Improving Partial Fingerprint Recognition

Costas, Andrea
ayc2@rice.edu
Rice University

Boult, Terrance
tboult@vast.uccs.edu
University of Colorado at Colorado Springs

Abstract— **In today's world, biometrics are becoming more and more popular for the levels of security and convenience they provide to users. Apple has been including fingerprint scanners on their most recent iPhones, and a variety of other big names in technology are moving in the same direction. With the increased use of fingerprints, the need to have better and smaller scanners is increasing. With that need comes an urgency to develop forms of authentication using smaller amounts of information. The ability to use partial fingerprints effectively would be incredibly beneficial moving forward. This approach would require finding a balance between user security and program efficiency. This work proposes an approach to finding that balance through the use of SIFT features.**

Index Terms— **fingerprints, partial fingerprints, SIFT features, fingerprint analysis**

I. Introduction

The rise[2] of biometrics as a means of security has been a long time coming. It has helped redefine user authentication procedures by antiquating the use of passwords and PIN numbers. Passwords and PINs can be stolen with relative ease, but fingerprints on the other hand are unique and fairly difficult to fake. Fingerprints provide added security with the promise of making sure no one can access an account or device that does not belong to them. Fingerprinting is far from a new concept, but one that has encountered and continues to encounter many challenges throughout the course of its development.

Maltoni et al. [1] pointed out that the adoption rate of biometrics has been slower than what was expected due to a general lack of awareness of its potential. For a long time, the use of biometrics has been greeted with hesitation. This has all been changing over the past few years now that cell phone manufacturers are placing fingerprint scanners in everyone's hands. The recent incorporation of fingerprint verification systems in cell phones has eased a lot of public concern leading to an increase in the popularity of biometrics in the commercial world. In this commercial world, the ability to reduce production costs is valuable, and one way to do that is by creating smaller products. So the ability to effectively use smaller scanners escalates in value. Smaller scanners mean smaller fingerprint regions which increases the need for programs that can authenticate users working with relatively small pieces of information while also ensuring that other users are not incorrectly authenticated.

There exist fingerprint matching algorithms already, however, none of them has proved very effective working with small prints. This work consists of testing current, widely-used matching procedures and then some that are not as widely-used to see what proves most effective and promising.

II. Problem Definition

Improving the means of working with partial fingerprints would have significant implications for the uses of biometrics, more specifically, the uses of fingerprints in commerce. There have already been applications of partials in consumer products such as cell phones, but there is potential for expansion into other products and fields. Fingerprint protected credit cards, USB drives, and more are all feasible through effective uses of partial fingerprints. But in order to use them, there must be ways to confidently authenticate them. Therefore, current authentication procedures must be assessed for their effectiveness, and then a new method may be necessary in order to see any actual improvement.

III. Previous Work

There has been plenty of research conducted in the past on fingerprints and fingerprint matching methods and their challenges. However, most of these methods are unsuitable when considering the properties (or lack thereof) of partial fingerprints. Many of the key features within fingerprints are lacking when working only with a small portion, which forces the development of a new field of thought when studying such prints. However, there is still not a lot of information regarding the best way to approach this.

A. Sensor Sizes

Mainquet et al. [2] studied sweep sensors and reasonable minimal sizes for those sensors. They used minutia-based identification software and found that 7mm was the minimum width for a sweep-type sensor to perform with acceptable accuracy. It is important to note that sweep sensor prints are quite different from partial fingerprints as they can often result with fairly full-sized images. These sensors essentially take multiple partial images and arrange them together to create a full, mosaic-like image.

[1] A. Costas is participating in a Research Experience for Undergraduates (REU) with the Department of Computer Science, University of Colorado Colorado Springs, CO, 80918 USE under NSF Award No. 1359275

[2] http://www.csoonline.com/article/2891475/identity-access/biometric-security-is-on-the-rise.html

[3] The term "partial fingerprints" is used to mean small fingerprint regions as technically speaking, all fingerprints are partial fingerprints.

[4] http://zwipe.com/news/zwipe-introduces-genuine-hid-technology-biometric-cards/

B. Single-Chip Sensor and Identifier

Shigematsu et al. [3] researched a chip architecture that was comprised of both a fingerprint sensor and a fingerprint identifier. The identifier works with an array of pixels that are processed in parallel, so each fingerprint is handled and studied pixel by pixel. Once the print was passed through the sensor, the chip would store the data of an initial print scan. Later, when a new finger was placed on the chip, it would generate an image of the fingerprint by sensing the shape of it. It proceeded to binarize that image and shift it around to allow for various finger positions. Finally, it would compare the two prints and report its result. In testing, the chip worked correctly 99% of the time. This study focused more on creating smaller scale fingerprint sensors and did not test how effective it was when working with partial prints but the rapid processing it provides makes it an intriguing option to look into in the future.

C. Multi-pass Matching Algorithm

Jea et al. [4] presented a multi-pass partial fingerprint matching algorithm that overcomes many of the typical challenges partials provide such as rotations and distortions by using localized features. Their algorithm is based on triangular matching and secondary features/ minutia points. It accounts for distortions of minutia and different orientations that could be caused by differences in pressure during the fingerprinting process. In many matching algorithms, many matching algorithms work by aligning two fingerprints and finding a direct correspondence between minutia points. In this algorithm, however, does not require alignment. It basically tries to find local matches, and then expand onto a more global-scale to try to conclude if the prints matched. This algorithm proved effective and provided tolerance of a large number of distortions and effective use of secondary and localized features.

This all sounds great and useful, however, the smallest size of prints they worked with were a 60% of the original full sized print. According to their graph, this means that they had anywhere from 25 to 55 minutia points, so the partials were still of significant size and much larger than the partials generated in section V-A.

D. Region of Interest Minutia Matching

Bhargava et al. [5] studied the limitations of image processing, particularly fingerprint processing, when it comes to image quality. They present a solution to this problem by choosing a Region Of Interest (ROI). ROI is essentially a segmentation procedure that helps represent the image in a simpler way. It can make analysis easier by making the image appear more meaningful. The minutia within the ROI is first marked, then the locations of those minutia points are compared for verification. This provides an interesting idea for if a partial can be considered an ROI and matched as such, however ROI picks the best possible, most representative region, and partials are not necessarily representative at all.

E. Minutia Recognition

There have been a variety of different studies along the lines of partial fingerprint recognition, but in this one in particular, Jea and Govindaraju [6] conducted research on a minutia-based partial fingerprint recognition system. They developed a system that uses localized secondary features of minutia and made use of a neural network. They acknowledged that a brute force approach to this problem would be unfeasible due to how many possibilities the program would have to consider. They also discarded approaches that make use of global features seeing as how partial fingerprints can often have no global features at all. The neural network they developed was based on a system of similarity scores and showed that the accuracy of the fingerprint matching improved when working with images larger than 0.32” x 0.46”. This size is approximately 60% of a full-sized fingerprint. When the prints were smaller than that, the performance dropped dramatically.

F. Fingerprint Recognition Using Robust Local Features

Mishra et al. [19] pointed out that there already exist recognition techniques for fingerprints and that most rely on minutia matching methods that are not rotation-invariant. For this reason, they often fail with transformed prints and partials. They tested SIFT’s performance matching prints and found that SIFT was effective in matching and feature extraction, and although they tested on smaller prints, they did not test on anything similar to the sizes seen in this paper.

G. SIFT Fingerprint Identification

Zhou et al. [20] discussed that the original SIFT algorithm would not be suitable for fingerprint identification due to the similar patterns of fingerprint ridges. They proposed a minutia descriptor based on SIFT in hopes of improving how quickly prints can be verified. They had good results not only working with regular prints but also cracked and low quality prints as well.

IV. Proposed Solution

The proposed solution is comprised of a few major steps. The first pertains to acquiring the data that is to be used for training and, eventually, testing. The second has to do with creating a baseline of sorts. This means testing the performance of pre-existing matching algorithms in order to have a comparison for performance later. The final step is dependent on the previous step. If the matching algorithms perform well, then machine learning can be used to create a program that automatically chooses the best matching algorithm for a task depending on the fingerprint. If the matching algorithms do not perform well, then machine learning will be used to create a new, more effective way of matching partial prints to their full-sized counterparts. The data created in the first step will be used to train a computer to identify matches and non-matches.

A. *Data Collection*

This step is primarily necessary for the training process. Full sized fingerprints from the 2000, 2002, and 2004 Fingerprint Verification Competitions (FVC) [1] will be the source of data for this research. Each year has approximately 3,500 different full-sized fingerprints. All of them will be used. These full-sized prints will be partitioned into a variety of smaller images of a size that would resemble the partial prints acquired by a small sensor. In this case, three different sizes are being generated. The largest size is 192 by 192 pixels. This is the size on MasterCard and Zwipe's new biometric credit cards[3]. There are also sizes of 128 x 128 and 96 x 96. Although these sizes are not regularly used, they would provide a useful comparison to possibly gauging how small is too small. The full-sized fingerprints will also be rotated and then partitioned in order to provide data on different transformations of prints that can occur. These prints will be used later to train a deep neural network to either choose which is the best matching algorithm to use, or to create a new network that learns when prints match and when they do not. Only about 80% of the generated partials will actually be used in training. The rest will be reserved for testing.

B. *Base Line*

This steps consists of creating a baseline or, in other words, testing the performance of existing matching algorithms on generated partial prints. A few different matching algorithms will be tested. Minutia Cylinder Code (MCC) [7], Protected Minutia Cylinder Code (PMCC) [8], and Bozorth3 [9] which is the NIST Biometric Image Software (NBIS) matching algorithm. These algorithms have proved as effective on full sized prints, however, they have never been tested on a smaller scale.

1) Minutia Cylinder Code: Capelli et al. [7] introduced the Minutia Cylinder Code (MCC) in a paper aiming to create a fingerprint matching algorithm that would focus solely on local minutia matching and combine the advantages of neighbor-based structures and fixed-radius structures while cutting out the drawbacks of each. The neighbor-based approach focuses on the K spatially closest neighbors while focusing on some central minutia point. This method is tolerant of sparse and missing minutia points. It comes with a couple of drawbacks such as sophisticated local matching and problems with the handling of radial angles. The fixed-radius approach can also lead to mismatch die to local distortions or location inaccuracy. The basic idea behind MCC is that a local structure is assigned to each minutia. These structures represent spatial and directional relationships between that minutia and those in its surrounding neighborhood. These parameters end up being represented by a cylinder where the base and height correspond to each parameter.

2) Protected Minutia Cylinder Code: Protected Minutia Cylinder Code (PMCC) is based off of the MCC algorithm but its main purpose is to keep minutia templates from being acquired from PMCC templates. It still uses the same system of cylinders to work, however, each cylinder is transformed permanently in order to provide that ”protection” it promises.

3) Bozorth3: Bozorth3 is NBIS's minutia matching algorithm [9][10][11]. Essentially, the algorithm computes relative measurements from each minutia and builds comparison and compatibility tables that can combine clusters to calculate a match score. The higher this score is, the more likely it is that fingerprints in question came from the same person.

C. *Learning and Moving Forward*

The final part depends on the previous part. If the matching algorithms perform well, then the next step would be to use machine learning to learn how to choose the best matching algorithm for a given task. If the matching algorithm performs poorly, this step could consists of searching for other methods of matching prints and analyzing them or creating a deep neural network and training it to classify partial fingerprints as matches or non-matches to a full sized fingerprint.

Some of the other alternatives to traditional fingerprint matching would be using and testing feature descriptors for their performance on prints. Using SIFT would be a good possibility simply because it has never been tested on prints as small as the ones studied in this paper. Other papers have achieved good performance using SIFT descriptors and matching on large prints.

V. Preliminary Results

A. *Data Generation*

Each image in the 2000, 2002, and 2004 FVC databases was passed through an algorithm that simply, after traversing every few pixels, an image was generated that was 96 x 96 pixels, another that was 128 x 128, and another that was 192 x 192. These sizes represent different portions of a full-sized print. The 192 x 192 is about $\frac{1}{4}$ of a full-sized print. The 128 x 128 is about $\frac{1}{8}$ and the 96 x 96 is just a bit bigger than $\frac{1}{16}$ of a full print. To provide some context, an Apple TouchID scanner uses prints that are 160 x 160 pixels which means they are approximately $\frac{1}{6}$ of a standard full print.

Not all of the generated partials were kept. An analysis was done to examine the average contrast over each generated image. This was done to get an estimate as to how much white space there was in each new partial. If an image had too much white space, it was discarded. See image . Such images basically did not have a large enough print portion on it for them to be worth keeping. They would not provide substantial information, so they would be essentially useless in the future when it came to testing and training. See Table 1 for more information on how many images were kept and how many were discarded on average.

B. *Feature Extraction*

In order to get the matching algorithms discussed in part IV-B working, they must be provided with a list of the minutia in each of the prints to be matched. Regularly, a full-sized fingerprint could have around 100 minutia points,

TABLE I

AVERAGE NUMBER OF IMAGES GENERATED AND KEPT FOR A SAMPLING OF FINGERPRINTS

Images kept with image size and shift size (5)			
	96 x 96	128 x 128	192 x 192
Generated	5616	4615	3016
Kept	3717	3305	2481

but with the smaller sizes in the generated data set, there are not nearly that many minutia points. So generally, fingerprint images must be passed through some kind of minutia extraction algorithm. The minutia extractor originally chosen was DigitalPersona's FingerJetFx. When trying to extract from the generated images, the FingerJetFX extractor failed completely. It worked excellently on full sized images, but upon trying it on the partials, it refused to run. It would not run even on the largest of the partial sizes. Another minutia extractor was tried, this time it was NBIS MindTCT extractor. The MindTCT was able to extract minutia. But the failure of the FingerJetFX raised a red flag.

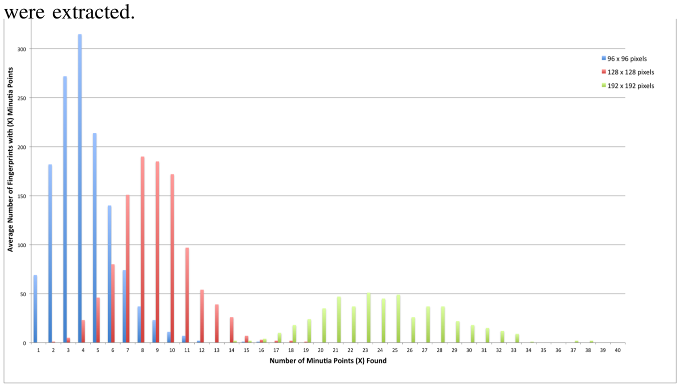
The matching algorithms most likely had a minimum number of minutia required to be able to try to match two prints. If the partials had less minutia than the lower bounds of the matching algorithms, then it would be impossible to test the effectiveness of those algorithms in the first place. Or rather, the algorithms would immediately be deemed to be ineffective on small prints. Looking into it further, the MCC is programmed to only run if there are at least 4 minutia points available. The PMCC requires at least 10 as does the Bozorth3 algorithm.

With this information on the limits of the matching algorithms, it was time to find out how much minutia could be extracted on average from the different sized of partial prints. Figure 1 shows the relationship between how many minutia points were found on average for partial fingerprints. The partials of size 96 x 96, generally found between 1 and 5 minutia points per image. The 128 x 128 found about around 6 to 11 minutia points and the 192 x 192 found anywhere from 20 to 28 minutia points.

When considering how many minutia points were extracted and the limitations of the different matching algorithms, it seems that creating a neural network to choose the best matching algorithm for a specific purpose may be, essentially, useless. None of the matching algorithms previously discussed would be able to process most of the 96 x 96 partials. The algorithms might be able to process some of the 128 x 128 partials, but only the MCC would really be able to work with most of them. This means that only the 192 x 192 holds good possibilities to be matched using the different matchers discussed.

The question now becomes what can be done to improve the matching processes of the smallest fingerprints in the database?

Fig. 1. Relationship between fingerprint sizes and how many minutia points were extracted.



C. SIFT Features

Following down the path of some other researchers, testing SIFT seemed promising. Before matching, partial prints were put through a contrast normalization algorithm and they were also blurred. This was done to remove any sort of randomness and inconsistencies between different print scans.

Here a couple of prints were compared with partials generated from seven different scans of the same finger. In other words, if finger A was scanned eight times, then finger A-1 is compared with the generated partials of finger A-2 through finger A-8. Due to limited times, this was only tested on 2 different fingers, but moving forward with this project, there will be many more sets of prints tested. But here are the results for how matching went with different sizes of partials.

In the first experiment, the threshold was set to 10. This means that in matching, there had to be at least ten good matches (where ”good” is defined as matches that perform well under Lowe's ratio test []). If there were less matches than that, then there would not be a conclusive match. See Figure 2 . Not surprisingly, the size 192 partials matched best here by matching just over 30 percent of the time. Yet, the results leave a lot to be desired.

The next experiment tested a lower threshold of 5 in hopes of getting improved performance on the matches. While the results improved, it was still not enough to be considered satisfactory. On average, the size 192 was accurate a little more than 50% of the time.

The threshold was lowered once more to 1. Only one match was necessary in order to match a partial to a full sized print. See Figure 3. Finally, there were better results.

It seemed concerning to lower the threshold so much, considering that one match is seemingly nothing to prove any degree of accuracy or security. However, a couple of brief preliminary tests were conducted with non-matching pairs. On both of the tests, the threshold was set to 1, the lowest of all the thresholds previously tested for true matches. When the results came back, all of them came back negative. That is to say that no prints or partial prints from different fingers matched at all. This is promising for the continued testing of this method, but not enough to prove anything definitely.

Prints will continue to be tested to be sure that this method does not pose any serious security risks for users.

Fig. 2. Percentage of partial prints accurately matched per partial size while under a threshold of 10.

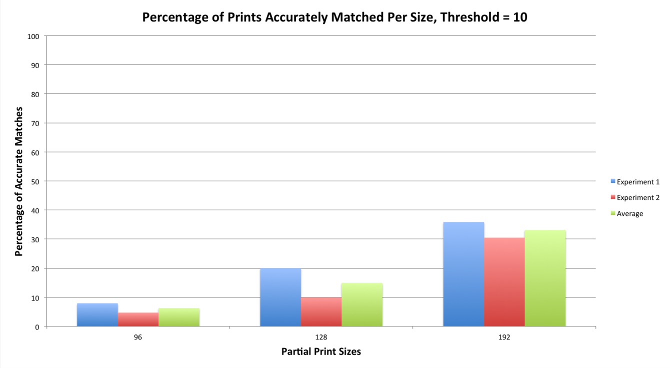


Fig. 3. Percentage of partial prints accurately matched per partial size while under a threshold of 5.

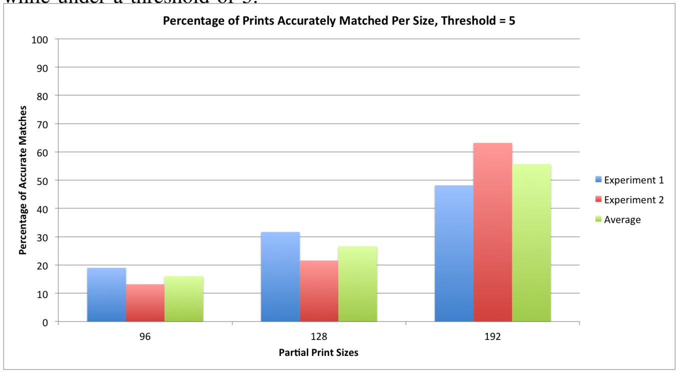
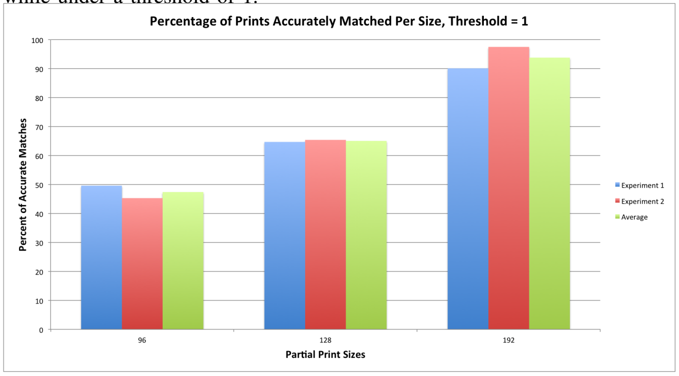


Fig. 4. Percentage of partial prints accurately matched per partial size while under a threshold of 1.



VI. Conclusion

It is far too soon to have any conclusions other than some educated guesses as to which path to take moving forward with this research. Definitively, using only minutia based matching methods would not be effective for such small prints, but that does not mean that they cannot serve some other purpose in the future.

VII. Future Work and Improvements

Work is far from over. Despite some challenges with testing, the use of SIFT has probed promising for authenticating fingerprints so it seems that the next step should be to try other robust feature transforms. Aside from Scale-Invariant Feature Transform (SIFT) [21], there is also Speeded Up Robust Features (SURF) [18], Binary Robust Invariant Scalable Keypoints (BRISK) [16], KAZE, and Fast Retina Keypoint (FREAK) [22]. All of these hold potential for improving partial fingerprint recognition.

Leutenegger et al. [16] proposed BRISK (Binary Robust Invariant Scalable Keypoints), another method for feature detection and matching. It primarily uses brightness comparisons to form a binary description. It compares well with SIFT and SURF, so it will be worth looking into in the future, but if it relies solely on the binary description, it may not be suitable for fingerprint matching simply because the prints are already represented in a binary fashion.

Proceeding on-wards, after testing other feature transforms, it may be interesting to combine a few methods. This could mean either combining a couple of feature transforms or combining a feature transform with minutia data or both. Now minutia points and keypoints have been used together in the past, but not at a scale as small as this one, which leaves the door open for more research. In the future, the use of a neural network for this will be considered as well. The hopes are to see more improvement in the matching of the smaller prints like the 96 x 96 and the 128 x 128.

References

[1] D. Maltoni, D. Maio, and A. K. Jain, Handbook of fingerprint recognition [With DVD ROM]. United Kingdom: Springer-Verlag New York, 2009.

[2] J.-F. Mainguet, W. Gong, and A. Wang, ”Reducing silicon fingerprint sensor area,” in Biometric Authentication, Springer, 2004.

[3] S. Shigematsu, H. Morimura, Y. Tanabe, T. Adachi, and K. Machida, ”A single-chip fingerprint sensor and identifier,” Solid-State Circuits, IEEE Journal of, vol. 34, no. 12, 1999.

[4] T.-Y. Jea, V. S. Chavan, J. K. Schneider, and V. Govindaraju, ”Partial Fingerprint Matching.”

[5] ”Region of interest minutia matching - Google Search.” [Online]. Available: https://www.google.com/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=region%20of%20interest%20minutia%20matching. [Accessed: 08-Jul-2016].

[6] .-Y. Jea and V. Govindaraju, ”A minutia-based partial fingerprint recognition system,” Pattern Recognition, vol. 38, no. 10, 2005.

[7] R. Cappelli, M. Ferrara and D. Maltoni, ”Minutia Cylinder-Code: a new representation and matching technique for fingerprint recognition”, IEEE Transactions on Pattern Analysis Machine Intelligence, vol.32, no.12, December 2010.

[8] A. Rozsa, A. E. Glock, T. E. Boult, ”Genetic Algorithm Attack on Minutiae-Based Fingerprint Authentication and Protected Template Fingerprint Systems”, CVPR2015.

[9] R. Vaan, ”NIST MINDTCT and Bozorth3 Review — SourceAFIS.” .

[10] ”Biometric System Laboratory.” [Online]. Available: http://biolab.csr.unibo.it/research.asp?organize=Activities&select=&selObj=81&pathSubj=111%7C%7C8%7C%7C81&Req=&. [Accessed: 03-Jun-2016].

[11] N. US Department of Commerce, ”NBIS.” [Online]. Available: http://www.nist.gov/itl/iad/ig/nbis.cfm. [Accessed: 10-Jun-2016].

[12] A. K. Jain, J. Feng, and K. Nandakumar, ”Fingerprint Matching,” Computer, vol. 43, no. 2, 2010.

[13] G. Hinton, ”Deep belief networks,” Scholarpedia, vol. 4, no. 5, p. 5947, 2009.

[14] ”CS231n Convolutional Neural Networks for Visual Recognition.” [Online]. Available: http://cs231n.github.io/convolutional-networks/. [Accessed: 10-Jun-2016].

[15] S. Chopra, R. Hadsell, and Y. LeCun, ”Learning a similarity metric discriminatively, with application to face verification,” in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05), 2005, vol. 1.

[16] S. Leutenegger, M. Chli, and R. Y. Siegwart, BRISK: Binary robust invariant scalable keypoints, in 2011 International conference on computer vision, 2011.

[17] F. Alcantarilla, A. Bartoli, and A. J. Davison, KAZE features, in European Conference on Computer Vision, 2012.

[18] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, Speeded-up robust features (SURF), Computer vision and image understanding, vol. 110, no. 3, 2008.

[19] R. Mishra and R. Mishra, Fingerprint recognition using robust local features, International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, no. 6, 2012.

[20] R. Zhou, D. Zhong, and J. Han, Fingerprint Identification Using SIFT-Based Minutia Descriptors and Improved All Descriptor-Pair Matching, Sensors (Basel), vol. 13, no. 3, Mar. 2013.

[21] T. Lindeberg, Scale invariant feature transform, Scholarpedia, vol. 7, no. 5, 2012.

[22] A. Alahi, R. Ortiz, and P. Vandergheynst, Freak: Fast retina keypoint, in Computer vision and pattern recognition (CVPR), 2012 IEEE conference on, 2012.