

A New Approach to Vein Pattern Authentication

V. Muralikrishna¹, Dr.B.Rama², Dr. Arvind Kumar Sharma³

¹ Research Scholar, Department of Computer science, OPJS University, Rajasthan.

²Professor, Computer Science Department, Kakatiya University, Warangal.

³Associate Professor, OPJS University, Rajasthan.

Abstract:

This paper presents a new approach to vein pattern authentication system. The various stages involved are binarisation of the ROI, thinning, feature extraction and matching. Our main focus is on Dynamic ROI Extraction and thinning. We got a database of hand images which are far infrared images of hand. For that images, we have done dynamic ROI extraction. The advantage of dynamic ROI extraction is that ROI extracted for different hand images varies in size as the size of the hand varies. As the size of ROI varies, we can extract more features from larger hand. We've proposed a new thinning algorithm which involves thinning the ROI to one pixel thick medial axis vein network and also removal of thinning artifacts. We used a sequential thinning algorithm for thinning. We used a flag map in addition to image to decide which pixel to delete during thinning process. An artifact removal algorithm is used to remove any extra artifacts which ends the thinning process. After that key features are extracted which are bifurcations and end points which are stored separately as templates. Finally a matching algorithm is applied for identification. The matching algorithm is a coordinate matching where in we match the coordinate positions of key features of the live image which are stored

in live template with the stored templates with certain fixed threshold. Our system is efficient as it got the lowest FAR which is necessary for security.

Keywords: hand vein, finger web, region of interest, binarisation, bifurcation, endpoint, morphology, template.

1. Introduction

A biometric system is basically an automated pattern recognition system that either makes an identification or verifies an identity by establishing the probability that a specific physiological or behavioral characteristic is valid [1]. Various types of biometric systems are being used for real-time identification and identity verification applications. Some of the biometric systems generally used are ocular recognition, facial recognition, fingerprints, hand geometry and vein pattern recognition [5]. Hand vein identification has emerged as a promising component of biometrics study. The human vascular structure is individually distinct and appears to be time invariant. Human blood vessels are formed during the embryo stage with a variety of differentiating features, rendering each pattern unique, and their patterns remain relatively constant over one's lifetime.

A unique network of veins and arteries exists in every hand and finger of each human being. The subcutaneous vascular network appearing on the back of hand, referred to as the hand vein in this paper, is extremely difficult to forge and, therefore, offers promising biometric which also ensures liveness. VPR technology was developed in the 1990s, and the first commercialized system emerged in 2004.



Fig 1. Vein Pattern Image.

2. Related work

We started the literature survey with the book *Vein Pattern Recognition* authored by Chuck Wilson [1]. The book provides an objective comparison of the different biometric methods in common use today including fingerprint, eye, face, voice recognition and dynamic signature verification. Primarily the book focuses on the introduction of VPR technology in biometrics, the advantages VPR systems has on other biometric systems, implementation of VPR systems, discusses about various types of vein patterns depending on from where it is collected, like finger vein, palm

vein, hand vein etc. Ajay kumar et al [7] presents a new approach to authenticate individuals using triangulation of hand vein

images and simultaneous extraction of knuckle shape information. Their method employs palm dorsal hand vein images acquired from the low-cost, near infrared, contactless imaging. Crisan et al [8] takes a new approach where the vein detection process consists of an easy to implement device that takes a snapshot of the subject's veins under a source of infrared radiation at a specific wavelength. By storing the relative angles between segments, the number of terminations and crossing points and the neighbors of each point, the comparison between the template values and the scanned values is faster than in the case of using images and the amount of data stored is kept to a minimum. Lin Yang et al [14] presents an algorithm for segmenting the dorsal hand vein image and extracting the vein skeleton. Here Gaussian lowpass filter and median filter are used to eliminate the speck noise and the horizontal strip scanning noise respectively. Yuhang Ding et al [17] presents the theoretical foundation and difficulties of hand vein recognition. The threshold segmentation method and thinning method of hand vein image are explained. The method of hand vein image feature extraction based on end points and crossing points is studied initially, and the matching method based on distances is used to match vein images. Zhang and Suen [15] explains about a fast parallel thinning



algorithm. It consists of two subiterations :
one aimed at deleting the south-east
boundary points and the north-west corner
points while the other one is aimed at
deleting the north-west boundary points
and the south-east corner points.

3. Proposed System

3.1. Segmentation and Thinning

After obtaining the ROI, binarisation of ROI is done. But before binarisation of the ROI, image enhancement techniques are applied on the ROI in order to increase the quality of the ROI, as the images are low quality far infrared images. The first step in enhancement is applying histogram equalisation technique [2] on the ROI.

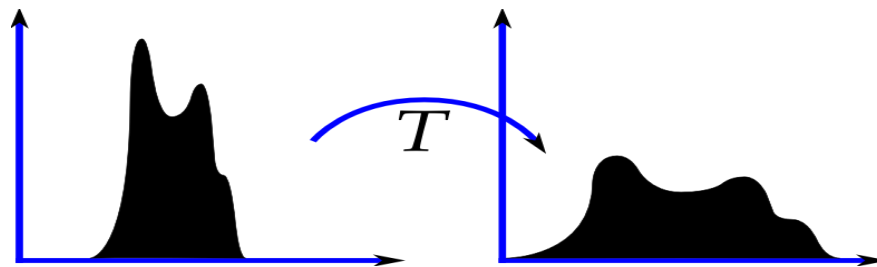


Fig 2. Histogram of an image before and after applying Histogram equalisation technique.

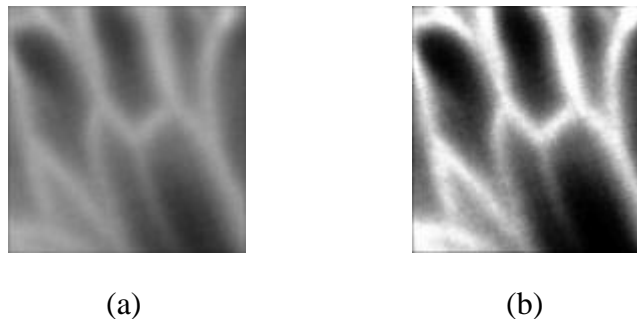


Fig 3. (a) Extracted ROI (b) ROI after Histogram equalisation

The resultant ROI after histogram equalisation is subjected to adaptive thresholding [2]. Thresholding is called adaptive thresholding when a different threshold is used for different regions in the image. Whereas the conventional [thresholding](#) operator uses a global threshold for all pixels, adaptive thresholding changes the threshold dynamically over the image [10].

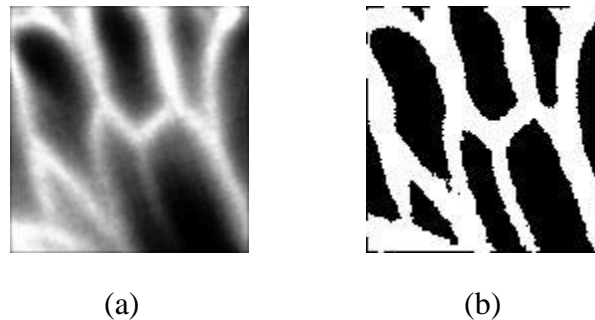


Fig 4. (a)ROI before thresholding (b) After thresholding

After thresholding process a noise reduction technique is applied on the ROI. Connected regions are identified in the obtained binarized ROI. Using a threshold value, regions are eliminated which are below this threshold value thereby reducing noise.



Fig 5. (a)ROI Before reducing noise (b)ROI after reducing noise

There are some obvious protuberances and many spurs at the boundary of vein pattern, especially in horizontal. Median filter of size 5×5 is used to reduce that noise. The median filter is a nonlinear [digital filtering](#) technique, often used to remove [noise](#)[9]. It preserves edges while removing noise.

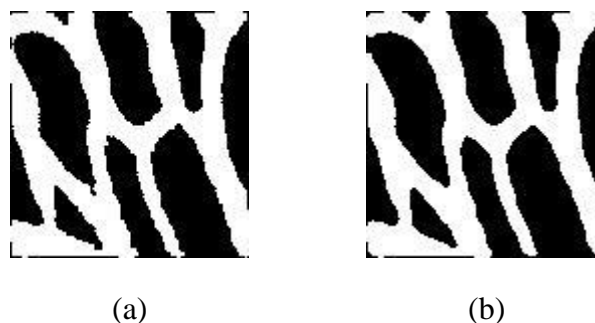


Fig 6. (a)ROI before filtering (b)ROI after filtering

Morphological opening [2] operation generally smoothes the contour of object, breaks narrow isthmuses, and eliminates thin protrusions. An opening is defined as an erosion followed by a dilation using the same structuring element for both operations. The opening operator therefore requires two inputs, an image to be opened, and a structuring element. the structuring element used is a diamond operator.

$$A \circ B = (A \ominus B) \oplus B$$

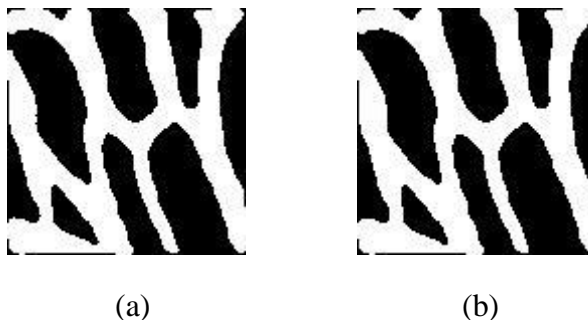


Fig 7. (a)ROI before opening (b)ROI after morphological opening

The main objective of thinning is to reduce data storage while at the same time retaining its topological properties and also to reduce transmission time as well as to facilitate extraction of morphological features from the digitized pattern. Thinning reduces the amount of data to be stored by transforming a binary image into a skeleton [15].

Ideally a skeleton must be topologically equivalent to the object. It must run along the medial axis of the object and must be one pixel thick. Maintaining connectivity and ability to handle boundary noise are the properties desired in thinning algorithms.

A skeleton should possess the following characteristics.

- Connectivity should be preserved.

- Excessive erosion should be prevented. The end point of a skeleton should be detected as soon as possible so that the length of a line or curve that represents a true feature of an object is not shortened excessively.
- The skeleton should be immune to small perturbations in the outline of the object. Noise or small convexities which do not belong to the skeleton, will often result in a tail after thinning. The length of these tails should be minimized. For this purpose deletion of loop points which will introduce extra holes in the skeleton, should be avoided.

In most existing thinning algorithms, when a pixel is decided as boundary pixel, it is either deleted or flagged and not deleted

until the whole image has been scanned. In the former case deletion of each boundary pixel will change the object in the image and hence affect the final skeleton. This kind of algorithms doesn't thin the object symmetrically. If the image is scanned in a row wise manner from left to the right, the thinned object lines are then located towards south eastern borders of the image, because the north western border points are removed first. In the latter case, as the pixels are only

flagged, the state of the bitmap at the end of the last iteration will be used in deciding which pixel to delete. If this flag map is not used to decide on whether to delete the current pixel then the information generated from processing the previous pixels in the current iteration will be lost. The final skeleton may be badly distorted. For example a two pixel wide line will be completely deleted.

In the algorithm we used, the image and the flag map are used together to decide on which pixel to delete. When examining a pixel and its neighbours to decide if it is to be flagged as a boundary pixel, both the flag map and the result from the last iteration are used in the decision making process.

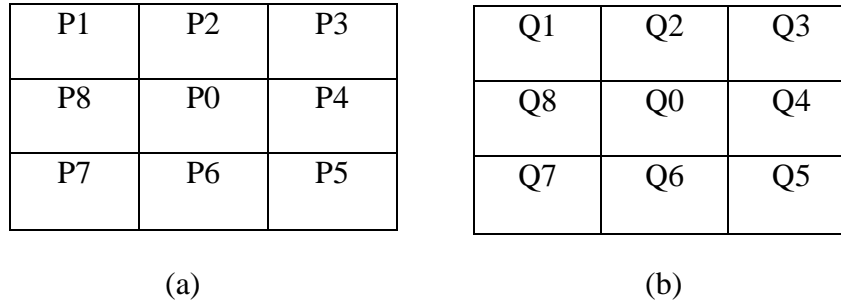


Fig 8. (a) Pixel notation for bitmap (b) Pixel notation for flagmap.

The flag map will be used for flagging those pixels that will eventually be deleted. The size of the flag map is the same as that of the image. Initially all pixels in the flag map are set to one. The value will be changed to zero as soon as that pixel is flagged.

Three functions are used in the algorithm. They are

- Previous Neighborhood function PN
- Current neighborhood function CN, and
- 0→1 transition function, Trans

The previous neighborhood function PN is defined as

$$PN(P_0) = \sum_{i=1}^8 P_i$$

It counts number of previous neighbors of pixel P_0 , that is the number of neighbors in the bitmap from the last iteration.

The current neighborhood function CN is defined as follows

$$CN(P_0) = \sum_{i=1}^8 P_i \times Q_i$$

It counts the number of current neighbours of pixel P_0 . If one foreground pixel in the neighbourhood of P_0 has been flagged, then it can no longer be considered to exist.

The 0→1 transition function Trans is defined as

$$Trans(P_0) = \sum_{i=1}^8 Count(P_i)$$

$$Count(P_i) = 1, \text{ iff } (P_i \times Q_i) = 0 \&\& (P_{i+1} \times Q_{i+1}) = 1.$$

$$= 0, \text{ otherwise.}$$

$$P_9 = P_1, Q_9 = Q_1$$

It gives the number of 0→1 transitions when traversing across the 8 neighbours of P_0 .

The thinning algorithm is described as follows

Step 1: For each pixel in bitmap do the following

- 1) Count $PN(P)$, $CN(P)$, $Trans(P)$
- 2) If P is a white boundary pixel, that satisfies condition 1 and (condition 2 or condition 3), flag it.
 Condition 1: $CN(P) > 1 \&\& CN(P) < 6$
 Condition 2: $Trans(p) == 1$
 Condition 3: P and its neighbours match one of the smoothing templates.

Step 2: Delete the flagged pixels

Step 3: Repeat step 1 and step 2
no pixel can be deleted.

x	1	0
0	1	1
0	0	0

0	1	0
0	1	1
0	0	x

until

0	0	x
0	1	1
0	1	0

0	0	0
0	1	1
x	1	0

0	0	0
1	1	0
0	1	x

X	0	0
1	1	0
0	1	0

0	1	0
1	1	0
x	0	0

0	1	x
1	1	0
0	0	0

Fig 9. Smoothing Templates

If the condition $Trans(P_0) = \min(CN(P_0), 8 - CN(P_0))$ is true, that means the central pixel P is a break point. The removal of break point will damage the connectivity of the image and hence must be preserved.

1	1	1
0	1	0
1	0	1

Fig 10. Central Pixel P is a break point

The thinned image still contains many artefacts. The artefacts must be removed after which we will get perfectly thinned image. In order to remove the artefacts, first the end points and bifurcation points[16] must be located in the thinned image. Then the following algorithm is used.

1. Initialise the values of length, current pixel, previous pixel and current vector to zero.
2. Start at any end point. Let it be current pixel
3. Check the neighbors of the current pixel in the 8 neighborhood, whose value is 'one', other than the previous pixel.
4. If the neighboring pixel is another end point, go to step 1, else if the neighboring pixel is any bifurcation point, check the value of length variable. If the value is greater than predefined threshold, then the pixel values whose coordinates are present in current vector are made to zero. Go to step 1.
5. If the neighboring pixel is neither end point nor bifurcation point, increase the value of length variable. Add the coordinates to Current vector. The current pixel now becomes previous pixel and this neighbor pixel is current pixel. Go to step 3..
6. The procedure should be followed until all the end points are scanned.

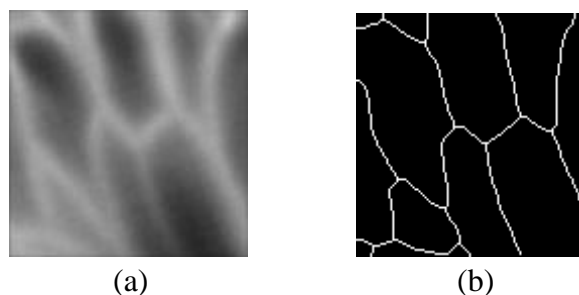
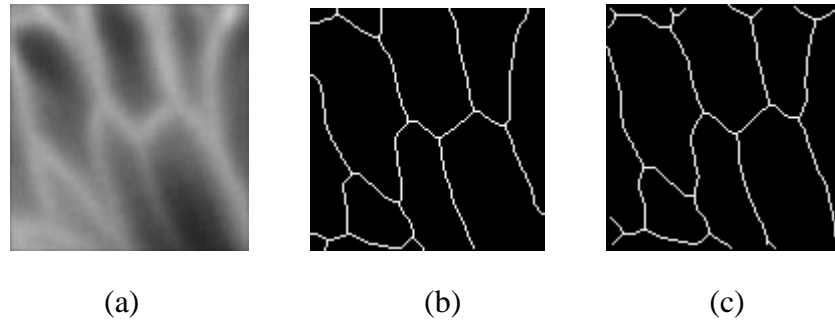


Fig 11. (a) ROI before thinning (b) ROI after thinning and artefact removal

Our algorithm gives the thinned image which is much better compared to the thinned image computed using the algorithm in MATLAB. The thinned image using our algorithm and using MATLAB is shown in the figure below.



**Fig 12. (a) ROI before thinning (b)Thinned image using our algorithm
(c)Thinned image using MATLAB**

3.2. Feature Extraction and Matching

After thinning the ROI, a one pixel thick medial axis of the vein network is obtained from the ROI. After this step, feature extraction is done and the resultant templates are stored. The final step is matching algorithm which is used to authenticate the user.

The extracted vein pattern is firstly used to locate the key points that are relatively stable, unique and repeatable. Vein endings and bifurcations are considered as key features. A vein bifurcation is defined as vein point where vein forks or diverges into branch veins, and the vein ending is the point at which vein ends or disappears abruptly[7]. This disappearance could be due to the abrupt ending of blood vessels or their poor visibility from the imaging system. In order to extract the vein endings and bifurcation points we examine the connectivity of every pixel and determine the crossing number $R(m)$ for every pixel m . The crossing number $R(m)$ [6] is the sum of differences between pairs of adjacent pixels in 3×3 window centered at m .

$$R(m) = \sum_{z=1}^8 |val(m_{z \bmod 8}) - val(m_{z-1})|$$

The pixel m with $val(m)=1$ corresponds to vein ending point if $R(m)=2$, and corresponds to vein bifurcation point if $R(m) \geq 6$.

Matching is the last phase in the vein pattern authentication system. Matching is used to authenticate the user, to know whether the user is already registered user by comparing the live template[1] generated by the user to the stored templates.

Matching is done by comparing the coordinates of the bifurcations and end points of the thinned ROI of the hand which are generated and stored in a template, with the already stored templates. A 1 to n matching is made for identification.

While comparing a live template with the stored template, the coordinate values present in the live template are searched for in the stored template. The threshold is fixed at two pixels, which means that if the difference between the coordinates is one or two pixels, then they are considered as matched. For every coordinate matched, a count value is increased. After that weight is calculated which is the ratio of number of coordinates matched to the total number of coordinates.

$$B_w = b_{\text{count}} / b$$

B_w = Bifurcation weight, which is the ratio of matched bifurcations to total bifurcations.

b_{count} = Total number of bifurcations matched between live template and stored template.

b = Total number of bifurcations obtained for the live ROI

In the similar fashion E_w is also calculated which is the Endpoint weight. The expression is given as

$$E_w = e_{\text{count}} / e$$

The final step is to calculate the matching percentage, by taking into consideration both B_w and also E_w .

We cannot give equal weight to both B_w and E_w in calculating the matching percentage. The reason is that end points can be obtained as a result of noise created while applying various image processing algorithms on the ROI[7]. So E_w is given less weight compared to B_w in calculating matching percentage.

$$\text{Matching Percentage} = (\text{weight 1} \times B_w + \text{weight 2} \times E_w) \times 100$$

$$0 \leq \text{weight1} \leq 1$$

$$0 \leq \text{weight2} \leq 1$$

$$\text{weight1} > \text{weight2}$$

The matching percentage is calculated for all the stored templates in comparison with the live template until the percentage is greater than or equal to a fixed threshold which confirms the identification.

4. Experiments and results

All the coding was done in MATLAB version 7.0 and the toolbox used in the MATLAB is Image Processing toolbox[19]. Identification is done by fixing the parameters for the matching algorithm as follows.

The threshold for the coordinate matching is fixed as 2 pixels which implies that the pixel coordinate if present anywhere in the window shown below is considered as matched.

x	x	X	x	x
x	x	X	x	x
x	x	P	x	x
x	x	X	x	x
x	x	X	x	x

Fig 20. Pixel coordinate positions

The weights given for bifurcations and end points while calculating matching percentage is as follows: weight given for bifurcations (weight1) is 0.8, weight given for end points (weight 2) is 0.2. The threshold for matching percentage is fixed as 80%. The FAR and FRR obtained for identification is as follows:

$FAR(\text{False Acceptance Rate}) = \text{Number of false acceptances} / \text{Number of attempts}$

$FRR(\text{False reject Rate}) = \text{Number of false rejects} / \text{Number of attempts}.$

The obtained values are FAR=0%, FRR=4.8%.

5. Conclusion

This paper proposes a new dynamic ROI extraction method and an advanced thinning algorithm with artefact removal included for vein pattern authentication. Our dynamic ROI extraction made it possible to grab more features from the hand and our thinning algorithm gave a perfect one pixel thick medial axis which was very helpful in feature extraction. We got a FAR which is very low which is perfect for security applications.

References:

1. Chuck Wilson. "Vein Pattern Recognition : A Privacy-Enhancing Biometric". Taylor & Francis Group, CRC Press, 1st edition, March 10, 2010.
2. Gonzalez, R.C. and Woods, R.E. "Digital Image Processing". Prentice Hall, Upper Saddle River, NJ, Second edition, 2008.
3. Gonzalez, R.C., Woods, R.E. and Eddins, S.L. "Digital Image Processing using MATLAB".

- Prentice Hall, Upper Saddle River, NJ, Second edition, 2010.
4. Kavati, I., Prasad, M.V. and Bhagvati, C., Vein pattern indexing using texture and hierarchical decomposition of Delaunay triangulation. In *International Symposium on Security in Computing and Communication* (pp. 213-222). Springer Berlin Heidelberg. 2013
 5. A. K. Jain, P. Flynn and A. Ross. "Handbook of Biometrics", New York: Springer, 2007.
 6. D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. "Handbook of Fingerprint Recognition". New York: Springer, 2003.
 7. Ajay Kumar and K. V. Prathyusha, "Personal authentication using hand vein triangulation and knuckle shape," *IEEE Trans. Image Processing*, vol. 38, pp. 2127-2136, Sep. 2009.
 8. S. Crisan, I. G. Tarnovan, and T. E. Crisan, "Radiation optimization and image processing algorithms in the identification of hand vein patterns," *Computer Standards and Interfaces*, vol. 32, no. 3, pp. 130 -140, 2010.
 9. C.L. Lin and K.C. Fan, "Biometric verification using thermal images of palm-dorsa vein patterns," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 2, pp. 199–213, Feb. 2004.
 10. L. Wang and G. Leedham, "A thermal hand-vein pattern verification system," in *Pattern Recognition and Image Analysis*, S. Singh, M.Singh, C.Apte, and P. Perner, Eds. New York: Springer, 2005, vol.3687, pp. 58-65.
 11. L.Wang, G. Leedham, D. Siu Yeung Cho, "Minutiae feature analysis for infrared hand vein pattern biometrics" in *Pattern Recognition* 41/3/pp.920-929 (2008).
 12. Prasad, M.V., Kavati, I. and Ravindra, K. Hand vein authentication system using dynamic ROI. In *International Symposium on Security in Computing and Communication* (pp. 203-212). Springer Berlin Heidelberg. 2013.
 13. <http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html>.
 14. Lin Yang, Xiangbin Liu, Zhicheng Liu. "A Skeleton Extracting Algorithm for Dorsal Hand Vein Pattern", *International Conference on Computer Application and System Modeling (ICCASM 2010)*, Page: V13-92 - V13-95.
 15. T.Y.Zhang and C.Y.Suen, "A fast parallel algorithm for thinning digital patterns", *Communications of ACM* 27, (3), pp. 236-239, 1984.
 16. S. Zhao, Y. Wang, and Y. Wang, "Biometric verification by extracting hand vein patterns from low-quality images," in *Proc. 4th ICIG*, Aug. 2007, pp. 667–671.
 17. Y. Ding, D. Zhuang, and K. Wang, "A study of hand vein recognition method," in *Proc. IEEE Int. Conf.*



Mechatronics & Automation ,
Niagara Falls, Canada, Jul. 2005, pp.
2106–2110.

18. Chih-Lung Lin, Thomas C. Chuang,
Kuo-Chin Fan, “ Palmprint
Verification using hierarchical
decomposition”, Pattern Recognition
38, No. 12, December 2005, pp.
2639-2652.

19. <http://www.mathworks.com>