Evaluation of Fusion Methods for Latent Fingerprint Matchers

V. N. Dvornychenko
National Institute of Standards and Technology (NIST), ITL
Gaithersburg, Maryland
USA

Abstract

Matcher fusion is a recognized approach for improving biometric system performance. Component matchers may be "encapsulated," reducing the need for understanding the inner workings of each matcher, and facilitating interchange of matchers. In large part, our interest in fusion was to determine how much performance "headroom" existed with current matcher technology. We employed five different latent fingerprint matchers. These matchers can use alternate input data (features), allowing the influence of "data type" to be investigated. Numerical results show it is possible to reduce the final candidate list to two to six candidates, with the probability that the true mate appears in the top (first) position boosted by 6 – 15%-points.

1. Introduction– Why Matcher Fusion?

Latent fingerprints have become increasingly important in criminal law enforcement, border security, and anti-terrorism. Traditional methods of latent fingerprint matching and identification require intensive human expert involvement. Automation is being called upon to reduce human dependence, increase throughput, and reduce turnaround time. Such matchers are referred to Automated Fingerprint Identification Systems (AFIS).

Presently automated matcher performance is still a limiting factor. Matcher fusion is an attractive method of improving biometric search performance: it can be implemented with relatively modest effort; the component-matchers can be "encapsulated," resulting in an architecture in which alternative matchers can be swapped in and out with relative ease; and multiple matchers, beyond two, can be accommodated.

As a result of testing under its ***Evaluation of Latent Fingerprint Technology*** (ELFT-EFS) project, NIST is in the favorable position of having performance data on an appreciable number of latent fingerprint matchers. Each matcher has the capability to accommodate a variety of input data types. These matchers were designed and implemented by professional biometrics companies, specifically for formal testing; and may be considered representative of the state-of-the art.

The general concept behind ELFT testing is found in [1]. Additional information on the testing methodology, the databases, the matchers, and the test results can be found in [2]. Steps in progressive automation are proposed in [3].

For this phase of the study we were most interested in the magnitude of performance gains achievable using different modes of operation (to be described). Significant improvements via such fusion are an indication that not all information is being used by each component matcher, and therefore there is "headroom" for further technology improvements.

2. Principles of Matcher Fusion

Matcher fusion works by adding information. This "added information" can fall into four basic categories: 1) additional external input; 2) additional extracted information (e.g., additional features used in searching); 3) additional information in the form of algorithmic improvements to the matcher (i.e., more "smarts"); and finally, 4) information regarding the strength and weaknesses of the component matchers.

Additional external information exists when several (different) copies of a search fingerprint are used during a single search. Section 6 focuses on this mode.

Four levels of fusion are commonly identified in the literature: 1) decision level; 2) score level; 3) feature level; and 4) image level. A somewhat different perspective, one focusing on implementation architecture, is shown by the following diagrams.

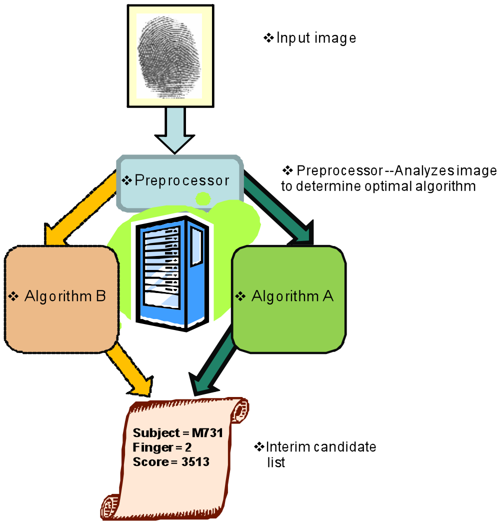


Figure 1: Fusion Architecture Employing Preprocessor

In Figure 1 a Preprocessor is used to select the

978-1-4673-0397-2/12/$31.00 ©2012 IEEE

appropriate matcher (called "Algorithm" in figures), following an examination of input image characteristics.

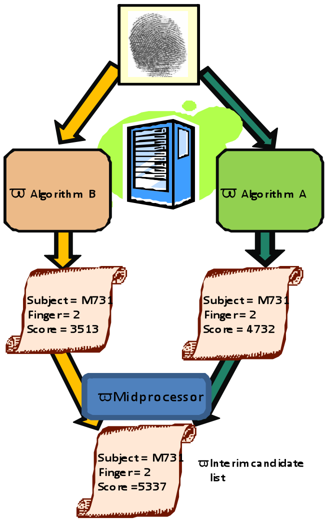


Figure 2: Fusion Architecture Employing Mid-processor

In Figure 2 a Mid-processor is used to combine the output of two matchers, and this is done immediately after each algorithm compares the search with a given file print. A generalization (not shown) combines both concepts: the preprocessor assists by determining the optimal weights used in merging the two outputs.

Figure 3 illustrates the use of a postprocessor whose function is to merge candidate lists produced by the individual matchers. While similar to Figure 2, there is one very important difference: In Figure 3 the "interim candidate lists" need not (and generally do not) contain all the file subjects looked at; only the highest scoring matches are placed on this list. The fusion algorithm must therefore handle not only cases where the same candidate appears on both lists, but also candidates appearing on one list only.

Figure 3 represents the type of system which we analyze in this paper. This choice was dictated mainly by the nature of the available data. By retaining only the highest ranked candidates a type of data compression has been performed, so information has been lost. This suggests the performance of the system of Figure 3 will often be lower than that of Figure2. However, for long candidate lists this effect can be expected to be small.

3. Weighting of Matchers

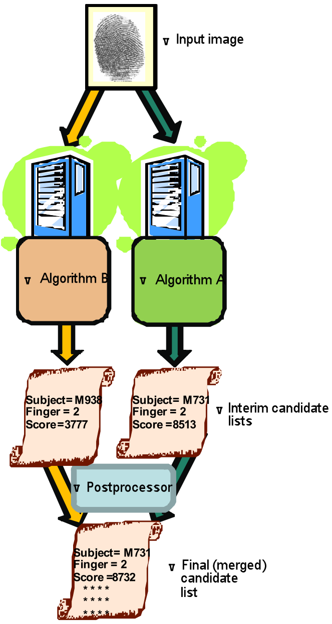


Figure 3: Fusion Architecture Employing Postprocessor

If poorer quality data are mixed with higher quality the conclusions are often degraded. Such an effect is observed with matchers. For best results it might be necessary to weigh the output of a matcher based upon its intrinsic merit.

A rigorous analysis is complicated, but insight can be gained from the following. Suppose two independent measurements/estimates, m_1 and m_2, are to be merged into a single estimate, m_c. A linear estimate for m_c takes the form

$$m_c = w_1 * m_1 + w_2 * m_2 \qquad \dots (1)$$

where w_1 and w_2 are weights having the properties, $w_1, w_2 > 0$, and $w_1 + w_2 = 1$.

It is known that for the minimal variance estimator the optimal weights are given by

$$w_1 = \sigma_2^2/(\sigma_1^2 + \sigma_2^2) \text{ and } \quad w_2 = \sigma_1^2/(\sigma_1^2 + \sigma_2^2) \dots (2)$$

where σ_1^2 and σ_2^2 are the variances of the two measurements. The variance of m_c is then given by $\sigma_1^2\sigma_2^2/(\sigma_1^2 + \sigma_2^2)$. This expression is always less than or equal to $\min\{\sigma_1^2 , \sigma_2^2\}$, showing that the combined estimate is usually better, and never worse, than the better of the two original estimates.

However, this is only true if the weights, w_1, w_2, are correctly chosen. If we have no estimates of the errors (i.e., sigmas), then it is best to use equal weights. But this can lead to a degraded estimate. This will occur if $\sigma_2/\sigma_1 >$

√3 (assuming “2” is the worse of the pair). We will return to this in section 8.3.

4. Prior Work

Sensor/matcher fusion is an active area of research, as shown by the numerous recent references, some of which are listed in Section 10. Many published papers have a very optimistic outlook on what gains are achievable. The paper by Daugman [4] is notable in this respect, and we found its cautious tone to be justified.

The papers by Vatsa, et al. [5]; Godil, et al. [6]; Scheirer, et al. [7]; Hong, et al. [8]; Ross and Jain, [9, 10]; and Grother and Phillips [11] provide a good cross-section of application-oriented papers.

High-level theoretical results are discussed by Ross et al. [12]; A. Jain [13]; K. Nandakumar, et al. [14]; Hube [15]; Baker and Maurer [16]; Dass et al. [17]; Thomopoulos et al. [18]. The paper by O. Melnik, et al. [19] is of special interest in that it attempts a kind of grand synthesis of several approaches.

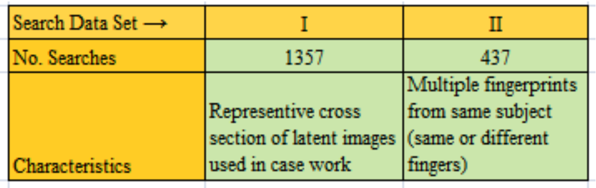
Recently there has been keen interest in the application of extreme value theory to score renormalization. References [20] through [24] provide some classical foundation material. Applications to the fusion problem are found in Scheirer, et al. [7].

5. Description of the Component Matchers and the Search Data

The data for this study were generated by five matchers submitted to NIST for use in ELFT/EFS testing. They are designated here, and in ref. [2], by A—E. It is not possible to go into the details of the matchers (partly for proprietary reasons, and partly for lack of space). However, much information concerning the test procedure and matcher performance will be found in [2]. In general, it is fair to state that Matcher A was the strongest, and the other matchers tended to have decreasing performance in alphabetical order, except that Matcher D (and not E) tended to be the weakest.

Matcher characteristics important to this study are: a) each matcher was able to execute a search using several types of input (search) data; b) each search looks at the entire database and nominally produces a candidate list of the one hundred highest ranking candidates; c) no matcher ever outputs more than one hundred candidates, but truncated candidate lists were not uncommon; d) in rare cases, some matchers produced no candidate list at all; e) candidates on the list were ranked by matcher score, highest score in top (first) position; f) the numerical range of matcher scores, and what constituted a “large score,” was not specified; g) consequently, the range of matcher scores varied over several orders of magnitude – a large score for one matcher might be 0.9, while for another it would be 100,000; h) in addition to the native score, the candidate list included an estimate of “probability this candidate is a true mate” in the form of a number between 0 and 100; finally, i) the manner of calculating “probability of true mate” was not prescribed, though it was suggested the calculations include information beyond raw matcher score.

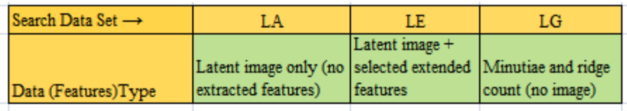
To avoid biasing the results toward the optimistic side, it was decided to retain and score all cases in which there was at least one meaningful candidate list. The number of such “degraded mode” cases was not large, on the order of 2 – 3%, but had a strong influence on the design of the algorithm. The following table summarizes the characteristics of the two principal search sets.



Search Data Set →	I	II
No. Searches	1357	437
Characteristics	Representive cross section of latent images used in case work	Multiple fingerprints from same subject (same or different fingers)

Table 1 – Search Data Sets

The next table summarizes the three types of input data (features) used in this study.



Search Data Set →	LA	LE	LG
Data (Features)Type	Latent image only (no extracted features)	Latent image + selected extended features	Minutiae and ridge count (no image)

Table 2 – Types of input data

The search space (database = foreground + background) was 100,000 subjects, resulting in one million fingerprints. During every search the input was compared with each of the one million database fingerprints. The one hundred largest scores were then placed on the output candidate list.

6. Summary of Fusion Algorithms Investigated

We looked at three types of fusion: 1) two different matchers, each using the same input data; 2) the same matcher for both, but using different input data (but the data are derived from the same latent image); and 3) two matchers employing two different images (or features derived from these images). In the last case, the two images belong to the same subject, but are not necessarily from the same finger. The rules for combining data are then somewhat different than for the first two.

6.1 Candidate List Reduction

The fusion was done in two steps. First a greatly reduced candidate list was generated from the two candidate lists. There were two reasons for this approach. The first was our desire to further explore "candidate list reduction" (see [1]). The second was to establish the average number of imposters that candidate lists tend to have in common. (By a "common imposter" we mean a candidate which is not the true mate but appears on both candidate lists. This number turns out to be surprisingly small, in the range of 1 – 3.)

We were also interested in the extent to which these common impostors would be high-scoring, and if so, what characteristics might account for this. The "goodness" of this intermediate candidate list was measured by comparing it to the candidate list of equal length from the better of the two matchers.

The rules for forming the intermediate (reduced) candidate list were: a) always included the two first-position candidates (one from each list); b) selectively include those in second positions; and c) include all common candidates found on the two lists. This sometimes results in duplications, which were eliminated.

6.2 Rescoring, FOMs

Latent matchers are typically graded by their ability to place the correct mate being in first place. This requires a re-ranking algorithm be applied to the intermediate candidate list. Four types were looked at. The first is rank-based, and similar to the familiar Borda count [25]. The second is probability based, and uses the "probability of true mate" found on the candidate list, and mentioned in Section 4, items (h) and (i).

The third and fourth were score-based. Combining scores from different matchers requires normalization, as scores produced by the individual matchers vary widely in magnitude. We considered two such normalizations, one based on the global mean (mean over all subjects on all candidate lists) and one a local mean (mean only for that search). (It is possible to avoid having to normalize by using the product instead of the sum. However, we did not consider this method because we wanted to consider unequal weighting of the two matchers.)

Much research has gone into the question whether the product rule or the sum rule is best for combining scores. Some of the early work tended to favor the product rule. But it appears that with proper normalization the sum rules is superior (ref. [7-9]). More recently extreme value theory has been receiving considerable attention (ref. [16-19]). This theory is excellent for identifying unusually high scores, and therefore potential hits. It is also good for score normalization, but it is not clear that by itself it can handle large differences in performance of two systems. Just because a system produces very large scores does not guarantee a hit – and we frequently encounter this.

To differentiate the fused score from the native matcher scores, we refer to this as figures-of-merit, or FOM. We considered four types of FOMs, and give representative equations for these:

1) Rank-based:

$$FOM_1 = ((101 - rank_a) + (101 - rank_b))/2 \quad \dots (3)$$

2) Probability-based:

$$FOM_2 = P_a + P_b - P_a{*}P_b/100 \qquad \dots \ (4)$$

(The division by 100 in the above is clarified by referring back to Section 4, item (h).)

3) Native-score-based

$$FOM_3 = 50{*}Score_a/Norm_a + 50{*}Score_b/Norm_b \quad \dots (5)$$

By "native score" we mean the score as it appears on the matcher candidate list. A candidate appearing on only a single candidate list is assigned zero for its second score. It will be noted that multiplicative factors have been inserted so as to make FOM generally lie between 0 and 100. This is of help to the analyst when perusing candidate lists. (Unusual values could indicate errors.)

"Norm" denotes the average score of all candidates (the majority of which are non-mates). We consider two types of norms. For the "global" norm we take an average over all candidates on all lists; while for a "local" norm we consider only the average over the current list.

6.3 Unequal Weighting

In section 3 we pointed out that when lower quality data are added to higher quality it is necessary to compensate for this imbalance. Accordingly, when fusing two matchers of unequal strength we modify (5) to read

$$FOM = 50{*}Score_a/Norm_a + 50{*}\mathbf{d_r}{*}Score_b/Norm_b \quad \dots (6)$$

where d_r is a "de-rating" factor to account for the fact that matcher "b" is the weaker of the two. Figure 7 shows the effect d_r has on performance.

7 Numerical Results

This section summarizes salient results obtained. We begin with the reduced candidate list.

7.1 Reduced Candidate Lists

Figure 4 shows typical candidate list lengths obtained. The letter pairs under each bar show the two matchers being fused. The average length (over all matcher pairs) is about 2.0. Since 0.7 of these are the true mate, the mean number of impostors is about 1.3 – a surprisingly low value, considering the original candidate list is 100 long. Note also that some matcher-pairs produce more candidates than others, suggesting these matchers are somehow more correlated; such an example is the triplet (B, C, E).

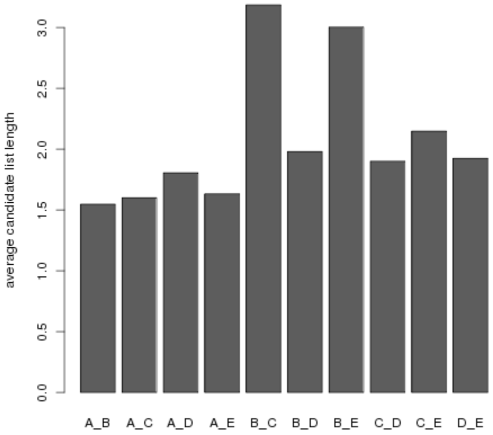


Figure 4: Average Reduced Candidate List Length

The next figure shows the performance gain at the candidate list level. (Recall these are computed by comparing performance to an equal length candidate list from the better of two matchers.)

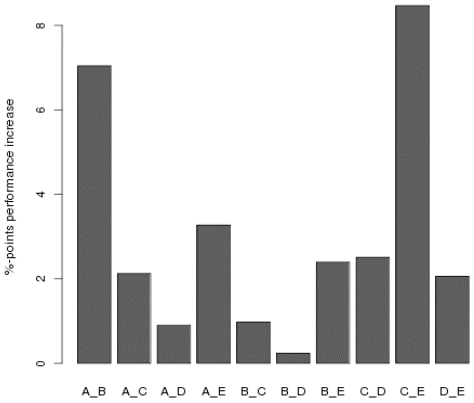


Figure 5: Average Performance Gain

The graph shows that performance gains can vary widely. Some combination such as A_B and C_E produce significant gains, while others such as A_D and B_D produce virtually no gain. The next figure shows the net performance of each pair on the candidate-list level.

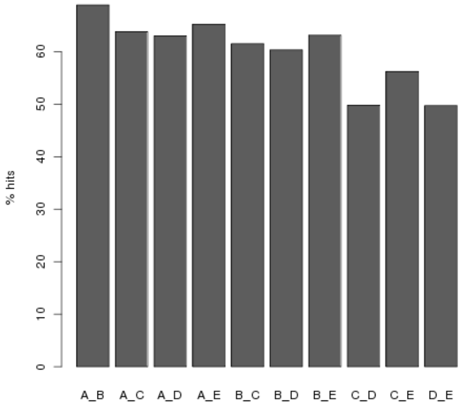
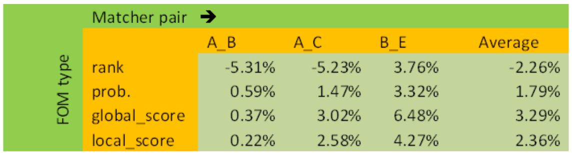


Figure 6: Average Performance

It will be seen that the combination A_B produced the best result – perhaps not surprising, as these are the two best matchers.

7.2 Top position

To investigate the effect of different FOMs, we selected a “bellwether” subset of cases which were challenging. The following table summarizes the numerical results.



	Matcher pair ➔				
FOM type		A_B	A_C	B_E	Average
	rank	-5.31%	-5.23%	3.76%	-2.26%
	prob.	0.59%	1.47%	3.32%	1.79%
	global_score	0.37%	3.02%	6.48%	3.29%
	local_score	0.22%	2.58%	4.27%	2.36%

Table 3 – Ranking FOM Results

Some of these results were surprising. The rank-based approach tended to produce poor results, probably because of coarse resolution provided by the ranking. The probability-based approach, while producing reasonable results, was somewhat disappointing. Possibly, this was because the matchers tended to assign zeros to lower ranked candidates. A zero value cannot help to elevate a candidate.

It was originally thought that local normalization would be superior to global, because it uses information more specific to a particular search. However, this did turn out to be the case.

The value for B_E of 6.5% appears to be a good representative of the types of gains achievable with fusion using a single finger; though in a few instances higher values, around 8%, were observed.

7.3 Unequal Weighting

We evaluated the result of unequal weighting of the two component matcher, along the lines presented in Section 6.3. Figure 7 summarizes the result. (Lighter line is experimental data; darker is curve fit.)

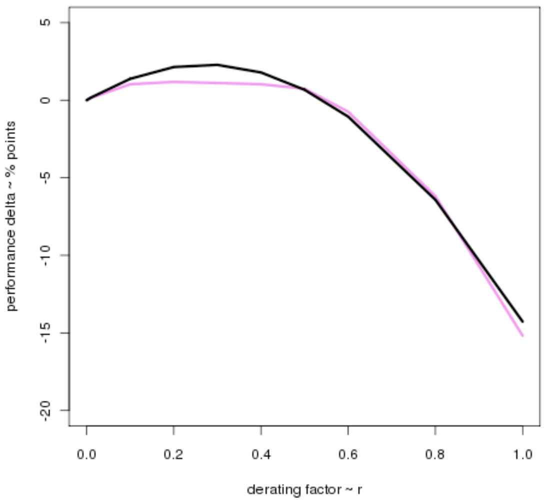


Figure 7: Performance vs. De-rating Factor

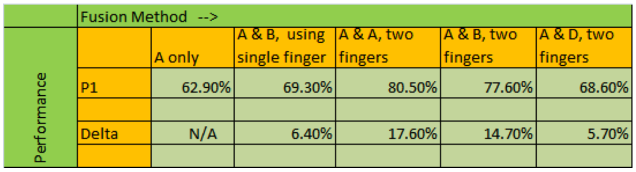
It will be noted that when $d_r = 1$ the weaker matcher “drags down” the result to a significant degree. On the other hand, when $d_r = 0$ the influence of the second matcher is nullified, so that we are essentially dealing with matcher “a” only. An optimum point is reached around 0.27, at which point the performance delta is estimated to be +2% points. To test whether this value is “robust” we tried substituting other matchers for the stronger matcher. In all cases performance was improved over $d_r = 1$. In four out of five cases the result was a positive performance delta, while in the fifth case it was just slightly negative. (Of course for each of these cases the true optimal d_r would be somewhat different; but the object was to test how sensitive the fused result is to d_r.)

7.4 Multi-finger Fusion

In previous sections the fusion was based upon one of the following modes: a) fusing the output of two different matchers employing the same input data; b) the same basic matcher is used for both, but it operates on two different datasets; or c) both matchers and data are different. The important point is that in all cases the search fingerprint image is identical for both matchers.

In this section we consider the case when the data (e.g., fingerprints) are obtained from two different fingerprint images, but from the same subject. These fingerprints might originate from the same finger, or from different fingers (more common). In either case it is assumed the matcher does not “know” whether they are from the same finger. This requires important changes in the fusion logic. Whereas in the previous cases scores were only combined if both the subject and the finger position were the same, in the present case only the subject need be the same.

Multi-finger fusion has not received as much attention as other cases, so that Ref. [21] is of special interest. The following table summarizes results we obtained:



	Fusion Method -->					
		A only	A & B, using single finger	A & A, two fingers	A & B, two fingers	A & D, two fingers
Performance	P1	62.90%	69.30%	80.50%	77.60%	68.60%
	Delta	N/A	6.40%	17.60%	14.70%	5.70%

Table 4 – Multi-finger Results

It will be seen achievable gains are significantly greater than for other cases. For a single finger gains beyond 6-8% were rare. For two-fingers gains of over 15% can be achieved. This is probably due to lower correlation.

It is interesting to see how far down in ranking candidates can be reclaimed via fusion. The most outstanding such case noted was when a candidate went from 98th place to second place. However, in this case the other mate of the pair was already a solid first place. In another case one of the candidates was ranked third and the other 21st. Following fusion they became first and second.

The most obvious approach is to use the best matcher and the best data type for both fingers. In our case this would be matcher A using dataset LE. It turns out this indeed produced the best overall results. For thoroughness we investigated other combinations. However, as the above table shows, using matcher B (the second best) for the second print resulted in a slight decrease in performance. Also interesting to note is that even when the second matcher is the weakest (D) improved performance was obtained.

8 Conclusions

Relative simple and robust algorithms can be used to fuse two candidate lists. The first step is to merge and reduce the candidate list down to a small size. Typically two to six candidates are sufficient. In unusual cases, where the matchers appear to be highly correlated, longer lists are produced.

The second step is to reorder the list based on all available information so that the best candidate will appear at the top. Performance increases of up to 6-9 percentage-points were observed when fusing two matchers operating on a single finger. When using two fingers from the same subject significantly larger gains, in the range of 15 percentage-points are obtained.

It is of interest to estimate how much information is added by the various fusion modes. A "back of the envelope" calculation suggests that when using two different copies of the same finger, information is increased by about 41%. When using two different matchers on the same information, the increase is only 17% -- or less than half of the first mode.

In interpreting these results, it must be kept in mind that matcher accuracy depends heavily upon the quality of the file-side (background/foreground) feature extraction – and the software performing this is not normally considered part of the latent matcher per se. Improvements in file-side extraction can, however, result in improved search accuracy, but it is unclear what effect this would have upon ten-print matching.

Finally, the fusion of matchers might result in more close non-mates (impostors) appearing on the final candidate list, possibly resulting in more Type I errors (false identifications).

9 References

[1] M. Indovina, A. Hicklin, G. I. Kiebuzinski, ELFT-EFS Evaluation of Latent Fingerprint Technologies: Extended Feature Sets [Evaluation #1], NISTIR 7775**,** March 2011, ➔ http://biometrics.nist.gov/cs_links/latent/elft-efs/NISTIR_7775.pdf

[2] Concept of Operations (CONOPS) for Evaluation of Latent Fingerprint Technologies (ELFT) (Rev. D, 21 June 2007), NIST online publication ➔ http://biometrics.nist.gov/cs_links/latent/elft07/elft07_concept.pdf

[3] S. Meagher, V. Dvornychenko, Defining AFIS Latent Print "Lights-Out", NIST IR, 2011

[4] J. Daugman. Combining Multiple Biometrics. ➔ http://www.cl.cam.ac.uk/users/jdg1000/combine/combine.html

[5] M. Vatsa, R. Singh, A. Noore, K. Morris, Simultaneous Latent Fingerprint Recognition: A preliminary Study, IEEE, 2009

[6] A. Godil, S. Ressler and P. Grother. Face Recognition using 3D Facial Shape and Color Map Information: Comparison and Combination.

[7] Walter Scheirer, Anderson Rocha, Ross Micheals, and Terrance Boult, Robust Fusion: Extreme Value Theory for Recognition Score Normalization, 2010

[8] L. Hong, Y. Wan, S. Pankati, Can Multibiometrics Improve Performance? (1999)

[9] A. Ross and A.K. Jain. Information Fusion in Biometrics. Proc. of AVBPA, Halmstad. Sweden. pp. 354-359, June 2001.

[10] A. Ross, A. K. Jain. Information Fusion in Biometrics (2003)

[11] P. Grother, P. J. Phillips. Models of Large Population Recognition Performance (2004)

[12] A. A. Ross, K. Nandakumar, and A. K. Jain. ***Handbook of Multibiometrics*** (2006)

[13] A. Jain, K. Nandakumar, and A. Ross. Score normalization in multimodal biometric systems," *PatternRecognition* 38, pp. 2270-2285, December 2005

[14] K. Nandakumar, Y Chen, S. Dass, and A. Jain. Likelihood ratio based biometric score fusion, IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(2):342-347, 2008.

[15] J. P. Hube, Neyman-Pearson Biometric Score Fusion as an Extension of the Sum Rule

[16] J. P. Baker, D. E. Maurer. Fusion of Biometric Data with Quality Estimates via a Bayesian Belief Network, Biometric Consortium Symposium (2005)

[17] S. Dass, K. Nandakumar, and A. Jain. "A principled approach to score level fusion in multimodal biometric systems," Proc. 5th Int'l Conf. on Audio- and Video-Based Biometric Person Authentication (AVBPA), pp. 1049-1058, (2005)

[18] S. Thomopoulos, R. Viswanathan, and D. Bougoulias. "Optimal decision fusion in multiple sensor systems," *IEEE Transactions on Aerospace and Electronic Systems,* AES-23, pp. 644-653,September 1987

[19] O. Melnik, Y. Verdi, C. H. Zhang; 2003. Mixed Group Ranks: Preference and Confidence in Classifier Combination

[20] R.A. Fisher and L. H. C. Tippett (1928). Limiting forms of the frequency distribution of the largest and smallest member of a sample, Proc. Cambridge Phil. Soc., **24**, 180–190.

[21] E. J. Gumbel, (1958), *Statistics of Extremes*, Columbia University Press, ISBN 0-483-43604-7, Pickands, J. (1975). *Statistical inference using extreme order statistics*, Annals of Statistics, **3**, 119–131.

[22] E. Castillo, 1988. *Extreme value theory in engineering.* Academic Press, Inc. New York.

[23] S. Coles. An Introduction to Statistical Modeling of Extreme Values (2001)

[24] S. Kotz and S. Nadarajah**.** Extreme Value Distributions: Theory and Applications (2001**)**

[25] Reference for Borda count ➔ http://www.ctl.ua.edu/math103/voting/borda.htm