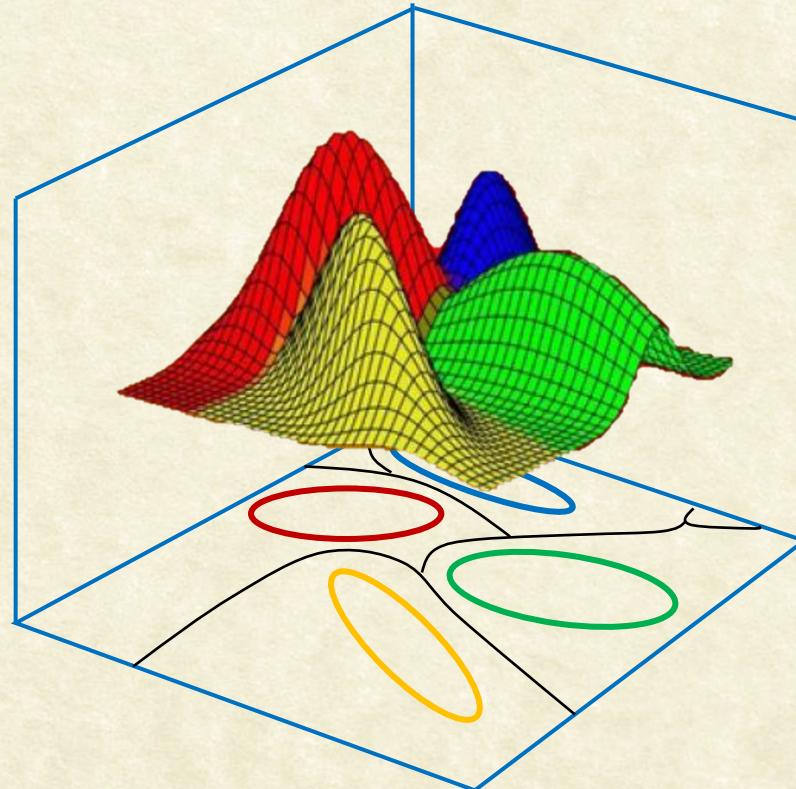




CS7.403: Statistical Methods in AI

Monsoon 2022:
Template Matching



Anoop M. Namboodiri

Biometrics and Secure ID Lab, CVIT,
IIIT Hyderabad



String Matching

Matching Sequences



Comparing Strings

- How to transform **file** to **farm** ?
 - **file** → **fil**m**** → **f**i**r**m**** → **f**a**r**m****
 - Edit distance = 3
- Given two words, how to compute the edit distance?
- Operations:
 - Add, Remove, Replace characters
 - Find the optimal set of operations

Dynamic Programming

	P	U	N	J	A	B
P	0	1	2	3	4	5
A	1	1	2	3	3	4
N	2	2	1	2	3	4
A	3	3	2	2	2	3
J	4	4	3	2	3	3
I	5	5	4	3	3	4



Intelligent String Matching

- We assumed each operation cost 1 unit
- In transliteration, some costs are less than others
 - Replacing '*oo*' with '*u*' or '*ee*' with '*i*' is not very costly: 0.1
 - Replacing 'a' with 'e' is costlier: 1.0
 - Replacing 'a' with 'x' is even costlier: 4.0
- Costs can depend on the applications
 - In typing '*e*' and '*w*' are closer than '*e*' and '*i*'.
- We can incorporate these costs into string matching



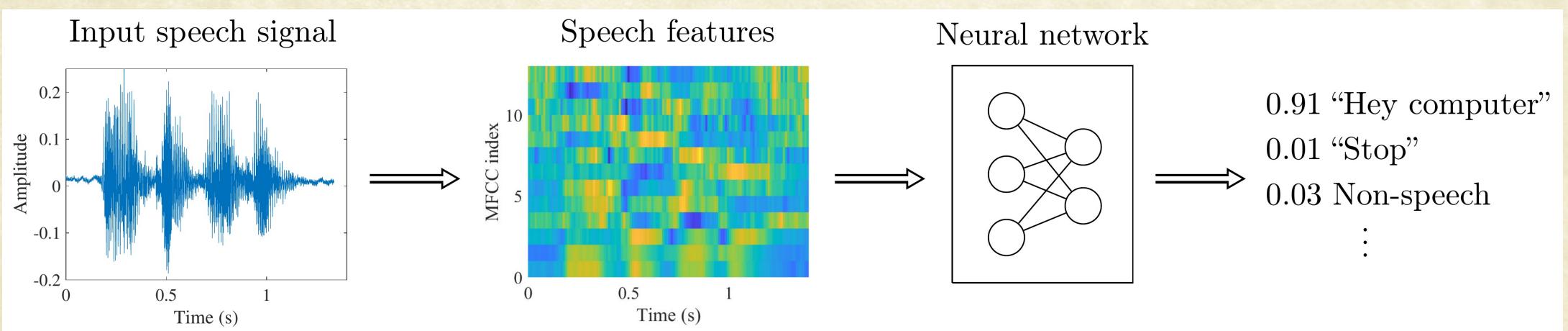
Application: Gene Matching

- DNAs are long sequences of **nucleotides** with A (adenine), T (thymine), C (cytosine) and G (guanine) bases.
 - Human genome is 3.2 billion nucleotides long
 - Divided into sub-sequences called chromosomes (23 pairs)
- Part of a chromosome that code for a protein is called a gene
- Problems
 - Given a protein, find the genes that code for it in a DNA sequence
 - Given two DNA sequences, find the similarity between them
 - Given a set of DNA sequences, find common genes in them
- All are sequence matching problems [**Basic Local Align. Search Tool**]



Other Sequences

- Matching feature vectors of variable length
- Word Spotting
 - Speech
 - Handwriting



పుత్రి సెలో మెయిట్ 15 రూజులకు
సుక్క పక్క మనియు , సింగావు 15 రూజులకు
క్రీప్ పక్క మనియు లెద్దు . సుక్క పక్క మనియు
బెస్ట్ కాయిలాడ్సు . క్రీప్ పక్క మనియు
చీకాటి వ్హిట్రియస్ రూజులు
ఎద్దులు మెయిలుయస్ ల్రూయస్ ఫ్లూమ్
ఇంగులను చెప్పు మిలీనిచూ హెచ్చులిమి , ఇంగులు

Search: "Roojulaku" in Telugu



Summary

- Matching sequences or variable length feature vectors can help in various applications
- Can be divided into several sub-categories
 - Global sequence matching
 - Local alignment
 - Sequence set alignment
 - Search for sequences (Word-Spotting)
- Efficient Methods exists
 - Dynamic Time Warping (DTW), Levenshtein dist., Needleman Wunsch
 - Smith Waterman Algorithm (local sequence alignment)



Questions?



Point Set Matching

The Hungarian Algorithm



The Assignment Problem

- You have 4 workers and 4 jobs.
- The cost of assigning each job to specific workers are as in the table.
- Find the optimal (least cost) assignment of jobs to workers.

	J_1	J_2	J_3	J_4
W_1	36	26	28	52
W_2	28	38	21	41
W_3	15	14	12	21
W_4	42	26	34	48



The Munkres Assignment Algorithm

1. Subtract Row-Min
2. Subtract Column-Min
3. Assign if possible
4. Subtract min(uncovered)
add to intersections
5. Assign if possible
6. Else repeat from step 4

	J_1	J_2	J_3	J_4
W_1	36	26	28	52
W_2	28	38	21	41
W_3	15	14	12	21
W_4	42	26	34	48



The Munkres Assignment Algorithm

1. Subtract Row-Min
2. Subtract Column-Min
3. Assign if possible
4. Subtract min(uncovered)
add to intersections
5. Assign if possible
6. Else repeat from step 4

	J_1	J_2	J_3	J_4
W_1	36	26	28	52
W_2	28	38	21	41
W_3	15	14	12	21
W_4	42	26	34	48



The Munkres Assignment Algorithm

1. Subtract Row-Min
2. Subtract Column-Min
3. Assign if possible
4. Subtract min(uncovered)
add to intersections
5. Assign if possible
6. Else repeat from step 4

	J_1	J_2	J_3	J_4
W_1	10	0	2	26
W_2	7	17	0	20
W_3	3	2	0	9
W_4	16	0	8	22



The Munkres Assignment Algorithm

1. Subtract Row-Min
2. Subtract Column-Min
3. Assign if possible
4. Subtract min(uncovered)
add to intersections
5. Assign if possible
6. Else repeat from step 4

	J_1	J_2	J_3	J_4
W_1	10	0	2	26
W_2	7	17	0	20
W_3	3	2	0	9
W_4	16	0	8	22



The Munkres Assignment Algorithm

1. Subtract Row-Min
2. Subtract Column-Min
3. Assign if possible
4. Subtract min(uncovered)
add to intersections
5. Assign if possible
6. Else repeat from step 4

	J_1	J_2	J_3	J_4
W_1	7	0	2	17
W_2	4	17	0	11
W_3	0	2	0	0
W_4	13	0	8	13



The Munkres Assignment Algorithm

1. Subtract Row-Min
2. Subtract Column-Min
3. Assign if possible
4. Subtract min(uncovered)
add to intersections
5. Assign if possible
6. Else repeat from step 4

	J_1	J_2	J_3	J_4
W_1	7	0	2	17
W_2	4	17	0	11
W_3	0	2	0	0
W_4	13	0	8	13



The Munkres Assignment Algorithm

1. Subtract Row-Min
2. Subtract Column-Min
3. Assign if possible
4. Subtract min(uncovered)
add to intersections
5. Assign if possible
6. Else repeat from step 4

	J_1	J_2	J_3	J_4
W_1	7	0	2	17
W_2	4	17	0	11
W_3	0	2	0	0
W_4	13	0	8	13



The Munkres Assignment Algorithm

1. Subtract Row-Min
2. Subtract Column-Min
3. Assign if possible
4. Subtract min(uncovered)
add to intersections
5. Assign if possible
6. Else repeat from step 4

	J_1	J_2	J_3	J_4
W_1	3	0	2	13
W_2	0	17	0	7
W_3	0	6	4	0
W_4	9	0	8	9



The Munkres Assignment Algorithm

1. Subtract Row-Min
2. Subtract Column-Min
3. Assign if possible
4. Subtract min(uncovered)
add to intersections
5. **Assign if possible**
6. Else repeat from step 4

	J_1	J_2	J_3	J_4
W_1	3	0	2	13
W_2	0	17	0	7
W_3	0	6	4	0
W_4	9	0	8	9



The Munkres Assignment Algorithm

1. Subtract Row-Min
2. Subtract Column-Min
3. Assign if possible
4. Subtract min(uncovered)
add to intersections
5. Assign if possible
6. Else repeat from step 4

	J_1	J_2	J_3	J_4
W_1	3	0	2	13
W_2	0	17	0	7
W_3	0	6	4	0
W_4	9	0	8	9



The Munkres Assignment Algorithm

1. Subtract Row-Min
2. Subtract Column-Min
3. Assign if possible
4. Subtract min(uncovered)
add to intersections
5. Assign if possible
6. Else repeat from step 4

	J_1	J_2	J_3	J_4
W_1	1	0	0	11
W_2	0	19	0	7
W_3	0	8	4	0
W_4	7	0	6	7



The Munkres Assignment Algorithm

1. Subtract Row-Min
2. Subtract Column-Min
3. Assign if possible
4. Subtract min(uncovered)
add to intersections
5. Assign if possible
6. Else repeat from step 4

	J_1	J_2	J_3	J_4
W_1	1	0	0	11
W_2	0	19	0	7
W_3	0	8	4	0
W_4	7	0	6	7



The Munkres Assignment Algorithm

1. Subtract Row-Min
2. Subtract Column-Min
3. Assign if possible
4. Subtract min(uncovered)
add to intersections
5. Assign if possible
6. Else repeat from step 4

	J_1	J_2	J_3	J_4
W_1	36	26	28	52
W_2	28	38	21	41
W_3	15	14	12	21
W_4	42	26	34	48

Cost: $28 + 28 + 21 + 26 = 103$



The Munkres Assignment Algorithm

1. Subtract Row-Min
2. Subtract Column-Min
3. Assign if possible
4. Subtract min(uncovered)
add to intersections
5. Assign if possible
6. Else repeat from step 4

- What if there are more jobs than workers?
- Time Complexity:
 - $O(n^2) * n = O(n^3)$



Fingerprint Matching

Matching Point Sets



Fingerprint Matching

Find the similarity between two fingerprints



Fingerprints from the same finger

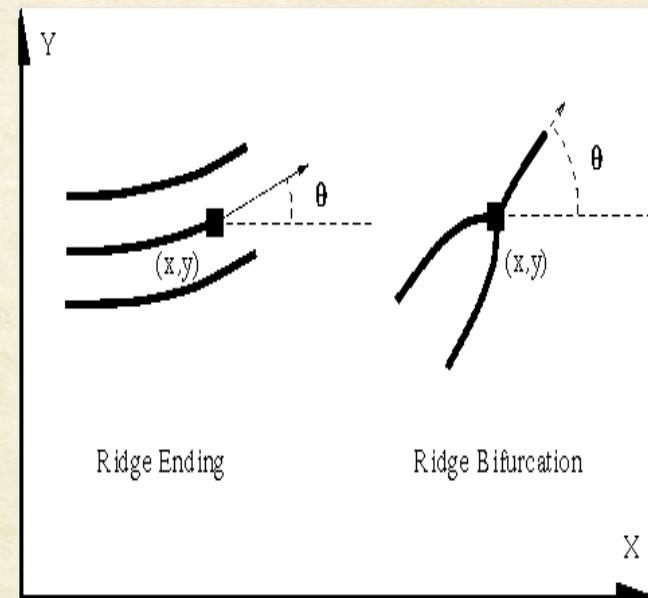
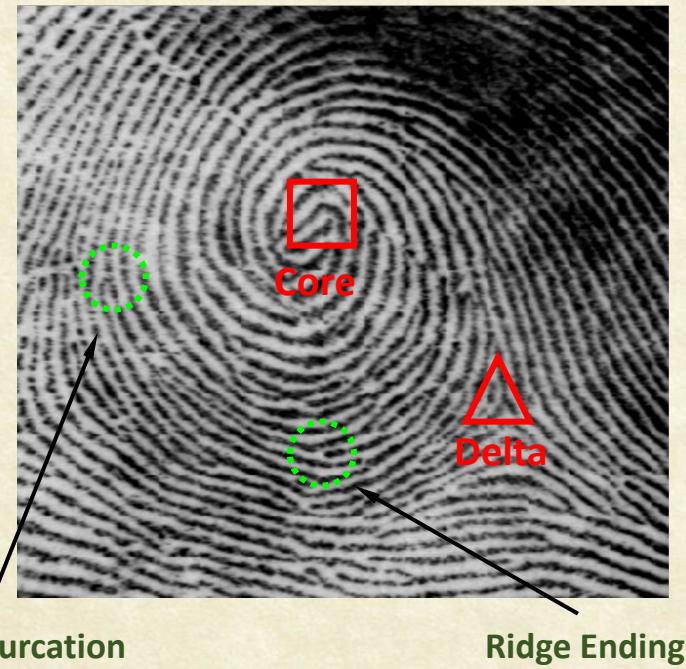


Fingerprints from two different fingers



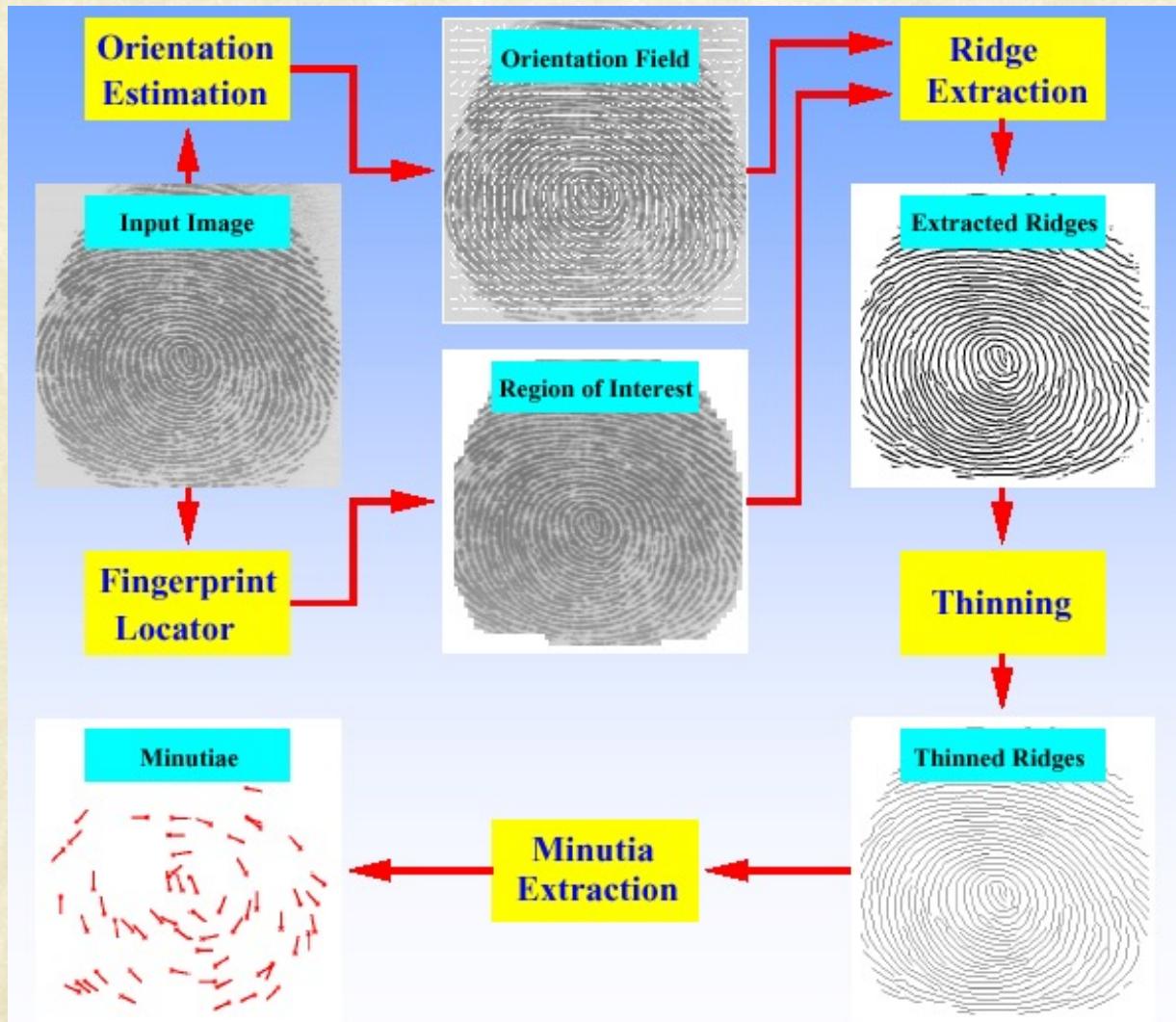
Fingerprint Representation

- Local ridge characteristics (minutiae): ridge ending and ridge bifurcation.
- Singular points: ridge orientation tendency not continuous.





Minutiae Extraction



Input live-scan image

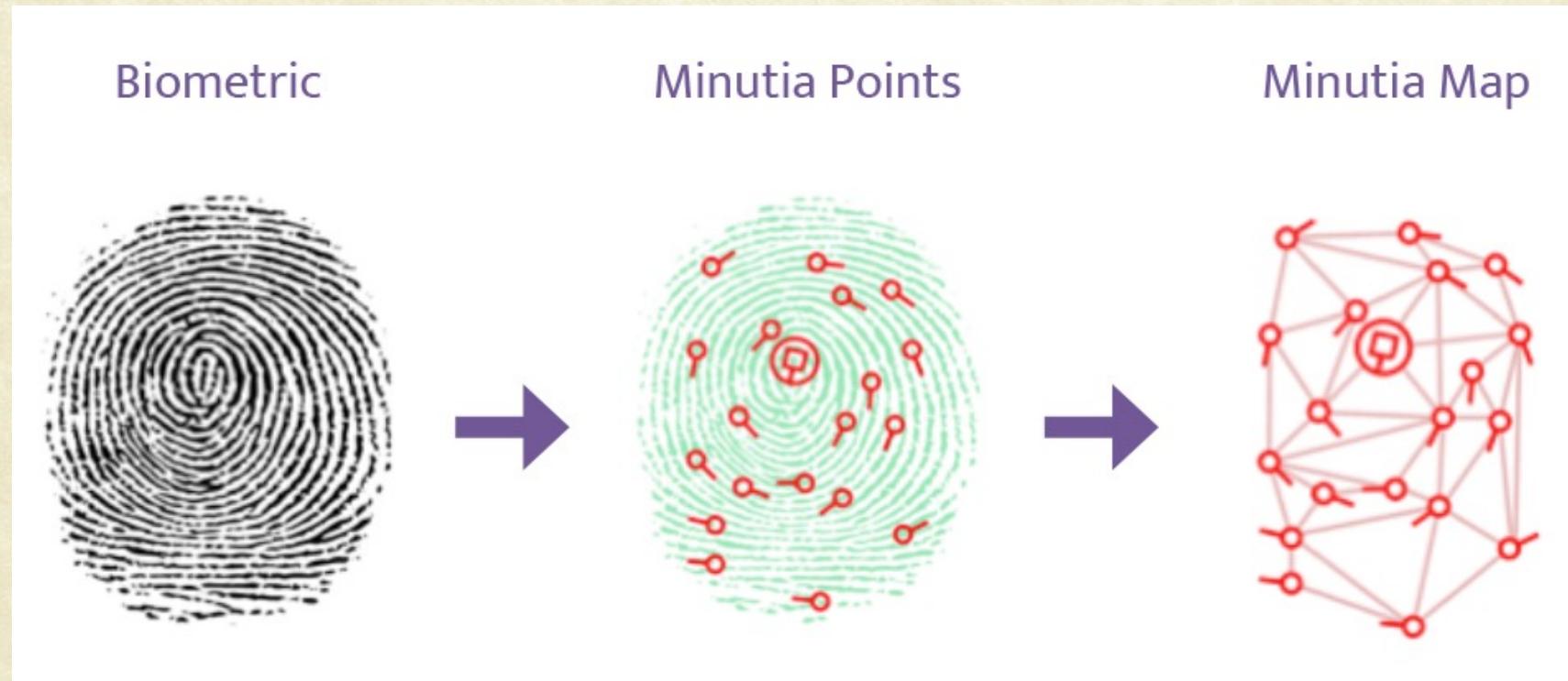


Extracted minutiae



Fingerprints

- Feature Representation of an individual's biometric trait
- Multiple templates are often stored to account for variations observed in practice

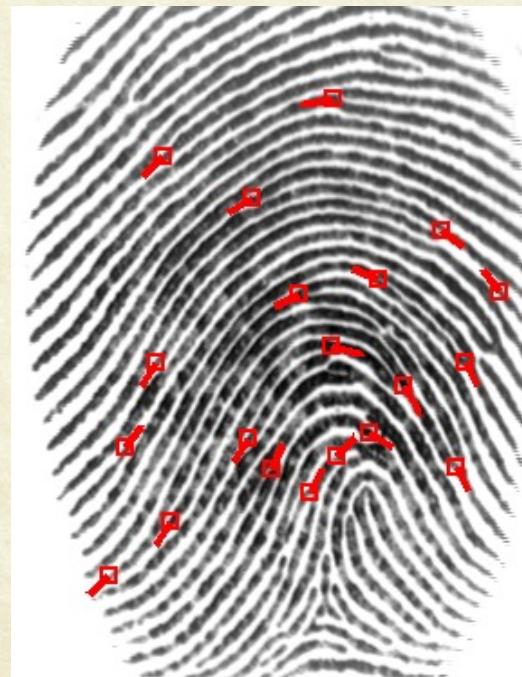




Fundamental Premise

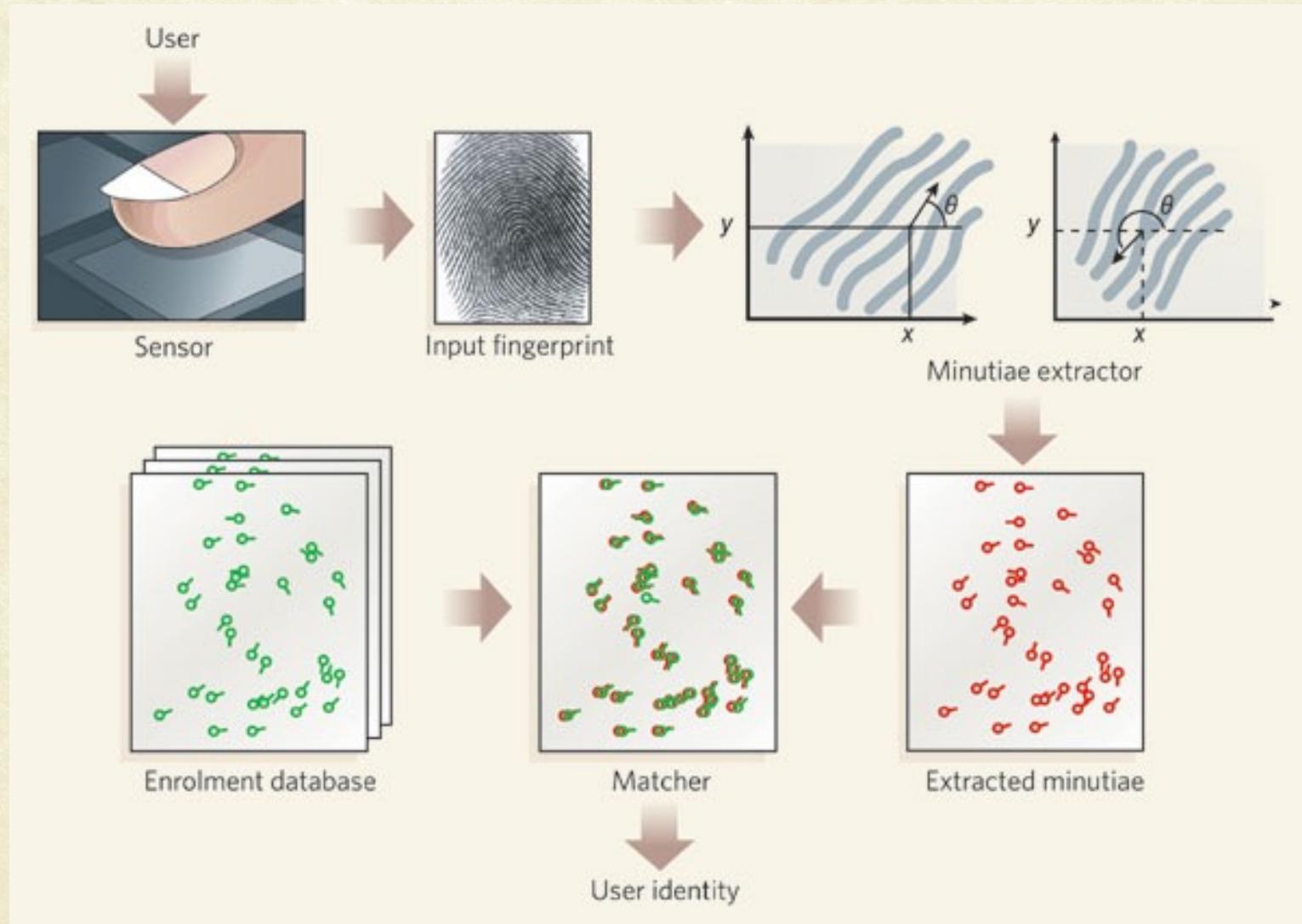
Minutiae distributions are **unique & permanent!**

- Intra-class similarity is extremely large
- Inter-class similarity is extremely small





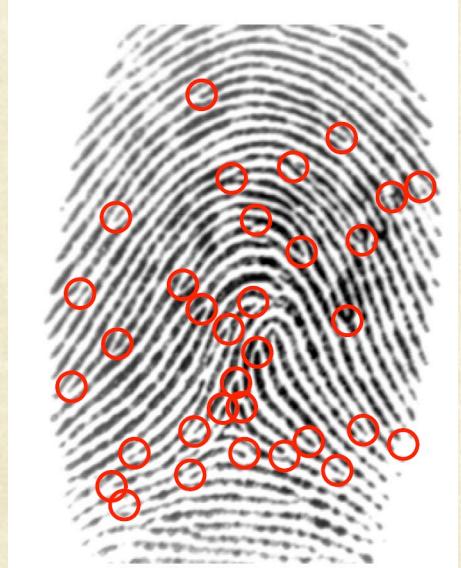
Biometric Recognition System



Enrolment and Authentication



Alignment



Three different impressions of the same finger

- Alignment based on
 - High-curvature points
 - Focal Points

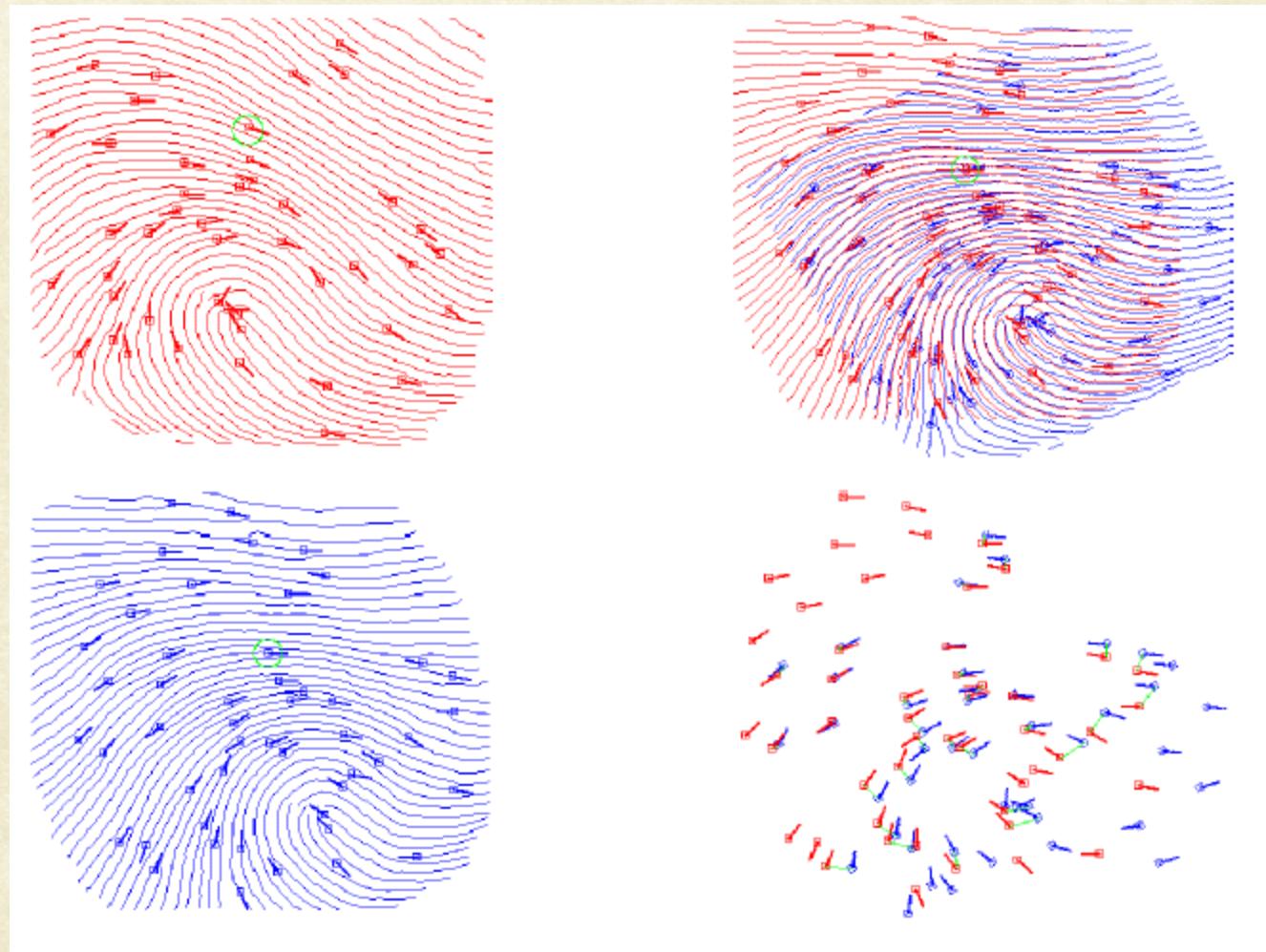


Two different impressions of the same finger using a Digital Biometrics scanner.

Non-rigid Deformation



Minutiae Matching Result





Matching Scores



(a)



(b)



(c)

$S_{ab} = 97$; $S_{bc} = 5$; $S_{ac} = 2$



Point-Set Matching

- Point set matching: Extension of sequence matching to 2D
- More degrees of freedom
 - Global alignment to start with (rot, trans)
 - Fine alignment using candidate matching
 - Non-rigid deformations
- Useful in several applications
 - Fingerprint Matching
 - Search for deformable objects in images (humans)
 - Pose Estimation



Questions?