geometry is one of the most recently used techniques of biometric authentication. This is an algorithm for matching a fingerprint image with previously given fingerprint images.
The main criteria are that, this algorithm is applicable only on binary form of fingerprint image. For fixing up a pixel region near the centre the dividend rule is taken. I.e. for selecting an image segment, we made four segments of the image using the centre position of the image, and we continued that technique until the certain criteria not be full filled. After reaching the criteria, using the fixed pixel region the traversing of DB Image would be started. After traversing the DB Image if we get the satisfactory result (that means, if the value above threshold value) it will proceed further and if in the next, it will give the positive result finally we can easily say that the fingerprints are matched.

Key Word: Biometric Authentication, DB image, Final image, image segment.

I. INTRODUCTION

This is an algorithm which can be used for matching two fingerprint geometry, one would be given previously and must be stored in the database called DB Image and the another would be the finally given fingerprint image for matching with previous one called Final Image.

This algorithm is only applicable on the binary conversion of fingerprint geometry. So for converting a fingerprint geometry first we have to convert the images gradually into gray scale image and then into binary image. It can be happened by using the threshold value over the black level intensity of gray scale image. So the binary image is nothing but a combination of 0 and 1. To apply this algorithm, both the DB Image and Final Image are required to convert into binary image.

To check the DB Image for matching purpose a small part of image segment would be chosen from Final Image. To choose the image segment we used the divide and conquer strategy. That means at first the Final Image would be divided into four segments, then it will check some given criteria. If the criteria would not be full filled it will break the south east corner image segment again into four segments. After this until the criteria would not be full filled it will break the North West corner image segment of the previous image segment in a nested way. If the finally selected image segment would match with any part of the DB Image above the threshold value, then the rest DB Image would be checked step wise as given below. If after the total checking the of pixel matching value is greater than the threshold limit then the Final Image would be matched with DB Image. That means we can easily say that two fingerprints are matched.

II. ALGORITHM

1st, Read 5 or 6 fingerprint images from different angle (say, DB Images).
1a: convert images into gray scale image.
1b: convert gray scale image into binary image.

Finally, for matching purpose one fingerprint image would be taken (say, Final Image)
2a: convert final image into gray scale image
2b: convert this gray scale image into binary image.

The DB Images and Final Image may not be taken from same device, so the images may not be of same size.

Then for matching purpose,

Step-1: Calculate the number of pixels along width and in length (say m and n) from Final Image.

Step-2: Using m, n derive the centre position (p, q) of Final Image by (m+1/2, n/2) or by (m/2, n+1/2) or by (m+1/2, n+1/2) or by (m/2, n/2).

Step-3: Using (p, q) as centre position break the image into four segments.

Step-4: Take the south east corner image segment (say A) of Final Image.

Step-5: If the number of pixel along width and length of the image segment A is greater than 10 then go to step1 and repeat step2 and step3.

Step-6: Take the North West corner image segment (say A_1) of the previous image segment A.

Step6a: If the number of pixel along width and length of previously selected image segment A_1 is greater than 10 then go to step5.

Step6b: If the number of pixel along width and length of previously selected image segment A_1 is less or equal to 10 then go to step7.

Step-7: Using the finally selected image segment (A_n), traverse one DB Image, pixel by pixel in row wise and column wise.

(Here A_n is said to be finally selected image segment because after breaking an image n+1 times, we will get the segments which can full fill the above criteria, that means after breaking an image 6 times if it full fills the above criteria then we can select the finally selected image segment say A_5)

7a: For each traverse store left most pixel position (a) of DB Image and percentage of pixel match (b) during that traverse.

7b: Store left most pixel position (a) and percentage of pixel match (b) of maximum matching of any two traverses.

Step-8: If the maximum matching of pixel is greater than the threshold value, then go to step 9,

Else go to step14.

Step-9: Using the left most pixel position fix up the pixel chunk of the DB Image using maximum match.

Step-10: Check the DB Image with the total area of four segment of final image segment (by which traversing of DB Image is occurred).

Step-11: If the percentage of pixel match is greater than the threshold value, then go to step 12. Else go to step 14.

Step-12: Next match the neighbor pixels row wise and then column wise.

Step-13: If the total pixel matching is greater than the threshold value, then continue and go to step 12.

Else go to step 14.

Step-14: Go to the next DB Image, and again check it from step1.

III. EXAMPLE

For matching two fingerprint images, first we need to convert images (both DB Image and Final Image) into gray scale image and then binary image, then we can enter into step1,because this algorithm is applicable on the binary format of images.

Step-1:

Derive number of pixel along length and width.

Here number of pixel along width= m = 40

Here number of pixel along length= n = 36

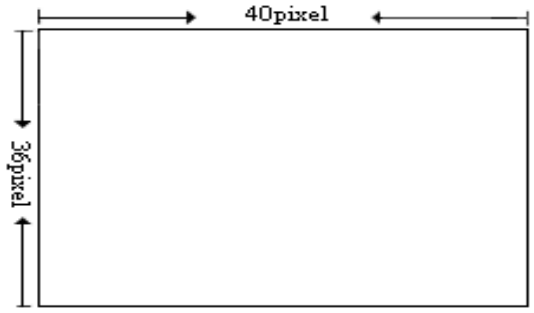

Fig.1. Derive number of pixels along width and length from Final Image

Step-2:

Derive the centre position of the image.

Here centre position= (p, q) = (m/2, n/2) = (40/2, 36/2)
= (20, 18).

Step-3:

Using centre position divides the image into four segments.

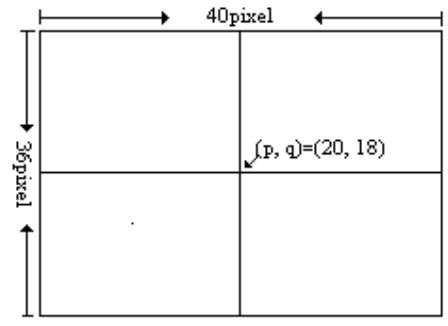

Fig.2. Devide the image into four segment using centre position

Step-4:

Take the south east corner of the four segments. The segment is said to be segment A.

Fig.3. Take the south east corner image segment of four segments

Step-5:

Derive number of pixel along length and width of the image segment A.

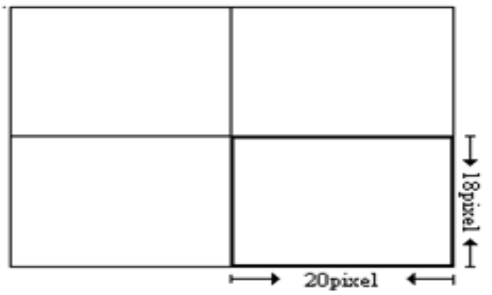

Fig.4. Derive number of pixels along width and length of the south east corner image segment(say segment A)

If the number of pixels along width or length is greater than 10 then go to step1 and repeat step2 and step3.

Here both in width and length number of pixel is greater than 10, so, repeat step1, step2 and step3. That means break the image segment into four segments again.

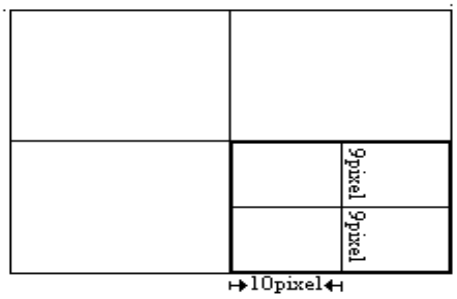

Fig.5. Break the image segment A again into four segments

Step-6:

Take the North West corner image segment (say A_1) of the previous image segment A.

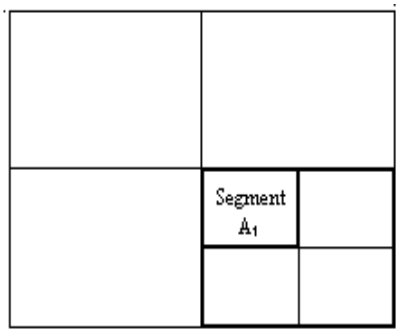

Fig.6. Take the north west corner image segment of segment A

Step-6a:

Here number of pixel along width and length of the image segment A_1 is not greater than 10.

So no need to go to the step 5.

Step-6b:

Here number of pixel along width and length of the image segment A_1 is less than and equal to 10.

i.e. for segment A_1 m=10 pixel

n=9pixel

So fix up the segment A_1 as final image segment.

So we have to go to the step 7.

Step-7:

Using final image segment A_1, traverse one DB Image pixel by pixel in row wise and then column wise.

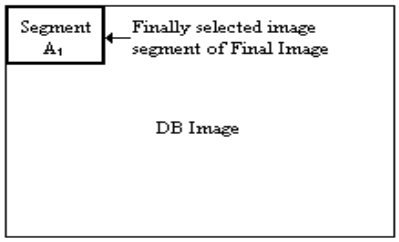

Fig.7. Traverse one DB Image by finally selected image segment (first step of traverse)

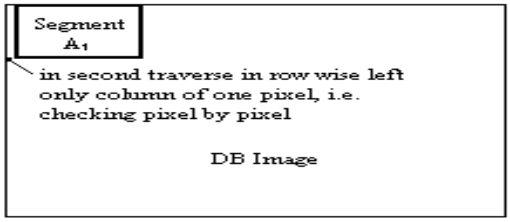

Fig.8. The second step of traverse

-

-

-

Fig.9. Final step of traverse

Step-7a:

For each traverse store left most pixel position (say a) and percentage of pixel match (say b).

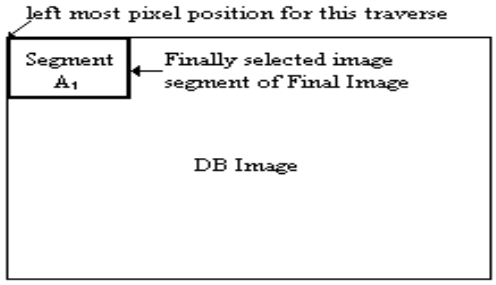

Fig.10. Store left most pixel position of that traverse

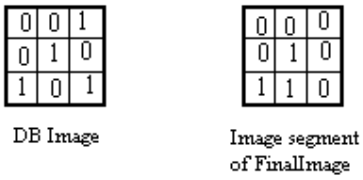

Fig.11. For calculating the pixel match

Here percentage of pixel match b = (6/9) * 100.

Step-7b:

For any two traverse store a where maximum (b) is received.

If for first step of traversing the value of q=80% and for second step of traversing the value of q=86%,then store the left most pixel position of second step traverse, and also store the maximum percentage of pixel match.

In this way the maximum matched area from DB Image would be gained.

Step-8:

If the maximum matched area is greater than the threshold value then goes to next step.

Else go to step 13.

Step-9:

Using the left most pixel position fix up the pixel region of DB Image using maximum matching.

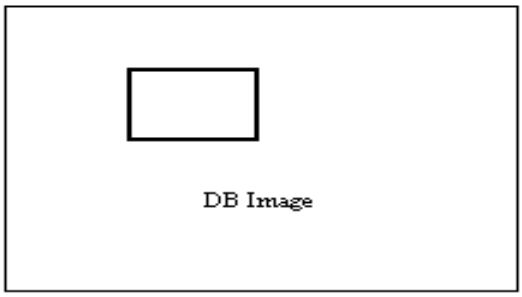

Fig.12. Let, this is the final fixed up pixel region after whole traverse

Step-10:

Check the DB Image with the total area of four segment of final image segment (by which traversing of DB Image is occurred).

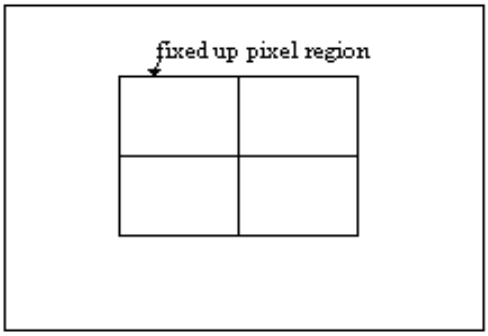

fig.13. Checking of DB Image using Four image segment

Step-11:

If the percentage of pixel matching of the above area (in fig.13) is greater than the threshold value then go to step12 and continue, else go to step 13.

Step-12:

Match the neighboring pixel row wise and then column wise.

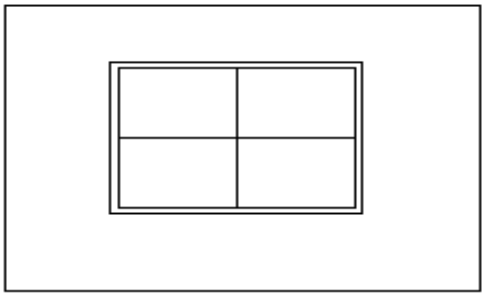

fig.14. Check the DB Image by neighboring pixel by row wise and then column wise

Step-13:

If the total pixel matching is greater than the threshold value, then continue and go to step 12.

Else go to step 14.

Step-14:

If the percentage of pixel match is less than the threshold value (at step 8 or at step11) then check the next DB Image.

IV. ANALYSIS AND CONCLUSION

To implement this algorithm, two main things cannot be ignored. First of all binary conversion of image is required, second, the image segment would be chosen from the nearby area of centre by the divide and conquer technique, because a fingerprint geometry always cover the centre part of device.

The main disadvantage of using this algorithm is that it will choose only one image segment from final image. Sometimes the fingerprint geometry can change by seasonal effect, so the small image segment may not be matched with any one of the DB Image. So for this reason the algorithm may be less effective sometime.

V. REFERENCES

[1]. J. K. Mandal, S. Dutta, "A 256-bit recursive pair parity encoder for encryption", Advances D -2004, Vol. 9 nº1, Association for the Advancement of Modelling and Simulation Techniques in Enterprises (AMSE, France), www. AMSE-Modeling.org, pp. 1-14

[2]. Pranam Paul, Saurabh Dutta, "A Private-Key Storage-Efficient Ciphering Protocol for Information Communication Technology", National Seminar on Research Issues in Technical Education (RITE), March 08-09, 2006, National Institute of Technical Teachers' Training and Research, Kolkata, India

[3]. Pranam Paul, Saurabh Dutta, "An Enhancement of Information Security Using Substitution of Bits Through Prime Detection in Blocks", Proceedings of National Conference on Recent Trends in Information Systems (ReTIS-06), July 14-15, 2006, Organized by IEEE Gold Affinity Group, IEEE Calcutta Section, Computer Science & Engineering Department, CMATER & SRUVM Project-Jadavpur University and Computer Jagat

[4]. Dutta S. and Mandal J. K., "A Space-Efficient Universal Encoder for Secured Transmission", International Conference on Modelling and Simulation (MS' 2000 – Egypt, Cairo, April 11-14, 2000

[5]. Mandal J. K., Mal S., Dutta S., A 256 Bit Recursive Pair Parity Encoder for Encryption, accepted for publication in AMSE Journal, France, 2003

[6]. Dutta S., Mal S., "A Multiplexing Triangular Encryption Technique – A move towards enhancing security in E Commerce", Proceedings of IT Conference (organized by Computer Association of Nepal), 26 and 27 January, 2002, BICC, Kathmandu.

[7]. Paul Reid, 2003, Biometrics for Network Security, Prentice Hall PTR, chapter-5

[8]. A white paper by the University of Southern California and VeriSign 2005 Building a Security Framework for Delivery of Next Generation Network Services United States.

[9]. L. Podio and Jeffrey S. Dunn 2002, Biometric Authentication Technology: From the Movies to Your Desktop, National Institute of Standards and Technology (NIST), Information Technology Laboratory 497

[10]. Edited by Lori Ayre, Infopeople Project, 2003, Library Computer and Network Security Infopeople Project, http://infopeople.org/howto/security/

[11]. Sarbari Gupta, 2004, Identity Authentication Identity Authentication using the using the PIV Token PIV Token, National Institute of Standards and Technology, India.

[12]. Secure Computing Corporation, 2001, Authenticating with one of the safest devices: the biometric Sony Puppy, Secure Computing Corporation, 4810 Harwood Road, San Jose, CA 95124 USA.

[13]. Biometric Consortium web site: http://www.biometrics.org 2006

[14]. International Biometric Industry Association, http://www.ibia.org 2005

[15]. Bioenable Technologies Pvt. Ltd. 2004-2005 http://www.bioenabletech.com/ biometrics_india_pune_contact.htm

[16]. Securitex Electronic Systems Engineering, 2006, Fingerprint Identification system http://www.securitex.com.sg/

[17]. Manvish Embedded Services, 2006, Finger print sensors technology overview http://www.manvish.com/embedded/miFAUN/techoverview.php

[18]. TopAZ Solutions Pte Ltd, 2006, Biometric Fingerprint Security, http://www.topazsol.com/bio_door_access.htm

[19]. Sanjukta Pal, Dr. Pranam Paul, "Cryptographic protocol Depending on Biometric Authentication", published in International Journal for Engineering Science and Technology (IJEST), February 2013 issue.

Short Bio Data for the Author

Sanjukta Pal is MTech (CST) final year student of Dr. B. C Roy Engineering College, Durgapur. She had completed MCA at West Bengal University of Technology.

Sucharita Pal is an Assistant Professor of Asansol Engineering College, Asnasol. She had completed her M.Tech in the field of Electrical Engineering at National Institute of Technology, Durgapur.

Dr Pranam Paul is an Assistance Professor of Narula Institute Technology, Agarpara. He had completed his Ph.D from Electronic and Communication Engineering department of National Institute of Technology, Durgapur in the field of Cryptography and Network Security and master degree in Computer Application in 2005 under West Bengal University of Technology, INDIA. He has total 50 International Journal publications among total 60 publications, except this one.