Latent Matcher Fusion
-- Lessons Learned

IAI
97th International Conference
Phoenix, Arizona

Vladimir Dvornychenko
NIST

July 2012

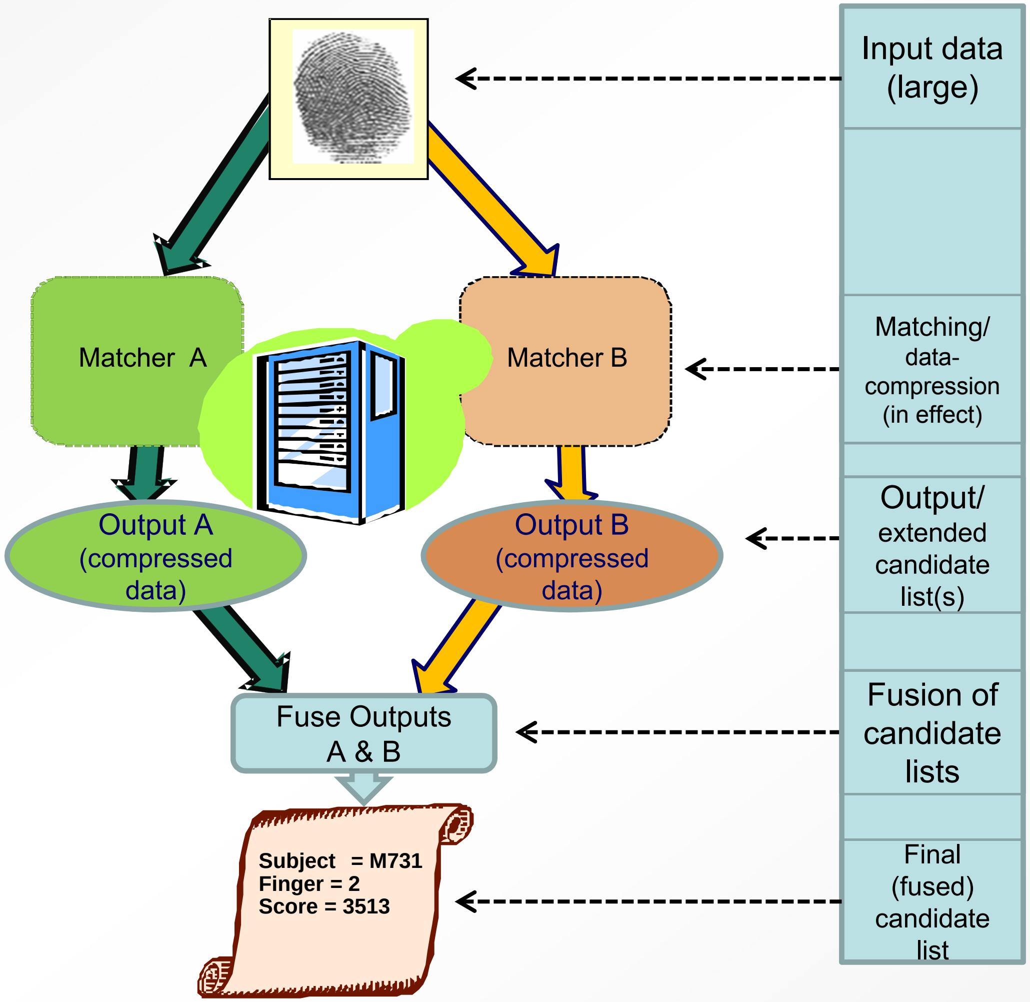
Why Look at Matcher Fusion?

- Matcher fusion provides a simple method for improving matcher performance
- What can be done via matcher fusion can, in principle, be done via a single “monolithic” matcher – but at potentially great increase in code complexity
- Fusion therefore provides us a means of assessing how much performance “headroom”/available-performance-margin exits with current technology
- This “available headroom/margin” is of interest to NIST in connection with its ELFT project

Principles of Matcher Fusion

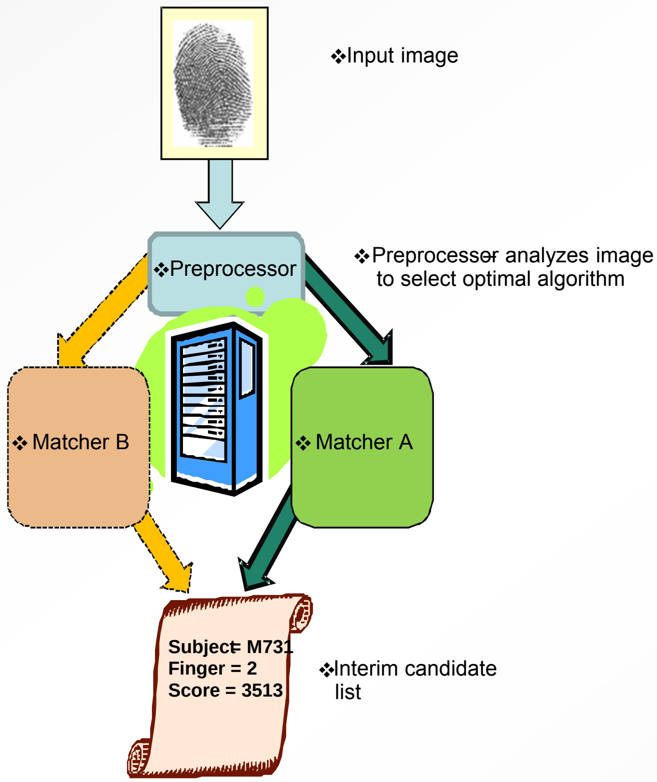
- From an information theoretic viewpoint, matcher fusion consists of two steps:
  - a) data compression – done by the original matchers
  - b) followed by data combining/fusion – performed by a separate algorithm
- The result of fusion is a “virtual matcher” – which for all purposes is indistinguishable from an “actual matcher”
- Because in the “data compression step” some information is theoretically lost, and the resulting matcher might not have the highest possible performance
- In practice, because of the complexity involved, the fused matcher might outperform the “monolithic” (integrated) matcher

Simplified Diagram of Fusion Process



Note: Inputs to the Matchers (A and B) need not be the same; for example: Matcher A could use latent image, while Matcher B uses human-extracted features.

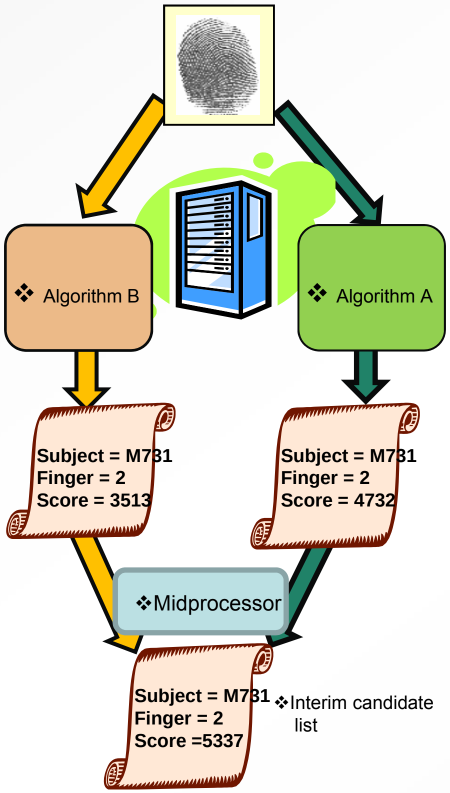
A number of architectures are available for implementing fusion – first architecture



This architecture uses a preprocessor to select the more/most appropriate matcher/algorithm.

In the form shown in diagram, either A or B would be selected, based on the analysis of the input image.
In a slight generalization, both A and B are used, then their outputs are fused using optimal weights, which were computed by the preprocessor.

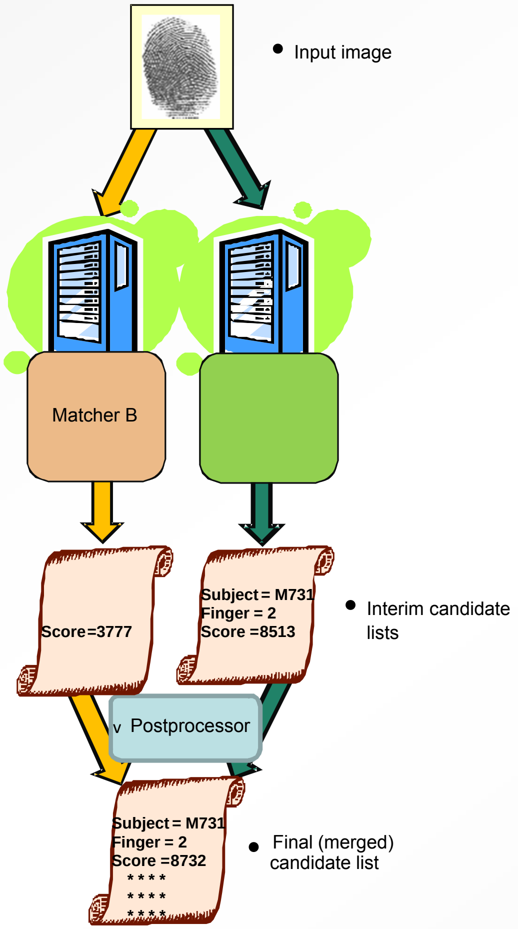
A number of architectures are available for implementing fusion – second architecture



This architecture employs a ***Mid-processor*** to fuse the outputs of the matchers immediately after each matcher has finished its comparison with the current file/known print.

Both matchers, A and B, output a summary of the comparison results. The output includes, as a minimum, the matching score of each matcher, but may include additional data (e.g., how many minutiae were actually mated, what was area of overlap, etc.) Since there is no preprocessor, the weights for fusion must be predefined (e.g., equal).
The “interim” candidate list is continually updated so as to retain only the highest running scores. Upon completion of the search the “interim” candidate list becomes The “final” output candidate list. .

A number of architectures are available for implementing fusion – third architecture



This architecture employs a ***Postprocessor*** to fuse the outputs of the component matchers (A and B) upon completion of the entire search by both.

Each (component) matcher outputs a candidate list of pre-specified length (e.g., 100, 500, 1000). These two “interim” candidate lists are then fused to create the final output list.

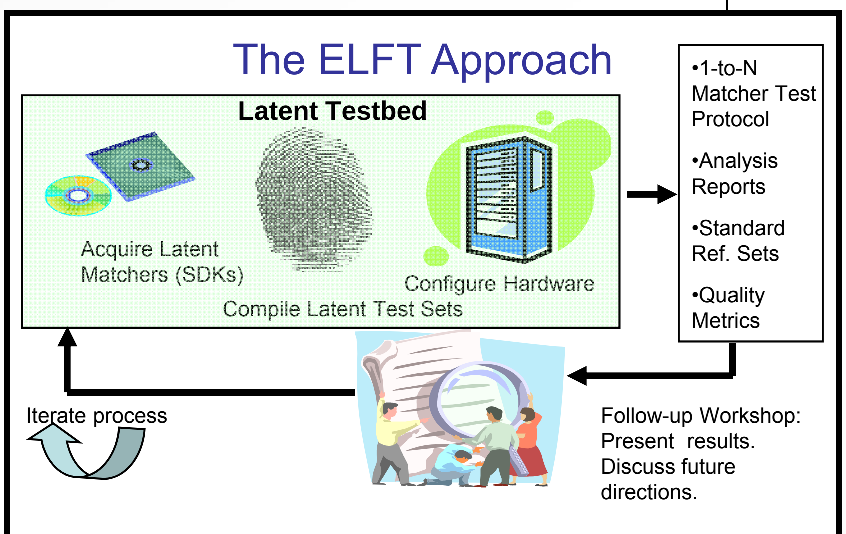
Architecture Employed in this Study

- The third architecture (previous slide) was the one actually used in this study
- This choice was largely dictated by the nature of the available input data

Matchers

- The candidate lists of five different matchers were used in this study
- These are identified as Matchers A through E (The companies supplying these are identified on the NIST website.)
- These matchers were supplied to NIST for feature/matcher performance evaluation
- Generally speaking, Matcher A was strongest, and D weakest

Evaluation of Latent Fingerprint Technology (ELFT)



Scope Statement:

•**The overall purpose of NIST's ELFT project is to advance the state-of-the-art in latent fingerprint searches via:** a) decreased dependence on human experts thru greater automation ; b) standardization of feature sets to facilitate data interchange; and c) standardized scores and performance measures

•To accomplish this, NIST has planned a series of tests for evaluating the state of the art in automated latent fingerprint matching. These tests will quantify the core algorithmic capability of contemporary matchers, and assist contractors/private-industry in improving their products. Some test have been completed; some are in progress; and some are future plans

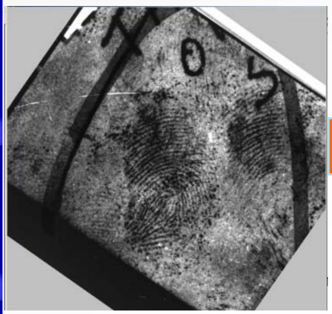
•Testing Phases I and II, and EFS Eval. #1 and #2 have been completed, and the results can be found on the NIST website

• Latent Website Mainpage ➔ http://fingerprint.nist.gov/latent/

How are Candidate Lists Generated?

AFEM = *Automated Feature Extraction and Matching*

Latent Fingerprint Image



Input

Fingerprint Matching System

Automatic Feature Extractor

features

Latent Fingerprint Matcher

Output

Candidate List

Rank	Subject	Finger #	Score	"Probability"
1	0731	2	2903	85
2	1303	7	1805	13
3	3950	1	1754	11
…	…	…	…	…
20	0121	4	350	0

Michael Indovina
IAD - Image Group

Input to Matchers

- The inputs to the matchers came in a number of "flavors":
  - ***Image only*** means the matcher is given only the image of the latent print – and no other information; the matcher must extract its own (native) search features
  - ***Features only*** means the matcher was given search feature extracted by human experts – and no other information (such as the image); as a minimum, these features would include the traditional features such as a) minutiae; b) core & delta; and c) pattern class; however in some cases these were augmented by extended features
  - ***Image plus Features*** means that both a) the latent fingerprint image, and b) the human-extracted features were supplied
  - ***Dual images*** means that two images from the same subject went to the matchers (one to each matcher); these might or might not have come from the same finger; if from the same finger they are different captures

Number of Searches and Background

- The principal search set consisted of 1357 latent fingerprints; these consisted of representative criminal casework, supplemented by special collections
- A second set consisted of 437 cases of multiple captures from subjects; the multiple capture might, or might not, be from the same finger
- The background consisted of 100K subjects = 1M fingers. (Actually this refers to background + foreground, where foreground are the mates of the searches.)

Input to Fusion Algorithm

- The input to the fusion algorithm nominally consisted of two candidate lists, each 100 candidates long
- In exceptional cases, the candidate list might be truncated or entirely missing; the fusion algorithm needed to handle these cases
- When the same input data is used for both matchers the two matchers need be different for meaningful results
- Otherwise, with different inputs, the two matchers being fused can be the same matcher (but operating on different data)

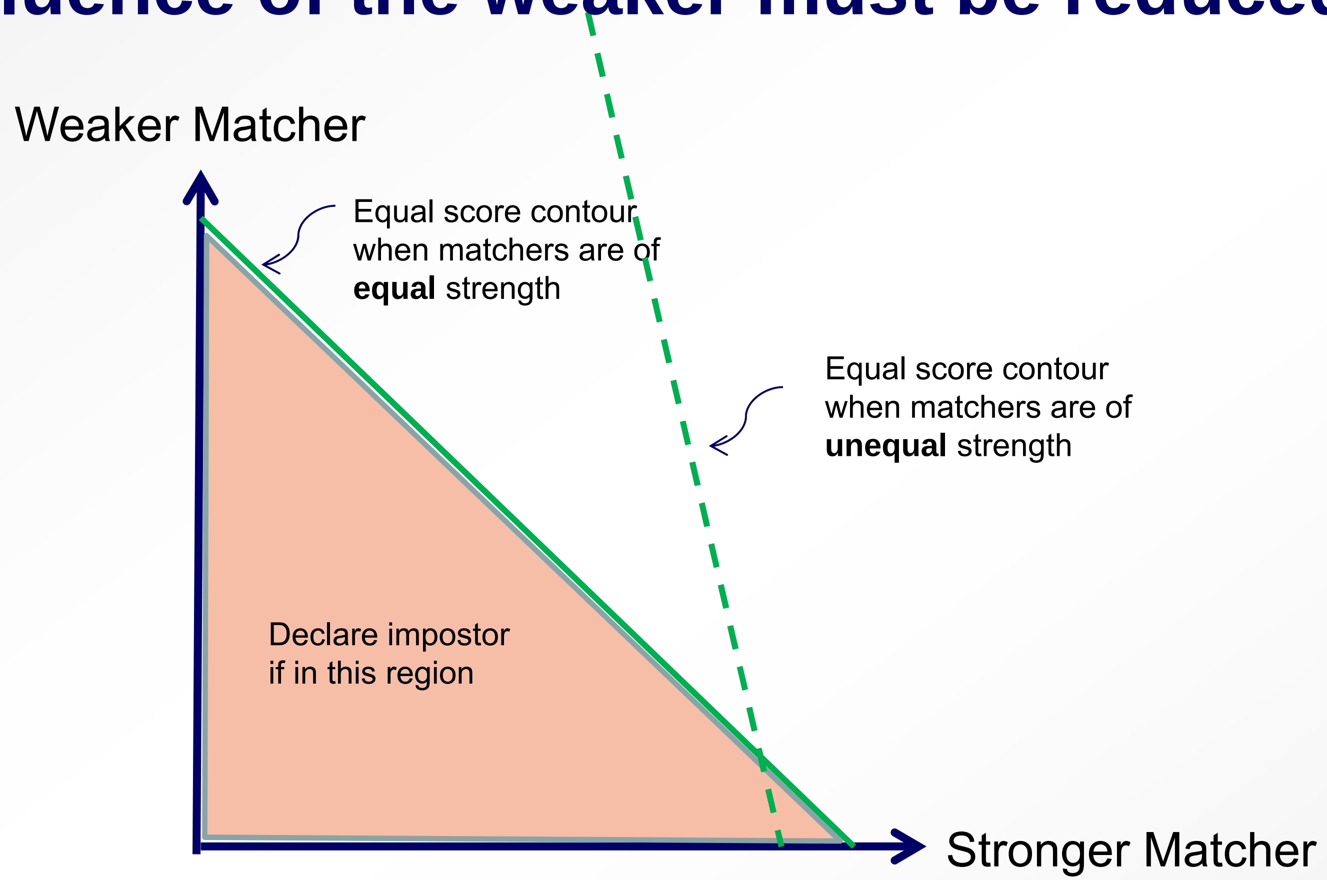
The Fusion was in Two Steps

- Step 1 – a reduced working candidate list was created consisting of:
  - First place candidates from both lists
  - Selectively, second place candidates form both lists
  - Any candidates appearing on both lists (subject and finger number same)
- Candidate list were then checked for duplications; these were eliminated

Step Two of Fusion

- **The second step** consisted of computing a new score; to avoid confusion with the original “native score” we called this fused score a “figure of merit” (FOM)
- Three different types of FOM were used: a) score-based; b)rank-based; and c) “probability”-based
- For subjects appearing on both lists, their FOM was boosted by adding the two FOMs (after suitable scaling)
- “Probability,” in the present context, refers to a special kind of normalized score appearing on candidate lists
- The final step was to reorder the list based on FOM

When matchers are of uneven strength, the influence of the weaker must be reduced

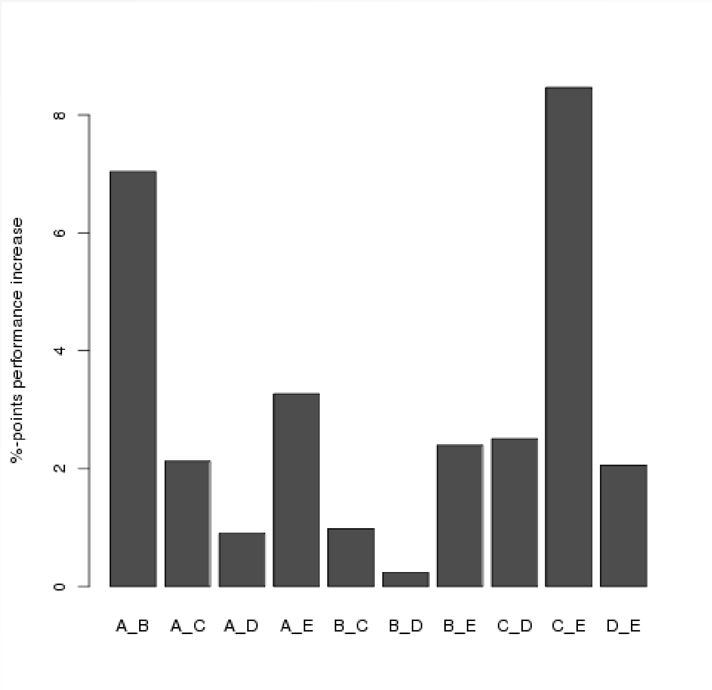


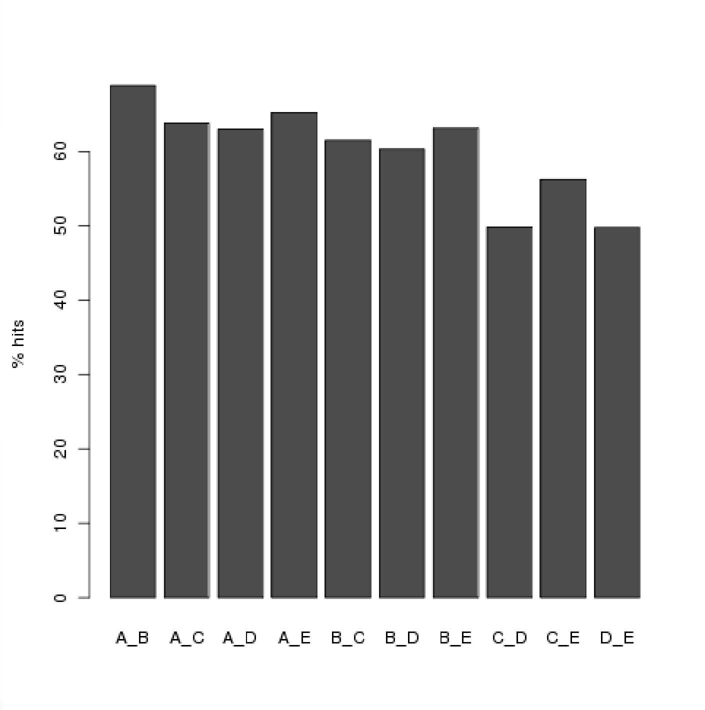
Method of Gauging Performance Gain

- Two different performance gains were computed:
  - a) candidate-list-level gain
  - b) first-position gain
- In each case the gain was based on the performance increase over the better of the two matchers
- For the candidate-list-level gain, we compared the probability that the true mate is on the reduced list with the probability it is on a list of equal length for the better matcher (this might require interpolation)
- A major reason for interest in the reduced candidate list was our “candidate list reduction” goal

Representative Gains from Fusion

-- gains range from over 8%-points to under 1%-point



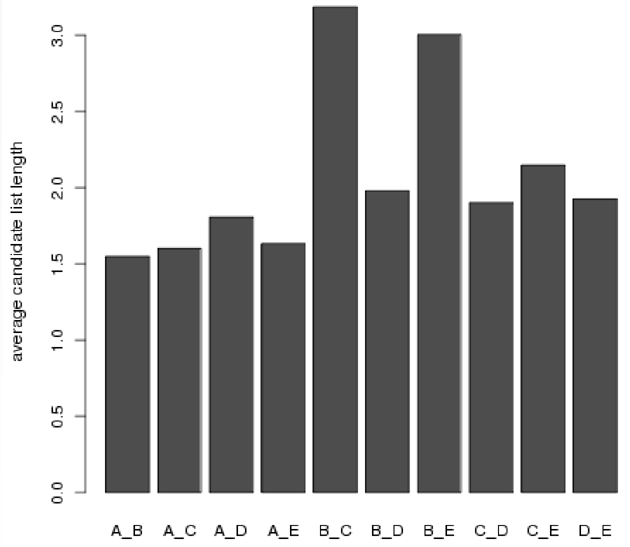


Performance Deltas for pairs of matchers, candidate-list-level, image-only data

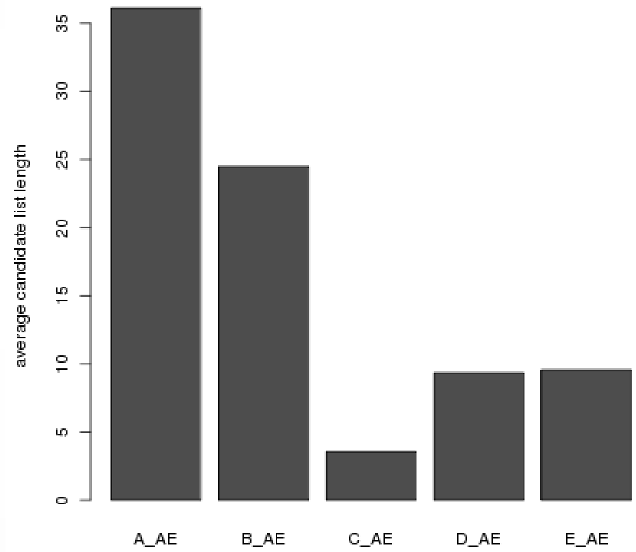
Net performance (fused) for pairs of matchers, candidate-list-level, image-only data
Note that although C-E showed highest gains, it did not have highest performance

Average Candidate List Length

-- In favorable cases the average length can be under three; in unfavorable cases it might be greater than 35



Average Reduced Candidate List Length
-- favorable case



Average Reduced Candidate List Length
-- unfavorable case

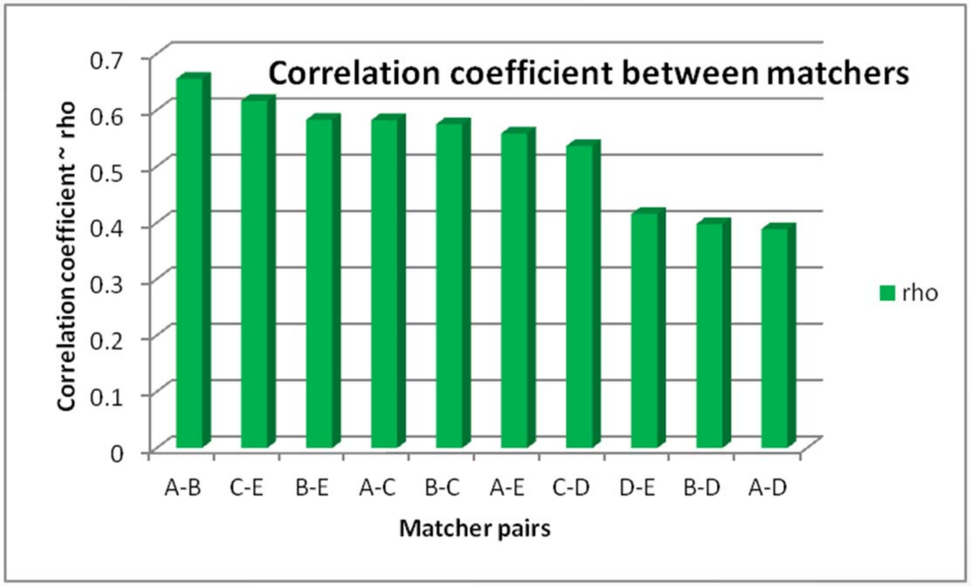
Notes:

1) Left graph shows two different matcher pairs working on image-only data – there a few impostors in common
2) Right graph shows a) matchers A-E working on image-only fused with b) matcher A using LE
– there are many impostors in common for the combination A/LA & A/LE

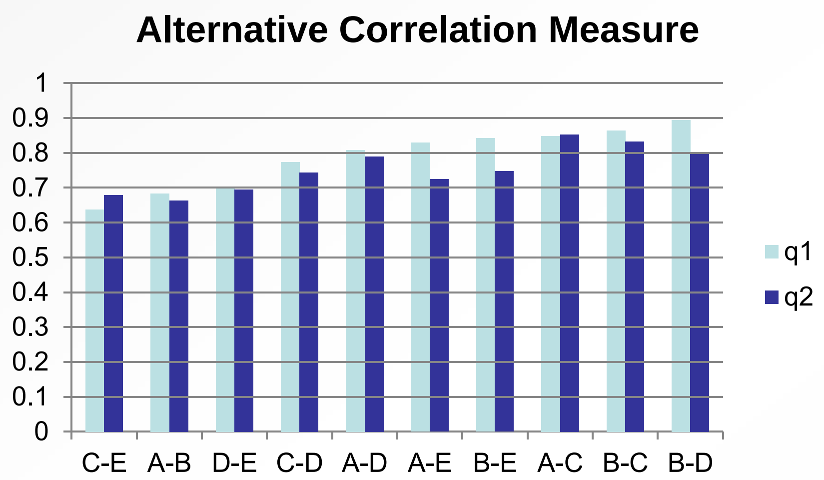
When operating on different input data, the stronger (of the two) matcher should get the better (more information) dataset

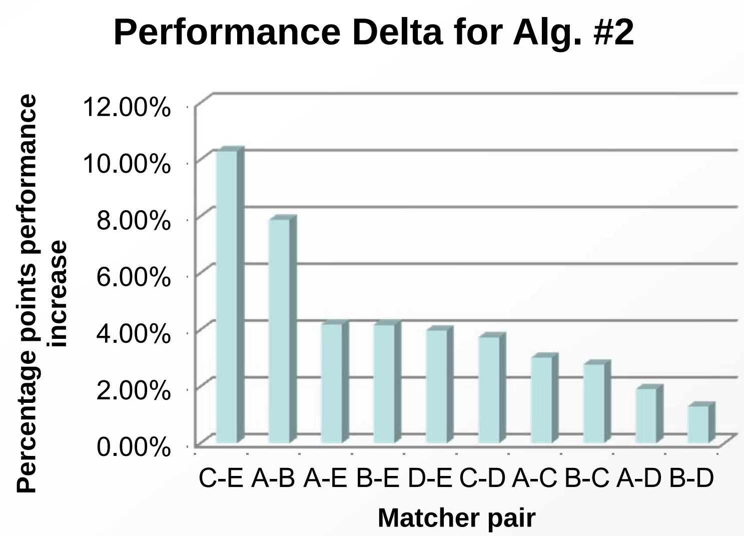
		Operating on Dataset LG				
	Matcher	A	B	C	D	E
Operating on Dataset LE	A	67.9	69	68.8	65.4	66.7
	B	64.8	65	65.1	61.5	63
	C	62.5	64.3	62.6	59.6	61.3
	D	45.5	48.7	48.7	22.3	35.1
	E	59.8	61	60.4	50.6	53.8

Pearson correlation measure was found NOT to be the most useful for predicting fusion gains



An “Alternative” Correlation Measure (q) is Proposed – and it does well!



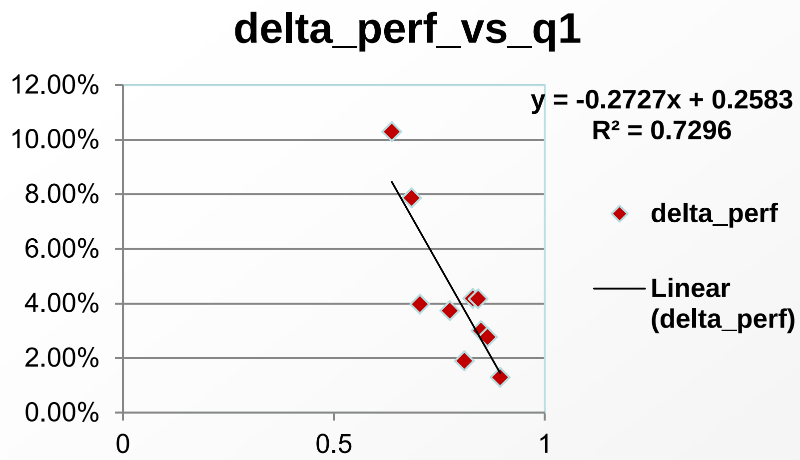


How is this "Alternative" correlation defined?

- Suppose we have two matchers, A & B, and A is the better performing of the two in terms of "hits" in first place
- Consider now creating another (virtual) matcher, call it B*, which is a "dumbed-down" version of A, having the same first-place performance as B
- We create B* by randomly switching between 1) Matcher A, and 2) a "random-guessing" matcher; the frequency of switches is so chosen as to reduce performance to B level.
- Now compute the (Pearson) correlation between A and B* (call it r) -- continued

-- More on “Alternative” Correlation

- Let ρ (rho) denote the Pearson correlation between A & B
- The new -- “alternative” – correlation is defined by ρ/r, and we denote this by q
- “q” does a good job in predicting performance gains, as seen below



Ranking FOMs/algorithms

- Four types of scores – referred to as Figures-of-Merit, or FOM – were looked at
  - 1) Rank based (Borda count)
  - 2) “Probability” based
  - 3) Native score based, global normalization
  - 4) Native score based, local normalization
- Giving both matchers equal weight sometimes produced negative gains; however, for simplicity, we retained this scheme in some cases
- Score-based generally produced largest gains

	Matcher pair -->					
		A-B	A-C	E_B		average
FOM type	rank	-5.31%	-5.23%	3.76%		-2.26%
	prob.	0.59%	1.47%	3.32%		1.79%
	score	0.37%	3.02%	6.48%		3.29%
	rel_score	0.22%	2.58%	4.27%		2.36%

Multi-finger Results

- Up to now we have considered fusion where the input comes from the same latent fingerprint image – whether features are extracted by machine, or by human experts
- We now consider the case where the input consists of two **different images** from the **same subject**; but the images which might be from the same finger or different fingers
- The matrix below shows that performance is more than doubled when two different images are used; this can be attributed to the influx of new information; alternatively, we can say S/N has increased by 41%

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

Conclusions:

- Matcher fusion can produce significant performance gains -- but do not expect "eye-popping" gains
- Gains are on the order of 6-8% points when based on data coming from a single input image
- As an independent check, we considered scoring a hit if *any of the five matchers placed the true mate in first position*; this resulted in 11% points improvement over the single best matcher
- For two different images (from same subject) gains are much higher, around 15%-points
- Candidate lists can be greatly reduced, 2-6 candidates, but still have performance exceeding 20 candidates from single matcher.

References/Links

- **Evaluation of Fusion Methods for Latent Fingerprint Matchers,** Dvornychenko, V. N.; 5th IAPR International Conference on Biometrics, March 2012 ➔ http://www.nist.gov/manuscript-publication-search.cfm?pub_id=910369
- Latent Website Mainpage ➔ http://fingerprint.nist.gov/latent/
- Final Report on Phase II Testing ➔ http://fingerprint.nist.gov/latent/NISTIR_7577_ELFT_PhaseII.pdf
- **NIST Latent Fingerprint Testing Workshop 2009, March 19 & 20, 2009 ➔** http://fingerprint.nist.gov/latent/workshop09/index.html
- **ELFT-EFS Homepage, April 2009 ➔** http://fingerprint.nist.gov/latent/ELFT-EFS/

- **NIST would like to thank its sponsors of this work:**

- DHS/Science and Technology Directorate (S & T)

- FBI/Criminal Justice Information Services (CJIS)

Thank you!

Questions?

Contacts:

Vladimir Dvornychenko
NIST
vdvorny@nist.gov