Center for Unified Biometrics and Sensors
University at Buffalo The State University of New York

A Framework for Efficient Fingerprint Identification using a Minutiae Tree

Praveer Mansukhani

February 22, 2008

- **Problem Statement**

  Developing a real-time scalable minutiae-based indexing system using a tree structure

- **Outline of the Talk**
  - Challenges & Motivation
  - Previous Classification and Indexing approaches
  - Our Method: Tree building and Searching
  - Handling Errors in Binning
  - Performance Analysis using Synthetic Datasets
  - Statistical Study of Minutiae Matching

Motivation : Why index?

- **2 types of Biometric systems :**
  - Verification : 1 – 1 Matching
    - Simple comparison between test and candidate template
  - Identification : 1 – N Matching
    - Test template must be compared versus N candidate templates
    - If 1-1 match takes time *t*, brute force identification takes *N * t*
- ***What if N is very large? N> 1K or even N>1M ?***

Effect of large N on Error Rates

- **When we use a verification (1:1) system for identification :**

$FAR_N = 1 - (1-FAR)^N$

$= N \times FAR$

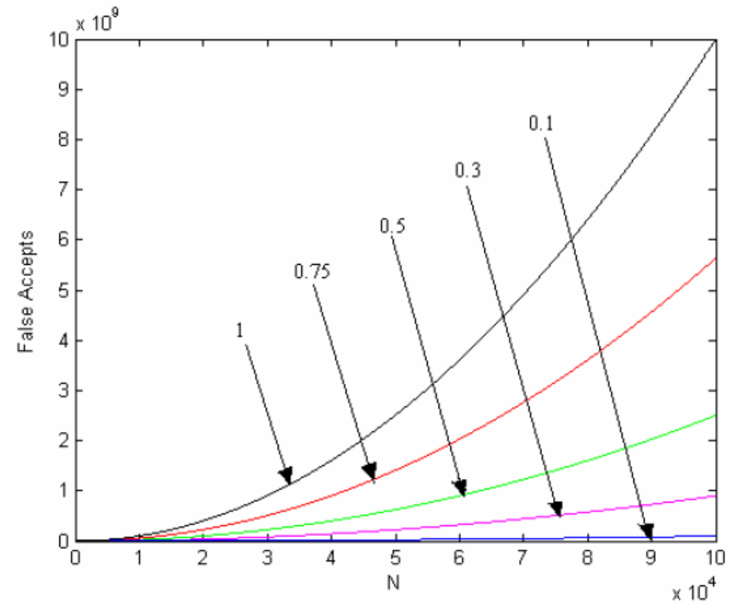
$FRR_N = FRR$

Hence, for larger values of N, FAR_N approaches 1

- Reducing size of search space to P_{SYS} of original …

$$FAR_N = 1-(1-FAR)^{N \times P_{sys}}$$
$$= P_{SYS} \times N \times FAR$$

$$FRR_N = FRR$$



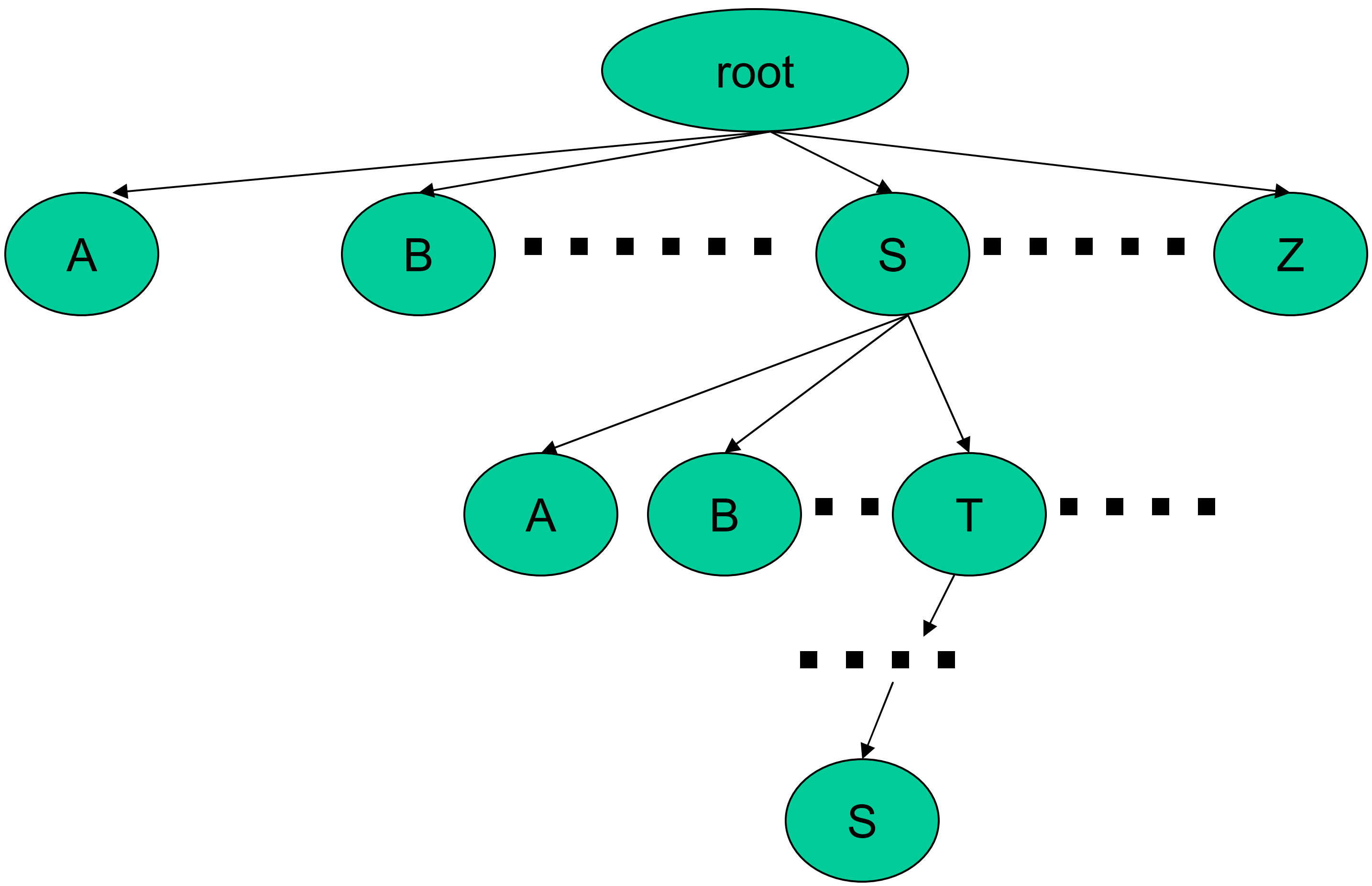
Effect of P_{SYS} on number of false accepts [Mhatre]

- Lesser number of false accepts generated.

- Thus indexing leads to ..
  - Better error rates
  - Less identification time

Basic (Text) Indexing Tree

- Searching **text** dictionary for '*starbucks*'



Simple Indexing for Biometrics?

- Text indexing requires exact match – *'starbucks'* wont match to *'statbucks'*
- Inherent variation present in biometric data
- Test & Reference templates are compared on the basis of similarities in values – exact match is not possible

Hence direct text-style indexing cannot be applied

Challenges

- **Lack of natural ordering of biometric data.**
- **Large datasets (eg FBI fingerprint database has ~47 million users)**
  - Time delays due to a large number of matches
  - Errors caused due to many prints similar to current test fingerprint
- **Different features used for recognition.**
- **Variation in calculated feature values (eg Two fingerprint images might have different orientation, and shear forces on skin leading to inexact images.)**

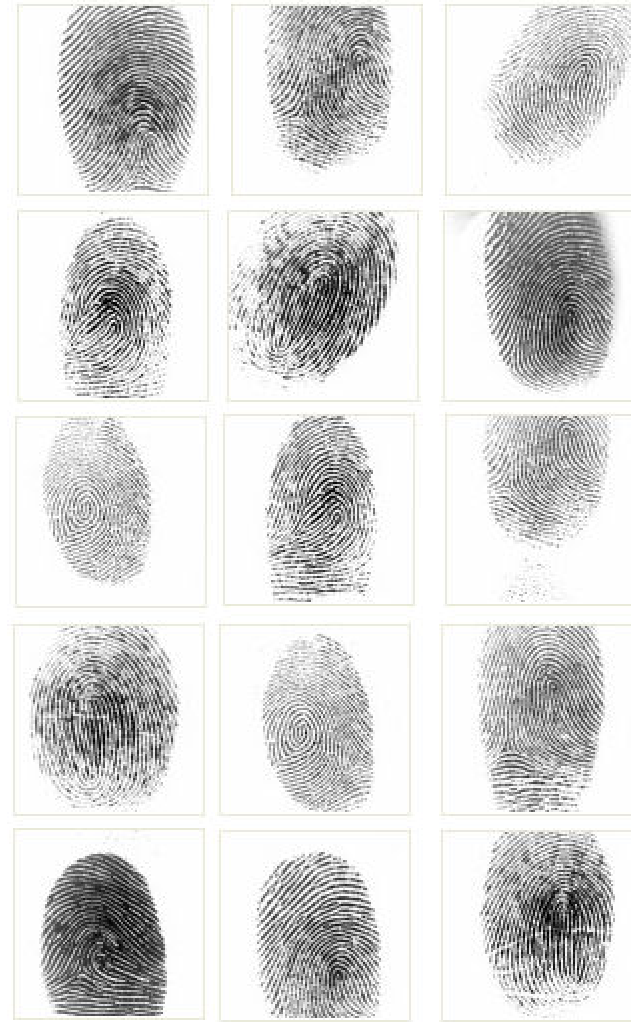


Fig: Typical Fingerprint Images

[Source: FVC 2002 Database #1]

Outline of Talk

- **Fingerprint Identification using a Minutiae Tree**
  - Challenges & Motivation
  - Previous Classification and Indexing approaches
  - Our Method: Tree building and Searching
  - Handling Errors in Binning
  - Performance Analysis using Synthetic Datasets
  - Statistical Study of Minutiae Matching

Fingerprint Classification : Reducing Search Space

- Earliest technique to reduce the search space was by dividing fingerprints into classes, depending on the basic pattern of the ridges.

- 6 fingerprint classes, at times reduced to 4 or 5 .

- Automatic classifiers reduce the search space. For greater accuracy 2 most probable classes may be searched.

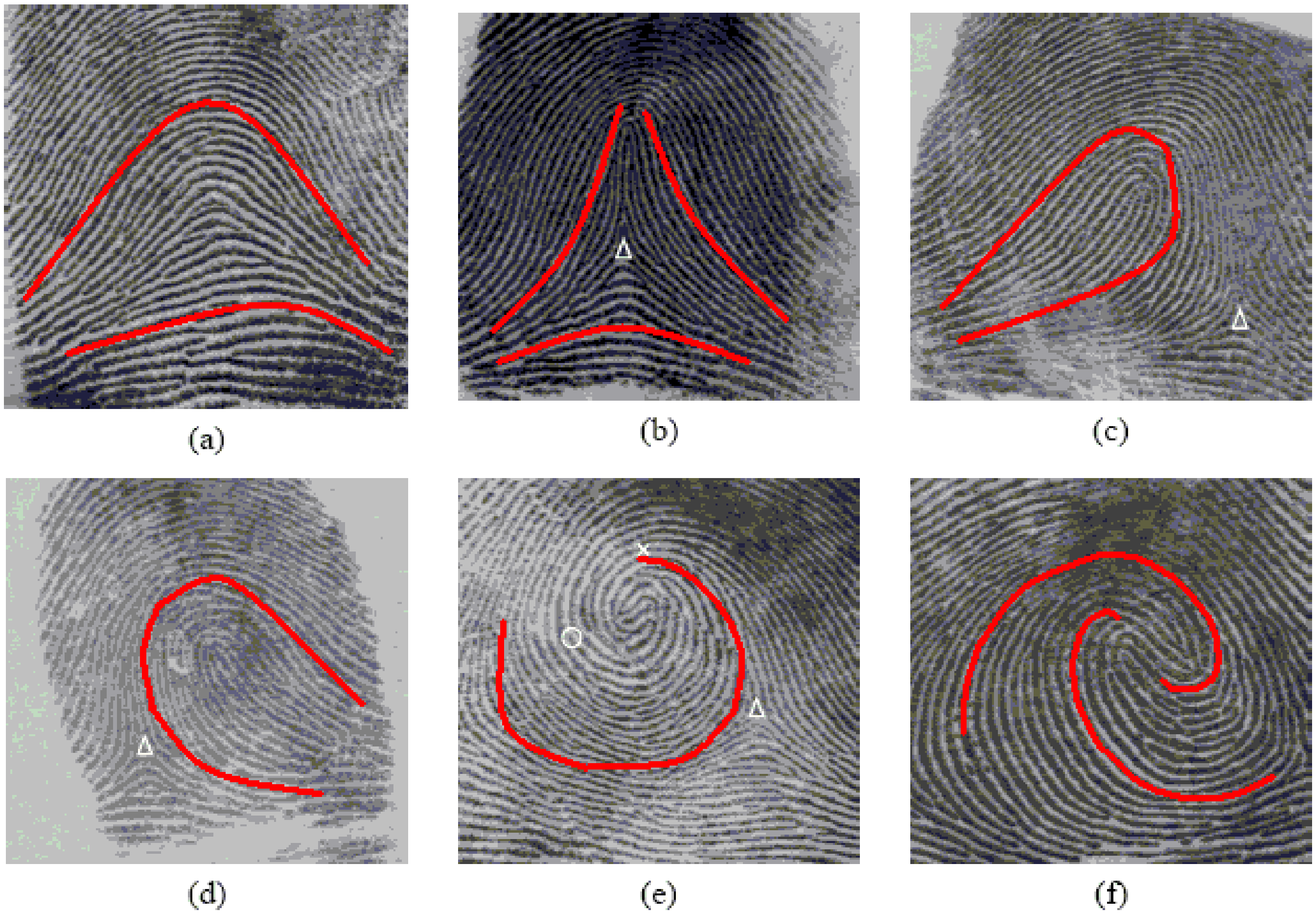
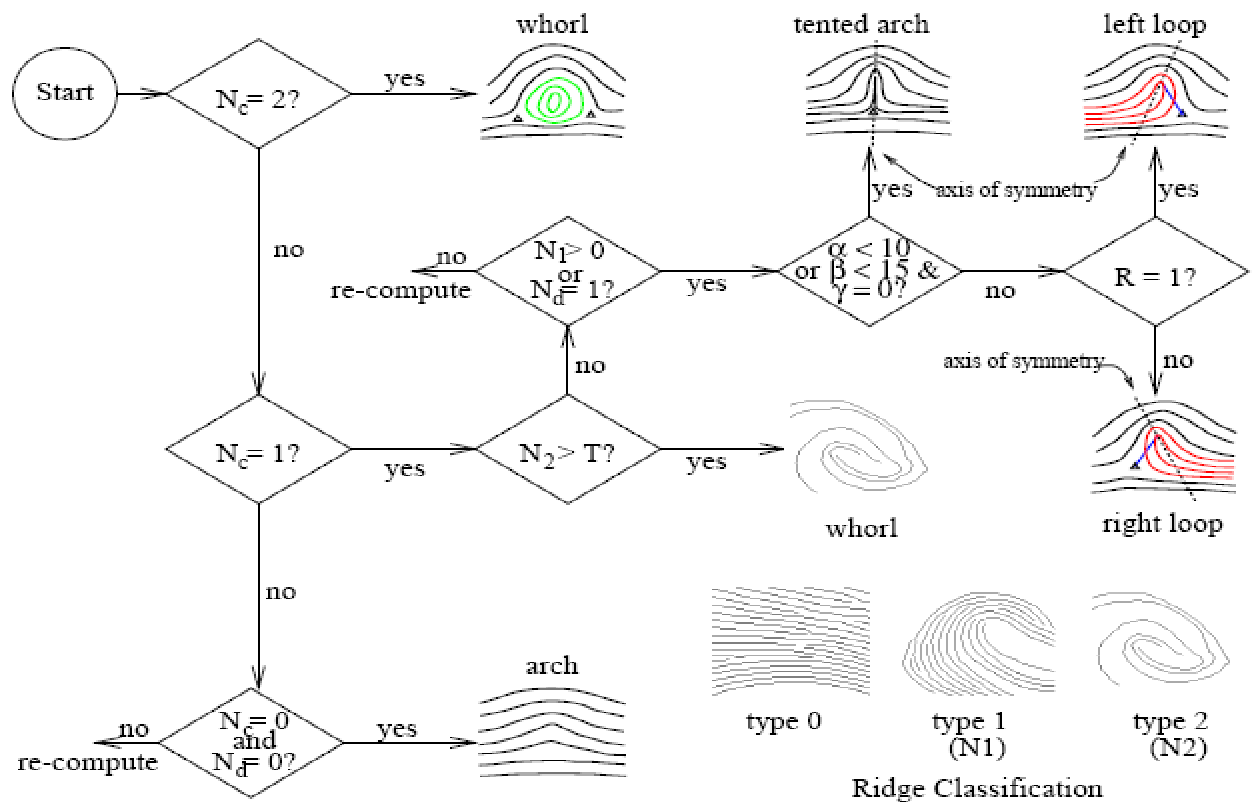


Fig: Various fingerprint classes – (a) Arch, (b) Tented Arch, (c) Right Loop, (d) Left Loop, (e) Whorl, (f) Twin Loop

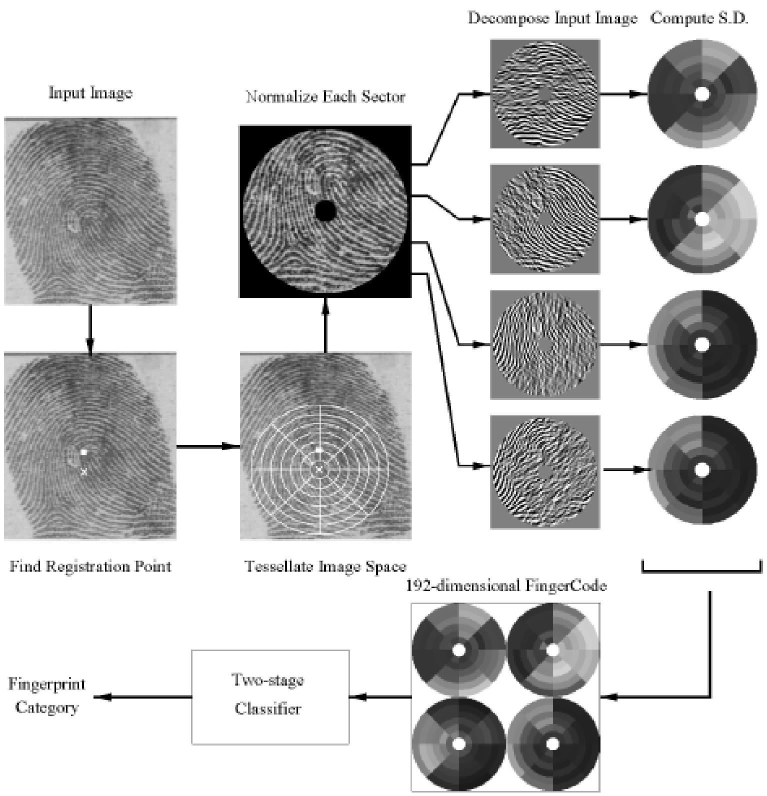
Classification Approaches

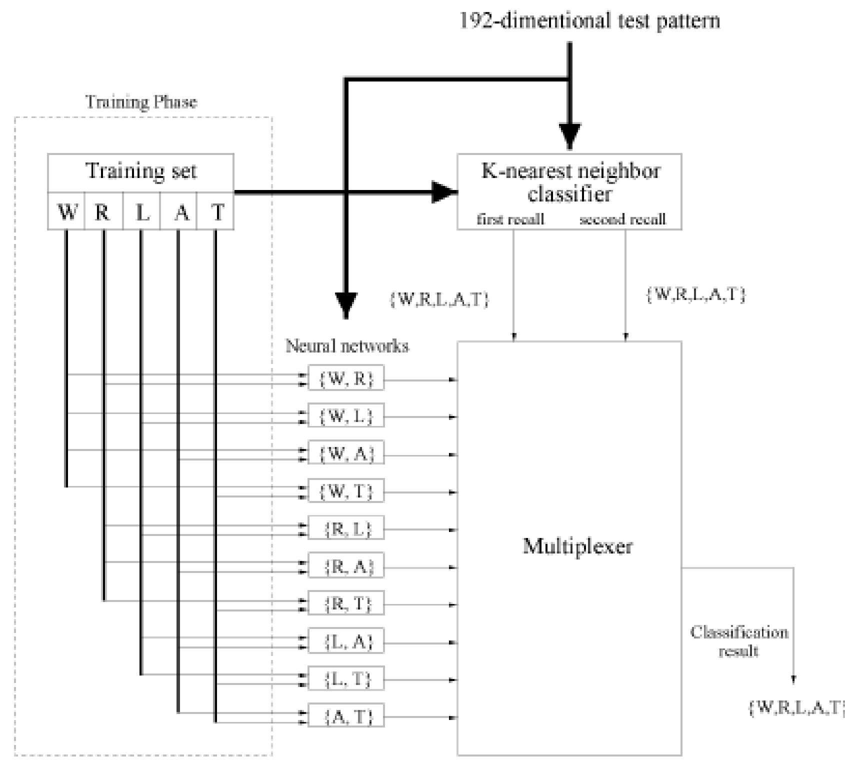
- **Rule based system.**

  Using location of Singular points and axis of symmetry to classify prints. [Jain/Pankanti]

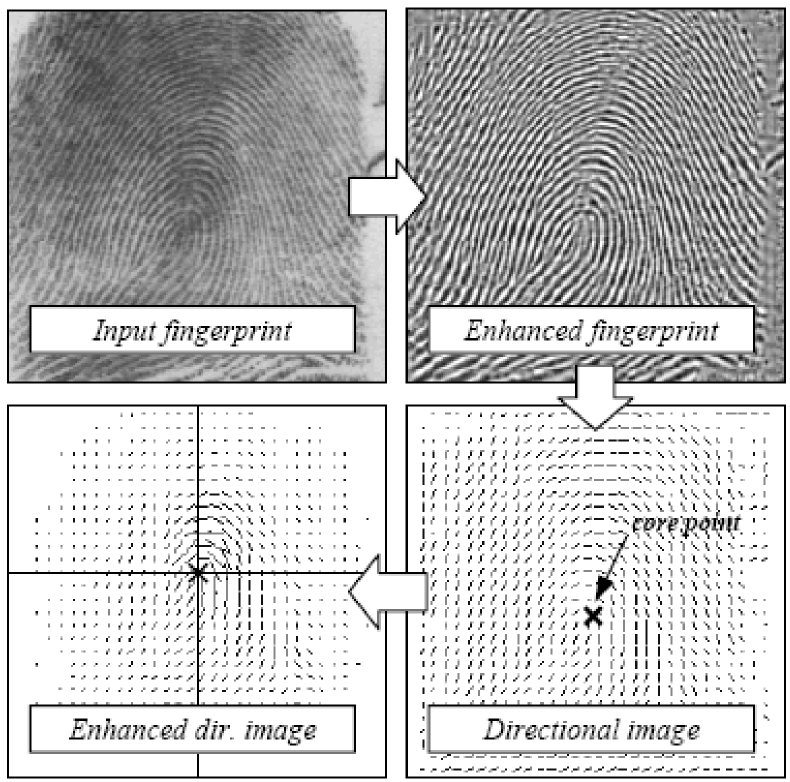


- **Multi-stage classifiers**
  - Using kNN to identify two candidate classes and Neural Networks for a final decision [Jain/Prabhakar]

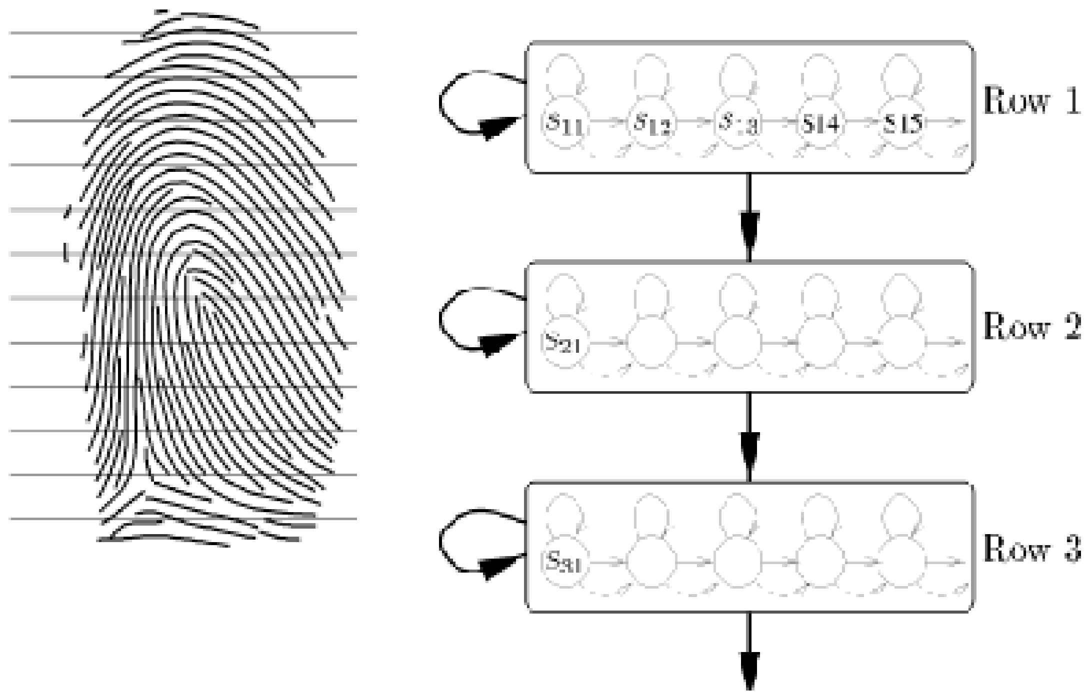




- **Multi-stage classifiers** contd..
  - Converts the image into a 28x30 grid and calculates orientation in each cell. Using MKL and SPD classifier combination [Capelli et al]



- **Stochastic Models for Classification**
  - 2 dimensional HMM [Senior]. Image is segmented and orientation of ridge at each segment is used.



Classification Results

Approach	# Classes	Misclassification Rate (%)	Dataset
Wilson [1993]	5	4.6*	Weighted NIST – 4 (2000 images)
Blue [1994]	5	7.2*	
Candela [1995]	5	9.5	NIST-14 (2700 images)
Karu [1996]	5	14.6	NIST-4 (4000 images)
Jain [1999]	5	10.0	NIST-4 (1000 images train + 1000 test)
Senior [2001]	4	8.5	
Yao [2003]	5	10.0	
Tan [2003]	5	7.2	
Cappelli [2003]	5	4.8	

Best error rate achieved is 4.8% for the 5 class problem (ATLRW) (Capelli’s method)

Disadvantages of a Classification-only approach

- Classification gives a significant speed-up, but greater speed-ups are needed for larger datasets. This is due to the separation of the dataset into only 5 (at times even 4) classes.
- Ambiguity between classes could mean that even 2 most probable classes are searched, increasing the size of the search space.
- Not all classes have equal size. Hence, for the more frequent classes, the reduction of search space is low.

Fingerprint Indexing

- **Using extracted features which provide higher discriminative power**
  - Greater reduction in size of search space

- **Approaches**
  - Filter based Indexing
    - Applies filters to image to get a feature-vector for the print
    - Matching is done by comparing feature vectors
  - Triplet based Indexing (Binning Approach)
    - Uses local arrangements of minutiae points
    - Fingerprints are enrolled in multiple bins based on presence of corresponding triplets

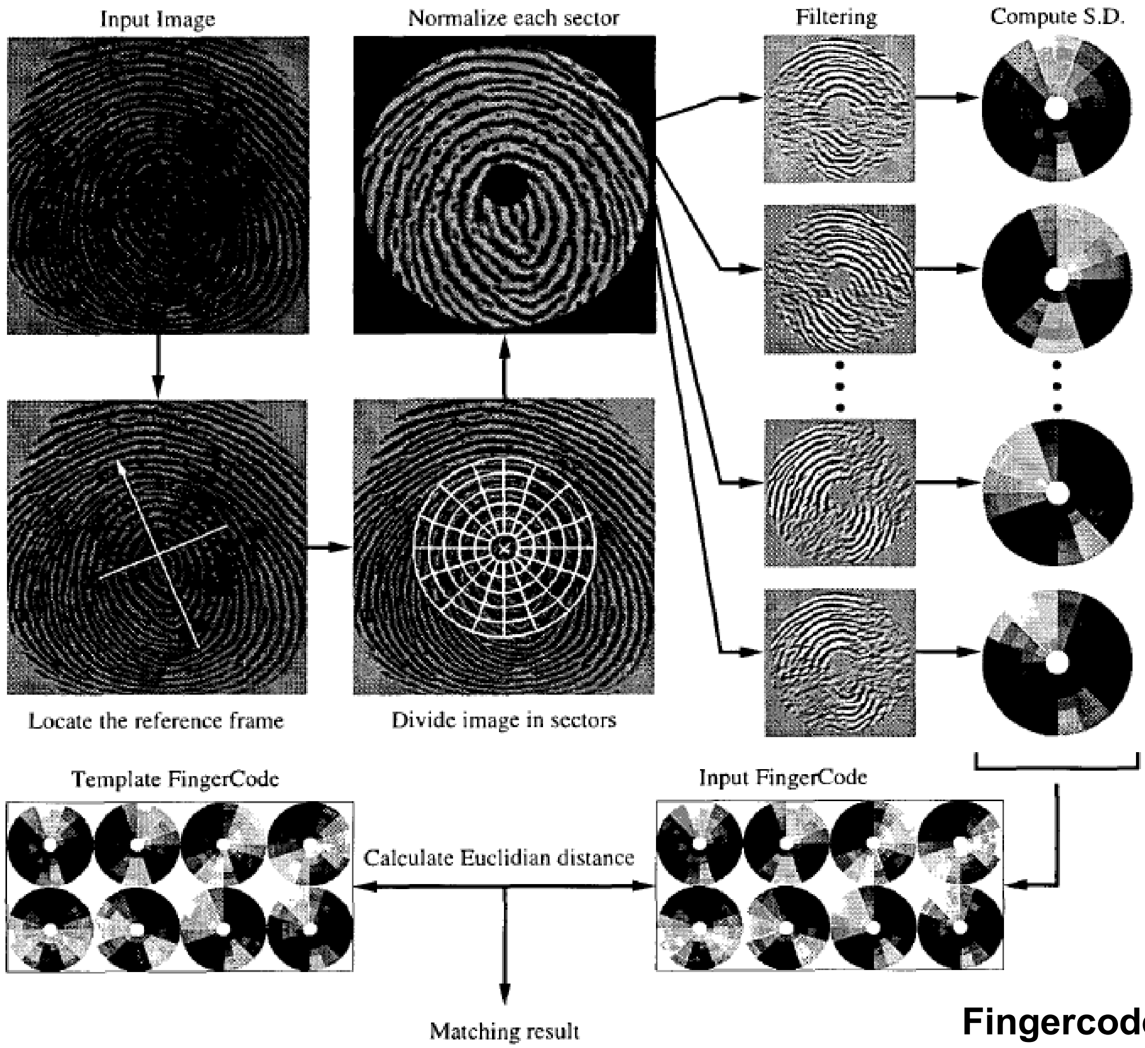
Filter based Indexing (FingerCode)

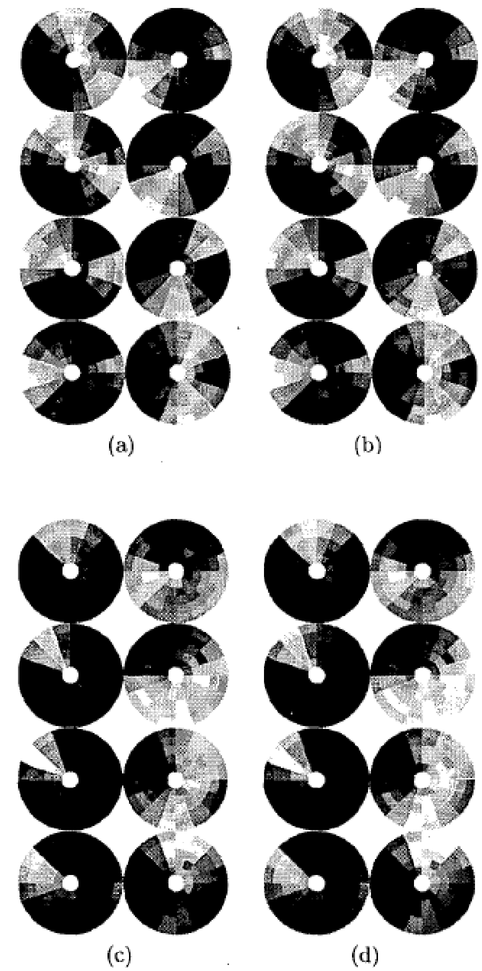
- [Jain] applies Gabor filters to each print to produce a 80 feature vector

- Each filter is applied in 8 directions to give us a 640 (80*8) feature vector called the FingerCode

- Matching score of two fingerprints is calculated using the Euclidean distance of their corresponding Fingercodes.

- Bit comparision based matching also makes Fingercode a good indexing scheme, ideal for large scale identification.

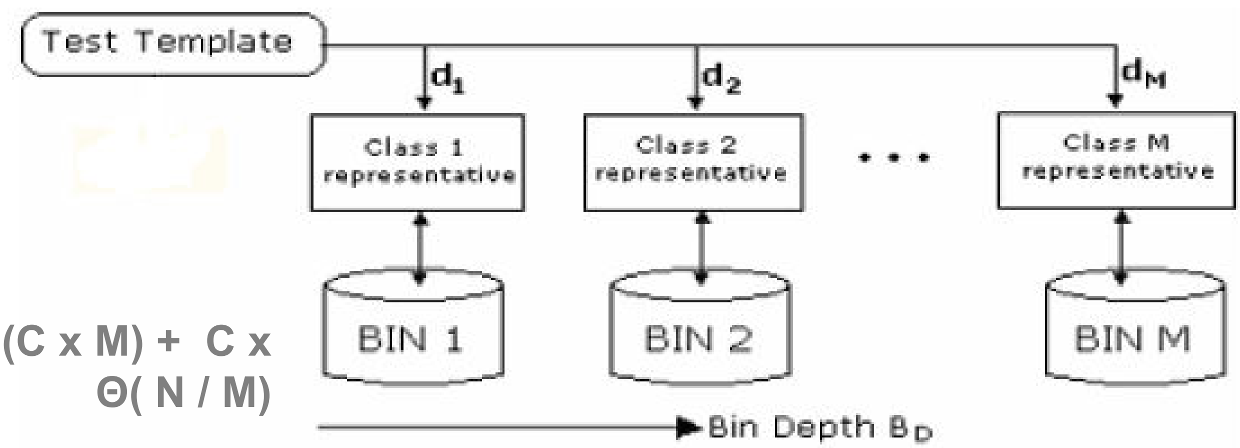




Fingercode representations of 2 fingers: (a) and (b) are calculated from different representations of the same finger, and (c) and (d) are calculated from samples taken from a different user.

Binning - Reducing Search Space

- Dataset is divided into *M* bins, and each template is enrolled into a particular bin
- For a test fingerprint, it is resolved to the nearest bin by comparing it against representative samples from each bin
- All templates from the nearest *C* bin(s) are compared with the test print



•**Average size of bin (N / M)**

•**Number of bins (C)**

Time to identify user = $\Theta(C \times M) + C \times \Theta(N / M)$

$\ll \Theta(N)$ – brute force

Minutiae Triplets

- Combinations of 3 neighboring minutia points
- High number of possible features
- Less prone to distortions
- Used for indexing & matching fingerprints

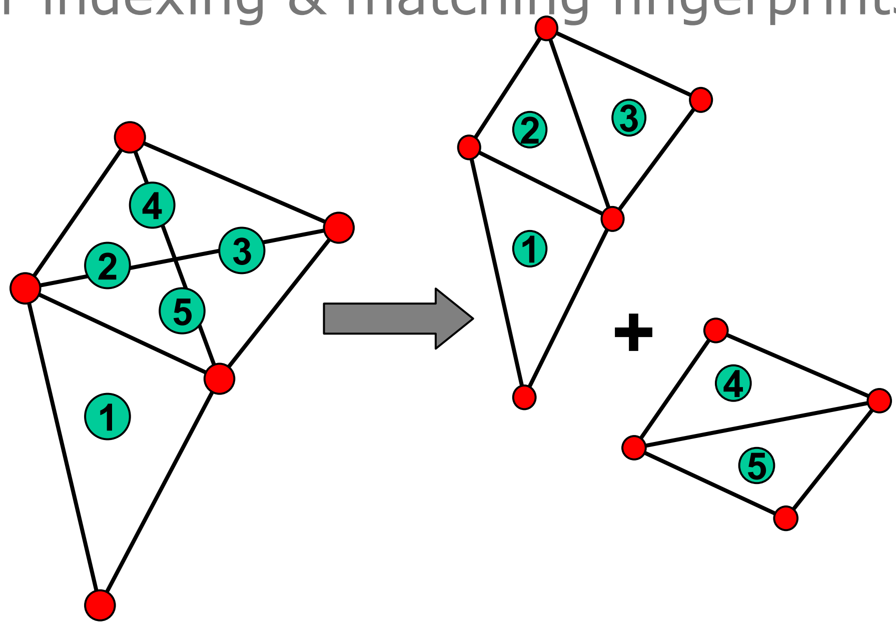
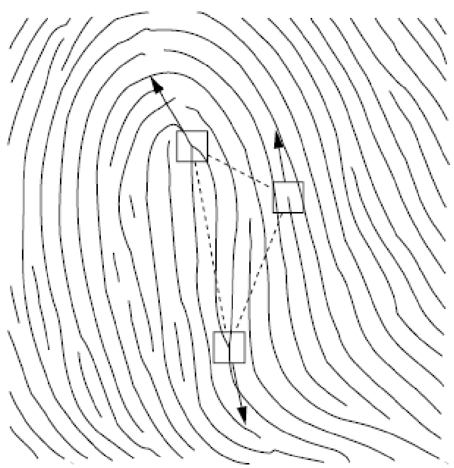
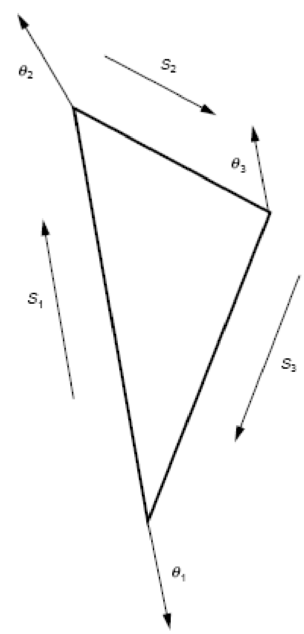


Fig: Different combinations of triplets [Choi 2003]

Triplet-based Indexing [Germain]

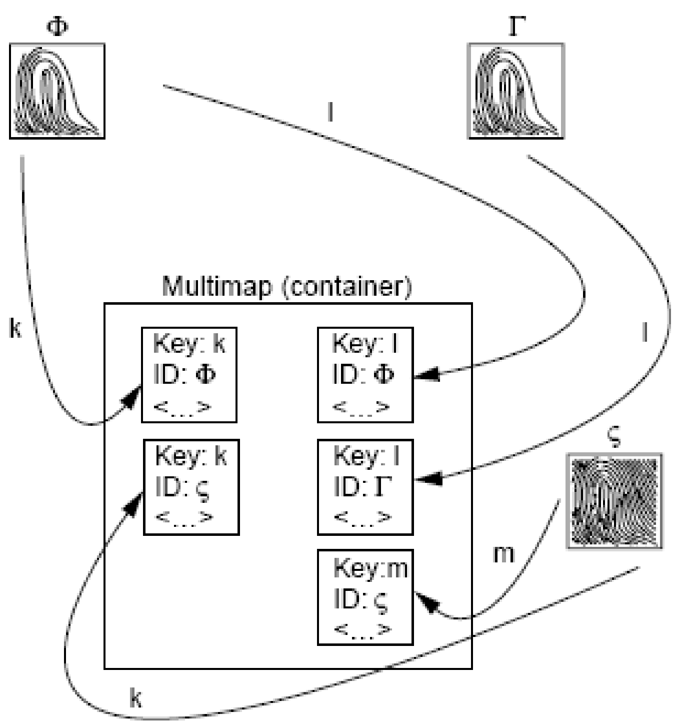
- 9 features are extracted for each triangle and are used to generate a key
  - Lengths of each side (3)
  - Orientations of ridge directions w.r.t. axis (3)
  - Number of ridges intersected by each side (3)



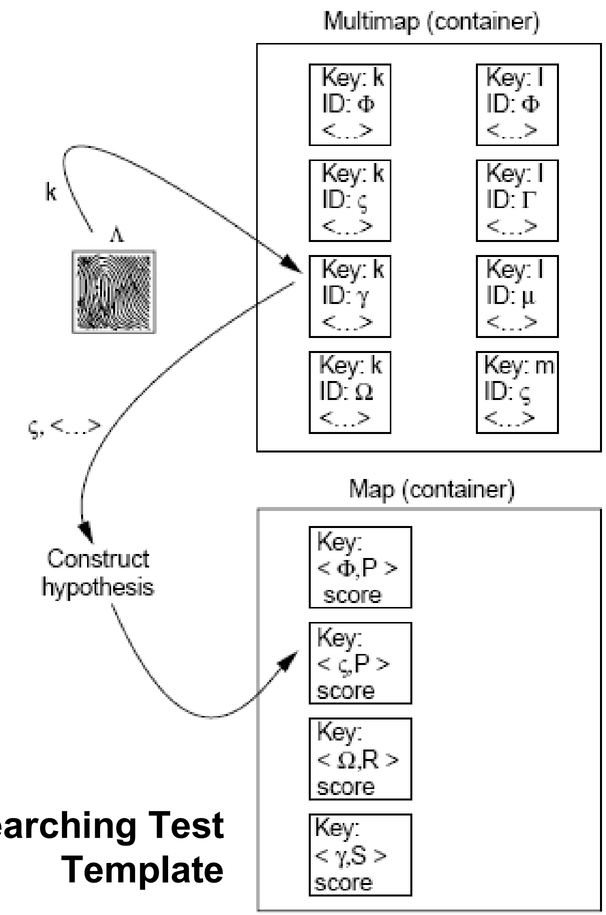


- **Enrollment** Bins triangles with similar features together.
- **Searching** For a test template, each triplet is used to retrieve a set of hypothesis (potential matching) prints.

  These are combined to give us the final identity of the user



Binning of Templates

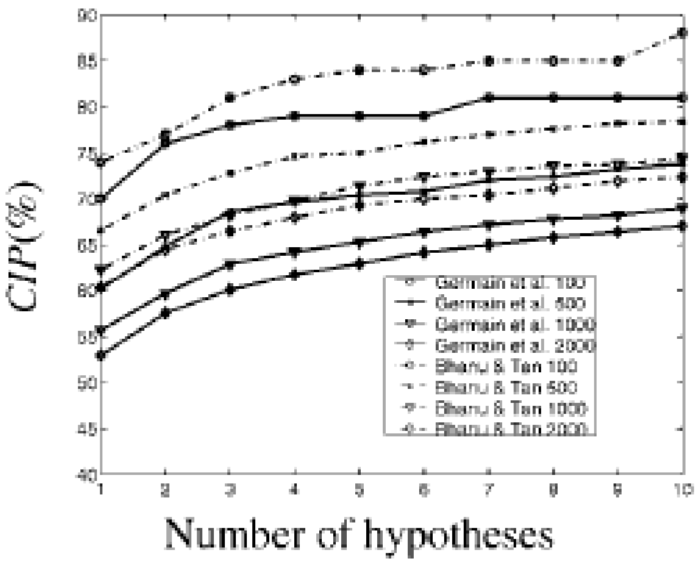


Searching Test Template

Center for Unified Biometrics and Sensors
University at Buffalo The State University of New York

Triplet-based Indexing [Bhanu][Choi]

- [Bhanu] uses similar triplet-based approach, uses “better” features
  - Max side, angles, (type, handedness ,direction) of triangle
- Fingerprint images are sorted based on the number of triangles they match, and a score is calculated for each candidate image.
- Gives a better performance than Germain’s approach



- [Choi] have taken the same approach, and added modifications to the system to get a better performance.
  - Weights to the matching pairs
  - Normalization of similarity scores.

Issues in Binning / Indexing schemes

- **Execution time is still large**
  - Even though the search space is reduced to a linear fraction of the total space.
  - Large execution times for bigger datasets.
- **Separate matching algorithm needed**
  - Most systems just list possible matches.
  - Matching / scoring system must be used on each candidate.
- **Significant overhead in building indexes / bins**
  - For static datasets, one-time cost
  - Dynamic datasets – need to update index for newly enrolled templates
- **Must handle variations in biometric features**
  - Searching in wrong bin would lead to errors
  - Features used should have minimum intra-class and maximum inter-class variance

Outline of Talk

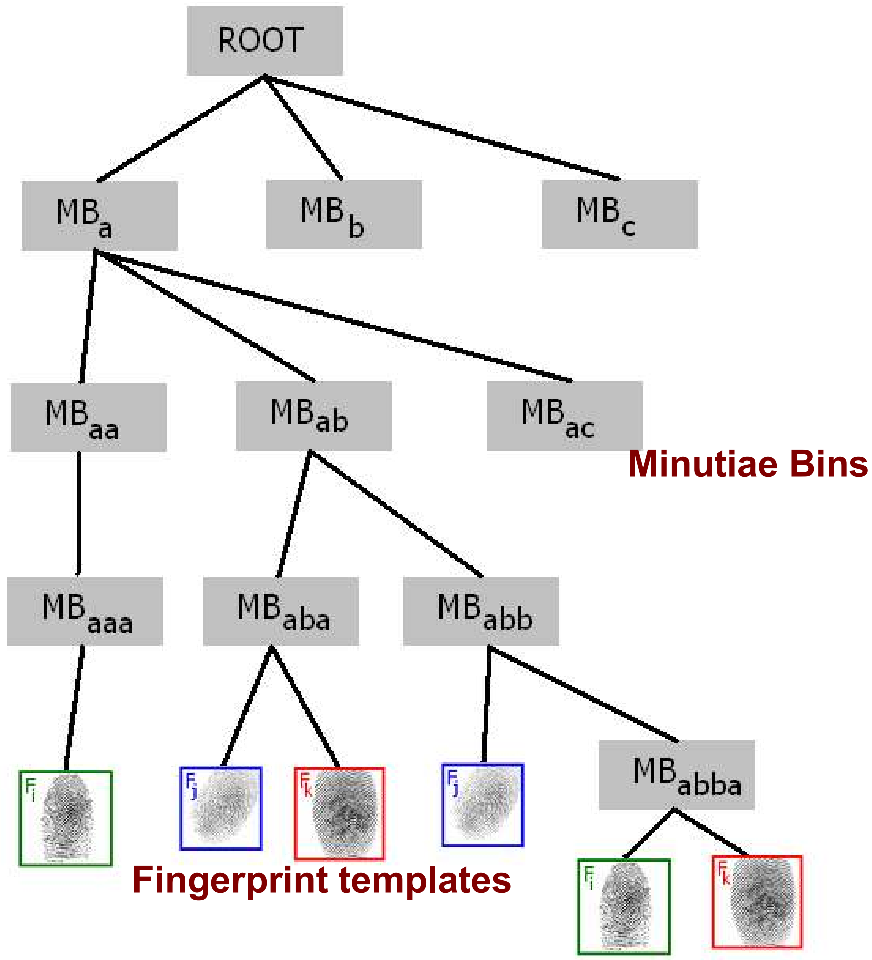
- **Fingerprint Identification using a Minutiae Tree**
  - Challenges & Motivation
  - Previous Classification and Indexing approaches
  - Our Method: Tree building and Searching
  - Handling Errors in Binning
  - Performance Analysis using Synthetic Datasets
  - Statistical Study of Minutiae Matching

Our Approach: Minutiae Tree Indexing System

- Fingerprint dataset is organized as a tree representing the arrangement of minutiae points.
  - ***Tree based search allows to search large datasets in real-time***
- Fingerprint templates are enrolled at multiple locations to compensate for the variations in feature values.
- Enrollment and search procedures use neighboring arrangements of minutiae points.
- Matching algorithm is optional.
  - ***Multi-level search results in automatic fingerprint matching.***

Tree Organization

- The dataset is represented as a tree.
- Similar minutiae points are binned into nodes.
- Fingerprint templates located at the leaves.
- Each path from root to a node represents an arrangement of minutiae points.



Branch Selection

- Branching on each level is done based the relative features of the current minutiae and its nearest neighbor.
- Finite number of bins are used to handle continuous-valued features.

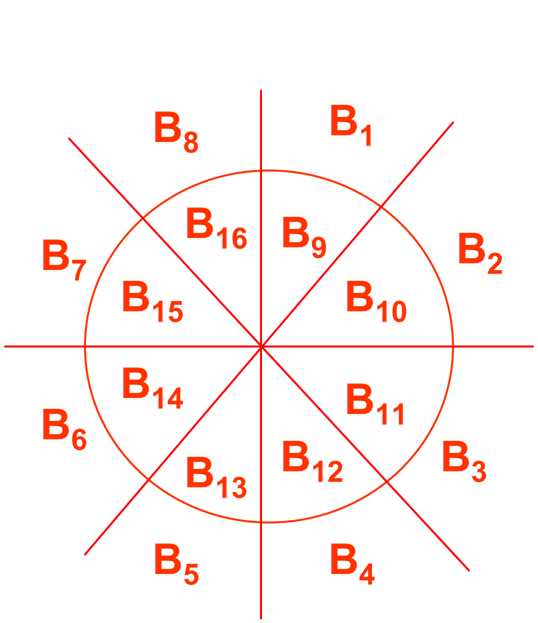


Fig: 16-bins based on minutia point position relative to centre point.

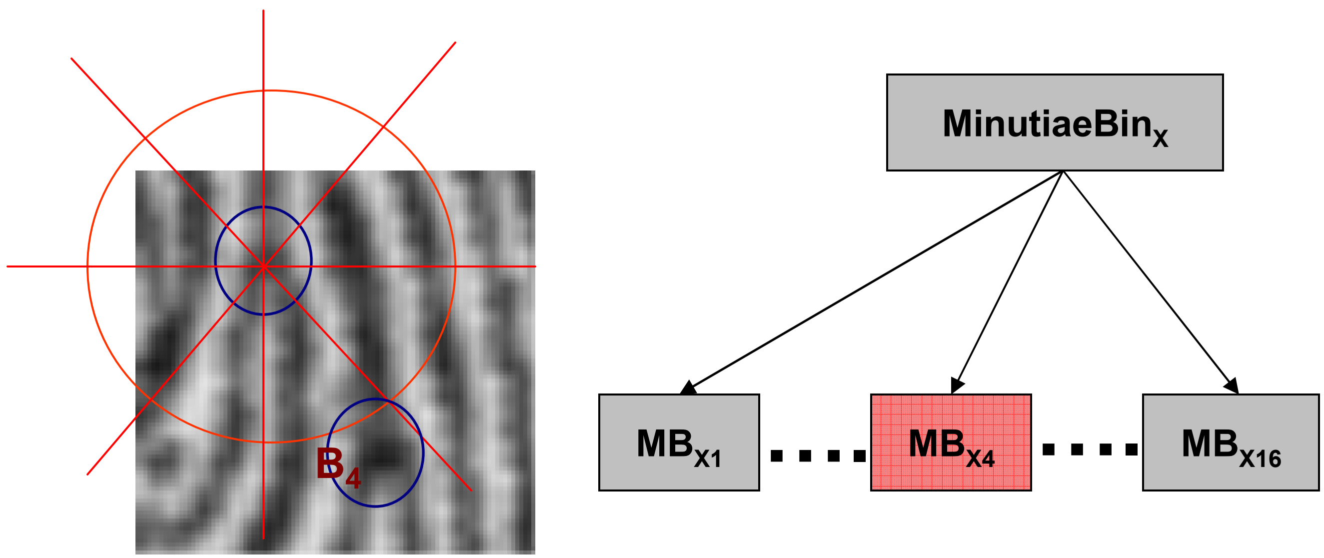
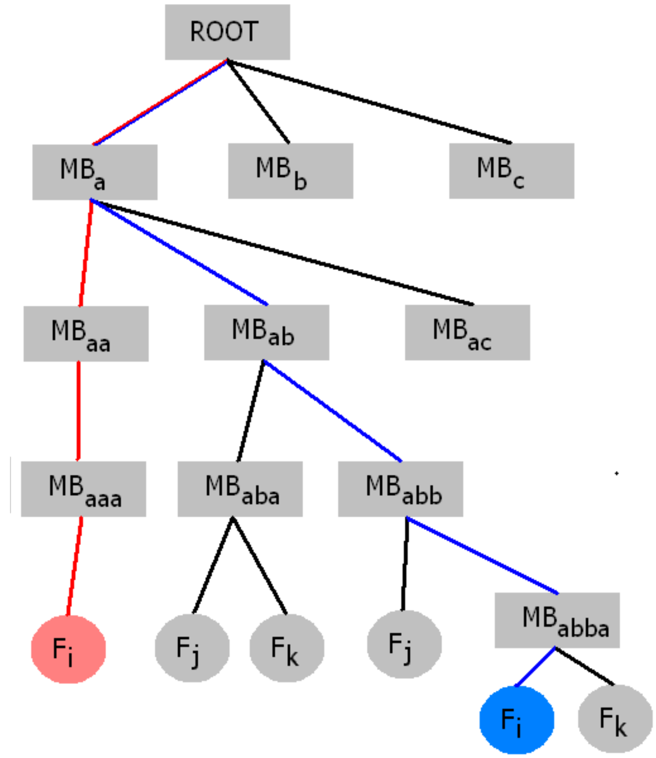


Fig: Bin selection and minutiae enrollment based on relative features.

Fingerprint Enrollment

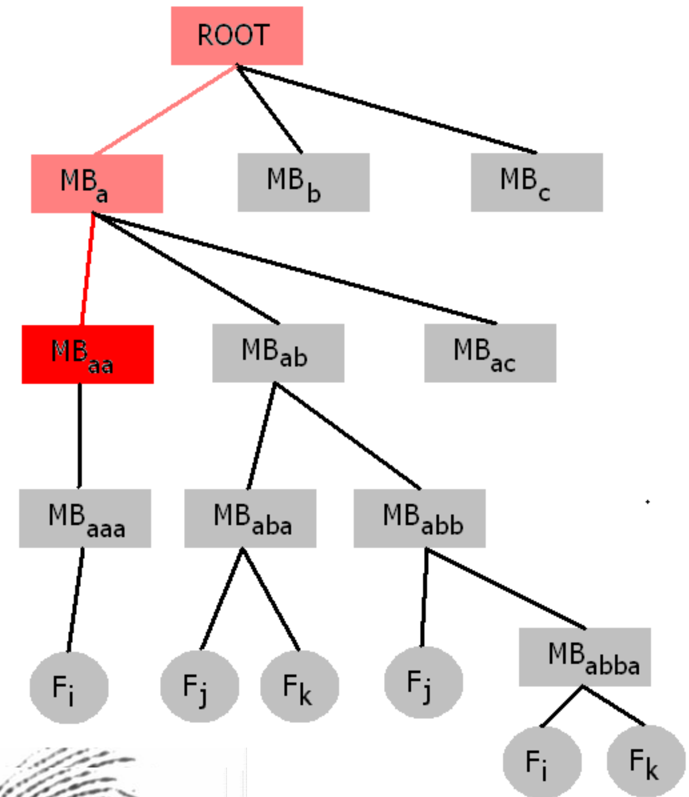
- **Fingerprint preprocessing and minutiae point extraction.**
- **One minutia point is selected as the root**
- **For each neighboring minutiae point, we traverse down the tree one level at a time and add the fingerprint at the appropriate leaf node.**
- **The process can be repeated for different points.**



Thus we see that we do not need to rebuild the tree a later stage while enrolling additional users into the system.

Fingerprint Matching

- Fingerprint preprocessing and minutiae extraction.
- Select one point as root and find nearest neighbor. Calculate features of this point (neighbor) w.r.t. the current minutia.
- Based on the feature values, traverse down the tree, taking one minutia point at a time.

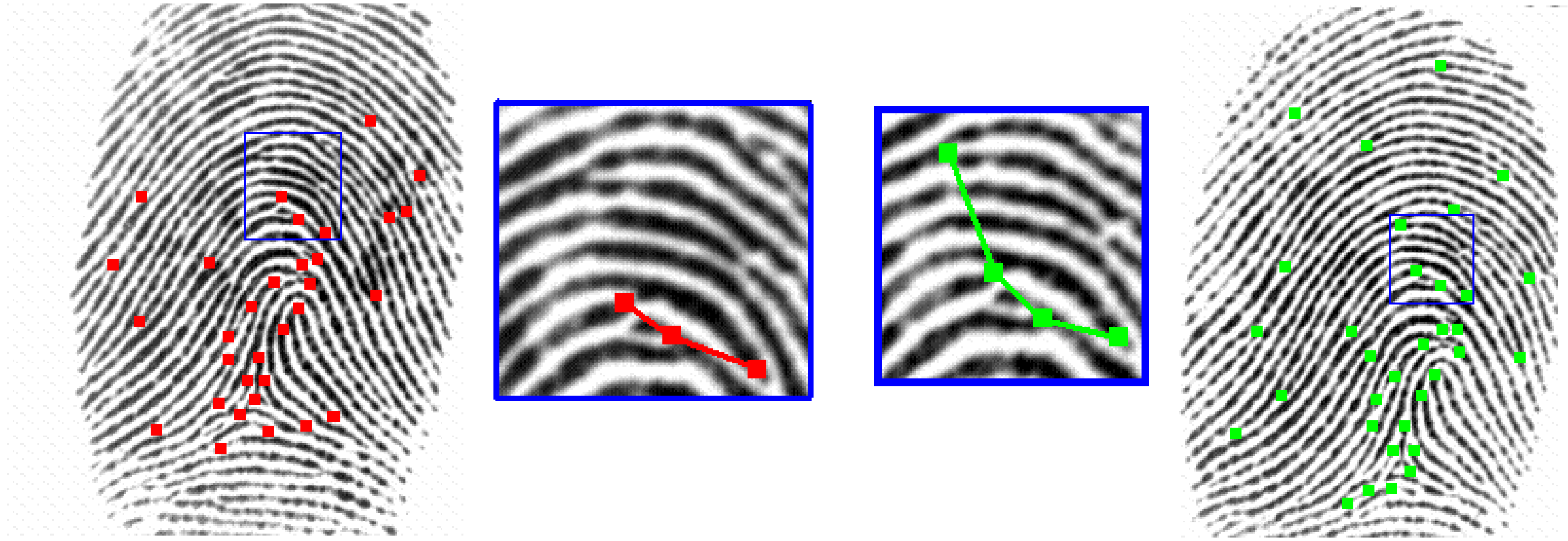


Outline of Talk

- **Fingerprint Identification using a Minutiae Tree**
  - Challenges & Motivation
  - Previous Classification and Indexing approaches
  - Our Method: Tree building and Searching
  - Handling Errors in Binning
  - Performance Analysis using Synthetic Datasets
  - Statistical Study of Minutiae Matching

Binning Errors

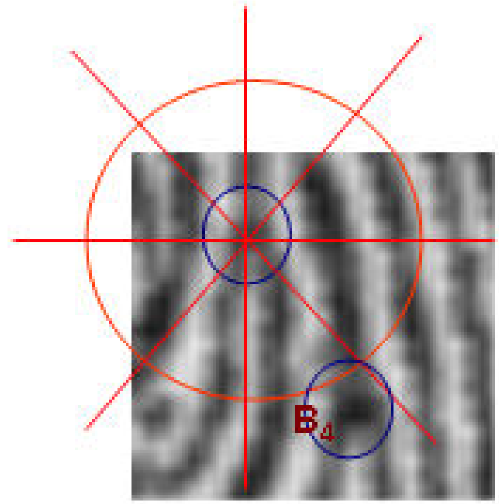
1. Spurious & Missing Minutiae



Presence of noise leads to some minutiae being missed and other points incorrectly classified as minutiae.

This may lead to variation in minutiae patterns while indexing

2. Variation in Minutiae Feature Values



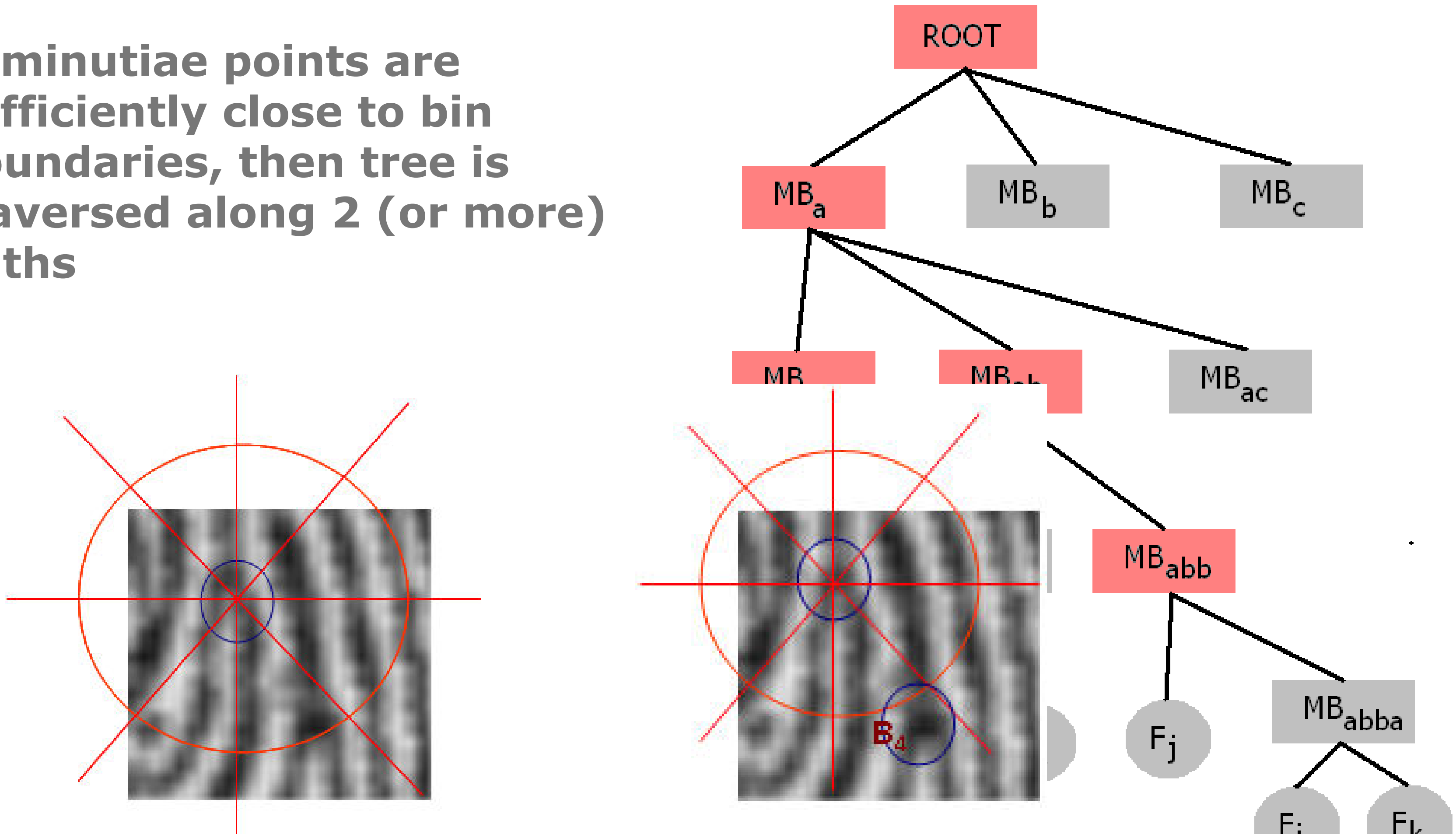
Ideal binning scenario: Samples of the same user always map to same bin

Binning Errors could be caused due to:

- Slight changes in feature values close to bin boundaries – due to distortion
- Missing or incorrect value/order of features – errors in feature extraction / noise

Searching in Multiple Bins

- If minutiae points are sufficiently close to bin boundaries, then tree is traversed along 2 (or more) paths



Searching Phase

Enrollment Phase

Experiments on FVC Datasets

- **Datasets : FVC 2002 DB1 and FVC 2004 DB1**
  - 100 users each * 8 prints per user
  - First 3 prints enrolled and 5 used for testing
- **N minutiae points used for building index**
  - 1 root (start) point
  - 1 point for aligning the bins
  - (n-2) points are compared with root to build tree
- **Binning based on 3 features**
  - Distance : 2 bins
  - Angle : 8 bins
  - Orientation : 8 bins
  - Total number of bins at each level : 2 x 8 x 8 = 128

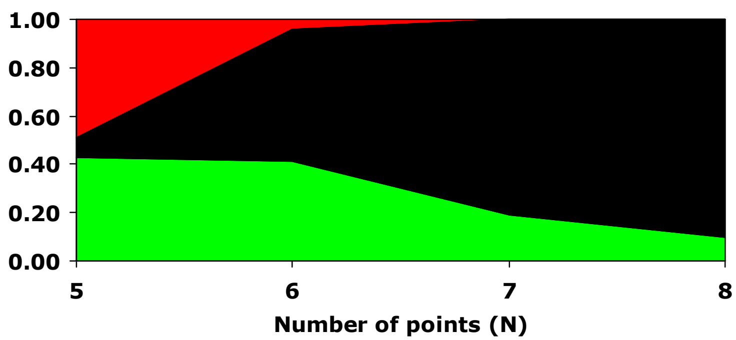
Center for Unified Biometrics and Sensors
University at Buffalo The State University of New York

Single Path Search

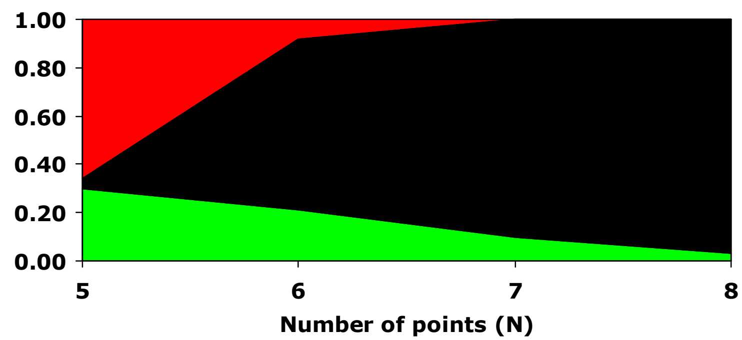
FVC 2002 DB1	N = 5	N = 6	N = 7	N = 8
Correct	0.43	0.41	0.19	0.10
No Matches Found	0.08	0.55	0.81	0.90
Incorrect	0.49	0.04	0	0
Average Returned Matches	1.42	1.03	1.00	1.00

FVC 2004 DB1	N=5	N=6	N=7	N=8
Correct	0.30	0.21	0.10	0.03
No Matches Found	0.04	0.71	0.90	0.97
Incorrect	0.66	0.08	0.00	0.00
Average Returned Matches	2.02	1.02	1.00	1.00

FVC 2002



FVC 2004



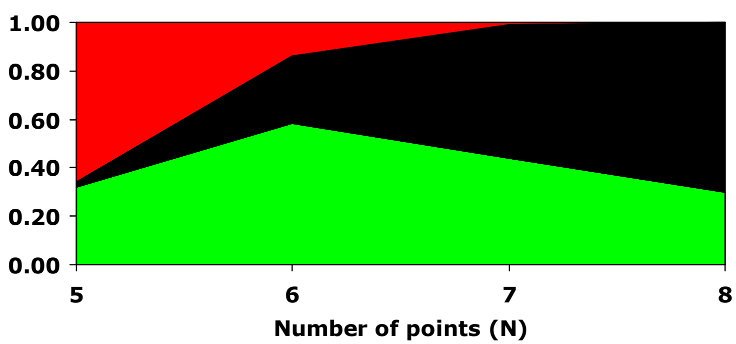
Center for Unified Biometrics and Sensors
University at Buffalo The State University of New York

Multiple Path Search

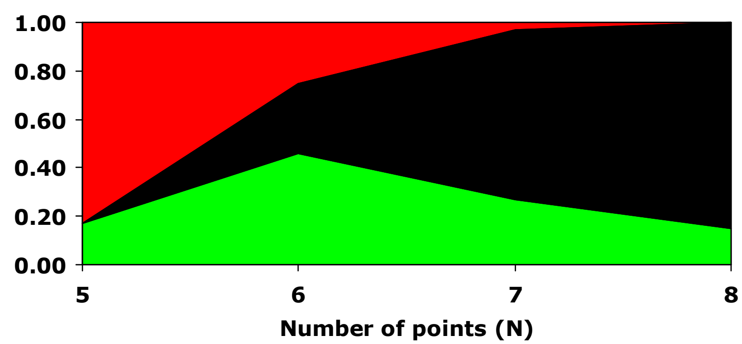
FVC 2002 DB1	N = 5	N = 6	N = 7	N = 8
Correct	0.32	0.58	0.44	0.30
No Matches Found	0.02	0.28	0.55	0.70
Incorrect	0.66	0.14	0.01	0
Average Returned Matches	1.29	1.08	1	1

FVC 2004 DB1	N=5	N=6	N=7	N=8
Correct	0.17	0.46	0.27	0.15
No Matches Found	0.00	0.29	0.70	0.85
Incorrect	0.83	0.25	0.03	0.00
Average Returned Matches	1.45	1.24	1.03	1.00

FVC 2002



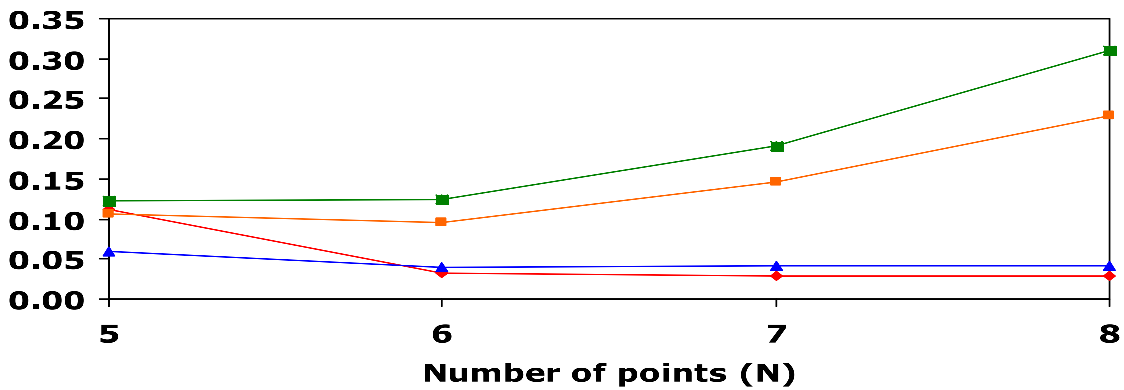
FVC 2004



Retrieval Time for Multi-Level Search

FVC 2002 DB1	N = 5	N = 6	N = 7	N = 8
Single Path Search	0.112	0.032	0.028	0.028
Multiple Path Search	0.106	0.096	0.146	0.230

FVC 2004 DB1	N=5	N=6	N=7	N=8
Single Path Search	0.060	0.040	0.042	0.042
Multiple Path Search	0.122	0.124	0.192	0.310



(Hardware: P4 2.2 GHz CPU, 512 MB RAM, MS Visual C++ 6.0 running WinXP)

Enrolling Multiple Templates

- **FVC 2002 – DB1**

Single Path

No. Templates Enrolled	1	2	3
Correct	0.23	0.35	0.41
No matches found	0.74	0.62	0.55
Incorrect	0.03	0.03	0.04

Multiple Path

No. Templates Enrolled	1	2	3
Correct	0.43	0.57	0.58
No matches found	0.48	0.33	0.28
Incorrect	0.09	0.10	0.14

- **FVC 2004 – DB1**

Single Path

No. Templates Enrolled	1	2	3
Correct	0.11	0.17	0.21
No matches found	0.87	0.78	0.71
Incorrect	0.02	0.05	0.08

Multiple Path

No. Templates Enrolled	1	2	3
Correct	0.24	0.38	0.46
No matches found	0.64	0.42	0.28
Incorrect	0.12	0.20	0.25

Enrolling multiple templates helps compensate for distortions in fingerprint images

Searching Multiple Templates

- **FVC 2002 – DB1**

Single Path

Templates Searched	1	2	3	4	5
Accuracy	0.81	0.91	0.91	0.95	0.96
Matching Rate	0.22	0.52	0.72	0.84	0.87

Multiple Path

Templates Searched	1	2	3	4	5
Accuracy	0.56	0.74	0.80	0.86	0.87
Matching Rate	0.32	0.63	0.76	0.84	0.85

- **FVC 2004 – DB1**

Single Path

Templates Searched	1	2	3	4	5
Accuracy	0.73	0.76	0.81	0.80	0.77
Matching Rate	0.19	0.32	0.50	0.57	0.58

Multiple Path

Templates Searched	1	2	3	4	5
Accuracy	0.66	0.69	0.73	0.73	0.64
Matching Rate	0.47	0.61	0.71	0.72	0.63

Probe multiple templates per user, aggregate the candidates returned and compare candidate with highest count against a threshold

Accuracy = Number of total correct users / Total users (candidates) returned
Matching Rate = Total user (candidates) returned / Total users probed

Searching multiple candidates reduces number of incorrect matches

Outline of Talk

- **Fingerprint Identification using a Minutiae Tree**
  - Challenges & Motivation
  - Previous Classification and Indexing approaches
  - Our Method: Tree building and Searching
  - Handling Errors in Binning
  - Performance Analysis using Synthetic Datasets
  - Statistical Study of Minutiae Matching

Synthetic Templates

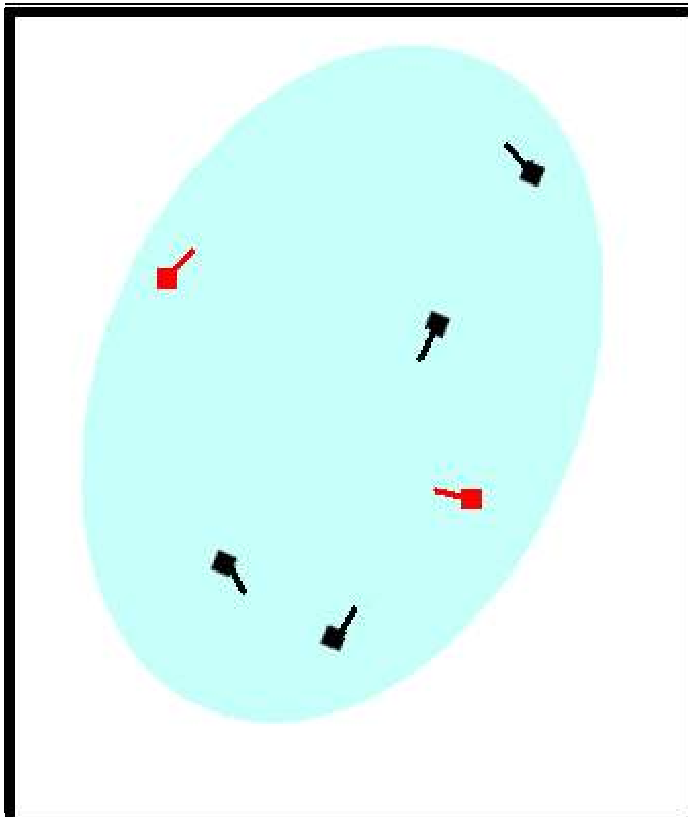
- Arrangement of minutiae points – (x,y,θ)
- Minutiae based matching: No images generated

- Advantages
  - Can control distortions applied
  - Generation of large sized datasets
  - Generation of multiple templates per user

- One master template per user
  - Minutiae points randomly generated
- Sample templates generated from master template
  - Created by applying distortions
  - Used for enrollment / testing

- Master template
  - Get width and height
  - Randomly distribute minutiae points in the area
  - Assign orientation value to each point, to get (x,y,θ) form

- Sample Template - *Apply distortions to master template*
  - Global Distortions – Whole Template
    - Translation
    - Rotation
  - Local Distortions – Individual Points
    - Shifting each minutiae point
    - Changing orientation of minutiae point
  - Point –based Distortions
    - Missing Points
    - Spurious Points



Experiments

- **Synthetic Datasets**
  - **8 templates per user – 3 enrolled, 5 test**

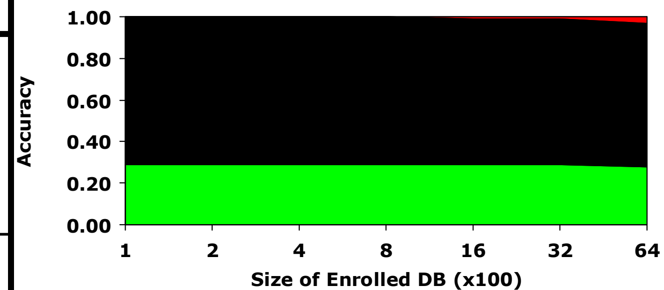
- **Testing for Distortions**
  - **One distortion parameter changes, others kept constant**
  - **100 users enrolled & tested**

- **Testing for Size of Dataset & Features**
  - **100 users tested, enrollment size changes**
  - **Distortion values kept to default**

Size of Dataset

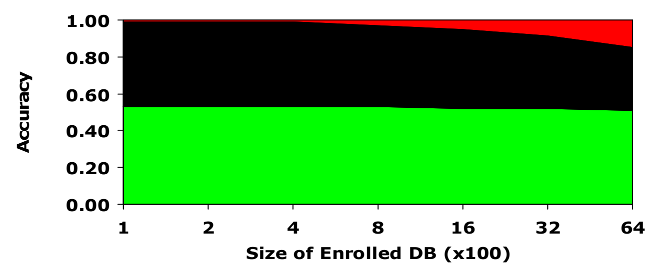
- ## Single Path Search

(x 100)	1	2	4	8	16	32	64
Correct	0.29	0.29	0.29	0.29	0.29	0.29	0.28
No matches found	0.71	0.71	0.71	0.71	0.70	0.70	0.69
Incorrect	0.0	0.0	0.0	0.0	0.01	0.01	0.03
Average returned matches	1.0	1.0	1.0	1.0	1.0	1.0	1.01



- ## Multiple Path Search

(x 100)	1	2	4	8	16	32	64
Correct	0.53	0.53	0.53	0.53	0.52	0.52	0.51
No matches found	0.46	0.46	0.46	0.44	0.43	0.39	0.34
Incorrect	0.01	0.01	0.01	0.03	0.05	0.09	0.15
Average returned matches	1.0	1.0	1.01	1.01	1.03	1.04	1.10



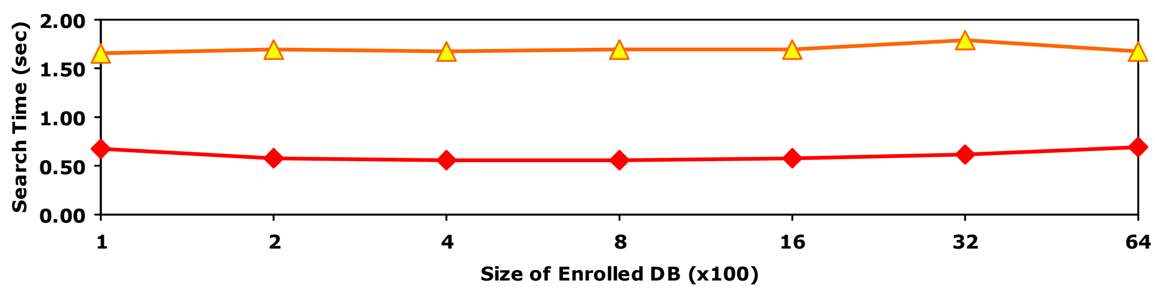
Number of correct matches remains constant with increase in database size

Center for Unified Biometrics and Sensors
University at Buffalo The State University of New York

Scaling with Large Datasets

- ## Single Path Search

(x 100)	1	2	4	8	16	32	64
Search Time	0.668	0.574	0.564	0.564	0.572	0.618	0.688
Average returned matches	1.0	1.0	1.0	1.0	1.0	1.0	1.01



- ## Multiple Path Search

(x 100)	1	2	4	8	16	32	64
Search Time	1.658	1.688	1.680	1.686	1.684	1.788	1.678
Average returned matches	1.0	1.0	1.0	1.0	1.0	1.0	1.01

Retreival time remains constant with increase in database size

Effect of Binning Features

SINGLE PATH SEARCH	All	~Dist	~Angle	~Orient
Correct	0.32	0.43	0.8	0.4
No matches found	0.68	0.55	0.03	0.09
Incorrect	0	0.02	0.17	0.51

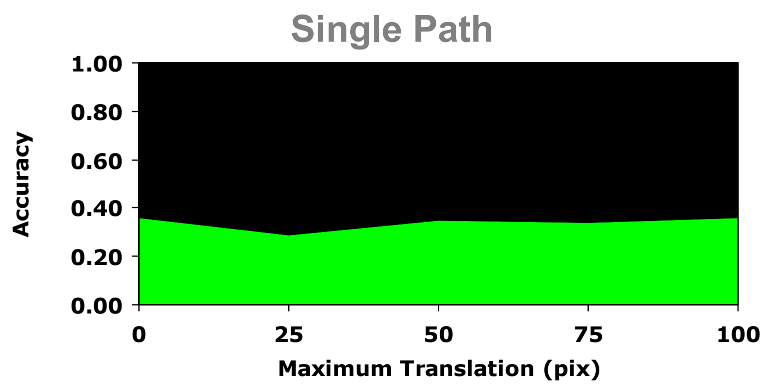
MULTIPLE PATH SEARCH	All	~Dist	~Angle	~Orient
Correct	0.59	0.62	0.75	0.27
No matches found	0.40	0.30	0	0.01
Incorrect	0.01	0.08	0.25	0.72

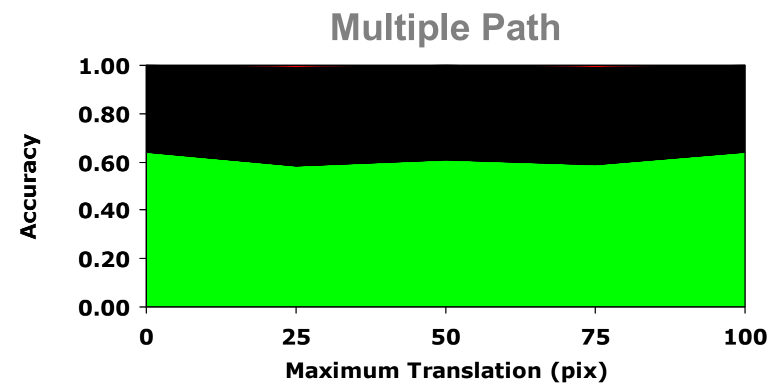
- *Elimination of even a single feature affects system performance*

- *Additional features might improve system accuracy*

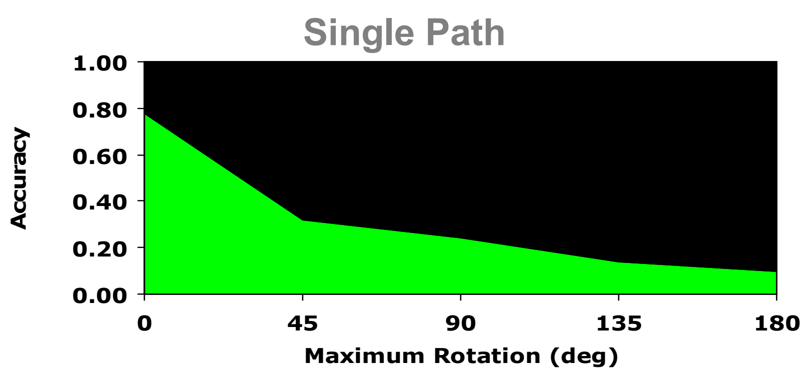
Global Distortions

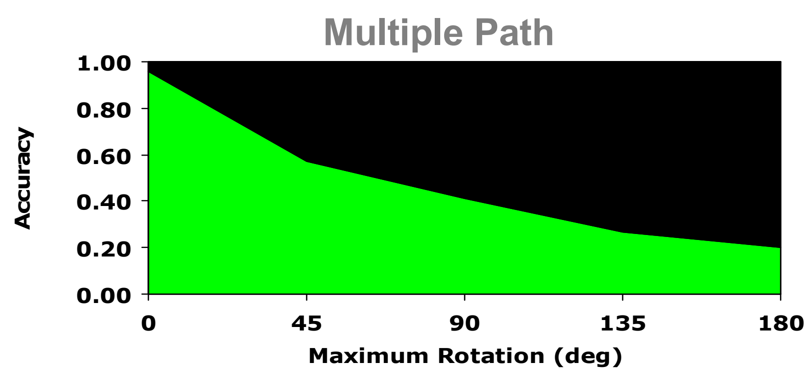
- **Translation of Template**





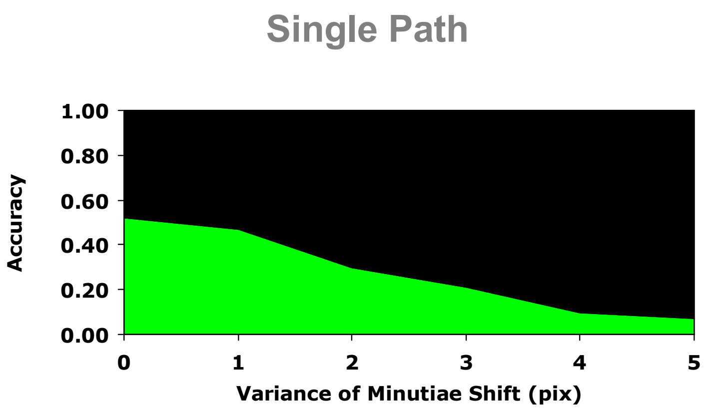
- **Rotation of Template**

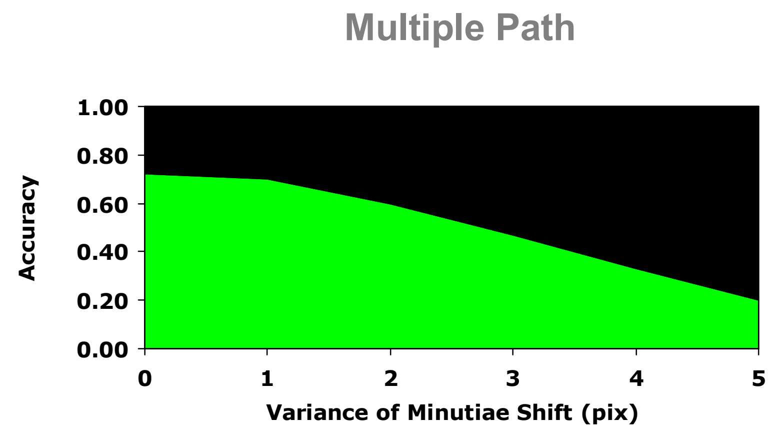




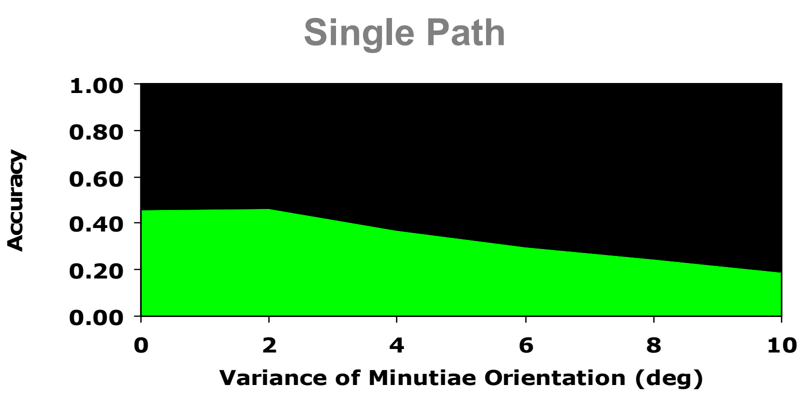
Local Distortions

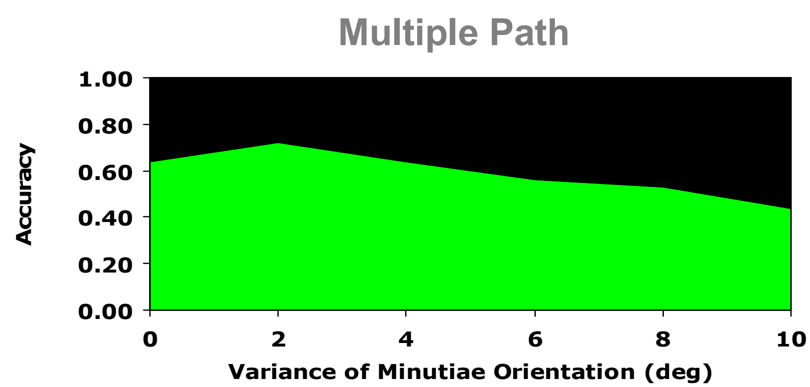
- **Shifting Minutiae Points**





- **Shifting Minutiae Orientation**

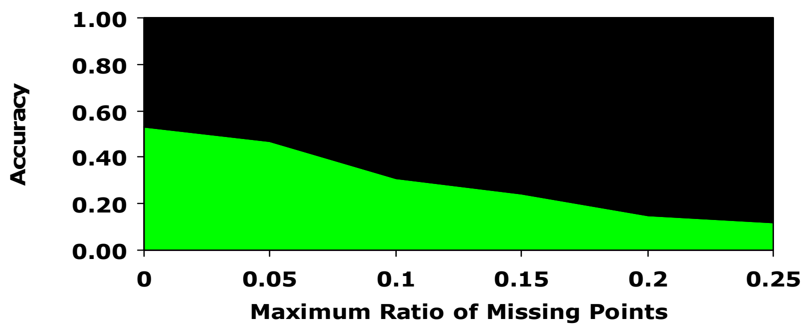




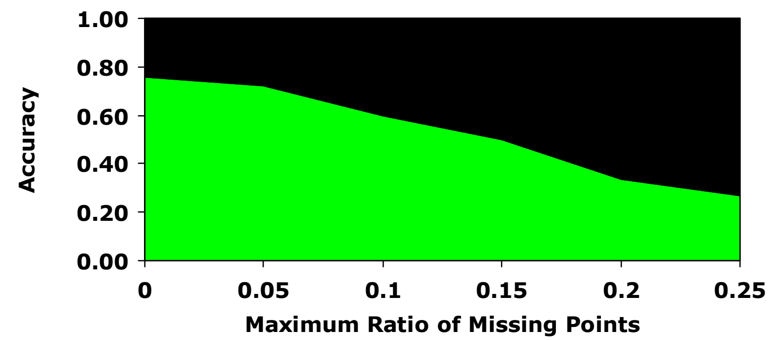
Point Distortions

- **Missing Points**

Single Path

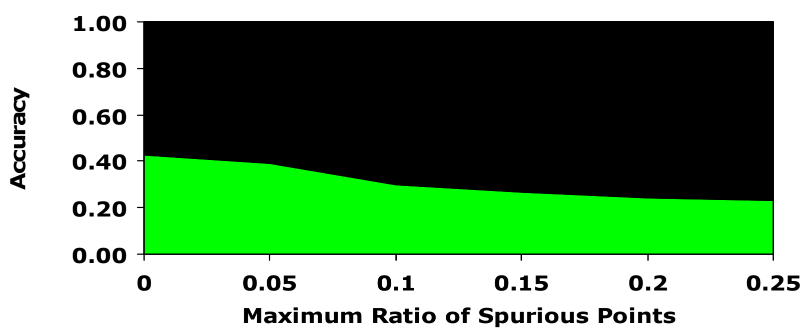


Multiple Path

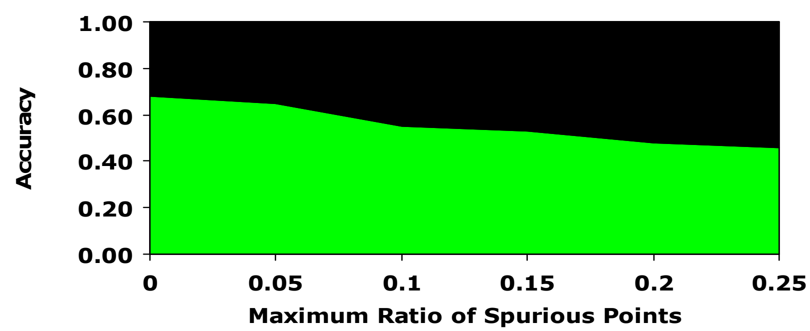


- **Spurious Points**

Single Path



Multiple Path

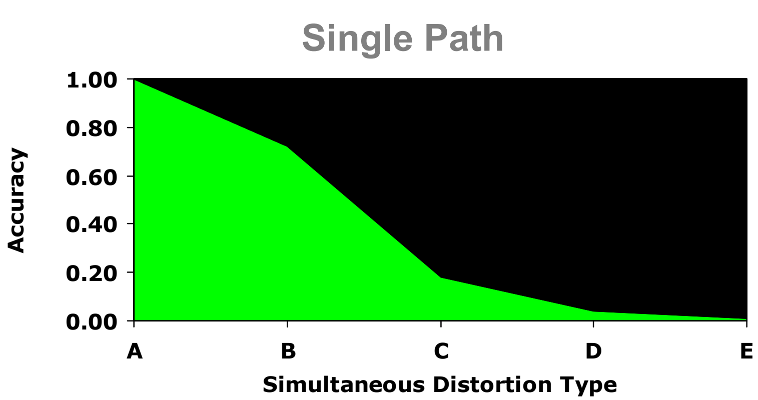


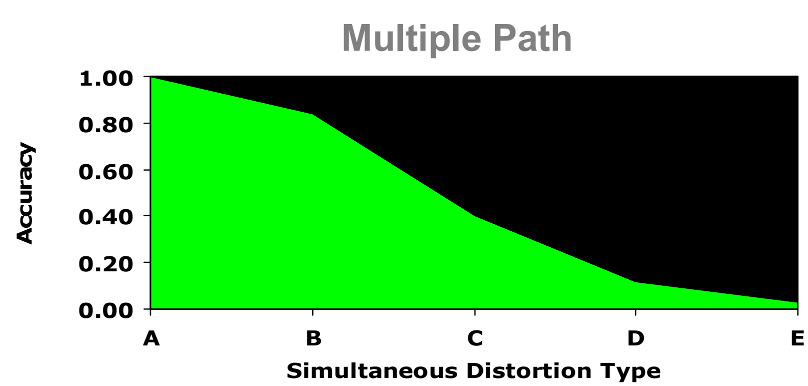
Simultaneous Distortions

- Varying all distortions together
- Indicator of system performance on highly degraded image sets

RED = Default

Distortion	Translation	Rotation	Shift	Orientation	Missing	Spurious
A	0	0	0	0	0	0
B	25	45	1	2	0.05	0.05
C	50	90	2	4 (Default=5)	0.10	0.10
D	75	135	3	6	0.15	0.15
E	100	180	4	8	0.20	0.20



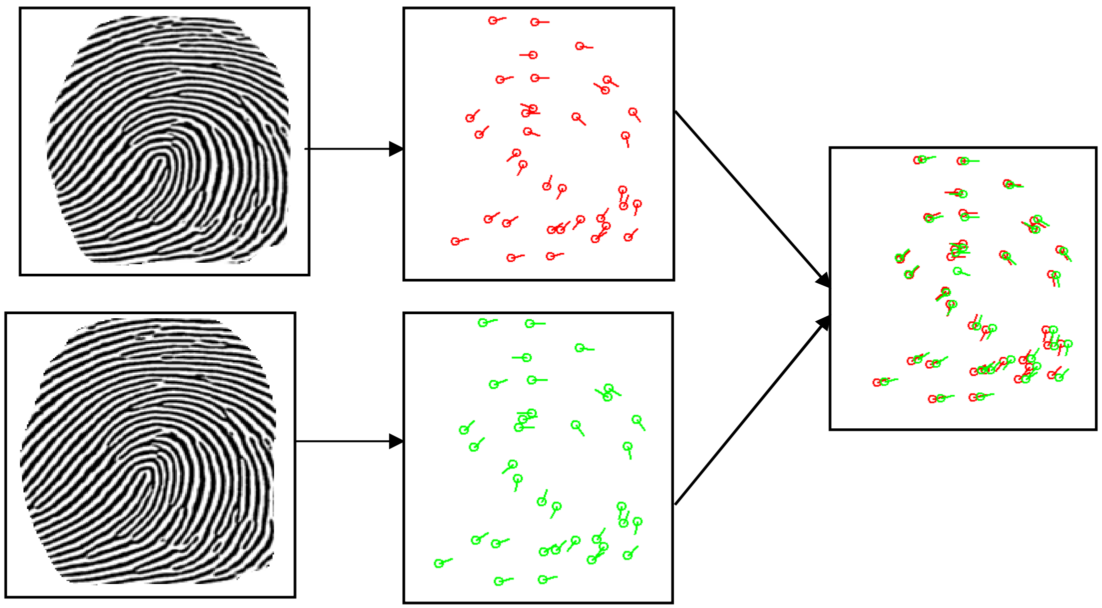


Outline of Talk

- **Fingerprint Identification using a Minutiae Tree**
  - Challenges & Motivation
  - Previous Classification and Indexing approaches
  - Our Method: Tree building and Searching
  - Handling Errors in Binning
  - Performance Analysis using Synthetic Datasets
  - Statistical Study of Minutiae Matching
    - **Using SVM to eliminate False Matches**
    - **Feature Selection**

Motivation

Getting optimal features and thresholds for matching & indexing minutiae points

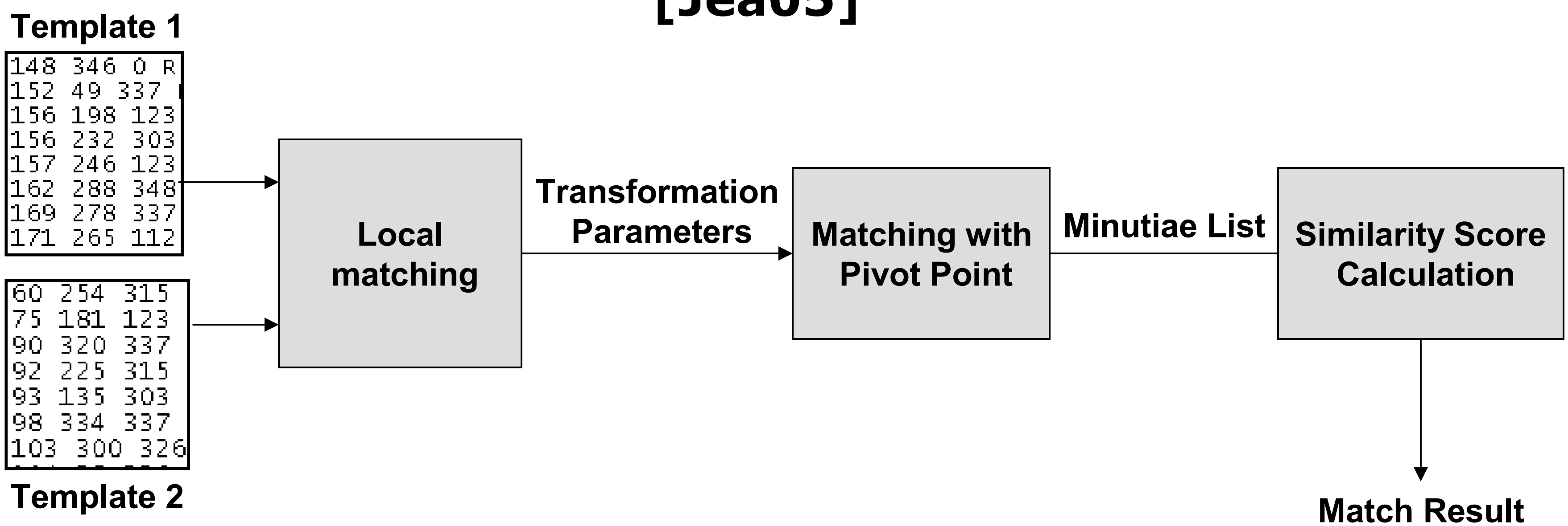


- **Matching between 2 minutiae points**
  - **Features ?**
  - **Thresholds ?**
  - **Score ?**

Proposed Approach

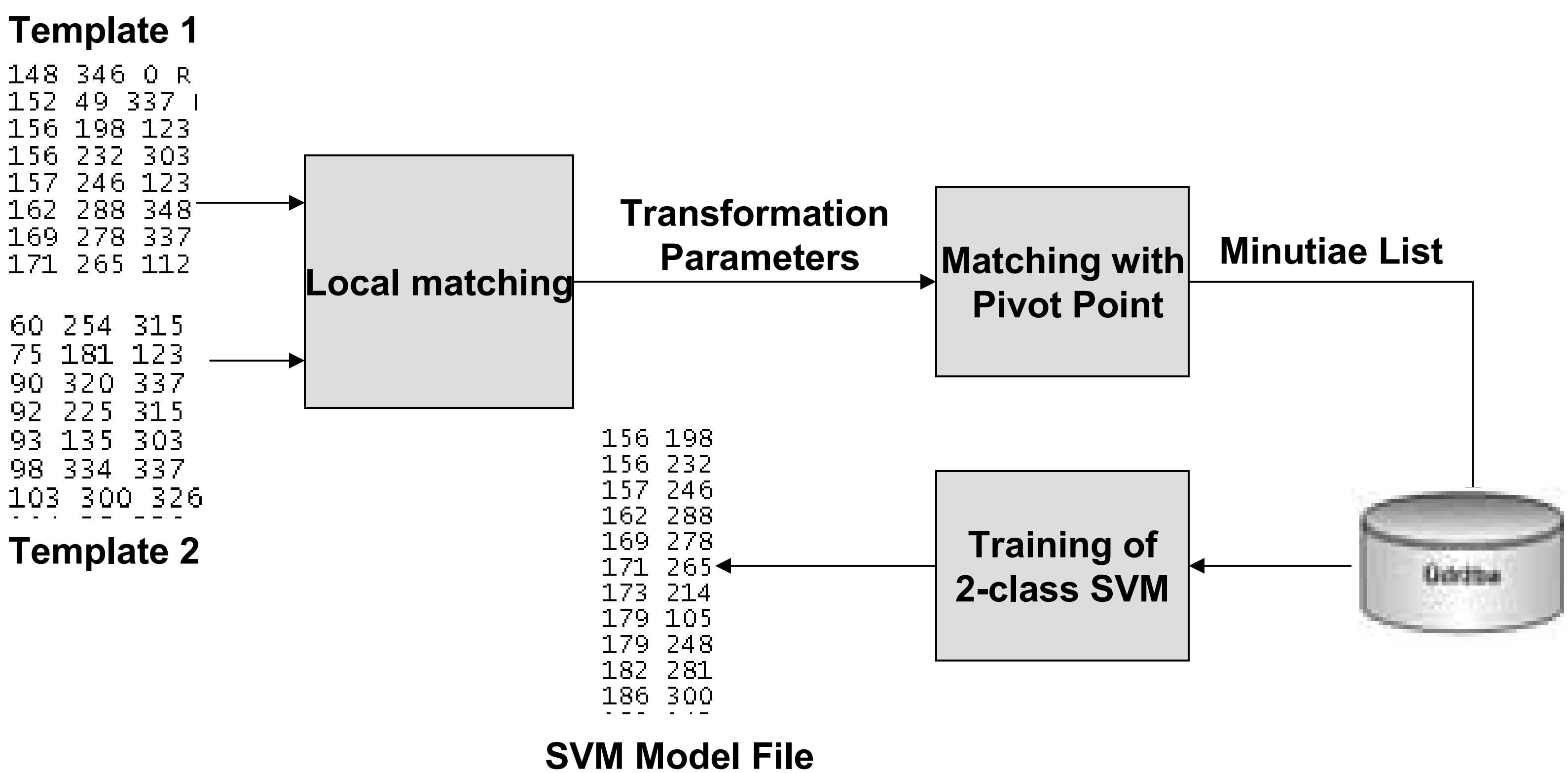
- Matched minutiae pairs are extracted from fingerprint pairs belonging to same and different users.
  - Genuine Matched Minutia vs. Impostor Matched minutia
- Best Matching Features are selected using a Feature Selection Algorithm
  - With respect to pivot points
  - With respect to neighboring points
- SVM is trained (2 class problem) for classifying match pairs: genuine vs impostor
  - Eliminate Imposter matching pairs
  - Update matching score

Basic Two Stage Minutiae Based Recognition System [Jea05]

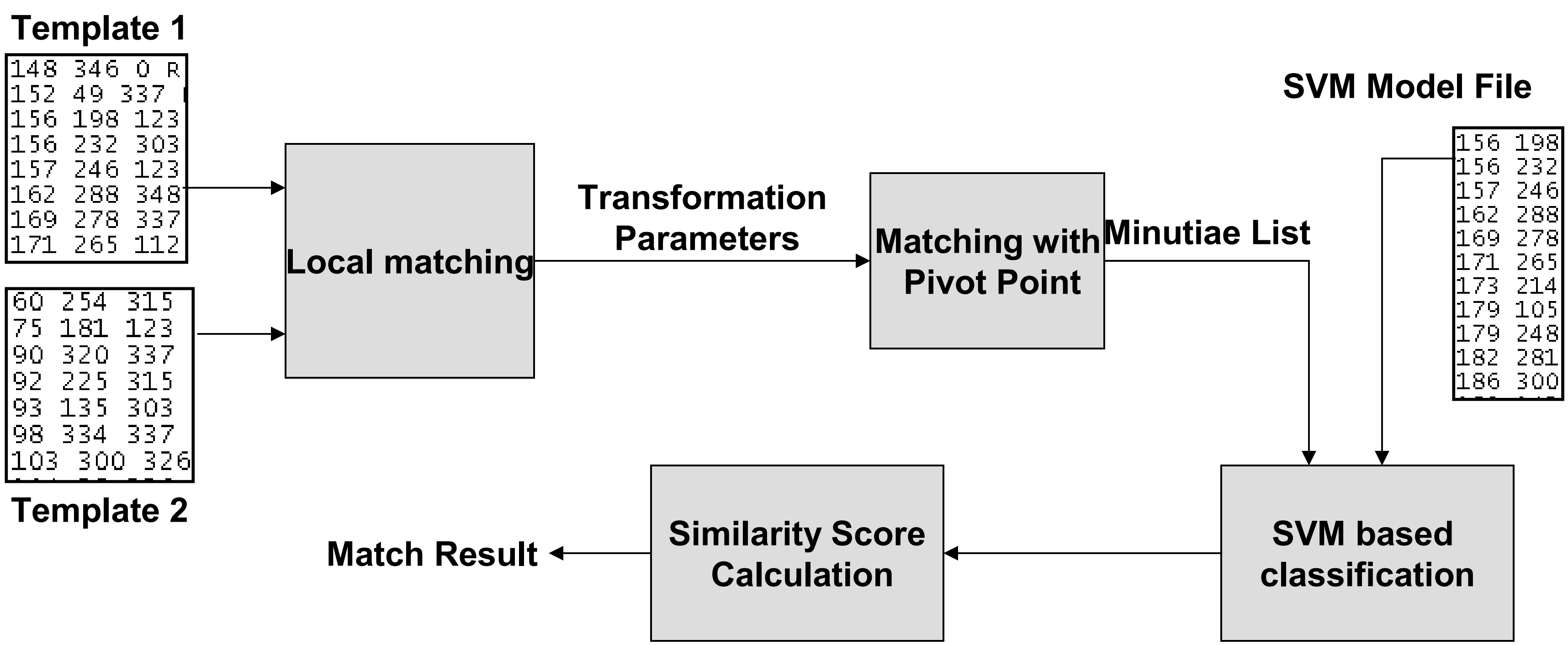


1. Compare template to get most likely transformation (pivot point)
2. Center on pivot point and compare minutiae pairs with respect to pivot
3. Score calculation based on individual minutiae comparision scores

Generating SVM model using Matched Minutiae Pairs



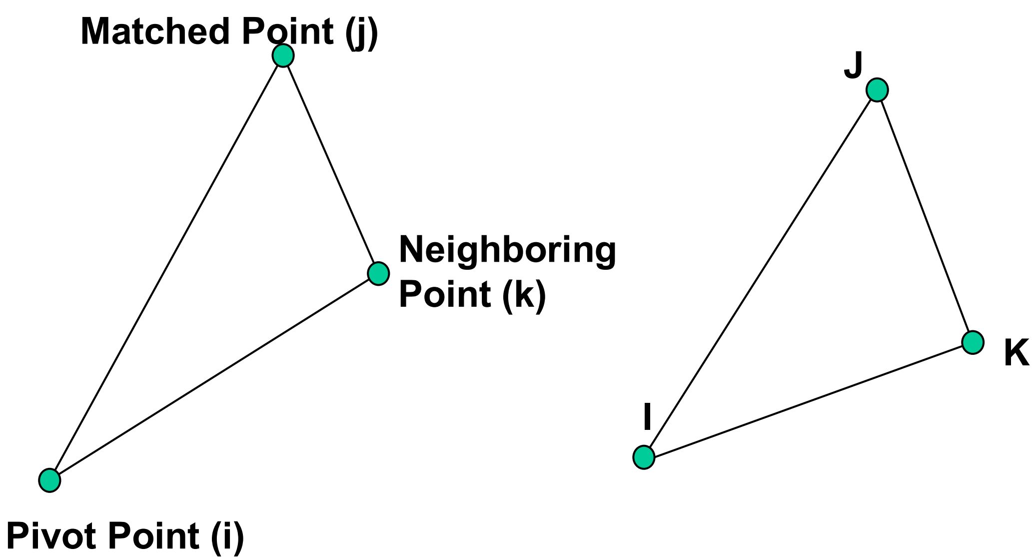
Test phase : SVM is used to eliminate false matches



Dataset and Features used

- FVC 2002 DB1 – 100 users * 8 prints
- Divided into train and test sets – 50 users each

- Matching pairs set – C(8,2) pairs * 50 users = 1400 comparisons , ~31K matched pairs
- Non-matching pairs set – C(50,2) pairs = 9800 comparisons = ~23K matched pairs



- 2 feature set
  - d_{jk}/d_{JK}
  - $(\theta_{jik} – \theta_{JIK})$
- 5 feature set
  - d_{ij}/d_{IJ}, d_{ik}/d_{IK}, d_{jk}/d_{JK},
  - $(\theta_{jik} – \theta_{JIK})$, $(\alpha_{ij} - \alpha_{IJ})$

Cross Validation Results

- **LibSVM used with Radial Basis Kernel**
- **5-fold cross validation**

Number of features	Ratio of genuine to imposter points	Cross – validation accuracy
2	57 : 43	64 %
5	57 : 43	67.05%

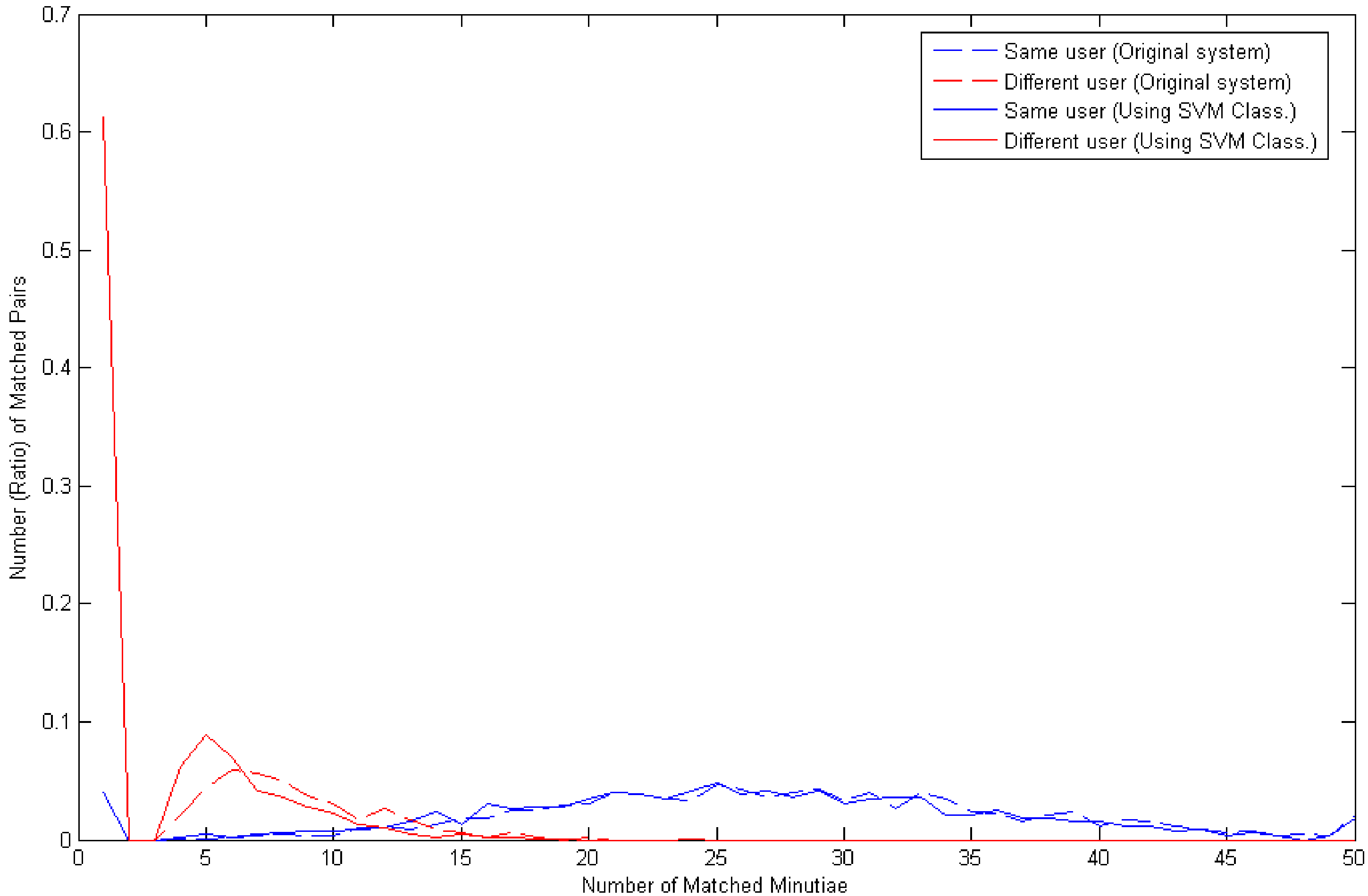
Results on Test Set

	No. of Fingerprint Pairs Compared	Total Matched Point Pairs	Rejected Matched Points	Accepted Matches
Same User	1400	37705	1831	95.14%
Different User	1225	3991	722	81.91%

Effect on Error Rate

- We have used the same scoring mechanism as original system.
- A slight decrease in accuracy has been observed
- If we use minutiae count, this method gives a slightly better result

	Using Minutiae Count		Score with Area & Individual Scores [Jea05]	
	Equal Error Rate (EER)	Improvement	Equal Error Rate (EER)	Improvement
Original System	7.59%	+0.31%	2.04%	-0.24%
Using SVM Classifier	7.28%		2.28%	



Distribution of number of matched minutiae pairs for original system, and using SVM

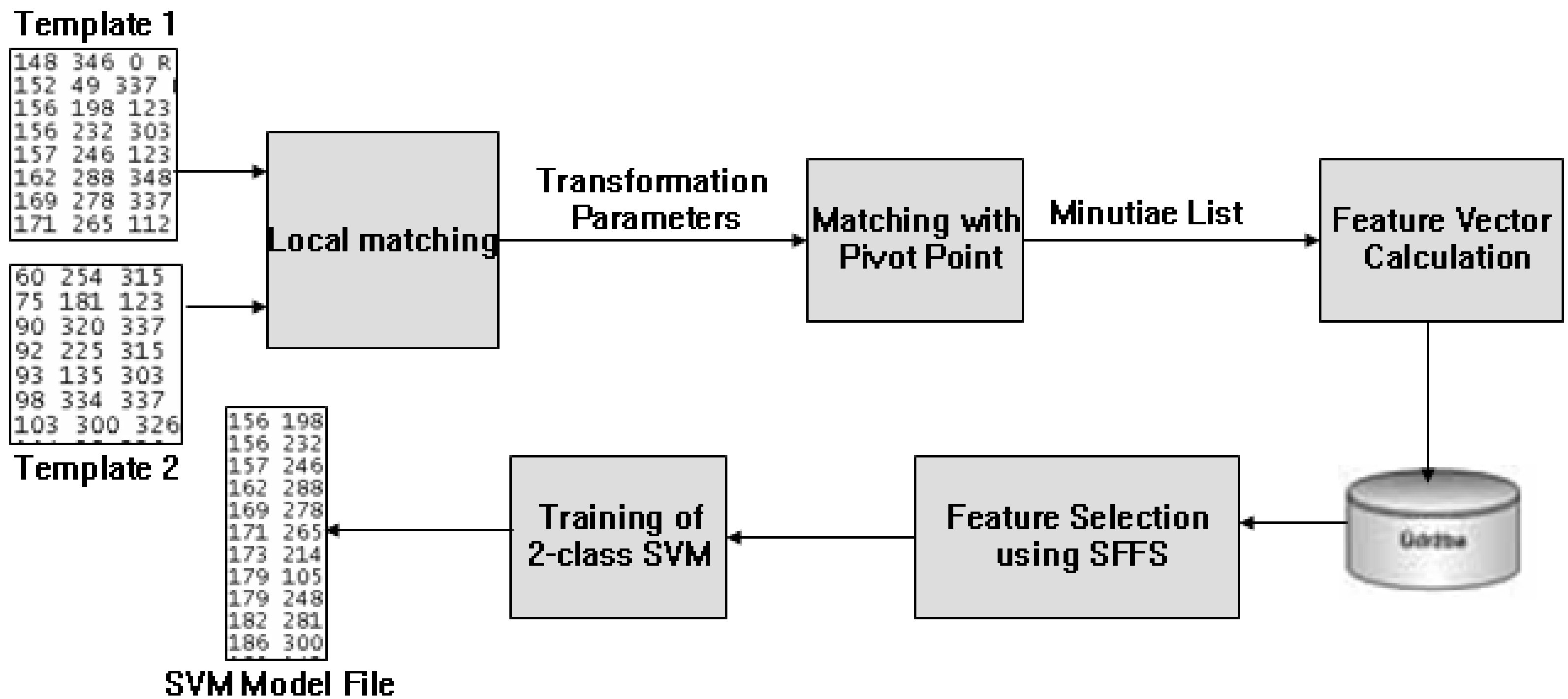
Feature Selection

- **Optimal feature set might not be the largest feature set**
  - Some feature might confuse the classifier
  - Larger feature set leads to greater overload

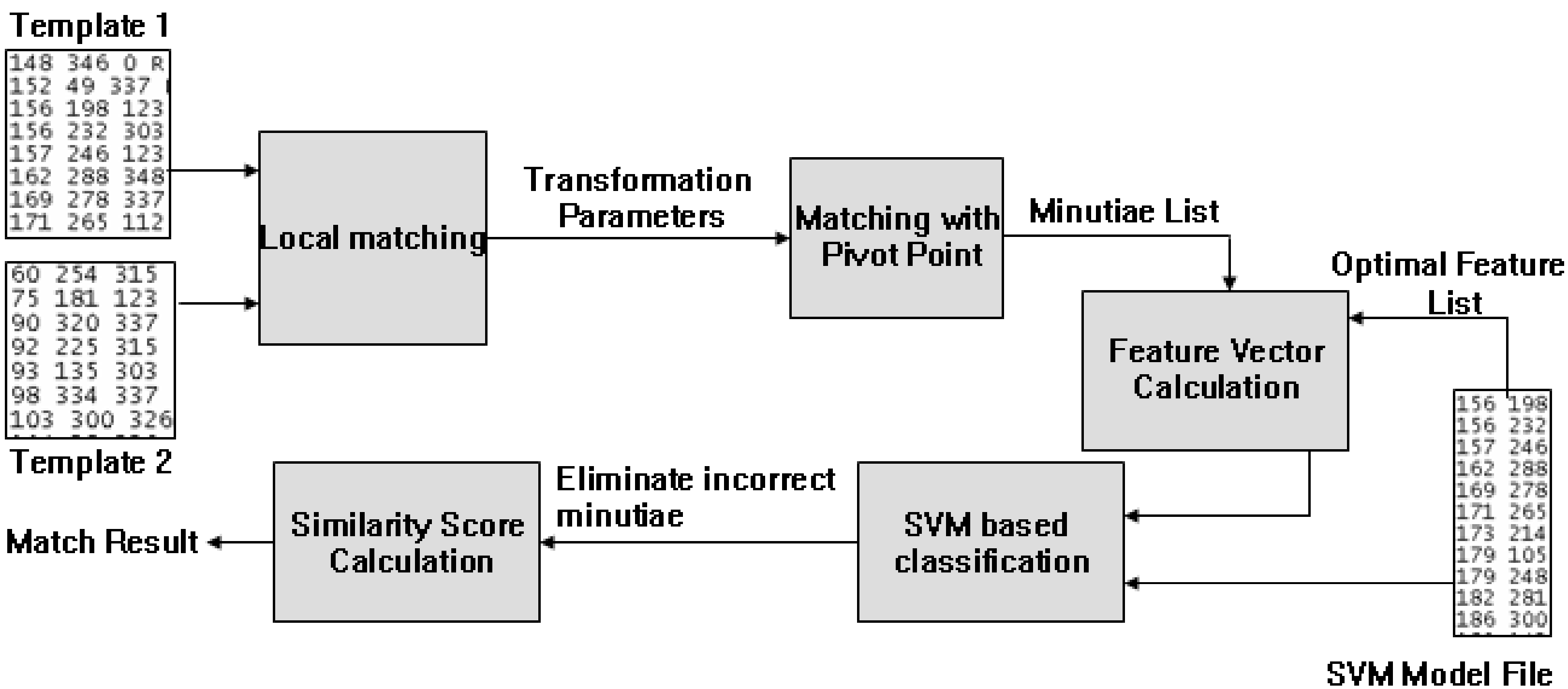
- **Stochastic Floating Forward Search(SFFS)[Pudil94] used**
  - Derive optimal feature set from arbitrary features
  - Starting point: Empty feature set. Ending Point: Target number of features
  - Adds features one at a time, check classification accuracy
  - Each stage, check if dropping a feature will increase accuracy

- **Use new optimal feature set for train and test**

Training: Feature Selection to derive Optimal Feature Set



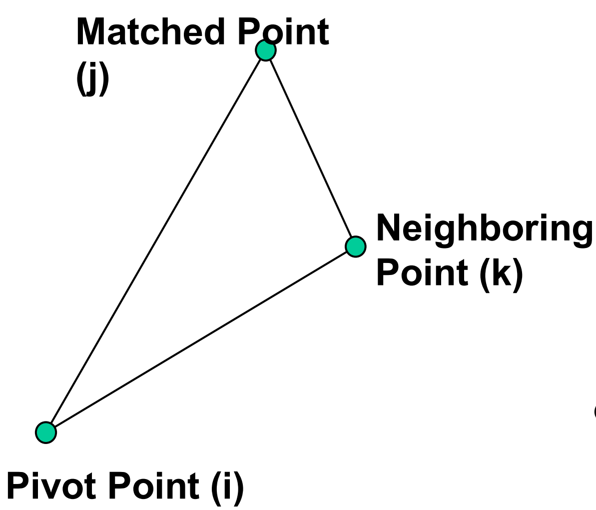
Test Mode: Classification using Optimal Features

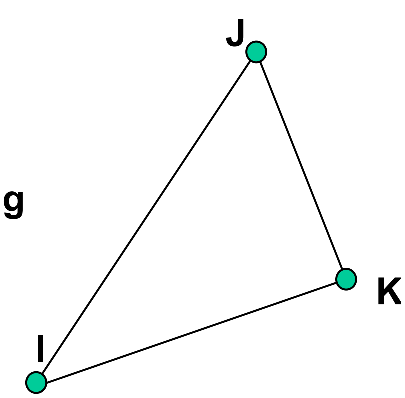


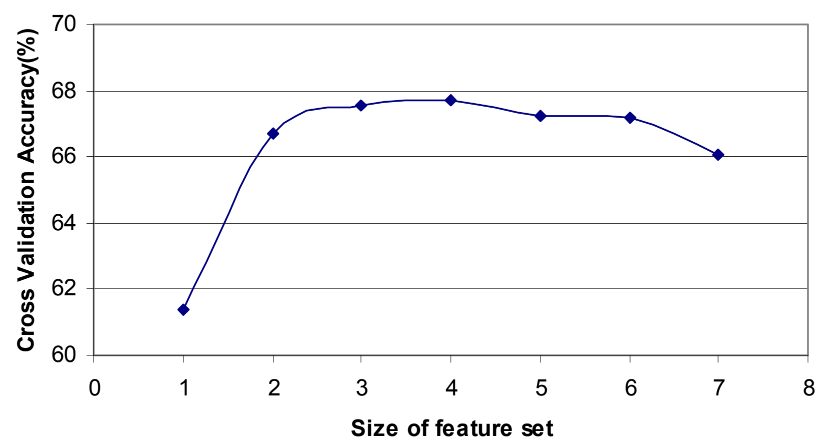
Training Results

Feature Number	Feature
1	d_{ij}/d_{IJ}
2	d_{ik}/d_{IK}
3	d_{jk}/d_{JK}
4	$(\theta_{jik} - \theta_{JIK})$
5	$(\alpha_{ij} - \alpha_{IJ})$
6	$(\theta_{ijk} - \theta_{IJK})$
7	$(\alpha_{ik} - \alpha_{IK})$

No of Target Features	Features Selected	Cross Validation Accuracy (%)
1	3	61.36
2	3,4	66.72
3	3,4,5	67.57
4	2,3,4,5	67.72
5	1,2,3,4,5	67.24
6	1,2,3,4,5,7	67.17
7	All	66.09

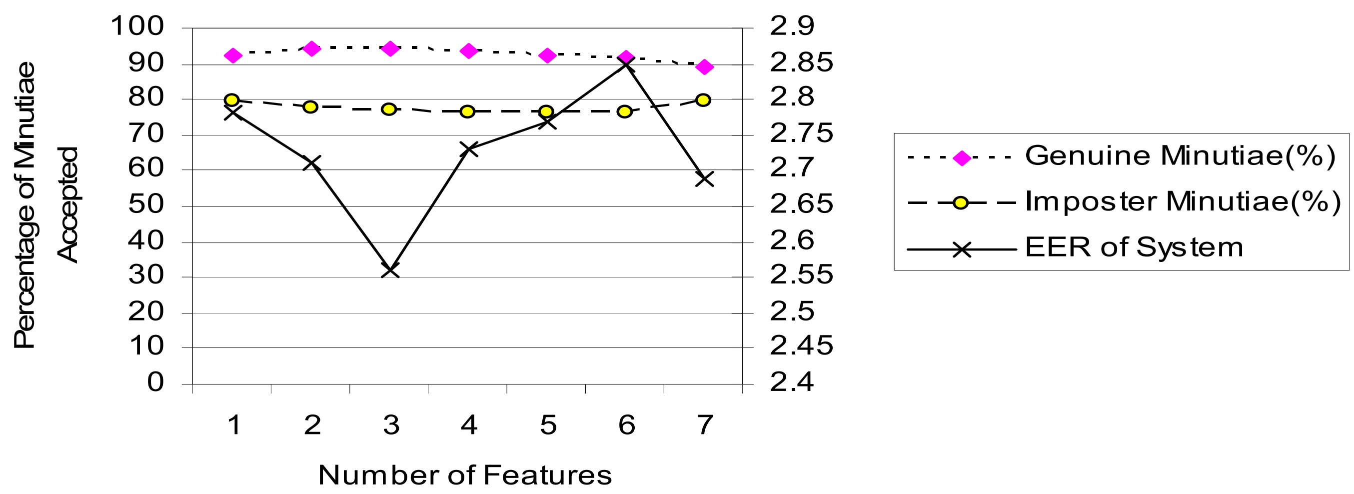






Results on Test Set

No of Features	Percentage of Genuine Matched Pairs Accepted	Percentage of Imposter Minutiae Pairs Accepted	Equal Error Rate
1	92.39	79.76	2.78 %
2	93.92	77.56	2.71 %
3	94.07	76.64	2.56 %
4	93.27	76.44	2.73 %
5	92.41	76.19	2.77 %
6	91.87	76.27	2.85 %
7	89.01	79.38	2.69 %



Contributions and Future Work

- **Developed a fingerprint indexing system based on minutiae binning**
  - **Scalable on large datasets**
    - **Fast enrollment time. Constant time per enrolled template.**
    - **Search time independent of index size.**
  - **Binning process allows for some amount of distortion without loss of accuracy**
  - **Can handle different feature sets – additional features could improve performance**
  - **Variable number of levels used for indexing**
    - **Bounds of the system could be set depending on number of levels indexed**
  - **Verification algorithm optional – Inherent verification provided, additional verifier could be added to reduce incorrect matches**

Contributions and Future Work

- **Developed a classification based minutiae matcher to eliminate false matches**
  - We have shown how SVM classifier can be used to eliminate spuriously matched minutiae
  - Addition of a feature selection algorithm improves performance of the SVM
  - Can study effectiveness of different feature sets for matching & indexing
- **What next? - To increase accuracy**
  - Statistical study of feature values to set optimal thresholds
  - Score generation based on similarity / number of minutiae trees matched
  - Incorporating the SVM classifier into the indexing system to eliminate incorrect matches

Thank you.