Improving Hand-Based Verification Through Online Finger Template Update Based on Fused Confidences

Gholamreza Amayeh, George Bebis and Mircea Nicolescu
Computer Science Department, University of Nevada, Reno, NV 89557

Abstract— **Since the biometric data tends to have a large intra-class variability, it is possible for the enrolled templates to be significantly different from acquired samples during system's operation. The majority of existing techniques in the literature, namely self update, update a template set by using a confidently verified input sample in order to avoid the introduction of impostors into the template set of a client. Therefore these techniques can only exploit the input sample very similar to the current template set leading to local optimization of a template set. To address this issue, this paper introduces a technique by decomposing the hand silhouette into the different parts (i.e. fingers) and analyzing the confidences of these parts in order to lead to global optimization of templates. In the proposed method, first the hand silhouette is divided in different parts corresponding to the fingers. Then the confidence of each finger, as well as its identity, is evaluated by a Support Vector Data Description (SVDD). The confidence of a query hand is determined by the maximum confidence of all fingers. If the maximum confidence is higher than a threshold, the boundaries of all fingers' SVDDs are incrementally updated to learn the variations of the input data. The motivation behind this technique is that the temporal changes that may occur in the fingers are uncorrelated in such a way that the confidence of each finger can be significantly different from the others. As a result those fingers with difficult intra-class variations (low confidence) can be used in the update process by this technique. The experimental results show the effectiveness of the proposed technique in comparison to the state of the art self-update technique specially at low false acceptance rates.**

Key words: Biometric, Hand Verification, Online Template update, Fusion.

I. Introduction

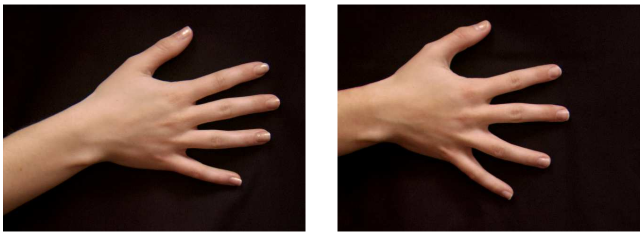
A typical biometric verification system operates by acquiring biometric data (i.e. hand) from a subject and comparing it against the template set of that subject, stored in a database, in order to verify a claimed identity. Most systems store multiple templates of a person in order to account for variations observed in biometric data. In fact the biometric measurements tend to have a large intra-class variability. Thus, it is possible for the stored template data to be significantly different from those obtained during system's operation, resulting in an inferior performance (higher false rejection rate) of the biometric system. Figure 1 shows an example of this variation. As you can see, shape of little finger in Figure 1(b) is curvier compare to one in Figure 1(a). Also thumb is bent more in Figure 1(b). More over cutting fingernails in Figure 1(b) causes some changes in the shapes of point, ring and little fingers (i.e. length of the finger).

Substantial intra-class variations are exhibited in the input data, non representative of the enrolled templates, which decreases the performance of the system. This issue has been recently faced by template update techniques. The earlier approaches, known as supervised learning methods, were based on enrolling multiple templates per person representing temporary variations, and by repeating the process of enrolment over period of the time to capture variations in the biometric data. Uludag et al. [1] proposed two simple methods to perform template update using the newly acquired data. In the first method, namely *Batch Update*, all current templates are replaced with templates selected from the newly acquired data set, thereby capturing temporal changes (i.e. in fingerprints) [1]. In the second method, namely *Augment Update*, both the current template set and the newly obtained data set are considered when performing template update [1]. These methods need a supervisor, who has to assign identity labels to the input data to be used for update, and it makes the update process very expensive, time consuming, and inefficient.

To over come the drawback of aforementioned methods, self update methods based on semi-supervised learning have been developed. They are self update systems as they update themselves by iteratively classifying the unlabeled samples and modifying the enrolled templates with highly confidently classified data, using their own knowledge gained from previously augmented template set. These techniques can be categorized as *Online* and *Off-line* methods. In online update methods [2][3], templates are updated as soon as an input data arrives, however in off-line update methods [4][5][6] templates are updated after the batch of unlabeled data is collected.

Jiang and Ser [2] proposed an online fingerprint template updating algorithm by merging the input data into the template database during the system's operation. Ryu et al. [3] proposed an online minutiae-based fingerprint template adaptation algorithm. The algorithm updates a template by using a query fingerprint, which is successfully verified by the fingerprint matcher as a high quality genuine input. Liu et al. [4] introduced an off-line update technique based more on the recent samples and less on the older samples with application in face recognition. Roli et al. [5] proposed an off-line semi-supervised version of the classical PCA-based face recognition algorithm to update the eigenfaces and the templates. Recently Rattani et al. [6] proposed an off-line graph-based approach to template update by its application to face verification, as a case study.

Self update techniques operate at high acceptance threshold in order to avoid the introduction of impostors into



(a) (b)

Fig. 1. Temporal change of a hand shape over 16 week time lapse.

the template set of a client. The impostors' introduction leads to the so called effect of *creep in* of errors which strongly decrease the effectiveness of update. Moreover, due to operation at high acceptance threshold, these approaches can only exploit the input data near to the current templates resulting in local optimization of the template set and non exploitation of many difficult and informative intra-class variations.

This paper introduces a global optimization approach to hand template update based on fusion. This method analyzes overall confidence of a hand by decomposing the hand silhouette in different parts (i.e. fingers) and fusing confidences of all fingers in order to better exploit difficult samples and use them in template update process. The maximum rule is employed to fuse confidences of all fingers. So, to update a hand template set, having a high confident finger in the verified hand sample is sufficient. The motivation behind this technique is that the temporal changes that may occur in the fingers are uncorrelated in such a way that the confidence of each finger can be significantly different from the others. As an example, consider the hand image in Figure 1(a) as a enrolled template and the other one in Figure 1(b) as a query sample. A typical self update system can not exploit the little, ring, index and thumb fingers due to having large variation from current enroled template, just middle finger very similar to the current template can be exploited. In our proposed method, since the middle finger is classified as a high confidence sample, therefore all fingers can be exploited by the system. However the proposed method has the potential to identify difficult intra-class variations compare to a typical self update system, the effect of imposter introduction in the proposed method can be worse than the typical self update system. The reason is that in the typical self method each introduced imposter affects only a specific finger of a client, while in our proposed method an introduced imposter affects all fingers of a client. To avoid that a tighter acceptance threshold, compare to the typical self update method, is chosen in the proposed system. The matching for each finger is performed by a Support Vector Data Description (SVDD). Support Vector Data Description (SVDD) is a technique which uses support vectors in order to model a data set [7]. The SVDD represents one class of known data samples (i.e. a subject templates) in such a way that for a given test sample it can be recognized as known (i.e. genuine attempt), or rejected as unknown (i.e. imposter attempt).

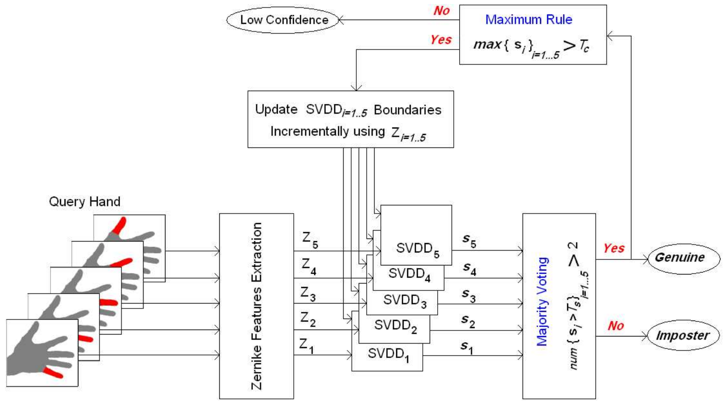


Fig. 2. The block diagram of the proposed hand template update scheme.

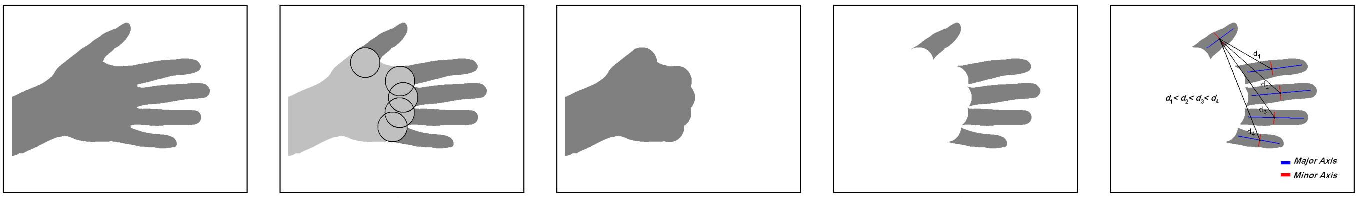
The rest of paper is organized as follows. The details of the proposed template update technique is explained in section 2. Section 3 shows the experimental results, followed by the conclusions in Section 4.

II. Proposed Method

Figure 2 shows an overview of the proposed template update scheme. First fingers are segmented from the hand silhouette using a morphological-based algorithm [8]. Then, in feature extraction module, the geometry of the fingers are represented implicitly using Zernike descriptors [8]. The matching score, s_i, of each part of the query hand (i.e. finger) is evaluated by a Support Vector Data Description (SVDD) which is one class classifier [7]. To validate the claimed identity, the matching scores are fused using majority voting strategy [8]. After verification of a claimed identity, to obtain the confidence of the query hand, the confidence of all fingers are fused using maximum rule. If the query hand recognized as a high confident genuine input,then the decision boundary of all SVDDs corresponding to the fingers are updated incrementally [9] using the current Zernike feature vectors. In the following subsections, we explain the main stages of the system in details.

A. Pre-Processing

This module includes the segmentation of the hand silhouette into different regions corresponding to the fingers. To segment fingers from the hand silhouette, we used the same algorithm described in [8]. The processing steps of the finger segmentation module are shown in Figure 3. A morphological closing operator based on a circular disk is applied on the hand image as shown in Figure 3(b). The radius of the structuring element was experimentally set to 45 pixels, making it thicker than the widest finger in the database. The closing operation filters out the fingers from the silhouette as shown in Figure 3(c). The remaining part of the silhouette corresponds to the forearm and palm, which is subtracted from the hand image to obtain the finger segments as shown in 3(d). To identify each finger automatically, major and minor axes of all fingers, as shown in Figure 3(e), are computed using first and second order of geometric moments [10]. Then the ratio of the length of major and minor axes



(a) (b) (c) (d) (e)

Fig. 3. (a) Hand silhouette; (b) structuring element; (c) the result of morphological closing; (d) the result of subtracting (c) from (a); (e) major and minor axes of the fingers and their distance to the thumb.

are computed. Our investigation on 688 hand images in our database indicated that the smallest ratio belongs to the thumb. The reason is that in terms of length it is comparable to the little finger, but it is thicker than that. The rest of fingers can be identified corresponding to the distance of their center of mass from the thumb's center of mass as shown in Figure 3(e). As a result the closest finger to the thumb is point finger and after that the middle, ring and little fingers respectively. After segmentation of fingers, their geometric features are extracted. Following subsection describes feature extraction step.

B. Feature Extraction

In this step, to capture the geometry of the fingers, we employed Zernike moments as region descriptors. Amayeh et al. [8] represented the geometry of the different parts of the hand implicitly using Zernike moments. Zernike moments are based on a set of complex polynomials that form a complete orthogonal set over the interior of the unit circle [11].They are defined as the projection of the image on these orthogonal basis functions. Specifically, the basis functions $V_{n,m}(x,y)$ are given by

$$V_{n,m}(x,y) = V_{n,m}(\rho,\theta) = R_{n,m}(\rho)e^{jm\theta} \tag{1}$$

where n is a non-negative integer, m is a non-zero integer subject to the constraints $n-|m|$ is even and $|m| < n$, ρ is the length of the vector from origin to (x,y), θ is the angle between the vector ρ and the x-axis in a counter clockwise direction, and $R_{n,m}(\rho)$ is the Zernike radial polynomial which is defined as follows:

$$R_{n,m}(\rho) = \sum_{k=|m|,n-k=even}^{n} \frac{(-1)^{\frac{n-k}{2}} \frac{n+k}{2}!}{\frac{n-k}{2}!\frac{k+m}{2}!\frac{k-m}{2}!}\rho^k \tag{2}$$

The Zernike moment of order n with repetition m for a *digital* image function $f(x,y)$ is given by [12]:

$$Z_{n,m} = \frac{n+1}{\pi}\sum\sum_{x^2+y^2\leq 1} f(x,y)V^*_{n,m}(x,y) \tag{3}$$

where $V^*_{n,m}(x,y)$ is the complex conjugate of $V_{n,m}(x,y)$. To compute the Zernike moments of a given image, the center of mass of the object is taken to be the origin. The magnitude of the Zernike moments is rotation invariant by its definition (See Eq. 3). Taking the center of mass of the object as the origin of the coordinate system makes them translation invariant as well. Additionally, to provide scale invariance, the object is scaled inside the unit circle. As proposed in [8], here Zernike moments are computed up to order 20 for the fingers resulting in 121 features for each finger.

C. Matching and Update through SVDD

As mentioned earlier, the matching score s_i of each finger is evaluated by a Support Vector Data Description (SVDD). A normal data description gives a closed boundary around the data (i.e. a finger template set) which can be represented by a hyper-sphere $F(R,a)$. The volume of this hyper-sphere with center a and radius R should be minimized while containing all the data (i.e. finger templates). As proposed in [7] the extension to more complex distributions is straightforward using kernels. For generalization purpose, slack variables $\varepsilon_i \geq 0$ are introduced. The error function to be minimized is defined as:

$$F(R,a) = R^2 + C\sum_i \varepsilon_i \tag{4}$$

subject to:

$$\|z_i - a\|^2 \leq R^2 + \varepsilon_i \quad \forall i. \tag{5}$$

Using Lagrange optimization the above results in:

$$L = \sum_i \alpha_i K(z_i,z_i) - \sum_{i,j}\alpha_i\alpha_j K(z_i,z_j) \quad \forall\alpha_i : 0 \leq \alpha_i \leq C \tag{6}$$

where α_i is a Lagrange multiplier and $K(z_i,z_j)$ is a kernel function. In this study we employed radial basis function as kernel function. When a sample falls in the hyper-sphere then its corresponding Lagrange multiplier is $\alpha_i \geq 0$, otherwise it is zero. After optimizing the function in (6) the following equality constrain must hold:

$$\sum_i \alpha_i = 1 \tag{7}$$

When a query sample is applied to a trained SVDD, the output is its distance to the center of the hyper-sphere. In the context of our application, this distance (multiply by -1) is considered as matching score s_i. As a result matching score s_i can be in the range of $(-\infty,0]$.

In the enrollment stage, using templates of a client a SVDD classifier is trained for each finger. Therefore a subject is represented by a set of 5 SVDDs corresponding to his/her fingers. The support vectors and their corresponding Lagrange multipliers are stored as the classifier information for each finger. This information is used later in the verification process.

During system's operation, hand template update is performed iteratively by adding verified hand sample with high confidence. Using high confident hand sample, we update the Lagrange coefficients α_i in trained SVDDs classifiers using an incremental learning algorithm [9]. This method is based on the theorem proposed by Osuna et al. in [13].

D. Fusion

In general, fusion can be implemented at different levels such as decision level, score level, and feature level. In this study, we used two different fusion strategies at decision level (in verification process) and score level (in update process).

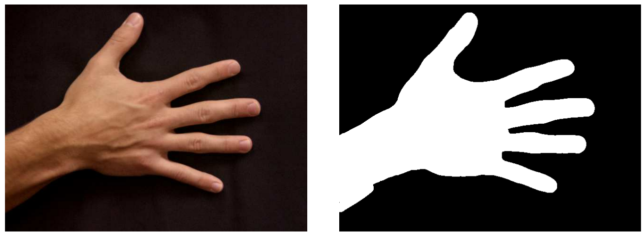
Majority voting is used in decision level to improve verification accuracy and robustness. Majority voting is among the most straightforward decision level fusion strategies. In this case, the final decision is based on the output results of several matchers. In the context of our application, first we verify each subject using different parts of the hand (i.e. fingers). Then, if three or more parts of the hand yield a positive verification ($s_i \geq T_s$), then verification is considered successful; otherwise, the subject is rejected.

After verifying an identity, maximum rule is used in score level to determine the overall confidence level of the query hand. Therefore the maximum matching score s_{max} of the fingers represent the confidence of the query hand. Higher matching score indicates higher confidence. If s_{max} is greater than a threshold T_c, then the query hand is used in update process. Usually this threshold T_c is much higher than the threshold T_s in verification process to avoid the introduction of impostors (false positive) into the system.

III. Experimental Results

We performed our experiments using a publicly available hand database provided by University of Notre Dame [16]. This database was created by collecting data on three different sessions. In the first session, two images from 132 subjects were collected. In the second session, which was conducted a week later, three images were collected from the same 132 subjects. The third session, which was conducted 15 weeks later from the second session, three images were collected from 177 subjects of which 86 people had participated in the first two data collections [16]. The database contains both range and color images, each being 640×480 in size. In our experiments, we used the color images of the same 86 subjects who participated in all three sessions. To extract the hand silhouette from a color image, we used the same algorithm described in [16]. As proposed in [16], we employed a combination of edge and skin detection techniques on the color image to extract hand silhouette from the intensity images. Figure 4 shows an intensity image sample and its hand silhouette extracted by aforementioned method.

In our experiments, all hand images in the first session were used as initial templates. Since only 2 templates represent the initial template set for each client, it can not exhibit a large intra-class variations of the hand. The rest of hand images in the second and third sessions, with 6 images per client, were used as queries. As a result the query images provided 516 genuine attempts and 43860 imposter attempts. It should be noted that a template image never becomes a query image even as an impostor for other clients.



(a) (b)

Fig. 4. (a) Color image sample and (b) extracted hand silhouette.

Due to training of the SVDD incrementally at each iteration, the order of training data may affect the learning process. To account for the variation in the order of the genuine queries, we repeated each experiment 10 times, each time choosing a random order of the genuine input for each client and reported the average performance. It should be noted that the relative order of genuine queries in second and third sessions does not change. In other words, a genuine sample from the third session never appears before any genuine query in the second session.

The threshold for self update techniques is always evaluated on initial template set, since it is the only set available in real environments. Threshold is evaluated on this template set by comparing each template to the templates of all the other clients thus estimating the impostor distribution and selecting a threshold value at a specific false acceptance rate (i.e. $FAR = 1\%$). Followed by the same methodology the threshold was chosen $T_c = -0.003$ in our experiments.

To make a base line for our experiments, we employed a conventional verification system which does not utilize any update scheme. In this system, SVDDs are trained once using initial templates in the first session. Therefore the order of query samples is not important in this case.

Figure 5 shows the performance of different fingers before and after utilizing a self update system at different confidence thresholds. Figure 5(e) indicates that the substantial intra-class variation in the thumb is so large. As a result its performance is the lowest one among other fingers and self update technique is not effective in this case. As you can see in Figure 5(a-f), adopting very high threshold (i.e. $T_c = -0.001$) limits the capability of the system to capture the substantial intra-class variations in the subject's input data. Also adopting low threshold (i.e. $T_c = -0.01$) introduces impostors into the template set of a client resulting in poor performance. Table I shows the true acceptance rate (TAR) of the conventional verification system before and after utilizing self update when false acceptance rate (FAR) is equal to 1%. As you can see in table I, self update technique improves the performance slightly as it can exploit only the patterns similar to the enrolled templates which leads to local optimization and non-exploitation of many difficult and informative intra-class variations.

TABLE I

TRUE ACCEPTANCE RATES OF DIFFERENT FINGERS BEFORE AND AFTER UTILIZING SELF UPDATE TECHNIQUE WHEN FALSE ACCEPTANCE RATE IS EQUAL TO 1%.

Method	Little	Ring	Middle	Point	Thumb
No Update	74.4%	76.2%	80.6%	74.5%	57.2%
Self Update	77.3%	77.0%	81.2%	75.7%	58.3%

For comparison purpose, we employed a typical self update system in which each part of the hand (i.e. finger) is updated independently using a high confident verified query finger (with respect to its matching score above a threshold T_c). As a result, each time the SVDD of one or more fingers might be updated using a query hand sample.

Figure 6 shows the performance of the hand verification system before and after utilization of the update techniques: typical self update and proposed fusion-based self update. Also table II shows the equal error rate and true acceptance rate (*TAR*) of the hand verification system, when *FAR* is equal to 0.1% and 1.0%, before and after utilization of the update techniques[1].

As it can be seen, the typical self update method improve the overall performance of the system slightly, while the proposed method improved the performance of system significantly. In fact the proposed approach can exploit difficult and informative intra-class variations. Our investigation indicated that there is no correlation between the confidence of different fingers in the hand, because the temporal changes that may occur in the fingers are uncorrelated. Therefore in a genuine hand, however some fingers may have high matching scores, some others may have very low confidence (i.e. thumb). In a typical self update system, these fingers with low confidence, which have informative intra-class variations, never can participate in the update procedure. In our proposed system, these fingers can contribute in the update process if and only if one of the fingers has a high confidence.

In the proposed method, the effect of imposter introduction is greater than the typical self update method. The reason is that in the typical self method each introduced imposter affects only a specific finger's SVDD, while in our proposed method an introduced imposter affects all fingers' SVDDs. To reduce the effect of imposters' introduction, a tighter acceptance threshold compare to the typical self update method is chosen in the proposed system. In figure 6 the threshold for the typical self update was -0.003 and for our proposed method was -0.0025.

IV. CONCLUSION

This paper introduces a technique by decomposing the hand silhouette and fusing the confidence of the fingers

[1]All these methods are implemented in MATLAB 7.6.0 using data description toolbox[15].

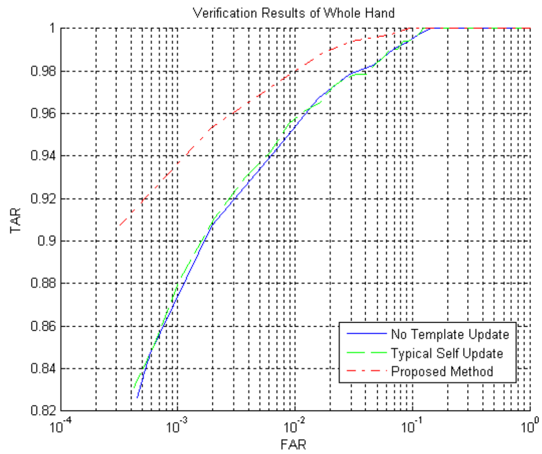


Fig. 6. Performance of the hand verification system before and after utilizing a typical self update and proposed fusion-based template update. In all methods, matching scores of the fingers were fused by majority voting technique in verification process.

TABLE II

TRUE ACCEPTANCE RATES AND EQUAL ERROR RATES OF THE HAND VERIFICATION SYSTEM BEFORE AND AFTER UTILIZING A TYPICAL SELF UPDATE AND PROPOSED FUSION-BASED TEMPLATE UPDATE.

Method	TAR(FAR=0.1%)	TAR(FAR=1%)	EER
No Template Update	87.4%	95.3%	2.50%
Typical Self Update	88.0%	95.8%	2.49%
Fusion-Based Update	93.6%	98.0%	1.50%

in order to lead to global optimization of templates. This update technique has the potential to identify difficult intra-class variations. The motivation behind this technique is that the temporal changes that may occur in the fingers are uncorrelated in such a way that the confidence of each finger can be significantly different from the others.

Experimental results in the case of hand verification have been very promising. Although this limited set of experiments does not allow to draw definitive conclusions, we believe that proposed approach to template update is worthy of further investigations. As a future work, we would like to investigate this frame work in multimodal biometric systems.

REFERENCES

[1] U. Uludaga, A. Ross and A. K. Jain, "Biometric template selection and update: a case study in fingerprints", *Pattern Recognition*, vol. 37, pp 1533-1542, 2003.

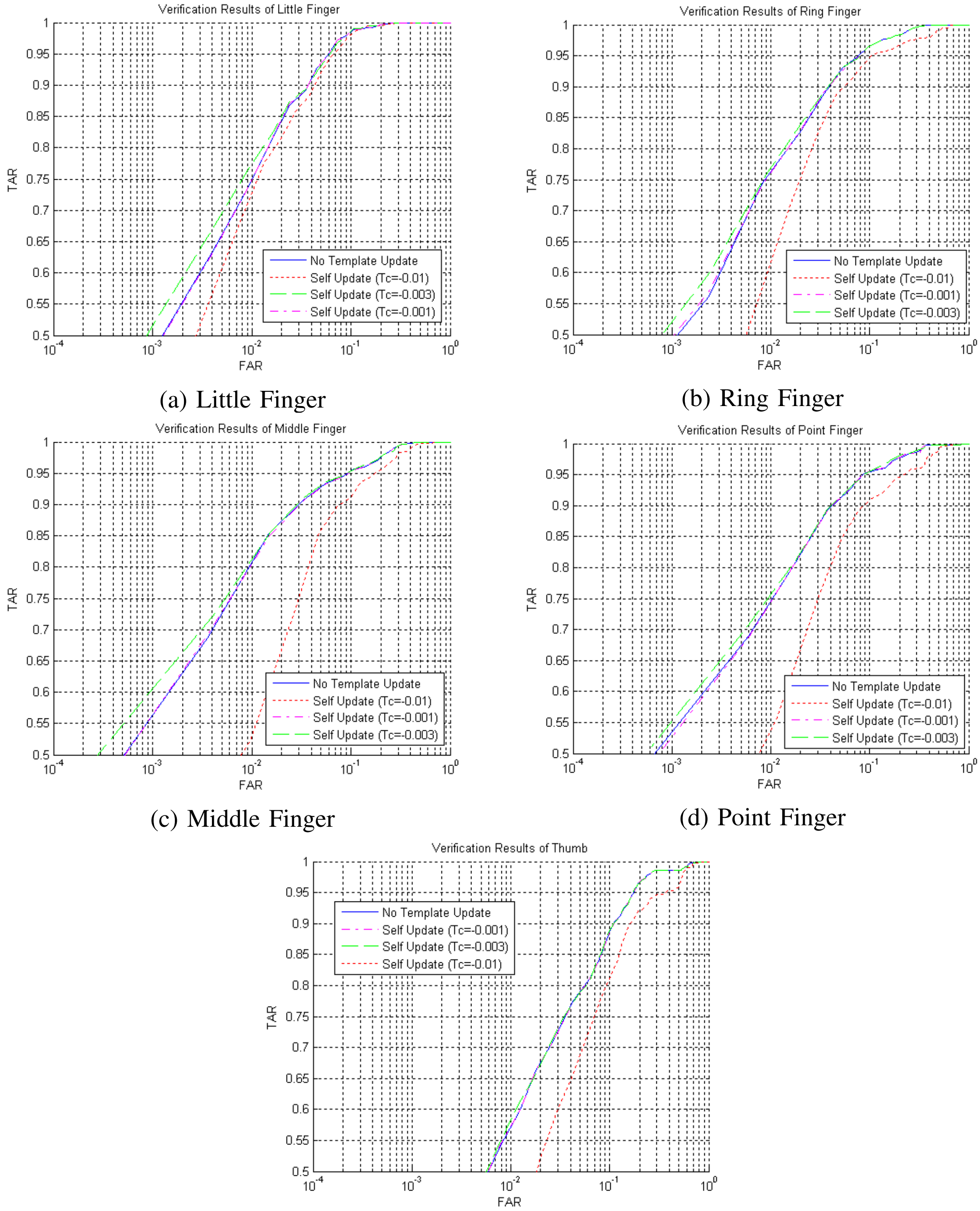
[2] X. Jiang and W. Ser, "Online Fingerprint Template Improvement", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp 1121-1126, 2002.

[3] C. Ryu, H. Kim and A. K. Jain, "Template Adaptation based Fingerprint Verification", *The 18th International Conference on Pattern Recognition*, vol. 4, pp 582-585, 2006.

[4] X. Liu, T. Chen and S.M. Thornton, "Eigenspace updating for non-stationary process and its application to face recognition", *Pattern Recognition*, pp 1945-1959, 2003.

[5] F. Roli and G.L. Marcialis, "Semi-supervised PCA-based face recognition using self training", *Int. Workshop on Structural, Syntactic and Statistical Pattern Recognition*, Springer LNCS 4109: pp 560-568, 2006.

[6] A. Rattani, G. L. Marcialis and F. Roli, "Biometric template update using the graph mincut algorithm : A case study in face verification", *6th IEEE Biometrics Symposium*, pp 23-28, 2008.



(a) Little Finger

(b) Ring Finger

(c) Middle Finger

(d) Point Finger

(e) Thumb

Fig. 5. Verification results of each finger using different confidence thresholds in a typical self update system. SVDD of each finger in a client profile is updates using a genuine finger input with matching score higher than T_c

[7] D. Tax and R. Duin, ”Support Vector Data Description”. *Machine Learning*, vol. 54, no. 1, pp 45-66, 2004.

[8] G. Amayeh, G. Bebis, A. Erol and M. Nicolescu, ”Hand-Based Verification and Identification Using Palm-Finger Segmentation and Fusion”, Computer Vision and Image Understanding (CVIU) vol. 113, pp 477-501, 2009.

[9] J. Platt. ”Fast Training of Support Vector Machines using Sequential Minimal Optimization”, *Advances in Kernel Methods - Support Vector Learning*, MIT Press, pp 185-208, 1998.

[10] Haralick and Shapiro, ”Computer and Robot Vision”, Ed. 1st, Addison-Wesley, 1992.

[11] F. Zernike, Beugungstheorie des schneidenverfahrens und seiner verbesserten form, derphasenkontrastmethode, Physica I, pp 689-704, 1934.

[12] A. Khotanzad and Y. H. Hong, ”Invariant image recognition by zernike moments”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp 489-498, 1990.

[13] E. Osuna, R. Freund and F. Girosi, ”Improved Training Algorithm for Support Vector Machines”, *In Proc. Neural Networks in Signal Processing*, 1997.

[14] A. Tavakkoli, M. Nicolescu, M. Nicolescu and G. Bebis, ”Incremental SVDD Training: Improving Efficiency of Background Modeling in Videos”, *Signal and Image Processing* , 2008.

[15] D. Tax, ”DDtools, the data description toolbox for matlab” version 1.7.3, 2009.

[16] D.L. Woodard and P.J. Flynn, ”Finger surface as a biometric identifier”, *Computer Vision and Image Understanding* vol. 100, pp 357-384, 2005.