Technical Disclosure Commons

Defensive Publications Series

May 29, 2019

Fingerprint-Matching Algorithms

Firas Sammoura

Kuntal Sengupta

Jean-Marie Bussat

Fingerprint-Matching Algorithms

Abstract:

This publication describes a fingerprint-matching algorithm that uses a fusion of minutiae and pattern-correlation matching. Instead of extracting minutiae points, the algorithm extracts small patches from the fingerprint image and transforms them into rotationally-invariant vectors. The algorithm divides each image to be evaluated, herein called an "enrolled" image, into "M" number of patches with a sliding distance of one (1) pixel. The algorithm also extracts "N" number of random patches from a stored image, herein called a "verify" image, and calculates the similarity between the "verify" rotationally-invariant vectors and the "enrolled" rotationally-invariant vectors. At this stage, the algorithm merges vectors from the "enrolled" images using a rotation and translation matrix and drops redundant vectors based on a quality score. The outcome of matching is the number or matching blocks that show similar translation and rotation. Finally, the algorithm generates a "Yes-or-No" outcome based on a predetermined threshold number of matching blocks.

Keywords:

Fingerprint, iris scan, palmprint, infant footprint, biometric, authentication, matching, cartesian coordinates, Fast Fourier Transform, FFT, Absolute-value Fast Fourier Transform, AFFT, rotational invariant vector, minutiae, feature, pattern, batch, image block, correlation, area sensor, sample image, verify image, enrolled image.

Background:

Since antiquity, various civilizations have used fingerprinting to identify individuals because human fingerprints are detailed, nearly unique, difficult to alter, and durable over the life of an individual, making them suitable as long-term markers of human identity. In modern times, virtually all countries use fingerprinting in some form or another to some of their citizens. Nevertheless, fingerprinting is more than a government or a legal tool for identifying alive or deceased individuals — many user equipment (UE) give a user a choice to use fingerprinting to gain access to a user device, door entrance, vault, application software, and other physical locations or virtual activities that the user wants guarded. Figure 1 illustrates an example of a fingerprint image.

Figure 1

The analysis of fingerprints for matching purposes generally requires the comparison of patterns and minutiae features. The three main patterns of the fingerprint ridges are the arch, the loop, and the whorl. An arch is a fingerprint ridge that enters from one side of the finger, rises in the center forming an arc, and then exits the other side of the finger. A loop is a fingerprint ridge that enters from one side of the finger, forms a curve, and then exits on that same side of the finger. A whorl is a fingerprint ridge that is circular around a central point. On the other hand, the minutiae

are features of fingerprint ridges, such as ridge ending, bifurcation, double bifurcation, trifurcation, short or independent ridge, island, lake or ridge enclosure, spur, bridge, delta, core, and so forth, as illustrated in Figure 2.

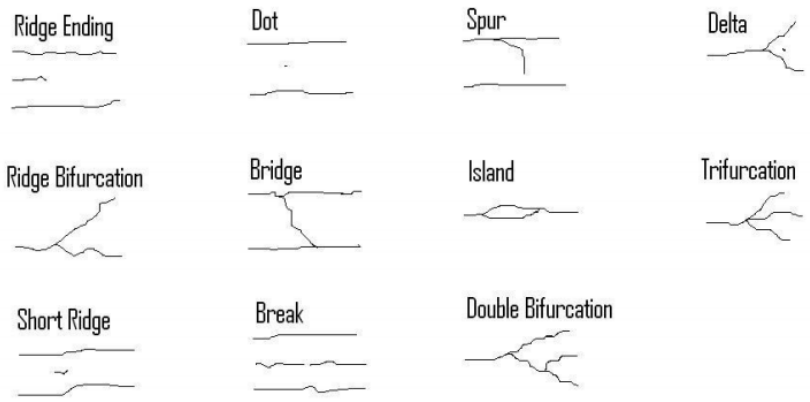

Figure 2

The UE may use a fingerprint sensor to capture a fingerprint image. When the UE uses a large fingerprint sensor, minutia matching is achieved with high success. Nevertheless, some UE, such as smartphones, use small fingerprint sensors, which decreases the number of minutiae being scanned. The UE struggles to make a positive minutiae matching when it can only scan a few number of minutiae (refer to Figure 2).

The small fingerprint sensor is one of the reasons why many UE manufacturers use pattern-correlation matching, and they often try to correlate the full fingerprint. First, the UE tries to match the alignment and the orientation of the fingerprint, then the UE correlates the full fingerprint image. This technique, however, cannot easily support 1000 Dots-Per-Inch (DPI) resolution fingerprint images. A higher-resolution image can increase the fingerprint-matching success rate.

It is desirable to use more than one type of matching to increase the image resolution and the fingerprint-matching success rate. To this end, this publication discusses a technique that fuses minutiae matching and pattern-correlation matching.

Description

This publication describes a fingerprint-matching algorithm that uses a fusion of minutiae and pattern-correlation matching. Instead of extracting minutiae points, the algorithm extracts small patches from the fingerprint image and transforms them into rotationally-invariant vectors. Rotationally-invariant vectors allow the algorithm to rotate a patch in any direction. The outcome of these vectors' transformation will be the same and mapped to the same vectors regardless of the orientation. Essentially, this algorithm replaces the minutiae with a pattern, treats the pattern like a minutiae, assigns a location and an orientation to the pattern, and proceeds with the minutiae-matching technique. To some, this algorithm may appear like a pattern-correlation technique, but this pattern-matching technique works as an instance's correlation-based matching. That is, this algorithm does not take an image and converts it to another form of that image. This algorithm can work with the raw image data, and it is more hardware-friendly.

The algorithm divides each image to be evaluated, herein called an "enrolled" image, into "M" number of patches with a sliding distance of one (1) pixel. For an incomplete image, one may alter the algorithm to extract a less than "M" number of patches with a sliding distance greater than one (1). Therefore, one may modify the algorithm to increase the computation speed by evaluating only part of the full image.

The algorithm extracts vectors from each patch by including the following:

- Rotationally invariant Absolute-value Fast Fourier Transforms (AFFTs) of each block;
- The patches' x-position and y-position — the Cartesian coordinates;
- The patches' polar representation of the Cartesian coordinates; and
- The patches' Fast Fourier Transforms (FFTs) of the polar representation with a high resolution in the theta (θ) direction.

Figure 3 helps demonstrate the relation between an image-block in Cartesian coordinates, the image-block's polar coordiantes representation, and the image-block's AFFT in polar coordinates.

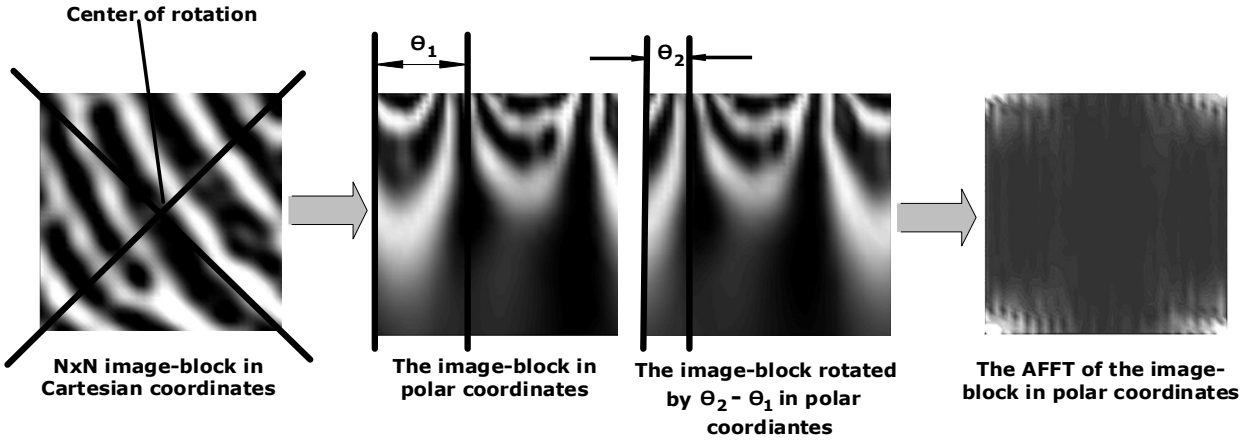

Figure 3

Figure 3 illustrates the polar representation of an "NxN" image-block. The angular rotation around the center point in the Cartesian coordinates transforms into translation along the theta (Θ) direction in the polar coordinate representation — this is called "phase shifting." The FFT assumes periodic boundary conditions. As such, the AFFT of the "NxN" image-block represented in polar coordinates is rotationally invariant, and the rotation angle is the location where the maximum correlation between the FFT of two (2) image-blocks represented in polar coordinates occurs.

The algorithm uses rotational and translation matrices, where the rotation and translation matrix between two (2) images, herein referred to as "image 1" and "image 2," can be defined as:

$$\begin{pmatrix} \cos(\phi) & \sin(\phi) & -T_x \\ -\sin(\phi) & \cos(\phi) & -T_y \\ 0 & 0 & 1 \end{pmatrix}$$

where "ϕ" represents the angle between the two (2) images, "T_x" represents the translation along the x-axis between the two (2) images, and "T_y" represents the translation along the y-axis between the two (2) images.

The x-coordinates and the y-coordinates of "image 2" can be transformed into the coordinate system of "image 1" using Equation 1.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\phi) & \sin(\phi) & -T_x \\ -\sin(\phi) & \cos(\phi) & -T_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad \textbf{(Equation 1)}$$

Furthermore, the rotation matrix between "image 1" and "image 2," herein called RM_{12}, is the inverse of the rotation matrix between "image 2" and "image 1," herein called RM_{21}, as shown in Equation 2.

$$RM_{12} = (RM_{21})^{-1} \quad \textbf{(Equation 2)}$$

Considering another property of the rotation matrix, RM_{12} can be determined from the rotation of a third image, herein called "image 3," with "image 1" and "image 2," as shown in Equation 3.

$$RM_{12} = RM_{32} * RM_{13} \quad \textbf{(Equation 3)}$$

where RM_{12} represents the rotation matrix between "image 1" and "image 2," RM_{32} represents the rotation matrix between "image 3" and "image 2," RM_{13} represents the rotation matrix between "image 1" and "image 3," and * denotes the mathematical operation of the convolution between RM_{32} and RM_{13}.

Rotation matrices can also be used to "stitch" images. For example, one may generate a "stitched image" of "image 1" and "image 2" by performing the following computations:

- Calculate the rotation matrices RM_{12} and RM_{21};
- Form the x-vectors, y-vectors, and z-vectors of "image 1" and "image 2;"
- Transform the x-coordinates, the y-coordinates, and the z-coordinates of "image 2" into the coordinates of "image 1;"

- Concatenate the x-vectors, the y-vectors, and the z-vectors of “image 1” and the transformed “image 2;”
- Define a new mesh grid, which is limited by the maximum and the minimum of “image 1” and the transformed “image 2;”
- Interpolate the z-vector on the new mesh grid; and
- Generate the stitched image, as demonstrated in Figure 4.

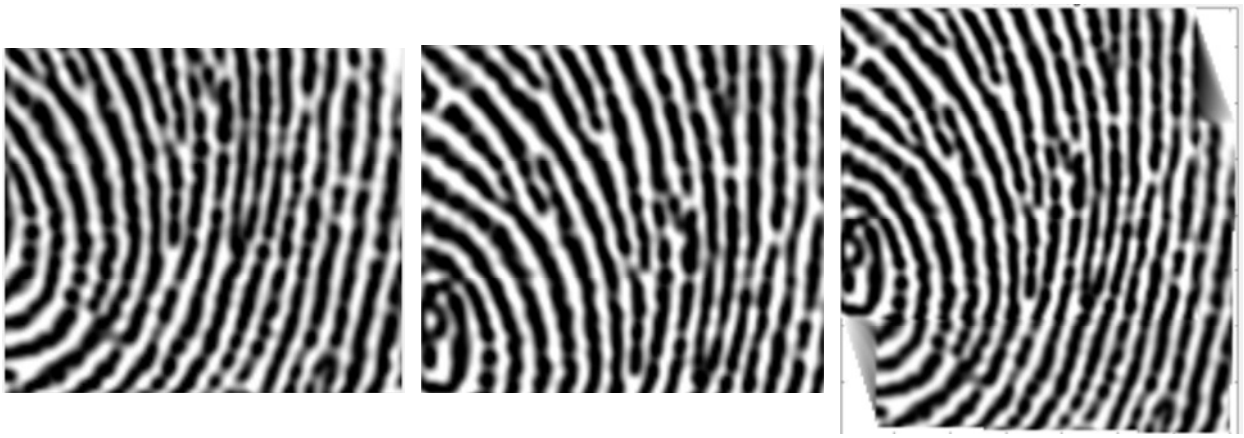

Figure 4

In addition to dividing each “enrolled” image to “M” number of patches, the algorithm extracts “N” number of random patches from a stored image, herein called a “verify” image. The algorithm calculates the similarity between the “verify” patch vectors and the “enrolled” image vectors along with the angle of rotation and the correlation. The algorithm, then, outlines the x-coordinate, the y-coordinate, and the angle correspondence between the “verify” blocks and their matching blocks in the “enrolled” image and calculates the translation in the x-direction and y-direction for each “verify” block. At this stage, the algorithm merges vectors from the “enrolled” images using a rotation and translation matrix and drops redundant vectors based on a quality-score. The algorithm drops redundant vectors using this quality-score to rank the top-ten

translation and rotation vectors. This algorithm, however, may rank a different number of translation and rotation vectors that are used in "voting" the most-likely translation and rotation vector.

Furthermore, the algorithm may be modified to "stitch" down-sampled images. For example, the algorithm may down-sample "verify" images by decreasing the number of patches (*e.g.*, from N=31 to N=23). This in turn, reduces the resolution of the "stitched" images, but it also reduces the processing time. Figure 5 demonstrates the effect of down-sampling "stitched" images.

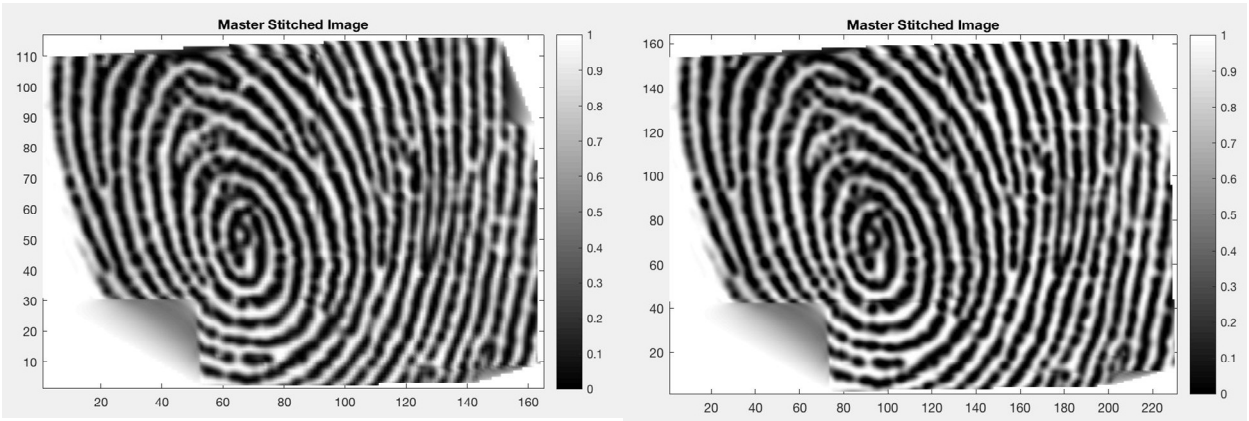

Figure 5

Lowering the number of patches (*e.g.*, from N=31 to N=23) reduces the resolution, but also reduces the matching time (*e.g.*, from 0.58 seconds to 0.28 seconds). As demonstrated in Figure 5, the algorithm that produces a "stitched" image with a slightly lower resolution reduces the matching time by almost fifty percent (50%). Therefore, the algorithm may easily be modified in the interest of reducing the matching time with little sacrifice to the matching success-rate.

The outcome of the matching is the number of matching blocks that show similar translation and rotation. To increase the robustness of fingerprint matching, a small error is allowed in the translation and rotation to account for variations due to skin-plasticity distortions.

Finally, the algorithm generates a "Yes-or-No" outcome based on the threshold number of matching blocks (*e.g.*, 9 out of "N" blocks).

This algorithm can be scaled to handle higher than 1000-DPI images and large area fingerprint images. Furthermore, this algorithm can be used for other forms of biometric matching, such as iris, palmprint, and baby footprint. In that aspect, this algorithm is versatile. One limitation of this algorithm is matching a perfect pattern (*e.g.*, a perfect zebra pattern) because in that case each block is identical. This limitation, however, becomes irrelevant because biometric patterns are not perfect. It is that imperfection and uniqueness that gives the biometric pattern its value.

To test the matching success rate of this algorithm, a biometric experiment was conducted, where four fingers from 50 users were evaluated. All "verify" and "enrolled" images were stitched for visualization purposes. For each "enrolled" image, a template vector was built. Then, each template vector was matched against all other finger images. Figure 6 demonstrates the success rate of the biometric experiment.

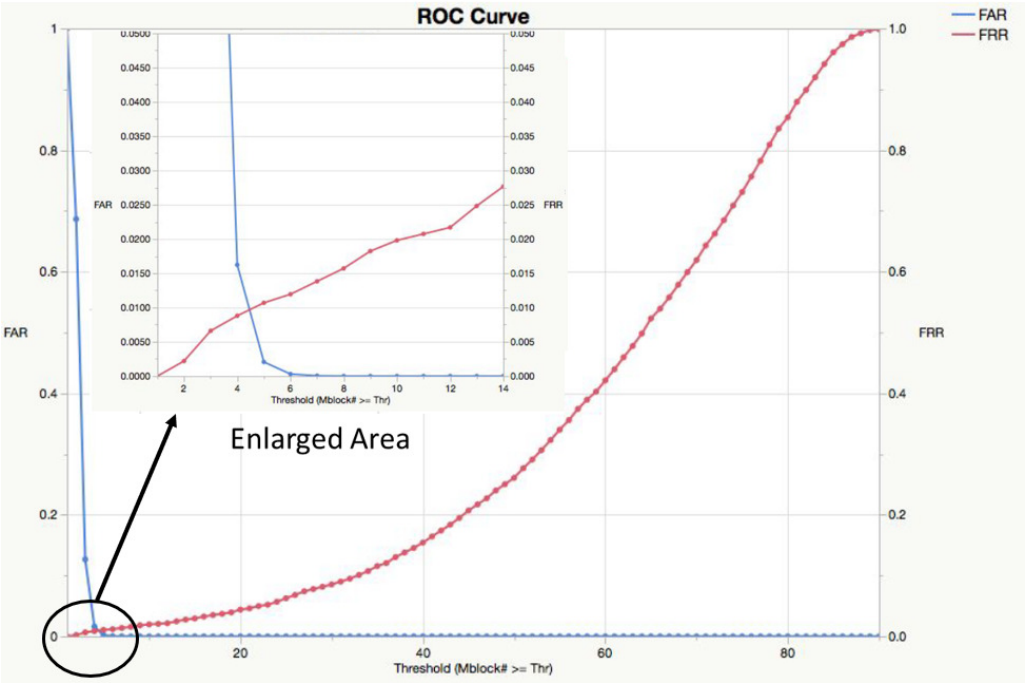

Figure 6

Figure 6 displays a receiver operating characteristic (ROC) curve, which is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. Biometric security uses false acceptance rate (FAR) for the proportion of times a system grants access to an unauthorized person, and false rejection rate (FRR) for the proportion of times a biometric system fails to grant access to an authorized person. The enlarged area of Figure 6 demonstrates that the experimental event rate (EER) of this biometric experiment is approximately 0.01 (or 1%).

In addition, a template update algorithm was used, which updates an existing template vector using the vectors of a matching image. The template update decreases the EER, as demonstrated in Figure 7.

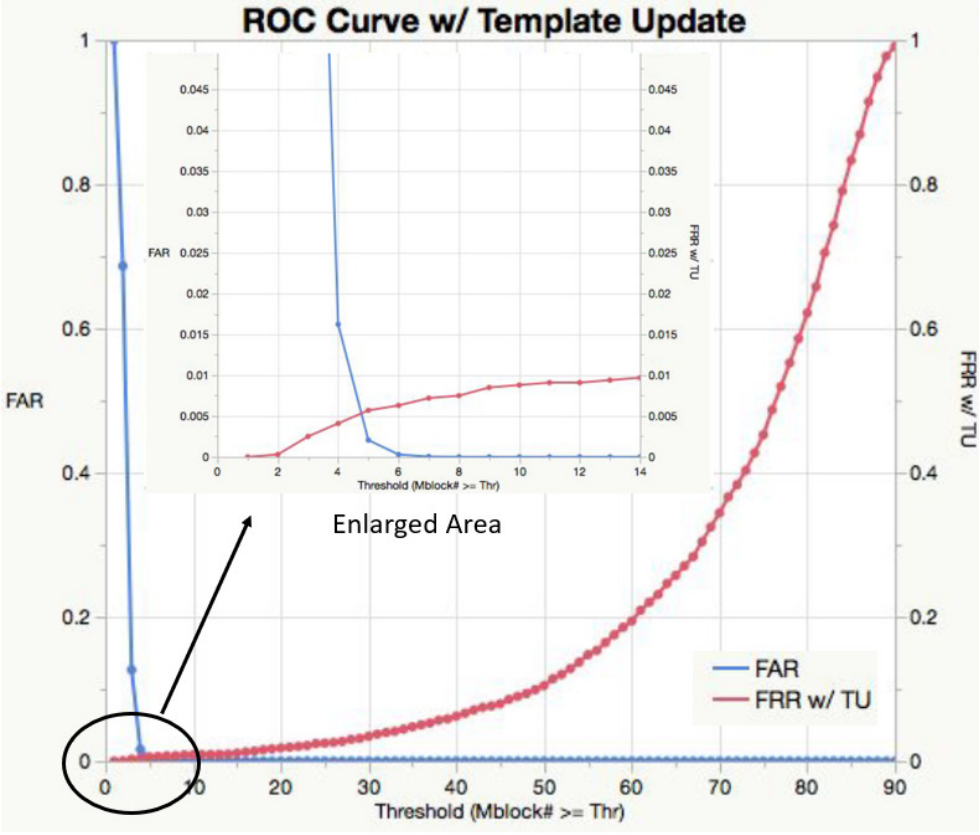

Figure 7

The enlarged area of Figure 7 demonstrates that using the template update the experimental event rate (EER) of this biometric experiment is approximately 0.006 (or 0.6%).

This algorithm increases the matching success-rate even in instances when a person may have cuts on their finger-pads. To show how this algorithm accomplishes this, consider Figure 8.

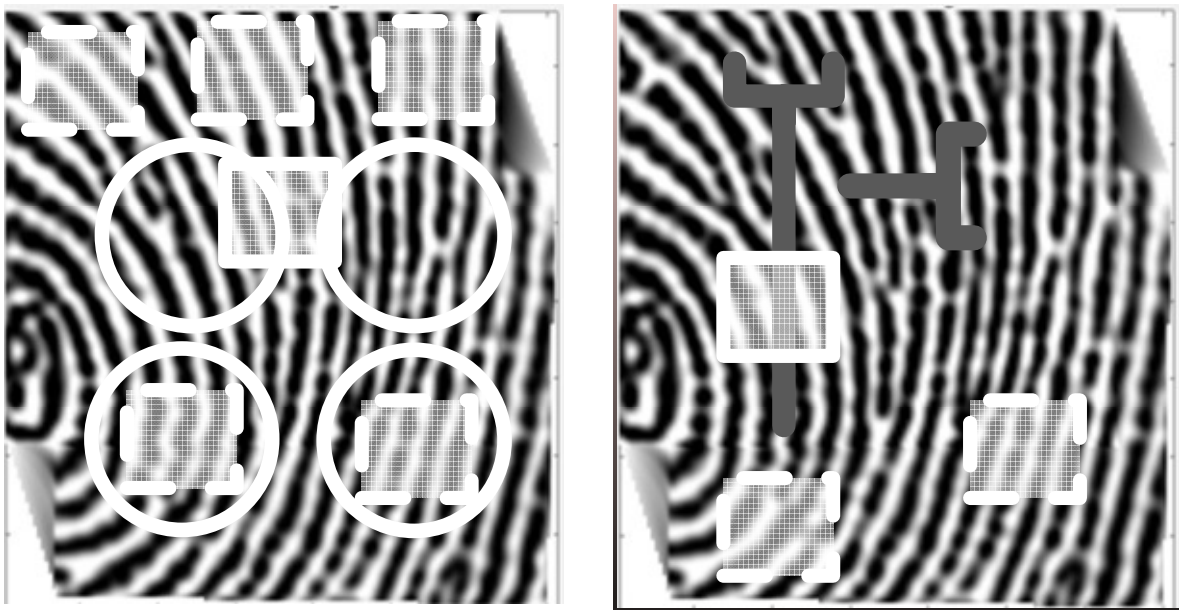

Figure 8 Legend

The two (2) solid-line squares are poor patches for matching

The seven (7) dash-line square are acceptable patches for matching

Figure 8

Figure 8 uses fingerprint images with a logo. In the Figure 8 example, the solid-line patches represent poor patches for matching, while the dash-line patches represent acceptable patches for matching. The logo on the image to the right may represent a recent cut on the person's finger-pad. For matching fingerprint images with a logo, the algorithm avoids patches that interfere with that logo. It does so by avoiding patches that interfere with the logo during the extraction of the "enrolled" images and avoids selecting the patches that interfere with the logo in the "verify" images. To increase the success-rate in matching the fingerprints of finger-pads with cuts, the logo

design may be optimized, such that it allows for the largest number of valid patches. This matching technique is advantageous over traditional minutiae matching or pattern-correlation matching because the logo can corrupt the directionality map.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs, or features described herein may enable collection of user information (*e.g.*, a user's preferences or a user's current location), and if the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user's identity may be treated so that no personally identifiable information can be determined for the user, or a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user. The user may also choose to disable the fingerprint-matching algorithms from their device.

In summary, the described fingerprint-matching algorithm, which uses a fusion of minutiae and pattern-correlation matching techniques is superior over minutiae matching or pattern-correlation matching.

References:

[1] Stoianov, Alexei. Fingerprint identification system for access control. US Pub. 2007/0248249, filed April 20, 2006, and published October 25, 2007.