Real-time Feedback for Usable Fingerprint Systems

Haiying Guan, Mary F. Theofanos, Yee-Yin Choong, and Brian Stanton

National Institute of Standards and Technology
Email: {haiying.guan, mary.theofanos, yee-yin.choong, brian.stanton}@nist.gov

Abstract

Compared with traditional password and other identification methods, biometrics such as face, iris, and fingerprints for automatic personal identification and verification have many advantages, and are increasingly gaining popularity in all kinds of applications. As the technologies mature, the community has begun to realize that usability has great impact on the final accuracy and efficiency of a biometric system. Although research has shown that effective user feedback can improve the quality of the fingerprint images captured and user satisfaction, currently user feedback information of fingerprint devices used in real world applications is very limited. We design a rich, quality-driven interactive real-time user feedback mechanism for unattended fingerprint kiosk. The system aims to improve the quality of biometric samples during the acquisition process by feeding rich information back to the user instantaneously by measuring objective parameters of the image. The paper proposes an innovative, cost-efficient, real-time algorithm for fingertip detection, slap/thumb rotation detection, and finger region intensity estimation. The paper provides detailed information on the technical solution and its implementation. Preliminary results show that the methodology can potentially increase efficiency, effectiveness, and user satisfaction of a fingerprint biometric system.

1. Introduction

The use of biometrics like face, iris, and fingerprints for automatic identification/verification has gained increasing popularity recently in all kinds of application domains, across government, business, industry, as well as our daily life. As the technologies have matured [1] [2] [3], it has been recognized that usability has a strong effect on the overall performance of a biometric system. As described in [4], a biometric capture process starts with the user interacting with the system, and the user's fingerprint sample determines the success of the process and the system as a whole. Thus, it is important to take a more holistic view and consider factors affecting usability.

Our goal is to develop a usable automatic comprehensive biometric application for screening, which self-captures fingerprints accurately; identifies individuals effectively and quickly. A recent handbook published by the National Institute of Standards and Technology (NIST) [5] provides a set of useful guidelines on how user-centered design (UCD) can be applied to biometric systems. It focuses on four aspects: context of use, user and organizational requirements, design solution (including interaction design, and interface design) and evaluation. Following these guidelines, we involve users in the design, development, and evaluation of the biometric application.

First, improving the system usability will provide high quality biometric data, and in turn, reduce the overall error of the data collection process. Though great attention and improvement [1] [2] [3] have been achieved in the past through technology, there is still room for more improvement through biometric usability. We would like to reduce the overall error at the beginning of the process: during data collection. In most cases, it is very difficult, almost impossible, for the users whose biometrics are being collected to assess if the image is good enough or not in a self-capture system. Currently, there are no systematic ways to provide users with feedback on positions and image quality, nor are there ways to help users recover from errors. Poor usability results in inaccurate recognition results, which decreases the system recognition rate, and also decreases the user's satisfaction (e.g. long waiting times for travelers in Customs and so on). As a result, there is an escalating need to improve biometric systems from the usability point of view for improved accuracy, efficiency, and user satisfaction [5].

The study in [6] indicates that effective feedback does translate into improvement in quality. In addition, the granularity of feedback of most current biometric systems is typically quite coarse. In our application, we would like to design an interaction system to effectively guide the user more directly by providing rich, real-time, feedback information to them during the biometric data self-capture.

The online feedback information provided by current commercially available fingerprint capture devices[1] is very

[1] Specific hardware and software products identified in this report were used in order to perform the evaluations described. In no case does such identification imply recommendation or endorsement by the National

978-1-4577-1359-0/11/$26.00 ©2011 IEEE

limited. In this work, we design a rich, objective, quality-driven interactive feedback mechanism for fingerprint self-capture based on the user's position status; such as "Move Up/Down", "Move Left/Right", "Rotate Clockwise/Counterclockwise", "Press More/Less" and "Less/More number of fingers" etc. The objective feedback aims to eliminate subjective judgment by the user, and improve the quality of biometric samples during the acquisition process by feeding the rich information back to the user instantaneously through measuring objective parameters of the image quality. Preliminary results show that the mechanism can potentially increase efficiency, effectiveness and user satisfaction of the fingerprint biometric system.

From the technical perspective the computational efficiency is also an important aspect that the real-time algorithms must address in order to provide the user with rich feedback information. In this paper, we propose an innovative, cost-efficient, real-time algorithm for finger location detection, slap/thumb rotation detection, and finger region intensity estimation. Our preliminary experimental results show that the proposed algorithm detects those parameters at a very fast speed and at a very low system cost.

The contribution of this work to the state of the art in biometric usability is threefold: (1) the design of a prototype usable biometric interaction system to bridge the information gap between the user and the system. (2) an innovative and cost-efficient real-time algorithm for fingerprint image analysis. (3) the integration of quality measurement into a rich feedback mechanism to improve the usability of fingerprint collection. .

2. Literature Review

ISO Standards [7] clearly state that the performance of a biometric system can be affected by the errors introduced in the data capture subsystem of the general biometric model. These factors can be attributed to either the user or sensor. With the advancements in biometric technologies, especially fingerprint technology, in order to continue improving the performance of biometric systems, it is necessary to take a more holistic view. Theofanos et al. [4] point out that we can no longer only focus on the system view (improving the performance, functionality, reliability and precision through technological means). It is necessary to examine the human factors and the usability point of view.

Prior studies [8] also showed that human factors, such as age, physical ability [9], etc. have significant impact on image quality and subsequently the system performance. [8] summarizes three studies: feedback and habituation, anthropometrics (influence and interaction of the height of the surface and the fingerprint sensor), and instruction modes in the emerging field of biometric usability. Each study focused on fingerprint recognition and the systematic uncertainty in system performance induced by different human factors. It is concluded that inherent characteristics such as gender and age do affect fingerprint image quality. Habituation with no feedback at all was not shown to affect the quality of prints. But, feedback and acclimatization did translate into improvement of quality.

In [10], it was found that the fingerprint image quality is highly dependent on the human–computer interaction and usability considerations of the acquisition system. The biometric system should view the user as an interactive component of the biometric system instead of as a passive source of the biometric sample. [10] also shows that how information is provided to the user on interacting with the fingerprint device does indeed affect image quality, and suggests that real-time feedback is necessary and must be integrated into fingerprint capture systems for users to ensure acceptable quality images.

Wong et al. [11] also aspire to improve the usability of biometric systems via Human-Computer Interaction (HCI) principles. They realized that although both engineering (focus on algorithm developments) and HCI approaches can contribute in complementary ways to biometrics, the contribution of HCI has been to date under-researched. A weak focus on human factors can lead to low user acceptance of biometric technologies, especially in the consumer market. Their work focuses on a design guideline which takes into account quality factors that can significantly affect the system performance, and examines ways to improve system usability through feedback mechanisms and user interaction. They illustrate the methodology using face recognition as a case study.

3. Method

We adopt the User-Centered Design (UCD) approach as proposed by [5]. In particular, we focus on the quality factors (those factors that directly affect the quality of the fingerprint capture such as finger position/placement, or pressure) and interactive feedback mechanisms to improve the system performance. In general, we focus on the use-case scenario of two-way communication and co-operation between the system and the user as the key. From the usability perspective, the system design needs to inform the user about the quality criteria the system uses to perform the tasks (e.g. recognition) in order to obtain the best overall performance. That is, "How to guide the user to help the system?" Firstly, we need to identify factors that may degrade the system performance. For example, if the fingerprint is out of the capture region, or the fingerprint's image intensity is too light/dark for the algorithm to perform feature extraction accurately, etc, the recognition

Institute of Standards and Technology, nor does it imply that the products and equipment identified are necessarily the best available for the purpose.

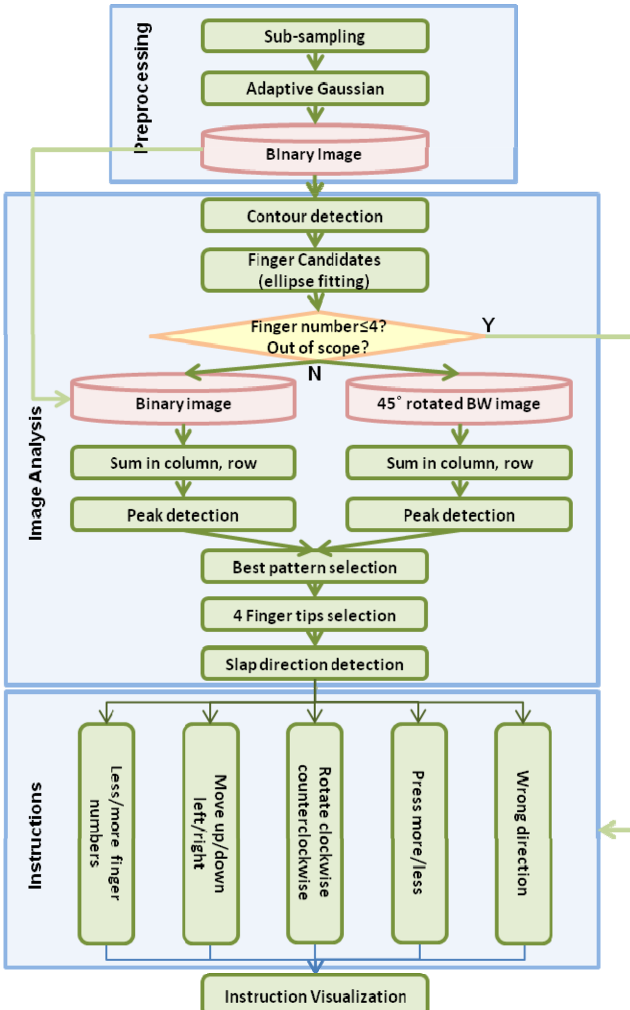


Figure 1: System flow chart in slap capture mode.

rate greatly decreases. After identifying the factors that may degrade the system performance, we measure objectively the extent of those degradation factor(s). We design a real-time user interaction mechanism to analyze the quality information, and translate them into rich feedback instructions. The feedback instructions guide the user to provide better fingerprint images.

In this section, we provide the detailed information on the technical solution about the fingerprint image analysis in our application and its implementation. We mainly focus on the algorithm design to measure objective degradation factors by detecting the fingertip locations, slap/thumb rotation, finger region intensity and so on.

3.1. General framework

Fig. 1 shows the proposed system framework in the slap capture mode. The prototype system consists of three major components: preprocessing, image analysis, and feedback instructions. After the input image is acquired by the fingerprint sensor, a preprocessing operation is performed to detect the regions of interest. In the next step, a contour detection operator is executed to find the fingertip candidates. Given the number and the locations of the candidates, a pre-filtering module, (which was designed to save processing time and cost), is used to decide if further image analysis steps should be taken. If any finger candidate is too close to the boundaries of the capture region, or the number of fingers is less than the desired number, the system jumps directly to the instruction step.

If the number of the fingertip candidates is more than four, the application continues to the fingertip selection step. The binary segmentation image is rotated by 45°. Then the algorithm projects both the rotated image and the original segmented image in the X and Y direction. The projection algorithm sums the number of finger pixels in the row and column to the local bins in each direction, and forms four one-dimensional curves. The peak detection module detects the main peaks and valleys of those four curves. The curve in the slap direction should contain four peaks and three valleys evenly distributed. The selection module selects the best curve according to this pattern and approximately estimates the slap orientation. Given the approximate orientation, the finger tip selection module estimates the four ellipses along this direction to represent fingertips. In order to make the algorithm more robust to noise, instead of detecting the orientation of each finger, the slap orientation detection module exploits the line fitting approach to estimate the slap direction given four selected fingertip ellipses.

In the last step, a post-processing algorithm translates all factors obtained from the image analysis to the feedback instructions. The instructions we implemented are: "Move Left/Right", "Move Up/Down", "Rotate Clockwise/Counterclockwise", "Wrong Direction", "Less/More Finger Number", and "Press Less/More". Finally, the feedback instructions are presented to the user.

3.2. Preprocessing modules

In the preprocessing phase, the first step is performed by the down sampling image module in order to save system cost. Given a 500 dpi fingerprint image with a resolution of 1600x1500, the module reduces the image size to 400x375 using a sub-sampling technique (e.g. Gaussian pyramid decomposition).

The next step is to extract the regions of interest. The literature on image segmentation methods is very broad. We propose a fast adaptive Gaussian threshold approach for image segmentation, based mainly on the trade-off between the speed and the accuracy needs specifically for our real-time application.

The adaptive Gaussian threshold algorithm consists of two steps: smooth and threshold. First, the algorithm exploits Gaussian smooth, which adaptively chooses the disk size of the Gaussian filter by a piece-wise linear function, given the mean intensity value of the candidate region (excluding the white background pixels) [2]. Then the algorithm finds the adaptive threshold for the segmentation algorithm. The threshold is determined by another piece-wise linear function, which takes the mean and standard derivation of the candidate regions as input parameters. Fig. 2 shows the comparison of the simple Gaussian threshold with fixed parameters and the adaptive Gaussian threshold algorithm. The result shows that the adaptive Gaussian provides better results in both low intensity and high intensity cases.

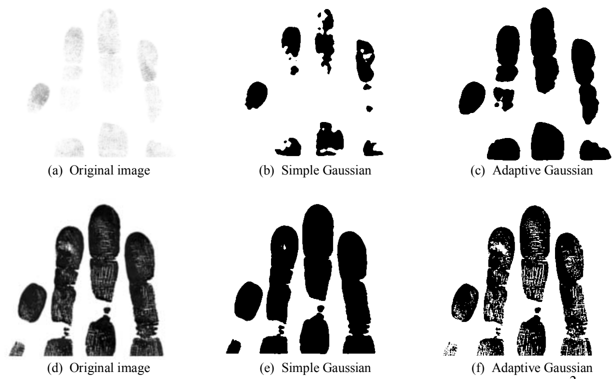


Figure 2: Simple Gaussian vs. Fast adaptive Gaussian[3]

3.3. Slap orientation detection module

The goal of this module is to automatically detect the orientation of the slap and provide corrective instructions if needed in real-time at low cost. For example, in the cases where the slap is not in the upright direction, the application gives feedback instructions to the user in real time, such as "Rotate the four fingers clockwise", or "Rotate the four fingers counterclockwise".

In the literature, there are several slap segmentation methods that perform orientation detection. For example, NIST proposed an offline slap segmentation algorithm, NFSEG [12] for the fingerprint image analysis of a ten-print card. In the NIST report, a finger rotation detection and segmentation approach for each finger is presented. The algorithm assumes that a left slap tends to be vertical or angled to the right. A right slap tends to be vertical or angled to the left. Then it does the exhaustive search on every possible direction using the given accuracy interval. The exhaustive approach provides an accurate result on the fingerprint segmentation.

Due to the speed requirement of our application, we propose a fast algorithm for the real-time slap orientation detection. The basic idea of the algorithm is shown in Fig. 3. Assume that the algorithm can detect the approximate

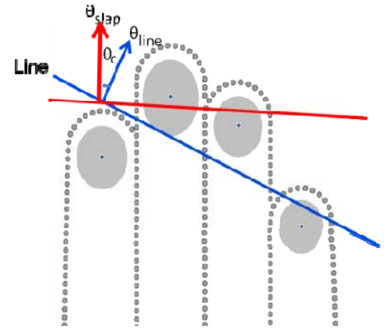


Figure 3: The basic idea of the slap rotation detection.

regions of the four fingertips, then it can fit the fingertips using ellipses fitting algorithm, and use the centers of the ellipses to represent the fingertip locations. After that, the algorithm fits the four finger tips to a line. The normal direction of the line, θ_{line}, can be used to estimate the slap direction: for the right slap, if we slightly rotate the line, θ_{line}, in a counter-clockwise direction with a small angle, θ_c [4], we will obtain the slap direction, θ_{slap}; Similarly, for the left slap, if we slightly rotate the line's normal direction, θ_{line}, in a clockwise direction with a small angle, θ_c, we will obtain the slap direction θ_{slap}.

3.3.1. Finger candidate detection module

The crucial part of the above idea is to detect the four fingertips. Given the segmentation image generated by the adaptive Gaussian threshold algorithm, the contour for each segmentation region is obtained by a contour detection algorithm [13]. After filtering out all the regions which are either too small, or too large to be fingertip regions, the fingertip candidates are obtained (Fig. 4). If the number of the ellipses is greater than four, the algorithm will need to select the four ellipses to represent the four fingertips.

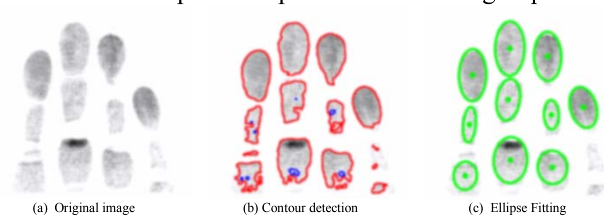


Figure 4: Contour detection and ellipse fitting algorithm

3.3.2. Fingertip selection module

In order to correctly pick the four fingertips from a set of candidates, we estimate the slap orientation for the approximate region of each finger. By analyzing the properties and features of the slap images, we made the following observations: for the upright slap image, if we project and sum the numbers of the slap image pixels along the column direction, the curve follows a very nice four-peak, three-valley pattern, as shown in Fig. 5. On the contrary, if we sum the image along other directions, the four-peak, three-valley pattern disappears.

[2] Given the input parameter, the piece-wise linear function outputs a linear value within a certain range, or a fixed value out of the range.

[3] Prints have been blurred for privacy protection.

[4] θ_c is determined by the hand geometry. We may obtain the range of this parameter according to statistic analysis. It is a fixed parameter in the current implementation.

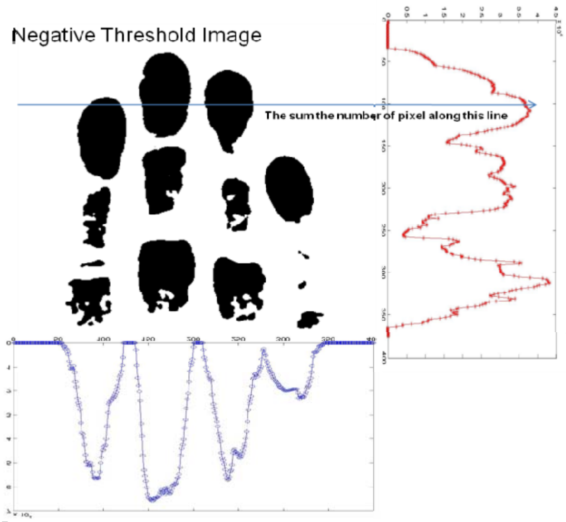
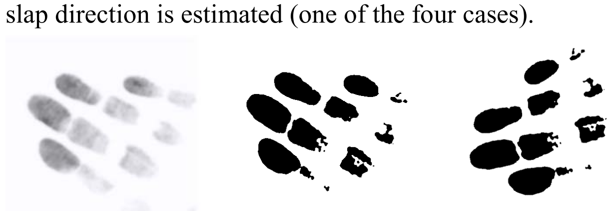
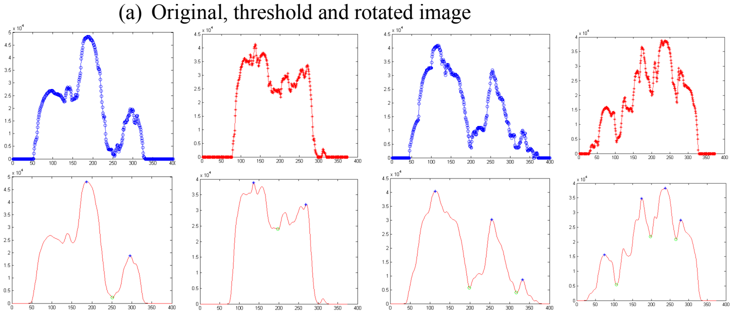


Figure 5: The intensity sum curves along column (blue), row (red) directions.

Based on this observation, given the threshold slap image, we rotate the threshold image by 45˚ and obtain the rotated image. For both the original threshold image and the rotated image, we project the pixels on the finger regions along both the X and Y directions, count the number of the pixels, and obtain four curves (illustrated in Fig. 6). The algorithm picks the curve among the four curves which is most similar to the four-peak, three-valley pattern. For example, the last curve in Fig. 6 follows the four-peak pattern the most, which is corresponding to the projection of rotated image along Y direction. Thus the approximate slap direction is estimated (one of the four cases).



(a) Original, threshold and rotated image



(b) X, Y, projection curves of the threshold and the rotated image

Figure 6: Four projection curves of original threshold image and the rotated image.

Table 1 illustrates all possible slap directions using the image captured (the rotated image is not included). Some cases are very hard to perform in practice, which are shown in the second and fourth rows. In the experiment, we assume that the fingers do not point downward, and thus we do not consider those cases. In addition, we also notice the difference between the left/right slap. For example, normally the right slap can rotate counterclockwise to a great extent, but cannot rotate clockwise greatly; and vice versa for the left slap.

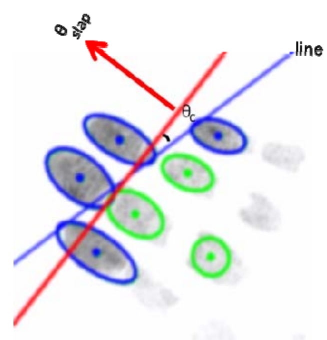


Figure 7: Line fitting (blue) to estimate the slap direction (red)

Given the assumption that fingers point upward only, the approximate direction of the slap is the one chosen from the four directions (cases 1 - 4) of the eight directions (cases 1-8) in the table. From the four curves that represent those four directions, the algorithm selects one direction which is most similar to the four-peak and three-valley pattern. Along this direction, given the locations of the four peaks and three valleys, the algorithm selects the four fingertips from the set of finger candidates.

Right Slap	Considered Cases	1 Wrong direction	2 Rotate clockwise	3. Ok	4. Rotate counterclo..
	Not considered	5	6	7	8
Left Slap	Considered Cases	1 Wrong direction	2 Rotate countercloc.	3. Ok	4 Rotate clockwise
	Not considered	5	6	7	8

Table 1 Slap directions and instructions

After that, the algorithm fits a line using the center of the four fingertips. Finally, the line direction is corrected by θ_c to obtain the slap direction (Fig. 7). The total computation cost of the rotation detection is very low. After an image rotation and four sum projections are performed, the algorithm's complexity is linear on four one-dimensional curves. The total average processing time of the orientation detection for a 400x375 image is about 6 ms in our experiments.

3.3.3. Orientation detection module

Having the objective, quantitative measure of the slap direction, the system translates it to the slap rotation instruction according to which zone the direction falls in (Fig. 8). For example, if the slap direction falls in the counterclockwise region, the instruction "please rotate your

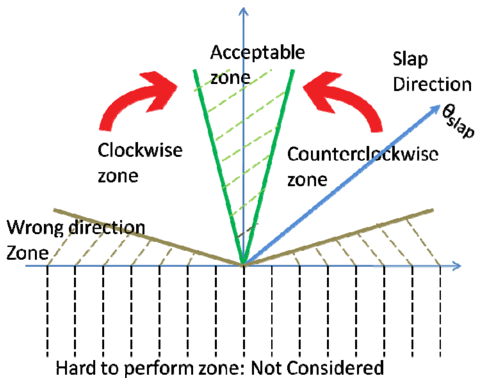


Figure 8: The slap orientation instruction zones

four figures counterclockwise" is given.

3.4. Thumb orientation detection module

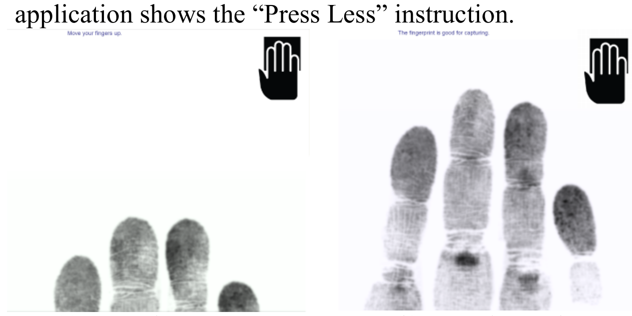
Thumb image analysis is similar to slap image analysis. The algorithm fits the thumb region with the ellipse. The major axis direction of the ellipse is considered as the thumb direction. Similarly, we assume that the thumb is not pointing downward. The instruction is given based on the instruction zone where the thumb direction is located.

3.5. Translation detection module

Once the four fingertip positions are known, the application also provides translation instructions as needed. The four slap translation instructions are "Move Left" (when one of the finger candidates is out of or too close to the right edge of the capture device), "Move Right", "Move Up", and "Move Down". By detecting whether there is any candidate in the border region, it is easy for the algorithm to give "Move Left/Right" and "Move Down" instructions. In those cases, it is not necessary for the algorithm to know where the fingertips are located, as long as any candidate is in the border region of the image, the instruction should be given. However, if any candidate is on the bottom region of the image, the algorithm needs to know if it is a fingertip: if so, the algorithm gives "Move Up" instruction; if not, the "Move Up" instruction should not be given. Fig. 9 gives an illustration. In case (a), the fingertips are in the bottom of the image, the instruction "Move Up" is given. In case (b), although the fingerprint is also presented in the bottom of the image, the fingertips are well located in the image. Thus the "Move Up" instruction should not be given.

3.6. Press more/less detection module

In some cases, when the user applies little pressure to the tips of the fingers, or the fingers are very dry, the sensor captures a low intensity image. According to [10], a minimum pressure of about 100 g is required to initiate the capture process, implying that instructing a user to push harder to trigger the capture process may be of value. Thus, the application checks the average intensity of each fingertip region. If the average intensity is lower than the threshold, the application presents the "Press More" instruction to guide the user to apply more pressure on the fingertips. On the other hand, if the fingers are too wet, the intensity of the image is saturated (very dark). The application shows the "Press Less" instruction.



(a) Instruction: "Move up" (b) Instruction : "Ok"

Figure 9: "Move Up" instruction.

4. Experiment

4.1. Experiment setup

Experiments have been carried out using a CrossMatch Guardian Fingerprint scanner with a fire-wire interface. The specifications of the scanner are: 500 ppi resolution, 3.2" x 3.0", single prism, single imager, uniform capture area. Based on the LScan Essentials SDK, we developed an application for a usability experiment of the fingerprint capture application. The proposed algorithm has been implemented in Visual C++ 2008. The implementation of many of the image analysis algorithms are based on OpenCV (Open Source Computer Vision) [13], a library of programming functions for real-time computer vision. Qt, a cross-platform application and a user interface (UI) framework is used for the development of the UI. The system runs on a 2.67 GHz, 2.75 GHz, Intel® core™2 CPU 6700 @2.66GHz processer PC.

During the experiment, users were seated in front of the desktop and were shown the fingerprints from the capture device on the screen in an indoor environment. We assumed a cooperative, overt, and non-attended (self-capture) environment. The real-time feedback, that is, text instructions, are directly given to the user on the screen. Without the help of an administrator or operator, the user responded to the system instructions. After the fingerprints passed the system evaluation criteria, they were automatically captured and saved by the system.

4.2. Bench testing

During the experiments, the system gives different feedback instructions to the user in real-time according to the fingerprint images captured online. We tested on both slap and thumb conditions. Due to the page limitation, we show the experimental results of the slap condition for the different instructions as follows:

4.2.1. Translation Instructions

Depending on the location of each fingerprint, the application provides "Move Left/Right, Up/Down" feedback to the user to make sure the fingerprints are all included in the capture area. Fig. 10 (a) - (d) shows the cases of the translation instructions.

4.2.2. Rotation Instructions

The application also makes sure that the fingerprints are approximately in the upright direction. According to the detected slap direction, the application provides "Rotate clockwise/counterclockwise" feedback to the user. Fig. 10 (e) and (f) show the cases of the rotation instructions.

4.2.3. Wrong Direction Instruction

If the slap is placed in the wrong way, the application provides "Wrong Direction; Put Four Fingers Upright" feedback to the user. Fig. 10 (g) and (h) show the cases of the wrong direction instructions.

4.2.4. Pressure detection modules

The application prescreens the fingerprint image quality by checking the average pixel intensity within the fingertip regions. If the user's fingers are very dry, or the user does not provide enough pressure to the fingertips, the region intensity becomes too light. The application shows "Press More" feedback to the user. On the other hand, if the user's fingers are too wet or greasy, the region intensity becomes too dark. The application shows "Press Less" feedback to the user (Fig. 10 (i) and (j)).

4.2.5. Finger Number detection modules

The application also detects if the number of the fingertips/thumb is correct, and gives the instruction if a finger is missing, or the number of candidates is more than expected. The following results in Fig. 10 (k) and (l) show that the application can correctly detect the correct number of the fingertips. For example, in Fig. 10 (l), even if the number of finger candidates described using ellipse fitting algorithm is greater than four, the number of the fingertips is less than four. The application correctly detects three fingertips, and shows that the finger number is not correct.

4.3. Speed evaluation

We also evaluated the algorithms' performance for speed during the experiment. The average speed for the different modules has been tested and evaluated individually. The results are shown in Table 2 (in milliseconds). It shows that the speed of the slap direction estimation for the fingertip selection algorithm is about 6 ms. The overall speed for the rotation instruction is comparable to the other instructions. On average, the overall time spent for the image processing is about 110 ~ 120ms for each image. The overall time used for the fingerprint capture device to capture a full resolution image (1600*1500) is about 230 ~ 550ms using the fire-wire CrossMatch Guardian. The image analysis algorithms consumes about 10-20% of the time in the overall tasks (500ms ~ 900ms) for the application to process a single fingerprint image.

5. Conclusion and future work

Applying a UCD methodology, the paper proposes a real-time usable biometric interactive system to improve the system accuracy, efficiency, and user satisfaction. We developed an innovative algorithm that counteracts potential performance degradation factors and translate them to corrective instructions for the user interaction. Our preliminary experimental results show that the proposed online algorithm can detect several factors such as finger/thumb locations, slap/thumb directions, and finger region intensity quickly at very limited system cost. Using our approach users no longer need to assess subjectively the quality of the captured biometric samples. In essence, our approach promotes more effective communication between the user and the self-capture system.

Based on the current system, we will continue our work in two aspects: first, we plan to study if providing an image overlay [14] affects user's interaction with a fingerprint system and the quality of the captured images; second, we will continue to study what kinds of instructional symbols [15] to replace the text instructions work best for the fingerprint capture user interaction interface.

6. Acknowledgement

The work was funded by the Department of Homeland Security Science and Technology Directorate.

Sub-sampling	4.71 ms
Adaptive Gaussian threshold	2.12 ms
Contour detection	0.48 ms
Ellipse fitting	1.27 ms
Overall finger selection	6.42 ms
(1) Rotate image 45 °	1.43 ms
(2) Sum image along four directions	4.24 ms
(3) Peak pattern selection	0.20 ms
Overall Rotation Instruction (≤ 4 candidates)	111.34 ms
Overall Rotation Instruction (> 4 candidates)	120.29 ms
Overall Translation Instruction	110.78 ms
Overall Press More/Less Instruction	119.38 ms
Overall Angle Lower Instruction	112.41 ms
Overall Less Finger Number Instruction	112.57 ms

Table 2: The speed performance of different modules [5]

References

[1] A. Jain, P. Flynn, and A. Ross. Handbook of Biometrics. Springer, 2007.

[2] A. K. Jain and D. Maltoni, Handbook of Fingerprint Recognition, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.

[5] The time measured is slightly different for different images. The numbers shown in the table are average numbers. For the overall performance, the time differences in the measurement is less than 15 ms. For the single module, the difference is generally less than 2 ms.

[3] R. Bolle, J. H. Connell, S. Pankati, N. K. Ratha, and A. W. Senior, Guide to Biometrics, Kluwer Academic Press, 1999.

[4] M. Theofanos, R. Micheals, and B. Stanton, Biometrics Systems Include Users, IEEE Systems Journal Special Issue on Biometric Systems, pp 461-468, January 2010.

[5] M. Theofanos, B. Stanton, and C. A. Wolfson, Usability & Biometrics - Ensuring Successful Biometric Systems, Handbook of National Institute of Standards and Technology (NIST), Gaithersburg, US, June 11, 2008, available at zing.ncsl.nist.gov/biousa/docs/Usability_and_Biometrics_final2.pdf.

[6] M. Theofanos, R. Micheals, J. Scholtz, E. Morse, and P. May, Does habituation affect fingerprint quality?, in CHI '06: CHI '06 extended abstracts on Human factors in computing systems, 2006, pp. 1427– 1432.

[7] ISO, ISO/IEC 19795-1: Information technology – Biometric performance testing and reporting - Part 1: Principles and framework, ISO/IEC, Editor. 2006, ISO/IEC JTC 1/SC37: Geneva.

[8] M. Theofanos, B. Stanton, R. Micheals, and S. Orandi, Biometric systematic uncertainty and the user, IEEE Biometrics: Theory, Applications, and Systems, 2007. BTAS 2007. First IEEE International Conference on, pp. 1–6, Sept. 2007.

[9] B. Stanton, M. Theofanos, and Charles Sheppard, A Study of Users with Visual Disabilities and a Fingerprint Process, NISTIR 7484, National Institute of Standards and Technology (NIST), Gaithersburg, US, Feb. 2008, http://zing.ncsl.nist.gov/biousa/docs/NIST-IR%207484%20access_study.pdf

[10] M. Theofanos, R. Michaels, S. Orandi, B. Stanton, N. Zhang, How the subject can improve fingerprint image quality, Journal of Electronic Imaging 17(1), 011007, Jan-Mar, 2008.

[11] R. Wong, N. Poh, J. Kittler and D. Frohlich, Interactive Quality-Driven Feedback for Biometric Systems, IEEE Biometrics: Theory, Applications, and Systems, 2010.

[12] C. I. Watson, M. D. Garris, E. Tabassi, C. L. Wilson, R. M. McCabe, S. Janet, and Kenneth Ko, User's Guide to NIST Biometric Image Software Export Control (NBIS-EC), National Institute of Standards and Technology, http://www.nist.gov/itl/iad/ig/nbis.cfm.

[13] G. Bradski, The OpenCV Library, Dr. Dobb's Journal of Software Tools, 2000.

[14] M. Theofanos, B. Stanton, R. Micheals, Y. Choong, Usability Testing of an Overlay to Improve Face Capture, IEEE Biometrics: Theory, Applications, and Systems, 2009, September 28-30, 2009.

[15] Y. Choong, B. Stanton, and M. Theofanos, Biometric symbol design for the public – case studies in the United States and four Asian countries, 3rd International Conference on Applied Human Factors and Ergonomics (AHFE), 17-20 July, 2010, Miami, Florida, USA.

(a) Right hand "Move Left" (b) Right hand "Move Right"
(c) Right hand "Move Up" (d) Right Hand "Move Down"
(e) Right hand "Rotate Clockwise." (f) Right hand "Rotate Counterclo."
(g) Left hand "Wrong Direction" (h) Right Hand "Wrong Direction"
(i) Right hand "Press More" (j) Left hand "Press Less"
(k) Right Hand "Less Finger" (l) Left Hand "Less Finger"

Figure 10: Instructions Results.