Line Extraction in Palmprint System

Fang Li, Maylor K. H. Leung, and Chan Sin Wai

***Abstract*—Today biometric-based systems have been widely used and tipped to replace the traditional methods and one of the emerging techniques is automatic palmprint authentication. To ensure that the matching process achieves an accurate result, the pre-processing of palmprint extraction plays an important role. A new line extraction technique using hysteresis thresholding is presented in this paper. It shows obvious improvement in line feature extraction and provides better base for further matching process.**

***Index Terms*—Line extraction, palmprint, hysteresis thresholding.**

I. Introduction

Computer-based personal identification known as biometrics computing begin in 1970s. Up till now, the most widely used biometric feature is the fingerprint [1], [2] and the most reliable feature is using the iris [3]. Due to the reason that fingerprints are used for criminal records and investigations, providing fingerprints for experiments is not a wise thing to do for private and safety reasons. Also due to skin problems, inborn or physical worn away of the fingerprint due to hand work or accidents, fingerprints may not be the best identification method. Using iris as identification has a very high recognition rate but considering the fact that the cost of sensors is relatively high and the iris patterns of eastern and western people are not the same, thus this method is not very universal as at this point of time, only the western people are adaptive to this method of identification.

Palm is the central region of the front of our hand and palmprint is the line pattern within the area of palm. There are a number of attributes that make the palmprint lines on a person's hand distinguishable from those on another. These include color, clarity, length, position, continuity and variation in thickness [2], [4]. It is proven that palmprint lines are stable features of human begins and no two palms have exactly the same line patterns [3], [4]. This makes palmprint a possible candidate in personal identification system.

A palmprint is mainly composed of lines and ridges. Researchers are working on palmprint matching systems based on texture features and line features [5]. Cognitive psychological studies indicated that human recognizes line drawing as quickly and almost as accurately as gray-level pictures [4]. Line pattern are highly effective for shape representation. With the advantages of simpler representation, low storage demand and consistency in detection, line is one promising primitive feature for shape matching involving large database. However during the extraction process of palmprint, there will always be a problem of missing or broken lines which causes difficulty in the matching process.

Canny edge detection is, at the moment, considered to be a powerful edge detection algorithm [6]. However canny edge detection does produce false edges on smoothly shaded surfaces. Palmprints can be considered as smoothly shaded surfaces [7], thus canny edge detection does not achieve desirable representations of the line structures in palmrprint images.

In view of this problem, there is a need for an efficient image pre-processing [8] stages in order to reduce the number of missing or broken lines in the binary images. This is greatly preferred over the attempt to solve broken lines issues in the post-processing stages. A new line extraction technique using heuristic threshold is presented in this paper. It shows obvious improvement in line feature extraction and provides better base for further matching process.

The rest of this paper is organized as follows: A survey of previous work is carried out in Section 2. Section 3 introduces the details of the proposed algorithm. Finally, the results and conclusion are highlighted in Section 4.

II. Image Pre-Processing Review

To begin with, palmprints to be verified are captured using digital devices [7]. Different palmprints captured do not have the same size and shape, fixed locations and orientations, thus we need to establish a coordinate system for palmprint alignment, and extract the Region of Interest (ROI) of the inner area of the palm based on the coordinate system. Based on this coordinate system, a window can be created to contain the information. In this application, the window size is set to 160×160 [9].The pixels within the ROI extracted are retained in this window for further processing, while the areas outside the window are ignored and discarded.

The first stage of feature extraction is to reduce noise, which is unimportant information, from the extracted image. Image thresholding is then required to convert the 8-bit image into a binary image, with white representing the pixels on the line and black otherwise. Subsequent to thresholding, the binary image is processed by a morphological thinning algorithm. The resulting image will contains lines of only single pixel width.

The main focus for this report is image thresholding. Thresholding is a very important stage in image pre-processing. The amount of extracted pixels depends on the threshold value. With a high threshold, only those pixels

Manuscript received May 25, 2012; revised June 25, 2012.

Fang Li is with the School of computer engineering, Nanyang Technological University, Singapore (e-mail: asfli@ ntu.edu.sg).

Maylor K. H. Leung was with School of computer engineering, Nanyang Technological University, Singapore. He is now with Universiti Tunku Abdul Rahman (UTAR), Malaysia (e-mail: asmkleung@ntu.edu.sg).

Chan Sin Wai is working in Industrial Automation domain as a software engineer in Singapore (e-mail: Chan0102@ntu.edu.sg).

with strong response to line detection will survive, so the number of extracted lines will be small. With a lower threshold, more lines can be extracted, since the weaker ones are also included.

Li. et al [9] have used a dynamic single-threshold determine the amount of pixels to be extracted, i.e. 5% of the total image pixels [9]. This algorithm is robust to luminance variance. However, for some images, e.g., the sample in Fig. 1, the binary image, although representing the image correctly, does have some broken lines along the principle lines, which may eventually cause some inaccuracy in matching. Thus we have proposed to modify it to improve the situation.

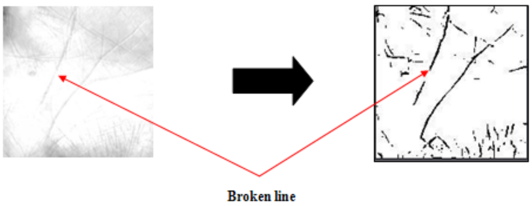

Fig. 1. Lines extracted using 5% single-threshold algorithm.

Canny edge detection [6] is, at the moment, considered to be a powerful edge detection algorithm. Stages of the canny algorithm includes: i) noise reduction, ii) finding the intensity gradient of the image, iii) non-maximum suppression and iv) tracing edges through the image and hysteresis thresholding.

However canny edge detection does produce false edges on smoothly shaded surfaces. Palmprints can be considered as smoothly shaded surfaces, thus canny edge detection does not achieve desirable representations of the line structures in palmrprint images. When canny edge detection is applied to the same image in Fig. 1, the intermediate processes and extracted lines are shown in Fig. 2. The extracted lines in Fig. 2(e) cannot present the line features well.

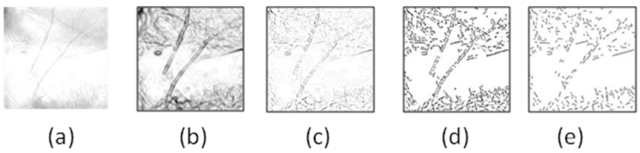

Fig. 2. Lines extracted using canny edge detection. (a) Original image (b) Gradient finding (c) Non-max suppression (d) Thresholding (e) Lines display.

In view of this problem, there is a need for an efficient image pre-processing stages in order to reduce the number of missing or broken lines in the binary images. This is greatly preferred over the attempt to solve broken lines issues in the post-processing stages. A new line extraction technique using heuristic threshold is presented in details in Section 3.

III. Proposed Line Detection Algorithm

The proposed pre-processing algorithm is simply described as follows:

1) Gaussian blurring technique to remove noise
2) Finding the intensity gradient of the image
3) Hysteresis thresholding to differentiate definite edges from noisy edges using two thresholds: the upper and lower threshold
4) Morphological thinning

A. Noise Removal

The palmprint lines are subtle features. They are usually thin and vulnerable to noise. Even a minor noise pixel may cause a palmprint line in the image to be broken. Hence, image pre-processing is necessary to remove the noise so that line extraction can become more accurate. Gaussian blurring technique is adopted for this noise removal process.

Gaussian blurring is efficient way of removing noise and detail level. Gaussian blurring is a filter that uses normal distribution to manipulate the transformation to be applied to each and every pixel of the image for blurring effect [8]. The equation of a 2-D Gaussian distribution is:

$$G(x,y) = \frac{1}{2\pi\sigma^2} exp^{-(x^2+y^2)/2\sigma^2}$$

where σ is the standard deviation of Gaussian distribution.

Fig. 3 shows a 2-D Gaussian distribution [8]. A Gaussian filter can be computed using a simple masking matrix and the matrix can be calculated from the values of the Gaussian distribution. Although a large width of a Gaussian filter reduces the sensitivity to noise of the edge detector, it increases the localization error in the detected edges [8]. Thus a 5 × 5 Gaussian filter is quite a common choice. In this application, we will use a Gaussian filter shown in Fig. 4.

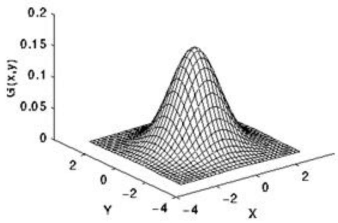

Fig. 3. 2-D Gaussian distribution with mean (0, 0) and σ = 1.

$$\frac{1}{115}\left\{\begin{array}{ccccc} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{array}\right\}$$

Fig. 4. Approximation of 5 × 5 Gaussian filter with σ = 1.4.

B. Finding the Intensity Gradient of the Image

Various sets of masks can be used to determine the intensity gradients of the image pixels. Each set of masks are four directional, which are capable of detecting horizontal lines, vertical lines and diagonal lines. For a single mask, the sum of weights equals to 0, so that it generates zero-response to the areas with uniform intensity level. The two masks tested in our project are shown in Fig. 5 and Fig. 6.

$$\begin{array}{|c|c|c|c|c|} \hline 2 & 2 & 2 & 2 & 2 \\ \hline -1 & -1 & -1 & -1 & -1 \\ \hline -2 & -2 & -2 & -2 & -2 \\ \hline -1 & -1 & -1 & -1 & -1 \\ \hline 2 & 2 & 2 & 2 & 2 \\ \hline \end{array} \qquad \begin{array}{|c|c|c|c|c|} \hline 2 & -1 & -2 & -1 & 2 \\ \hline 2 & -1 & -2 & -1 & 2 \\ \hline 2 & -1 & -2 & -1 & 2 \\ \hline 2 & -1 & -2 & -1 & 2 \\ \hline 2 & -1 & -2 & -1 & 2 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|c|c|} \hline -2 & -1 & 1 & 2 & 2 \\ \hline -1 & -2 & -1 & 1 & 2 \\ \hline 1 & -1 & -2 & -1 & 1 \\ \hline 2 & 1 & -1 & -2 & -1 \\ \hline 2 & 2 & 1 & -1 & -2 \\ \hline \end{array} \qquad \begin{array}{|c|c|c|c|c|} \hline 2 & 2 & 1 & -1 & -2 \\ \hline 2 & 1 & -1 & -2 & -1 \\ \hline 1 & -1 & -2 & -1 & 1 \\ \hline -1 & -2 & -1 & 1 & 2 \\ \hline -2 & -1 & 1 & 2 & 2 \\ \hline \end{array}$$

Fig. 5. 5 × 5 mask.

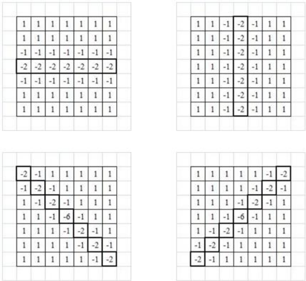

Fig. 6. 7 × 7 mask.

Fig. 7 shows the comparison of determined intensity gradients of the image pixels using above two masks. The result shows the latter one is slightly better after comparing the histograms. This is concluded from the following two points:

1) The contrast between the strong lines and weak lines are higher.
2) The overall gradient intensity of the result when using the set of 5 × 5 masks is lower than that of the result when using the set of 7 × 7 masks.

Based on the above comparison, the set of 7 × 7 masks is more preferable in performing line detection.

C. Hyteresis Thresholding

Hysteresis thresholding is required as it is not easy, or rather quite impossible to specify a threshold that determines a pixel if it carries important information or useless information. Thus, using a single threshold does not allow flexibility. Hysteresis thresholding, although allows more flexibility, too high a threshold results in a risk of losing important information, in contrast, too low a threshold will regard useless information as important information.

Hysteresis thresholding makes use of two thresholds: upper threshold and lower threshold. The assumption that important edges should be connected along consecutive pixels in the image enables us to disregard noisy pixels that have high response but do not connect to any line.

An upper threshold is first determined and applied to mark out edges that are sure to be genuine that contain important information. Starting from these edges, using the directional information obtained earlier, edges are traced through the image. The lower threshold is applied at each edge, tracing edges until a starting or ending point is found. The process flow is shown in Fig. 8.

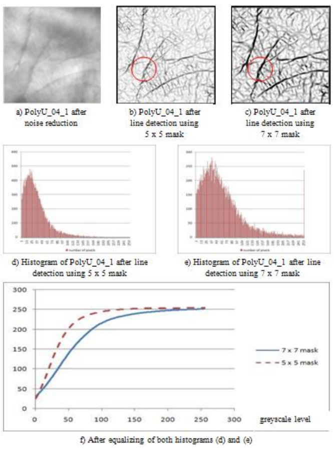

Fig .7. Comparison of line detection using different masks for PolyU_04_1.

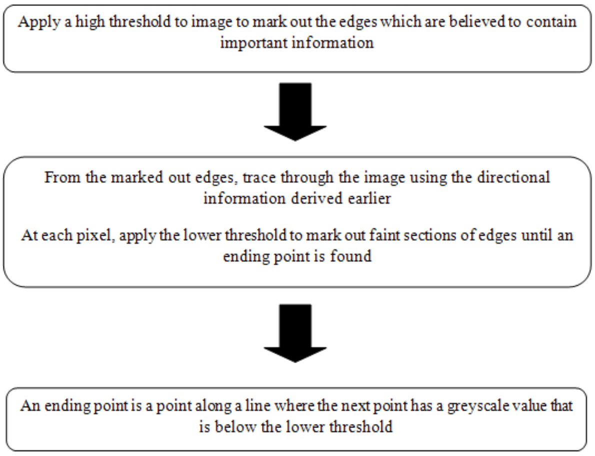

Fig. 8. Hysteresis thresholding using upper threshold and lower threshold.

1) Threshold selection

In our project, dynamic thresholds are used as in [9]. However, two thresholds are used instead of one. We can denote them as the upper threshold and lower threshold. Different single dynamic thresholds are tested and the experimental results are shown as in Fig. 9.

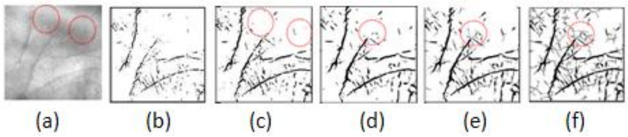

Fig. 9. Edge pixels detected using different single dynamic thresholds on PolyU_04_1 (a) Original image (b) 5% (c) 8% (d) 12% (e) 20% (f) 30%.

Experimentally, 8% of the total image pixels is used as the optimal upper threshold in our discussion.

Having set the upper threshold value, we will now select a lower threshold value, aiming to improve the connection in principle lines. 12% lower threshold, 20% lower threshold and 30% lower threshold are experimented as shown in Fig. 9.

As can be seen in Fig. 9, visually there are not much difference in the amount broken lines between 9(e) and 9(f). However, in 9(f), there is more unwanted information being represented such as fine wrinkles or ridges, which will affect the accuracy in palmprint matching. A 12% lower threshold can also be employed, as can be seen in the images of 9(d). However, there is not much chance for improvement in the connection of principle lines comparing with just using the 8% upper threshold, and that defeats the objective of using two thresholds. In the portion circled out in 9(a), there is some important information. In 9(d), this information is lost; however in 9(f), not only the important information is being represented, but corrupted with some fine wrinkles as well, which is not what we desire. For this instance, 9(e) gives the best representation of the circled out portion.

Thus after visual analysis, 20% of the total image pixels will be used as the optimal lower threshold in our discussion. However, not all the pixels that satisfy the lower threshold will be marked as an edge. We will discuss how to decide if a pixel is an edge in the later section.

2) Identifying definite edges and potential edges

As illustrated in Fig. 10, the pixels in image are categorized into three types as follows:

– Definite edge point: pixel's gradient value exceed upper threshold.

– Potential edge point: pixel's gradient value in between upper threshold and lower threshold.

– Unimportant point: pixel's gradient value below lower threshold.

In order to convert to binary image, the pixels need be classified into two classes, foreground pixel with value 255 and background with value 0. We first set all definite edge points as foreground because they are most obvious pixels in the images, at the same time; we set the unimportant points as background pixels since they are too faint to be useful. A unique algorithm will be used to decide the final category of potential edge points.

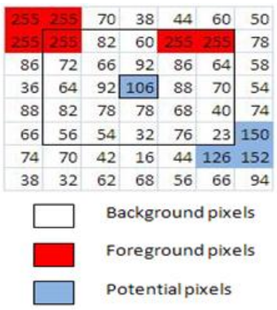

Fig. 10. Illustration of three categories.

3) *Potential edge point categorization using a 5 x 5 sliding window*

We believe that all potential edges are attached to either potential edges or definite edges. However due to the positioning of the palm, part of the principle lines may result in a lower response. A pixel with low response will usually have a greyscale value that is lower than the upper threshold. In order to connect broken lines efficiently a unique 5×5 sliding window algorithm is proposed in this paper.

Fig. 11 shows the working principle of the 5 × 5 sliding window. The seed pixel is a potential edge. The 4 images show 4 different directions that definite edges can be connected. The asterisks (*) denote the pixels that the algorithm will check. The algorithm will start in the direction shown in Fig. 11(a), the algorithm will visit the 4 other pixels and check their pixel values. If there are at least 2 definite edges or potential edges out of those 4 pixels, the algorithm will break and regard the seed as a foreground pixel. Otherwise the algorithm will continue with 11(b), 11(c) and 11(d) if necessary. The algorithm will reiterate until there are no more changes to the pixel values, and after which all the pixels which do not have values of 255 will be set to 0.

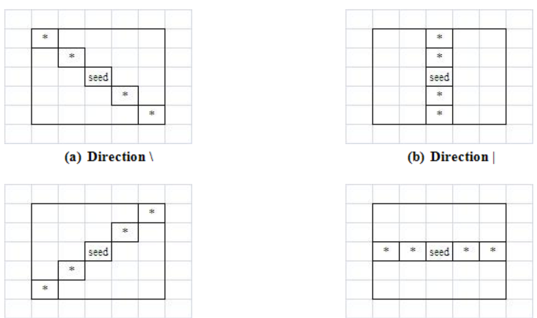

Fig. 11. Working principle of 5 × 5 sliding window.

Using the concept of the sliding window, we extended the window to a 7× 7 sliding window, 9 × 9 sliding window and so on, finding the optimal sliding window to be used in this application. Experimentally, 5 × 5 sliding window achieves optimal results. Thus we adopted the 5 × 5 sliding window in this application.

D. *Morphological Thinning*

Thinning is the operation to generate skeleton, that is, all lines in the image are reduced to single pixel width. There are several methods for thinning operation. Morphological thinning algorithm is the most commonly used one.

In this algorithm, a set of eight structuring elements are defined as shown in Fig. 12. The thinned image is calculated by translating the origin of the structuring element to each possible pixel position in the original image, and comparing the pixel pattern of the structuring element with the underlying part of the image. At each position, if the pixel patterns match, the image pixel underneath the origin of the structuring element is set as background, in other words, a redundant foreground pixel is removed. If the pixel pattern of the structuring element does not match with the image, the corresponding pixel remains unchanged.

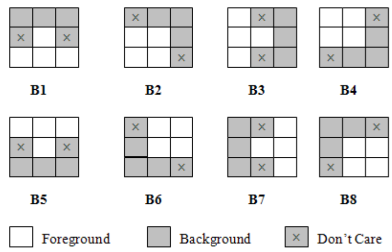

Fig. 12. Structuring elements for thinning operation.

After thinning by structuring element $B1$, the same process is repeated with $B2$, using the previous output as new input. The process is executed eight times, until all structuring elements are applied. This finishes one cycle of the thinning operation. The operation of each cycle can be expressed as:

$$A \otimes \{B\} = ((...((A \otimes B1) \otimes B2)...) \otimes B8)$$

where A is the input image, B is the set of structuring elements.

The entire operation is repeated until it causes no further changes. The final output is the image containing the hand's boundary of single pixel thickness. The thinned edge image makes the analysis of boundary information much easier, which helps to extract the feature points more efficiently.

IV. Results and Conclusion

A series of experiments were carried out using a palmprint database from the Biometric Research Centre, The Hong Kong Polytechnic University. The palmprint images were collected from 100 individuals (6 images for each person) [10] using a special palmprint capture device. The subjects mainly consisted of volunteers from students and staffs at the Hong Kong Polytechnic University with a wide range of age distribution and different genders. The images were captured under various illumination conditions resulting in different qualities.

We compared the different results obtained using the three different techniques adopted for this palmprint application as shown in Fig. 13. We have chosen 3 very different palmprint images for result comparison. First sample has relatively high luminosity than other images, second is a dark one, and the last one is a normal one. The proposed algorithm consistently preserves the structure of the original image better than other algorithms.

The objective of this project is to represent the original image by a binary image, extracting the pixels of principle lines to be important information and ignoring the unimportant information With this objective, the main priority of the final binary image is to maintain the structure of the original image, followed by solving the broken line issue and finally reduce as much unimportant information as possible.

Canny edge detection is no doubt by far one of the best techniques for edge detection. However in this application, using Canny edge detection for the image pre-processing stage cannot achieve the result we desire where we were unable to visually trace the structure of the principle lines from the final binary image. That means using Canny edge detection is not able to preserve the structure of the original image very accurately in this particular application. Upon proposing the new algorithm, we were able to better represent the raw images of the palmprints in terms of reducing the number of broken lines in the final binary images and in the meantime preserving the structure of the raw images.

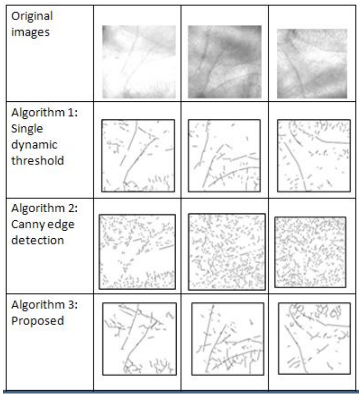

Fig. 13. Comparison of different algorithms.

REFERENCES

[1] *The Hand Reveals*, Warren-Davis, D. Great Britain: Element Books Ltd., 1993.

[2] *Automated Biometrics - Technologies and Systems*, D. Zhang, s.l.: Kluwer Academic Publishers, 2000.

[3] *Hand-based biometrics*, Biometric Technology Today. 2003.

[4] J. You, W. Li, and D. Zhang, "Hierarchical Palmprint Identification via Multiple Feature Extraction," *Pattern Recognition*. 2002.

[5] A. J. M. Traina, J. C. Felipe, and C. T. Jr., "Global Wrap Metric Distrance: Boosting Content-based Image Retrieval through Histograms," *Proc. IEEE*, 2005.

[6] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vols. PAMI-8, 1986.

[7] V. S. Nalwa and T. O. Binford, "On Detecting Edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vols. PAMI-8, 1986.

[8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. s.l: Prentice Hall.

[9] F. Li and M. K. H. Leung, "A Two-level Matching Scheme for Speedy and Accurate Palmprint Identification," *Lecture Notes in Computer Science*, vol. 4352, pp. 323-332, 2007.

[10] The Hong Kong Polytechnic. (2006). PolyU Palmprint Palmprint Database. Biometrics Research Centre. [Online]. Available: http://www4.comp.polyu.edu.hk/~biometrics/

Fang Li is currently a Lecturer at School of Computer Engineering, Nanyang Technological University, Singapore. Before coming to Singapore, she ever served the Electronic Research Institute of Ministry of Information Industry. In the Institute, she worked for several National Large-Scale Electronic System Engineering projects and became one of the members in management committee. Since 1999, she was in Nanyang Technological University, and involved in the research of image retrieval. She received B.Eng degree in Computer Application and Engineering, and M.Eng degree in Information Engineering, respectively in 1995, and 2001. Thereafter, she joined I-Sprint Ltd as a senior member of research staffs to provide network security solutions. She is now actively participating in Biometrics Technology and focuses her research on Pattern Recognition. She is a member of IEEE and PREMIA(Singapore).

Maylor K. H. Leung received the BSc degree in physics from the National Taiwan University in 1979, and the BSc, MSc and PhD degrees in computer science from the University of Saskatchewan, Canada, in 1983, 1985 and 1992 respectively. Currently, Dr. Leung is a Professor at Universiti Tunku Abdul Rahman, Malaysia. His research interest is in the area of Computer Vision, Pattern Recognition and Image Processing. Particular interests are on object recognition, video surveillance for human behavior detection, robot navigation, line pattern analysis, and computer aids for the visually impaired.

Chan Sin Wai graduated with Bachelor of Engineering (Computer Engineering) from Nanyang Technological University, Singapore in 2008. Currently he is working in Industrial Automation domain as a software engineer. In the meantime, he pursuing his Masters at National University of Singapore, majoring in Knowledge Engineering. My interest includes iOS mobile development and cloud computing.