

Umeå University
Department of Applied
physics and electronics

Bachelor thesis
(C Level)

Hand Reader

A hand Geometry Based Personal Verification System

Sani Anani
baiani02@student.umu.se

2010-12-01

Supervisor: Haibo Li

Abstract

This report presents my bachelor thesis performed at the Department of Applied Physics and Electronics at Umeå University. The Digital Media Lab was the assigner, supervising and assisting me during the project.

A simple and smart system to verify a person's identity using hand geometry as a biometrical identity is developed. The system should tolerate some deviation in hand placement and will not use pins or pegs. A complete system, hardware and software, is delivered. A webcam based hand-scanner is constructed and designed to operate on a PC utilizing Matlab for image processing.

Sammanfattning

Denna rapport beskriver mitt examensarbete utfört vid institutionen för Tillämpad Fysik och Elektronik vid Umeå Universitet. Uppdragsgivare var Digital Media Lab som har stöttat och assisterat mig under hela projektet.

Målet med projektet var att utveckla ett system som ska verifiera en persons identitet genom att använda handens geometri som en biometrisk identitet. Systemet skall tolerera några avvikelser i handplaceringar och kommer inte att använda stift eller pinnar. Systemet består även av olika bildbehandlingsmetoder programmerade i Matlab. Systemet är byggt för att fungera på en PC.

Keywords

Biometrics, Hand Geometry, Peg free, Digital Image Processing, chain coding, Dynamic Programming, verification, identification.

Table of contents

1	Introduction	4
1.1	Background	4
1.1.1	Biomedica	5
1.2	Problem Description	5
1.3	Biometrics	6
2	Hand Reader	7
2.1	Hand geometry	7
2.2	The hand verification system	9
3	Method	11
3.1	Theory	11
3.1.1	Binarization	12
3.1.2	Feature extraction	13
3.1.3	Chain coding	14
3.1.4	Normalization	15
3.1.5	Dynamic programming	16
3.1.6	Applying DP	18
4	Experimental Results	23
5	Concluding Remarks	26
6	Acknowledgement	27
7	References	28

1 Introduction

This is the final report for the bachelor thesis where the aim is to develop a system for verifying a person's identity using hand geometry. The project was developed at the Department of Applied Physics and Electronics, University of Umeå.

1.1 Background

Biometrics is being used more and more in security systems for identifying or verifying persons especially after the September 11 attacks in New York 2001. It is said that the 9/11 attacks has changed the world, at least that's the case in biometrics security applications.

Biometrics security systems are pattern recognition systems using components and methods common to those in machine vision systems (cameras, A/D, edge detection, thresholding, etc.) to compare features of a person's biological or behavioral characteristics to stored features in a database in order to confirm their identity.

Such a method of identification requires the person's physical presence at the point-of-identification. Identification based on biometric techniques obviates the need to carry a security pass card.

In this work the hand geometry is used. It involves the measurement and analysis of the shape of one's hand.

Hand geometry as a biometric identity is considered to be accepted by users, not associated with criminology as fingerprints and easy to perform. It is useful in civil applications where a reasonable level of security is satisfactory. The aim of the study is to build a low cost, reliable personal verification system based on the hands geometry. The intended system is to be peg free and accept some variation in hand placement.

If a false acceptance takes place, no catastrophe will occur, but possibly a minor economic loss. In case of a higher security requirement a pin-code can be used as in the case of a key card and a pin-code, in this case the hand would be a key card that you never forget at home because it follows you where ever you go.

It also requires of the person to be physically present at the point-of-verification which guaranties that only he can get access.

1.1.1 Biomedica

While studying at the Department of Applied Physics and Electronics I studied an advanced level course “Biomedica” where I was introduced to biometrics and the interaction between humans and computers. I became interested in the subject which led me to do this project. In this project I have used my knowledge which I have gained from courses like Biomedica, Image Processing and Media Signal Processing.

1.2 Problem Description

The aim of the study is to build a low cost, reliable personal verification system based on the hands geometry as a biometric identity. The system should tolerate some deviation in hand placement and will not use pins or pegs.

To be able to measure the similarity/difference between geometrical characteristics of two hands, in most commercially available systems the hand should always be placed at the same position and to accomplish that the systems on the market use pegs to guide the hands into the correct position.

The challenge is to compensate for the absence of the peg oriented hand placement to make sure that any comparison is only made between two corresponding points in two hands.

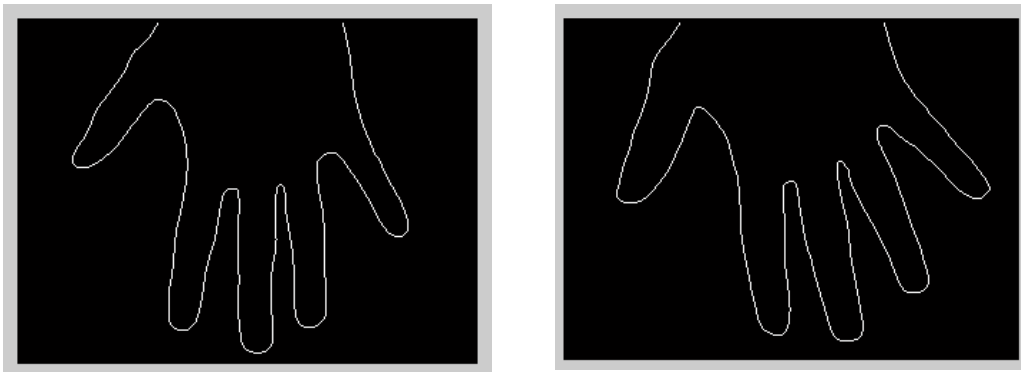


Figure.1. Two different enrolments of on hand.

Already existing systems/solutions use pegs to ensure that the hand is placed at the exact same place making the signal processing less complicated when it concerns feature extraction.

The challenge in the case of the present study is to have a peg free system that is able to process the extracted information even if the person can place the hand more freely meaning that it's not sensitive to some changes in the placement of the hand from time to time.

1.3 Biometrics

Biometrics refers to both characteristics of humans and a process where that specific human characteristic is processed. A biometric is a measurable biological (anatomical and physiological) characteristic such as fingerprints, hand and facial features, and facial patterns or behavioral characteristics learned or acquired, such as how a person signs his name, utilizes the keyboard or even walks. The term is derived from the Greek words bio (= life) and metric (= measure).

As a process a biometrics is an automated method of recognizing an individual based on measurable biological (anatomical and physiological) and behavioral characteristics. ^[1]

Biometrics is a rapidly evolving technology which has been widely used in forensics such as criminal identification and prison security. Recent advancements in biometric sensors, matching algorithms and the increasing need of more reliable and effective security methods have led to the deployment of biometric authentication in a large number of civilian applications. Biometrics can be used to prevent unauthorized access to premises ATMs, cellular phones, laptops, desktop PCs, workstations, and computer networks. It can be used during transactions conducted via telephone and Internet (electronic commerce and electronic banking). Biometrics can replace keys with key-less entry and key-less ignition. Due to increased security threats, many countries have started using biometrics for border control and national ID cards.

A biometric system is essentially a pattern recognition system which makes a personal identification by determining the authenticity of a specific physiological or behavioral characteristic possessed by the user ^[2]. An important issue in designing a practical system is to determine how an individual is identified. Depending on the context, a biometric system can be either a verification (authentication) system or an identification system.

Examples of physiological characteristics are fingerprints, retina and iris patterns, facial characteristics, hand geometry, palm print, signature and vein geometry. Examples of behavioral characteristics are voice, signature and keystroke dynamics.

There are two different ways to resolve a person's identity, verification and identification. Verification tackles the question (Am I whom I claim I am?) which involves confirming or denying a person's claimed identity. In identification, one has to establish a person's identity (Who am I?).

2 Hand Reader

2.1 Hand geometry

Hand geometry is a biometric where the user's identity is verified by measurements of the spatial geometry of the hand and finger and doesn't include prints of the palm or fingers.

Hand geometry systems measure a user's hand along many dimensions making each human hand unique. The length of the fingers, the width of the fingers/hand and contour/edges of the hand, the thickness and surface area are all characteristics that can be extracted to use in a verification process. Hand geometry is occasionally misunderstood as "palm reading" does not involve extraction of detailed features of the hand like skin patterns unlike palm print technologies and fingerprints. In IR based systems the feature to be extracted is the vein pattern.

It is well known that the individual hand features themselves are not very descriptive and that hand geometry authentication has relatively high False Acceptance Rate (FAR) and False Rejection Rate (FRR) rates. Devising methods to combine these non-salient individual features to attain robust positive identification is a challenge pattern recognition problem in its own right. Yet despite these error rates, hand recognition systems are surprisingly widespread, a testament to their user-friendliness.^[3]

When the features in interest are extracted after the enrolment of a hand they are compared to those measurements saved in a database.

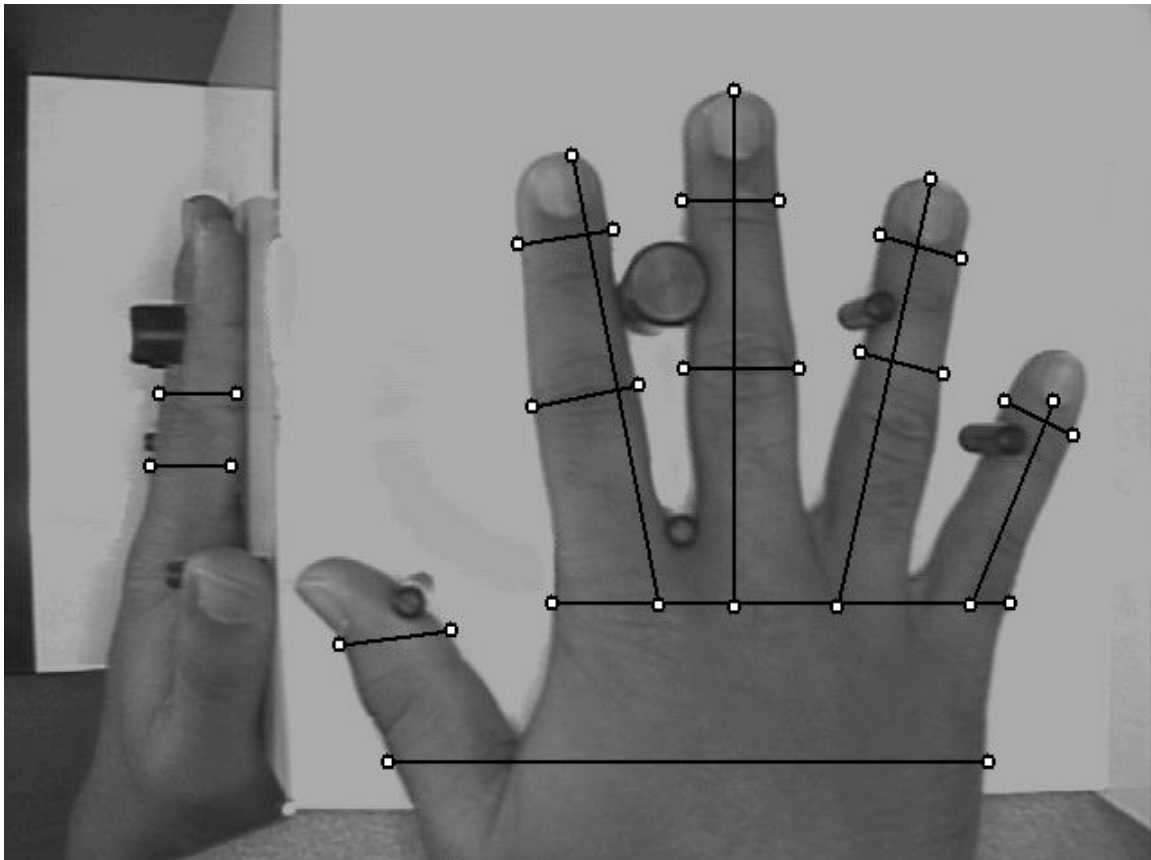


Figure.2 Several measurements that can be extracted from the hand^[4]

This method of identification is preferred over traditional methods involving passwords and PIN numbers, because biometrics requires being physically present at the point of identification and because identification based on biometric techniques eliminates the need to remember a password or carry a token.

Hand geometry as a biometric identity could be seen as more user friendly for people than fingerprint because it's not associated with criminology as fingerprints. Fingerprints are usually linked to forensics in general which can make it sensitive to give out your fingerprints.

2.2 The hand verification system

The hand verification system designed in this project consists of two parts: a hardware part and a software part.

The first part is the hardware part for capturing the image of the hand during enrolment. Enrolment is the process of capturing an image of a user's hand template. The second part of the system consists of a software part for hand feature extraction written in Matlab code.

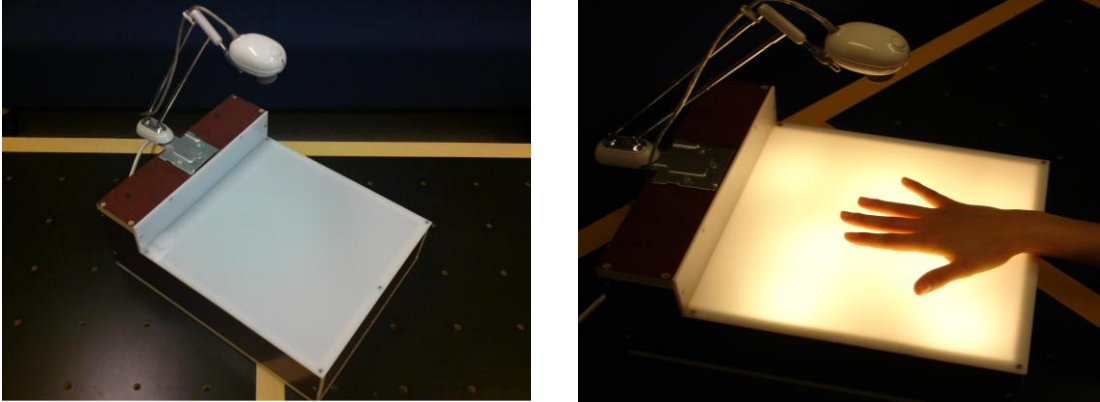


Figure.3

The hardware part consists of a wooden box containing low energy lamps. The lamps are installed in a way to make the light emitted as uniform as possible to avoid defects in the acquired image, the box is then covered with white Plexiglas. When a person as in Fig.3 places his hand on the lightened surface a web camera takes the image which will later be processed. Making the image processing easier requires a uniform light to make the edges of the hand more clearly visible.

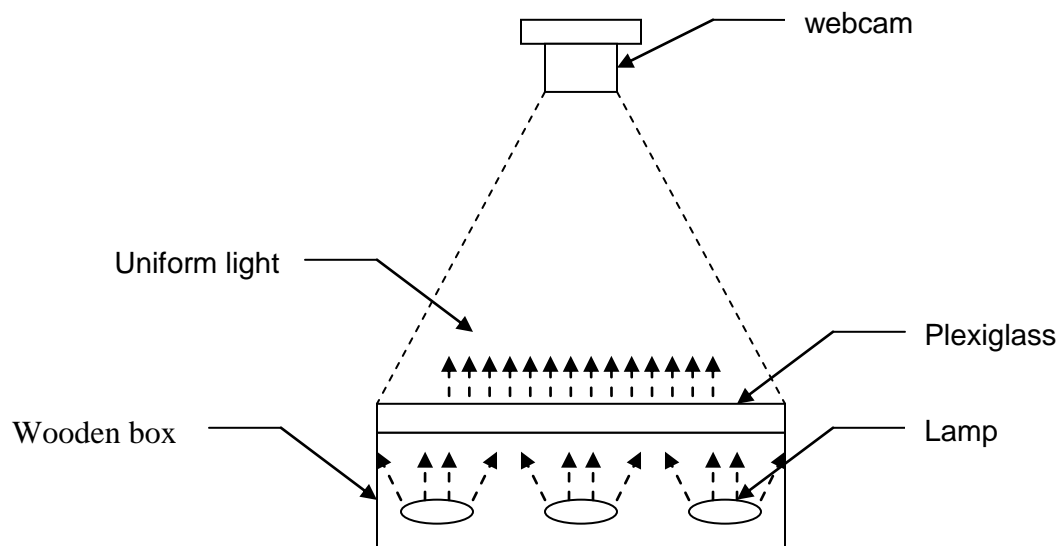


Fig.4 Schematic presentation of the hand scanner

A database contains a number of templates for a number of users. When a user wants for example to enter a facility with this system he places the hand for scanning and the features of the hand are extracted to be compared to his template from the database.

The challenge here is as shown in the picture is that there are no pegs to fix the placement of the hand. This means that the person has a little bit more freedom to place his hand for example one might place his hand rotated 10 degrees to the left or right relative to the first time an image of his hand was captured. Of course every system has its logical limits for example one can't place a hand where one of the fingers or a part of the hand is outside the range of the camera or putting two fingers together.

In the case where pegs are used, then all what is needed is to compare the features of the enrolled hand with the features of the template because in that case the only change would be that the hands are different. In the pegs free case there will always be a change even for the same person's hand due to that the possibility of the user to place the hand exactly the same way as the first time is small. To enable a correct calculation of the similarity/difference between the two images we have to extract the feature from the same place that which is correspondent in both hands. This is solved with Dynamic Programming.

If for example a man claims he is Eric. He enrolls by placing his hand on the scanner for feature extraction. When the feature is extracted it's processed and then compared to Eric's template from the database. If the difference between the two does not exceed a certain threshold then we have a confirmed identity and if not then this person is not who he claims to be.

3 Method

3.1 Theory

To be able to verify a users hand the system must have had scanned his hand to create the template which will be saved in a database. The next time his hand is enrolled, it can be compared to an already existing template. When an image of the hand is captured by the camera it will be a colored image. This image will go through a number of steps before becoming a template. First the system wants to get rid of the noise in the picture by applying a filter. The image is then converted from a colored image to a grayscale-image and then binarized to be black and white. The edge is detected and chain coded to a vector which will result in a template. After doing that with several users the database will consist of a number of templates representing several users. When a person claims an identity he places his hand and it will go through the same steps so it can be compared to the template of the claimed identity.

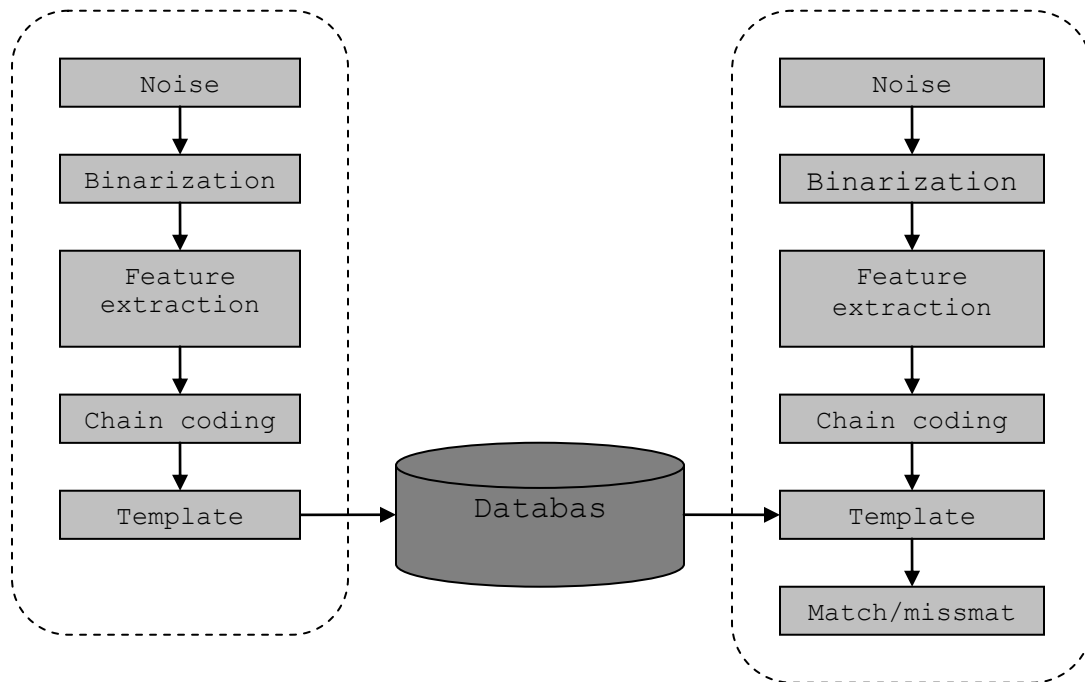


Figure.5

3.1.1 Binarization

When placing the users hand on the Plexiglas an RGB image of the hand is acquired. RGB is a color model and it stands for (Red Green Blue). These three colors are mixed together in different intensities to get different colors. Every color has the intensity scale 0-255. All three colors combined with full intensity gives the color white. When all colors have the intensity 0 the result will be the color black. See the following table for more examples.

Red	Green	Blue	Result
255	255	255	White
255	0	0	Red
0	255	0	Green
0	0	255	Blue
0	0	0	Black
255	255	0	Yellow
255	0	255	Magenta

Table.1

The image is converted in Matlab from a colour image to a greyscale image. A greyscale image contains 256 gray levels. Each pixel in the image consists of an intensity value from 0 to 255.

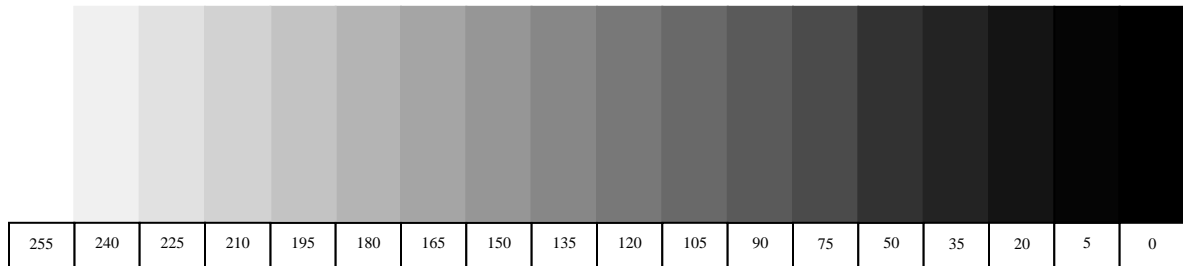


Figure.6

Converting the gray-scale image to a strictly black and white image is called binarization. In a binary image a pixel can only have one of two values, one and zero. A black pixel has the value “0” and a white pixel has “1”.

Binarization is done by choosing a threshold value for example 128. All pixel values above 128 are classified as white and all pixels values under 128 as black.

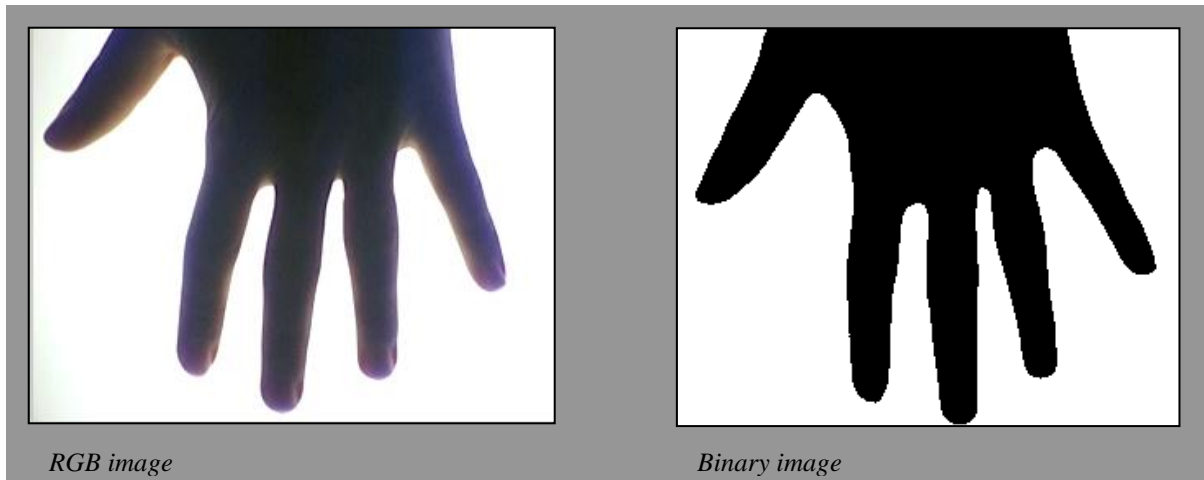


Figure.7

3.1.2 Feature extraction

The feature I need to extract in this image is the contour of the hand and to do that I need to apply an edge detection algorithm on the image. The image now has a white background and a black foreground. To extract the contour of the hand, the foreground should be only an edge of the hand and should only be one pixel thick. The rest of the image will be the background. I have chosen the color white to the foreground and black to the background, it doesn't really matter.

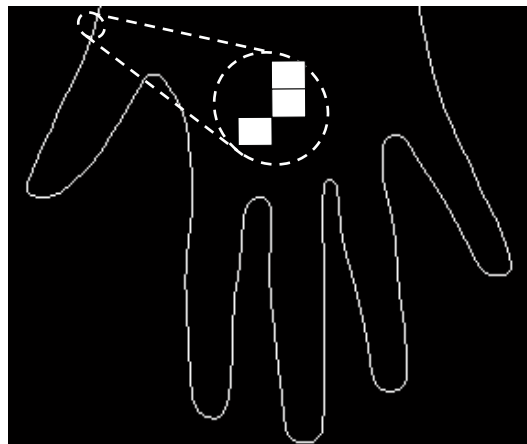


Figure.8

3.1.3 Chain coding

Chain codes are used to represent a boundary by a connected sequence of straight-line segments of specified length and direction ^[5]. This is based on the 8-connectivity of the segments (8-directional chain code). The change in direction when moving from one pixel to its neighbor is coded. As you see in *fig.9(a)* every pixel has 8 neighboring pixels. When the edge/boundary of the hand is detected it's followed in a clockwise direction while assigning a direction to the segments connecting every pair of pixels, forming the Freeman chain code.

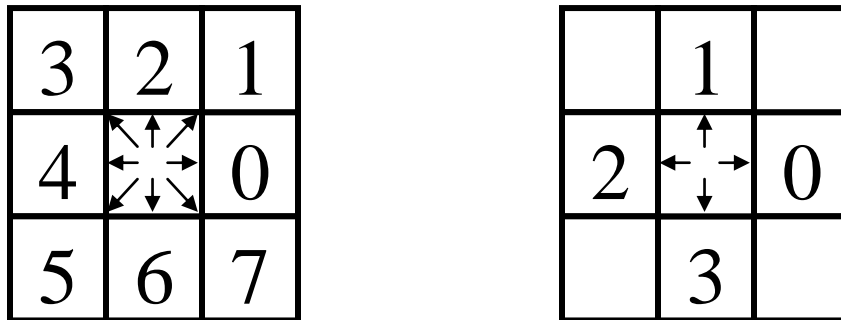


Fig.9 (a) Eight directional chain code (b) four directional chain code..

If we use the line in the figure below to the left as an edge we follow the change in direction of the path from the starting point to the end. The result will be an 8-direction chain code vector with the values {1, 2, 1, 2, 1, 3, 2}. The vector should have the same length as the number of white pixels representing the edge of the hand in the image minus the starting point pixel.

We can consider the image to the left as the template acquired from the first enrollment and the image to the right as the second enrollment. As mentioned before the hand will probably not be placed the exact same way as the first time. To show this I have rotated the edge 90 degrees to the left to give a simple demonstration of the arising problem that definitely needs to be solved. The chain code of the second figure is {3, 4, 3, 4, 3, 5, 4} which is apparently totally different from the first one in spite of the fact that they both represent the same line segment but one is tilted. So how can any calculation be done to see the degree of resemblance between the two to determine whether the two hands belong to the same user or not?

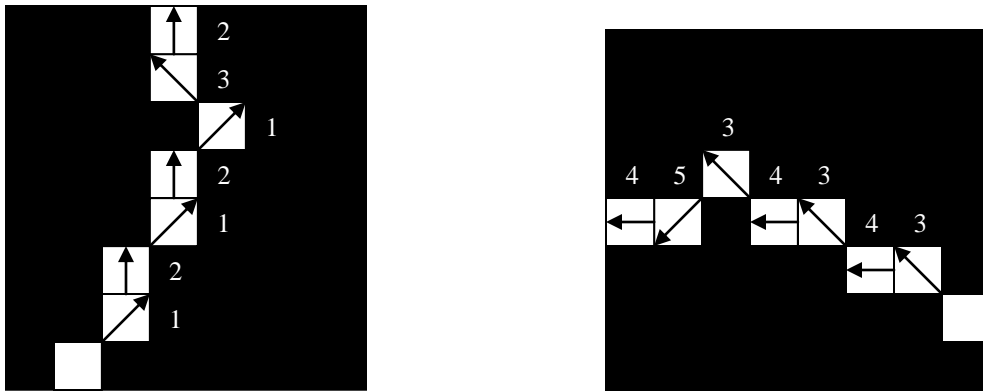


Figure.10

3.1.4 Normalization

Normalizing for rotation is done by using the “first difference” of the chain code instead of the chain code itself. This is computed by counting counter clockwise the number of directions between two adjacent elements in the chain code.^[7]

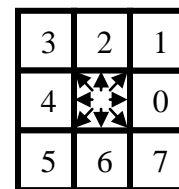
The difference in the first chain code between the first element (1) and the second (2) in direction is 1 step, the difference between the second element (2) and the third (1) is 7 steps. In the end the result will be a new representation of the edge looking like this {1, 7, 1, 7, 2, 7}. If we do the same for the rotated edge then the result is exactly the same {1, 7, 1, 7, 2, 7}. So now instead of doing the comparison between {1, 2, 1, 2, 1, 3, 2} and {3, 4, 3, 4, 3, 5, 4} it will be done between {1, 7, 1, 7, 2, 7} and {1, 7, 1, 7, 2, 7}. As we can clearly see the difference will be zero meaning that those two edges came from the same hand and we have a match, of course in reality there will be a difference although it's a match but the difference should not pass a certain threshold.

Chain code 1:
{1, 2, 1, 2, 1, 3, 2}

New code:
{1, 7, 1, 7, 2, 7}

Chain code 2:
{3, 4, 3, 4, 3, 5, 4}

New code:
{1, 7, 1, 7, 2, 7}



3.1.5 Dynamic programming

Dynamic programming “[solves] an optimization problem by caching subproblem solutions (memorization) rather than recomputing them.”^[6] This method is used in mathematics and computer science for solving complex problems by dividing them into smaller simpler problems/sub problems.

Originally the word “programming” came from the term “mathematical programming” a synonym for optimization.^[8]

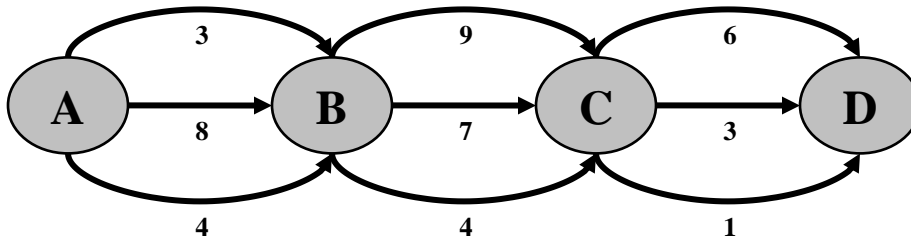


Figure.11 Directed acyclic graph (DAG).

Let’s consider the directed acyclic graph in figure.11. You are standing at station A and your destination is station D, in the way there are several stations and paths to take, each path has a cost. The question is how you get from A to D with minimum cost. One way to get the correct answer is called the greedy algorithm. The solution is $3+4+1 = 8$ which is correct. The greedy algorithm cannot be applied in all cases. We will now try to solve the DAG in figure.12 and see what happens. The first step from A is $\min(\text{dist}(E), \text{dist}(B), \text{dist}(F))$

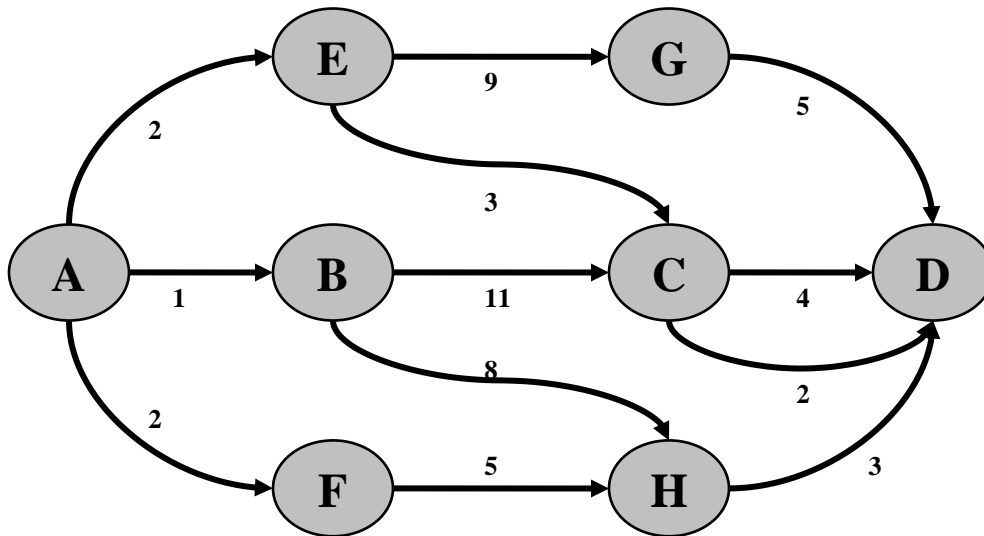


Figure.12 DAG.

I will use dynamic programming to find an approximate match to a pattern in a string. The pattern in this case will be the “template” which is the extracted feature of the first time a hand was enrolled to the system. The string would be the new enrolled hand that needs verification also called “target”.

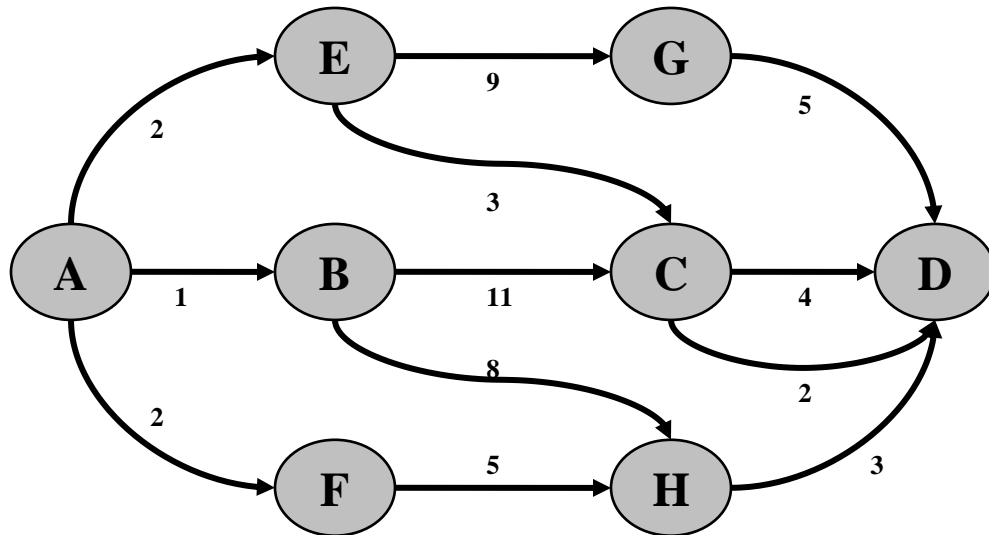


Figure.12 DAG.

In this section I will use the word “match” when describing in dynamic programming how the algorithm finds a sequence of numbers in the target similar to the template. A match in this case means that the algorithm found the part of the target hand that corresponds to the template so that measurements can be taken for comparing. The measurements to verify whether the template and target belong to the same person will be done in future work.

3.1.6 Applying DP

A matrix is created with the size of $m \times n$, where n and m are the lengths of the two strings. The image below to the left represents the first enrollment of the hand to the database (template) and the image to the right represents the hand enrolled for the second time for verification purpose (target). The feature extracted from the image to the left is saved after chain coding as a vector. The same is done with the second hand.

We want to see the difference between the two hands to be able to verify whether both hands belong to the same person or not.

Since the user has more freedom in placing his hand on the surface it will result in some difference not only in rotation but even when it comes to how much of the hand is placed within the frame of the camera/image. The user must be reasonable and place the most important part of the hand, which should contain all five fingers and most of the hand. The beginning of the hand from the wrist will be always out of range but how much will be different from time to time. I manually cropped the template from the first image giving me resulting in a template vector shorter than the vector of the second image where I take the whole chain code without cropping.

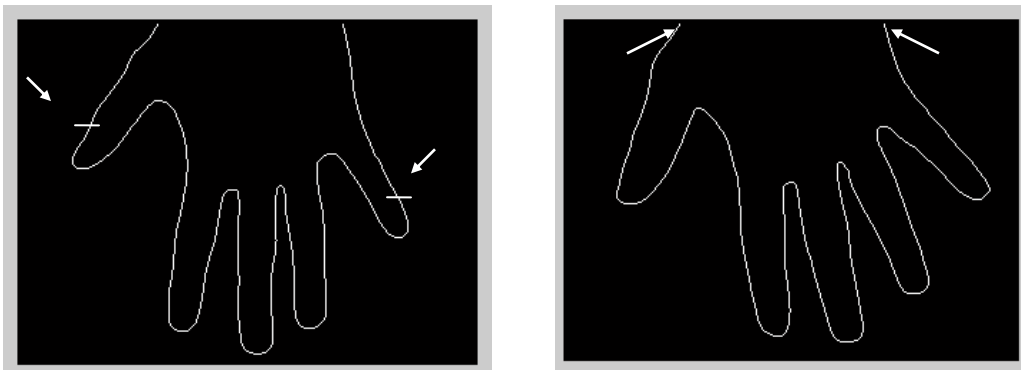


Figure.13

An example of the matrix is applied here with two vectors representing the template and the target. A matrix is created with size 15×8 and is initially filled with zeros as in fig.14

Template = $n = \{6, 7, 7, 7, 6, 6, 7, 6\}$

length = 8

Target = $m = \{7, 0, 6, 7, 7, 7, 6, 6, 7, 6, 5, 5, 6, 7, 0\}$

length = 15

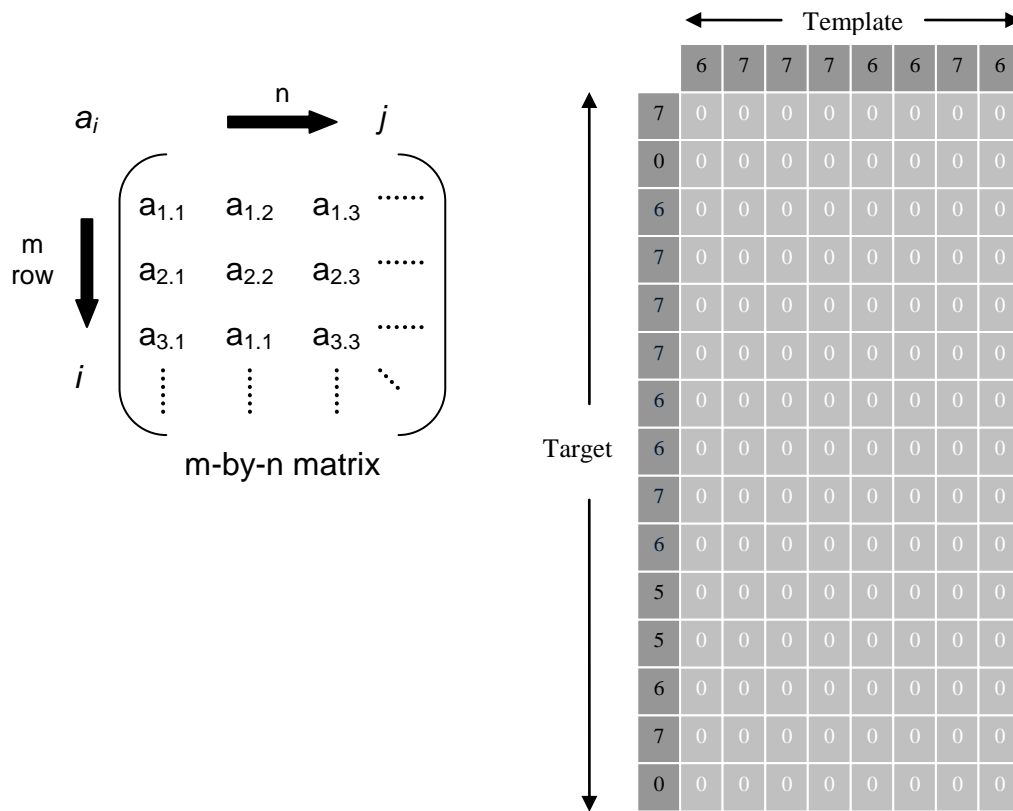
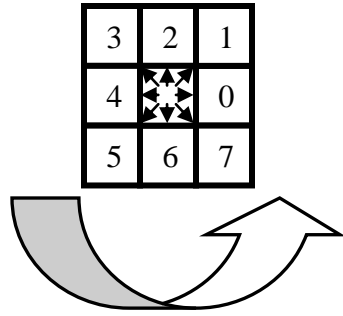


Figure.14

Step two is represented in fig.15 where the matrix is filled with the cost resulting from the difference between the chain code values when comparing the difference between the values in the template and target. The difference is calculated by counting counter clockwise the direction change needed to transform one direction into the other. Let's consider filling the first row $[a_{1,1} a_{1,2} a_{1,3} a_{1,4} a_{1,5} a_{1,6} a_{1,7} a_{1,8}]$. The first element in the row is $a_{1,1}$ and has the initial value zero. Because the first element of the first row $a_{1,1}$ corresponds to the first elements in vectors m and n it should have the value of the cost calculated between n_1 and m_1 . n_1 has the value 6 and m_1 has the value 7. The cost in direction changes is the steps counted clockwise from 6 to 7 which is 1 giving the same result in $a_{1,1}$ and $a_{1,8}$. Continuing to the second element $a_{1,2}$ gives the cost 0 because there is no change in direction since both m_2 and n_2 have the value "7" and the case is the same for $a_{1,3}$ $a_{1,4}$ and $a_{1,7}$. The result is $[1, 0, 0, 0, 1, 1, 0, 1]$ as presented in the matrix below.

Another example can be demonstrated to show a single case like $a_{11,2}$ where n_2 has the value 7 and m_{11} the value 5. For $a_{11,2}$ count the steps from 7 to 5 and don't forget it should be counter clockwise. The steps are 7, 0, 1, 2, 3, 4 and 5, so $a_{11,2}$ will have the value 6.



	6	7	7	7	6	6	7	6
7	1	0	0	0	1	1	0	1
0	2	1	1	1	2	2	1	2
6	0	7	7	7	0	0	0	0
7	1	0	0	0	1	1	0	1
7	1	0	0	0	1	1	0	1
7	1	0	0	0	1	1	0	1
6	0	7	7	7	0	0	7	0
6	0	7	7	7	0	0	7	0
7	1	0	0	0	1	1	0	1
6	0	7	7	7	0	0	7	0
5	7	6	6	6	7	7	6	7
5	7	6	6	6	7	7	6	7
6	0	7	7	7	0	0	7	0
7	1	0	0	0	1	1	0	1
0	2	1	1	1	2	2	1	2

Figure.15

The next step is to add total cost so that the last column will contain the total costs. Starting from $a_{2,2}$ the value is the summation between the current element $a_{2,2}$ and the minimum value of elements $a_{1,1}$ to the upper left and $a_{2,1}$ the element to the left. The calculation continues so that $a_{2,3}$ will have the value of $a_{2,3} + \min(a_{2,2}, a_{1,2})$. The path from which the value is added is saved. This is done to all elements starting from $a_{2,2}$ to all the elements in the sub matrix A {2:15;2:8}. The above can be described mathematically with the formula in (eq.1) where $l_m=15$ and $l_n=8$ in this case and in general the length of vector m respectively the length of vector n .

$$c(m, n) = a_{m,n} + \min(c_{m,n-1}, c_{m-1,n-1})$$

$$c(m, 1) = a_{m,1}$$

(eq.1)

The figure below 16(a) is the matrix before adding the total cost and the figure 16(b) is the result of adding the total cost. In order to find the match as we have to find the least cost in the last column in the matrix to the right. In this case I chose the values of both vectors so that the dynamic programming would find an exact match of the template in the target.

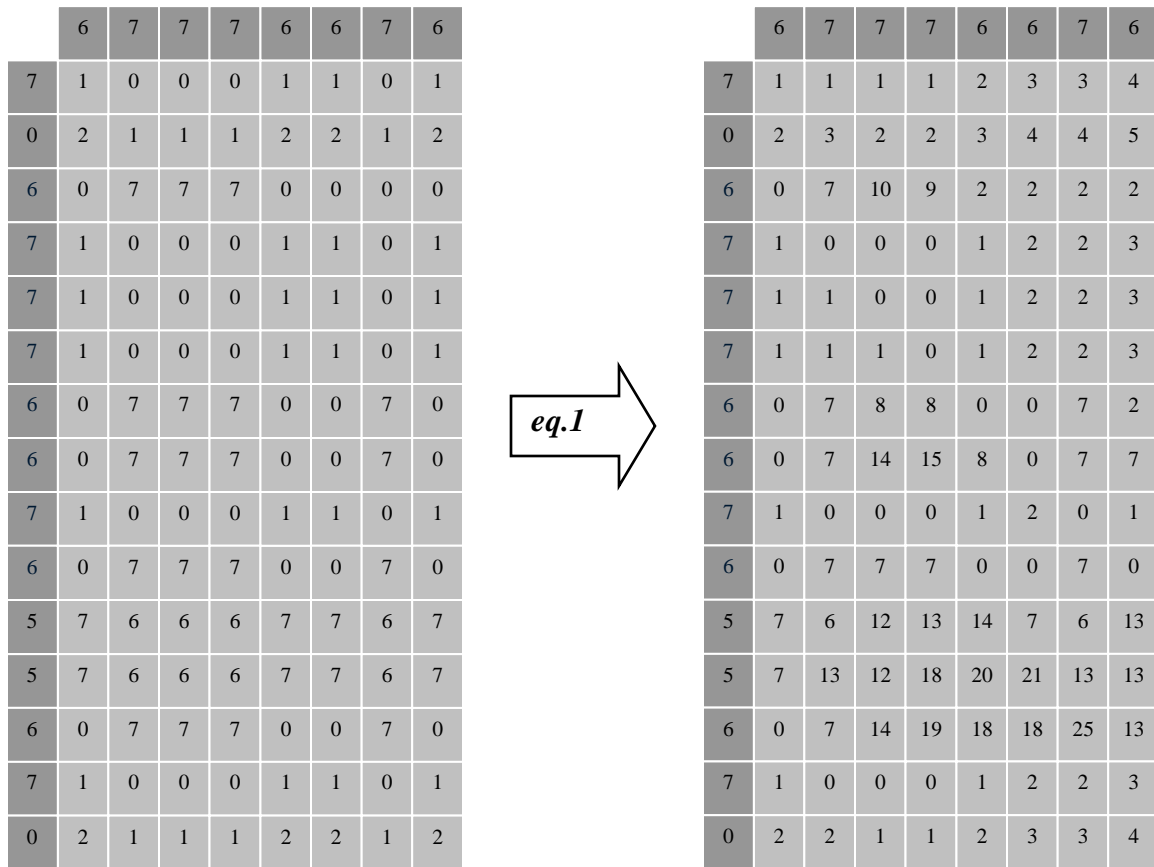


Figure.16 (a) and (b) after applying (eq.1)

The last column in the matrix of *figure.16b* has the minimum cost value 0 at index nr 10. If we follow through the saved path backwards starting from $a_{10,8}$ it will lead us to $a_{3,1}$. In *figure.17* the corresponding match to the template is found in the target from index 3 to 10. This case was perfect as the cost is zero but in other cases where the template is not an exact copy of a part of the target the cost will be higher but still in my experiments after locating the minimum cost a very acceptable match of correspondence is found. See *figure 18*.

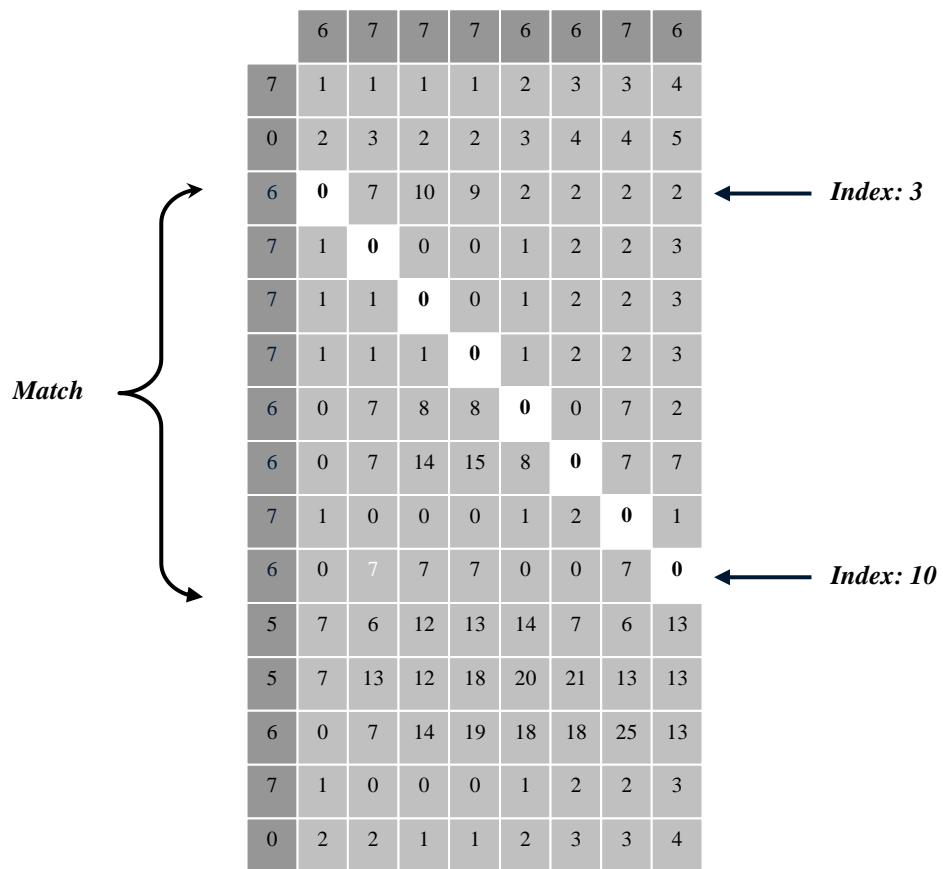


Figure.17

4 Experimental Results

To illustrate the results, samples of test is presented. I enrolled 6 hands of different persons and applied all image processing steps mentioned earlier. The part of the hand to be a template was chosen manually and can be extracted automatically in future work. For the target I enrolled the same 6 persons hands but with some change in position and rotation.

The hand to the left in the figure.18 below is the template extracted from my hand redrawn in the same window as the target only to easily show the results. A number of elements were chosen manually in the template vector and crosses were drawn on these positions on the image as markers. Since we know at exactly what index the marks were drawn in the template then we can use this information to draw similar marks on the other vector “the target” by using the same indexes and see how it turns out. The result shows that the markers falls on almost the same positions making it much easier to compare the features of the template and target by applying different measurement calculations.

The results are promising because the algorithm has managed to find the corresponding area of the feature in the target.

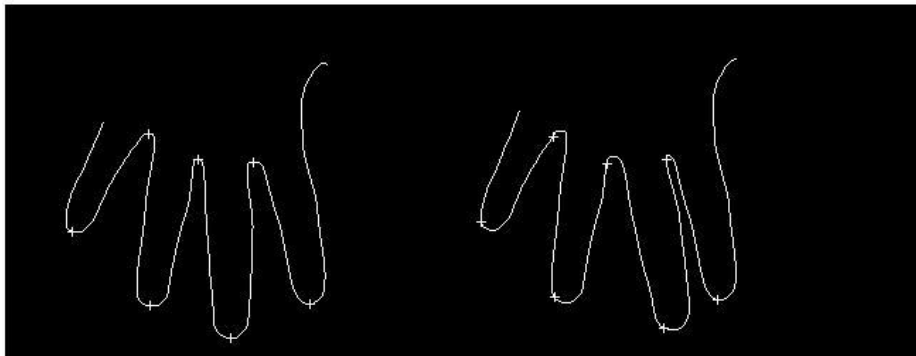


Figure.18

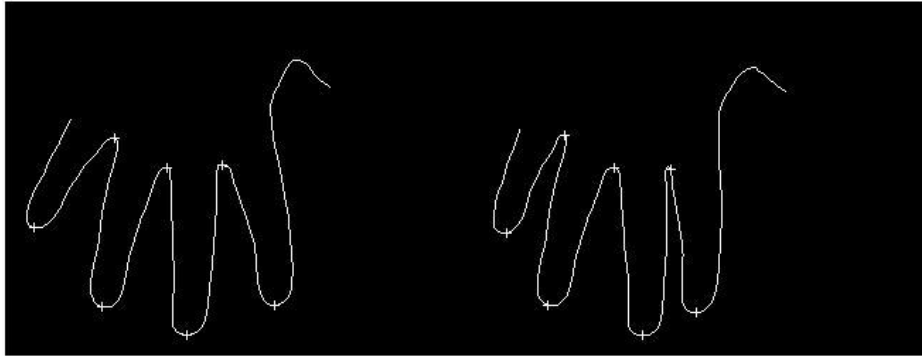


Figure.19

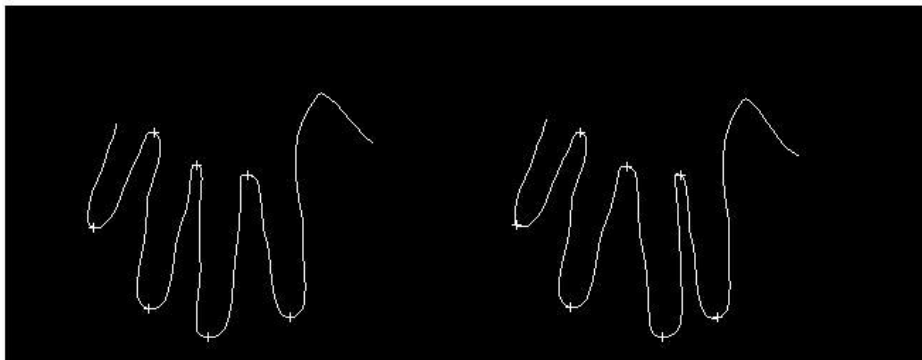


Figure.20

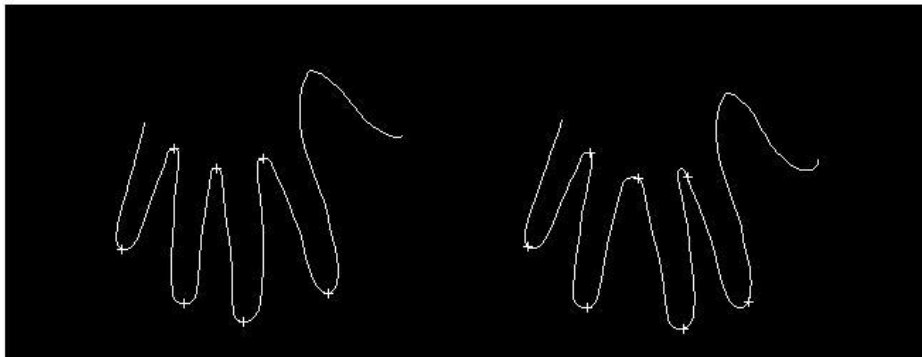


Figure.21

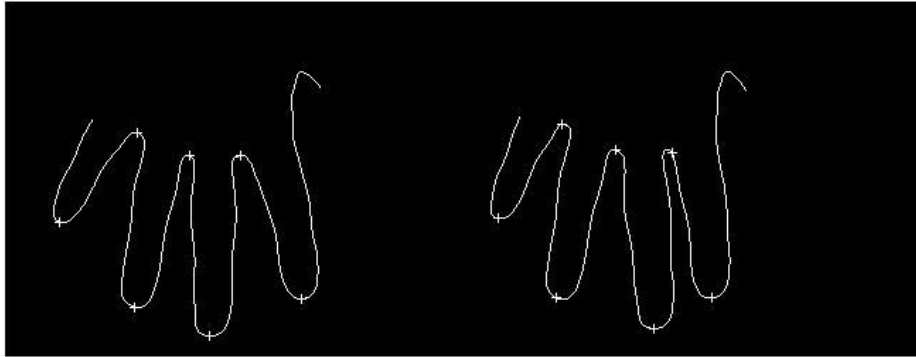


Figure.22

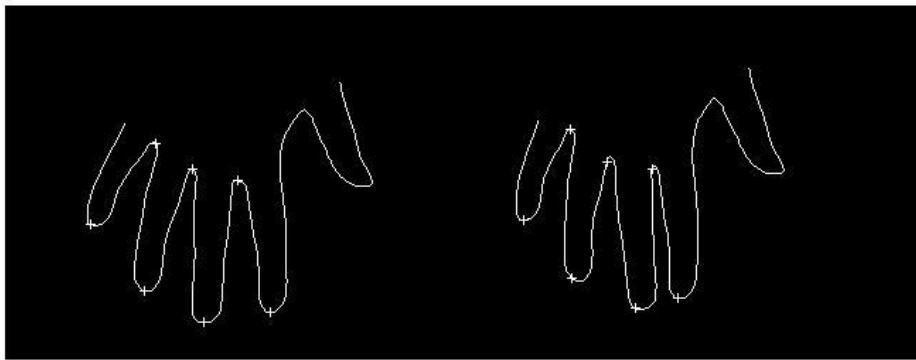


Figure.23

It's very important that these two hand features resemble each other. For example I can measure the width of the index finger in the template and the target to calculate the difference between them. If the measurements are not extracted in the same place in both hands then the result would be no verification even if the target and template belongs to the same person because one could be measuring between the width of the template hand's index finger and the targets middle finger or anything else for that matter.

5 Concluding Remarks

In this bachelor thesis, I have built a peg free system to verify a person's identity using hand geometry as a biometrical identity.

An evaluation of the systems performance was made to explore the possibility of using dynamic programming to find corresponding points between two hand images.

After studying the results I can conclude that the system I built can scan the hand and apply the necessary image processing. The system manages to find the part of the target that corresponds to the template which is proved by manually choosing certain points of interest in the template and finding approximately the corresponding ones in the target by applying dynamic programming. The system is sensitive to deviation in hand placement but due to time limitations for this thesis I will do more extensive tests after creating a database of hand images and improve the algorithm.

Future work will include a database and a study to verify the results. In addition to that, features will be extracted automatically and there will be some fine tuning of the system to make it less sensitive to rotation.

6 Acknowledgement

I would like to thank my supervisor Haibo Li for making this project possible and for his help and advice during its implementation. Also thanks go to the fantastic Digital Media Lab members and Applied Electronics employees for all the help and support I have received along the way. I would also like to thank Adi Anani and Ville Jalkanen for useful discussions.

7 References

- [1] NSTC Subcommittee on biometrics, Frequently Asked Questions.
- [2] Encyclopedia of Multimedia Technology and Networking, Volume 1
By Margherita Pagani
- [3] Guide to Biometrics. Ruud Bolle et al, Springer.
- [4] Digital image processing, By Rafael C. González, Richard Eugene Woods.
- [5] NIST. Dictionary of algorithms and data structures, 2005.
<http://www.nist.gov/dads/>
- [6] Arun Ross, "A Prototype Hand Geometry-based Verification System", M.S.
Project Report, Michigan State University, 1999.
- [7] An approach to image retrieval based on shape. Guojun Lu, Journal
Information Science
- [8] Richard Bellman on the Birth of Dynamic Programming, Stuart Dreyfus,
University of California.