

Template Matching

- Einleitung
 - Database Clustering
 - Minutia-Point Based Algorithms
 - Filterbank Based fingerprint matching
 - Iris matching
-

René Zorn

Elmar Weber

Robert Meiche



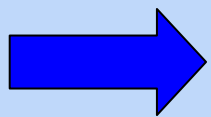
PADERBORN
CENTER FOR
PARALLEL
COMPUTING

- ▶ Einleitung
- ▶ Fingerabdrucksuche per Database Clustering
- ▶ Möglichkeiten der Klassifizierung
- ▶ Database Clustering
- ▶ 1. Phase: Offline Database Clustering
- ▶ 2. Phase: Online Query processing
- ▶ Ergebnisse

Unterscheidung zwischen 2 Modi von Systemen bei der Fingerabdruckerkennung

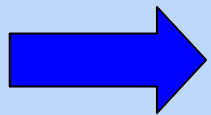
- ▶ Überprüfung der Identität (Authentifizierung)
- ▶ Identifikation

- ▶ Benutzer gibt seinen Fingerabdruck und seine Identitätsinformationen an das System
- ▶ Das System muss diese Merkmale nur noch verifizieren



Geringe Komplexität

- ▶ Nur der Fingerabdruck wird an das System übergeben
- ▶ Die Identität zu diesem Abdruck muss in der Datenbank gesucht werden



Hohe Komplexität

Verschieden Möglichkeiten, Fingerabdrücke zu repräsentieren

- ▶ Minutienpunkte
- ▶ Distanz- und Orientierungsvektoren
- ▶ Filterbasierte Merkmale

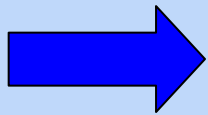
Problem: Fingerabdruck muss mit sehr vielen Einträgen (> 50 Mio.) verglichen werden.

Lösungsansatz: Klassifizierung der Datenmengen um die Anzahl der Vergleiche zu reduzieren

- ▶ Exklusive Klassifikation
- ▶ Kontinuierliche Klassifikation
- ▶ Database Clustering

Einteilung in Klassen (Bsp. Henry Classes)

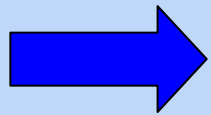
- 5 Klassen die sich nach menschlichen Merkmalen der Fingerabdrücke gliedern
- sehr ungleichmäßige Aufteilung der FA über die Klassen
- Im Durchschnitt muss jeder FA mit ca. 30% aller Templates über „Fine-Matching“ verglichen werden



Keine zufriedenstellenden Möglichkeiten der Optimierung der Datenbanksuche

Indexieren der FA über Trippel von Minutenpunkte

- Schnelleres vergleichen von Vektoren gegenüber „Fine-Matching“
- Effektivere DB-Suche da größere und bessere Klassifizierung



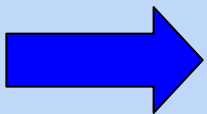
Sehr viel effizienter wie die Exklusive Klassifikation allerdings immer noch nicht schnell genug für große Datenbanken

Prinzip:

- ▶ Daten werden in Cluster eingeteilt
- ▶ Cluster haben Repräsentanten und sind wiederum klassifiziert

Vorteil:

- ▶ Ein FA muss nicht mit der gesamten Datenbank verglichen werden
- ▶ Einschränkung auf einen/mehrere Cluster



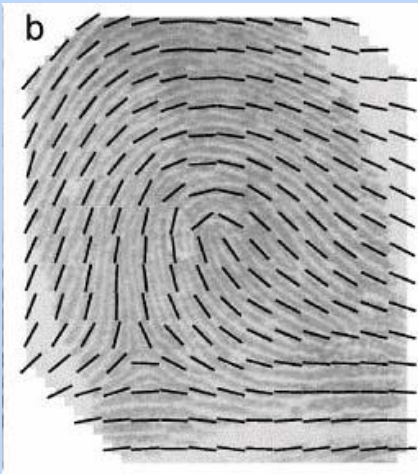
Weniger Vergleiche. Lässt sich dadurch auf größere Datenbanken anwenden.

- ▶ Gliederung in 2 Phasen
 - ▶ Offline DB Clustering
 - ▶ Online Query Processing
- ▶ Als Sucheigenschaft werden ein Orientierungs- (156 Einträge) und ein Average-Distanz-Vektor (1 Eintrag) definiert.

1. Phase: Offline DB Clustering

- ▶ Es gibt 2 Möglichkeiten der Merkmalsgewinnung für FA
- ▶ Globale Merkmale (Beschreibung der Flußstruktur, Orientierungsfeld)
- ▶ Lokale Merkmale (Bsp. Minutenpunkte)

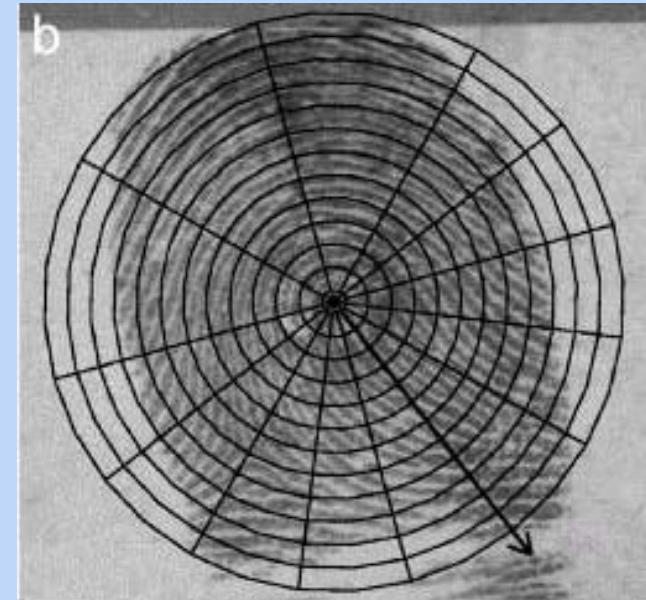
Dieses Verfahren benutzt globale Merkmale die in 2 Vektoren zusammengefasst werden.



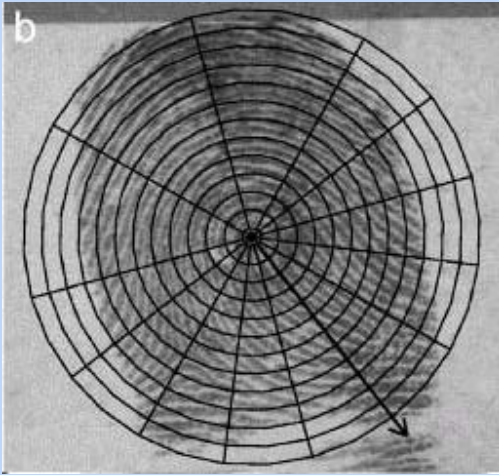
→ Orientierungsfeld (Abb. Links) wird in Sektoren eingeteilt
→ Referenzpunkt für die Ausrichtung und eine Referenz-Richtung für die Rotationsausrichtung werden ermittelt

2 Möglichkeiten das Orientierungsfeld zu erstellen:

- Uniform-spacing
- Non-uniform-spacing (siehe rechte Abb.)



Non-Uniform-Spacing (Berechnung des Vektors)



- ▶ Jeder Sektor von jedem Ring wird berechnet und im Orientierungsvektor angegeben
- ▶ Berechnung des j -ten Sektors des i -ten Rings wie folgt:

$$S_{i,j} = \{ (x, y) \mid (i-1)b + b_0 \leq r < i*b + b_0, \varphi_{j-1} \leq \varphi < \varphi_j, 1 \leq x \leq X, 1 \leq y \leq Y \}$$

$$(1 \leq i \leq E, 1 \leq j \leq F)$$

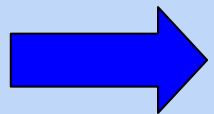
$(x_r, y_r) \longrightarrow$ Referenzpunkt

$$r = \sqrt{(x - x_r)^2 + (y - y_r)^2}$$

$$\varphi = \tan^{-1} \left(\frac{y - y_r}{x - x_r} \right) - \theta_r \bmod 2\pi$$

φ ist Rotationswinkel von Referenzrichtung θ_r

- ▶ Lokaler Kantenabstand: Abstand von 2 zentralen Punkten zweier benachbarter Kanten
- ▶ Eine weitere, wichtige Eigenschaft eines Fingerabdrucks, allerdings nicht so beständig wie der Orientierungsvektor

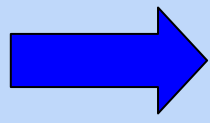


Zusätzliche Sucheigenschaft

- ▶ Orientierungsvektoren werden in sich nicht überschneidende Cluster eingeteilt
- ▶ Modifizierter K-Means Algorithmus

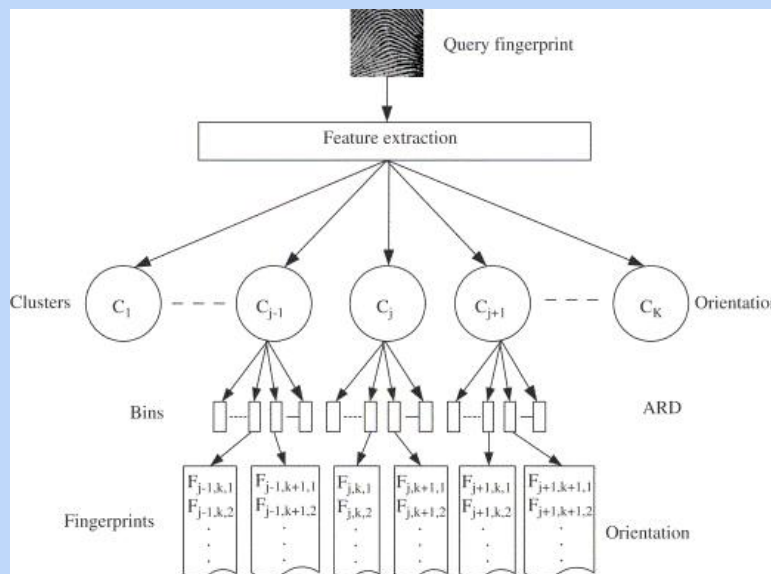
Wenn N Muster in K Cluster eingeteilt werden:

- ▶ Durchschnittliche Anzahl von vergleichen: $(K + N) / K$ (um einen entsprechenden Cluster zu finden auf dem dann die FA-Suche erfolgt)
- ▶ Wird minimiert wenn $K = \sqrt{N}$
- ▶ Um die Genauigkeit zu erhöhen werden mehrere ähnliche Cluster gesucht

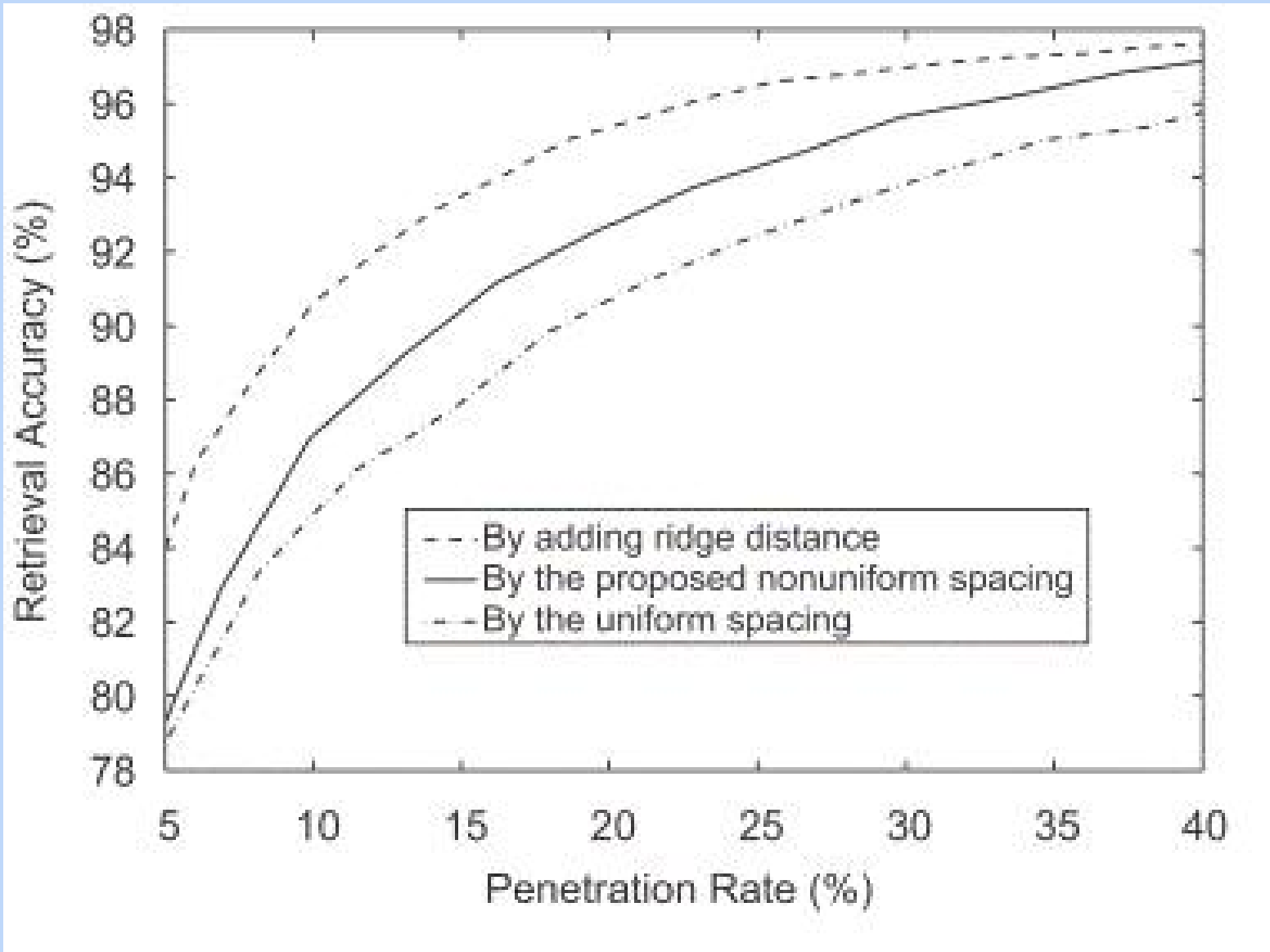

$$K = \tau \sqrt{N}, (1 \leq \tau \leq 3)$$

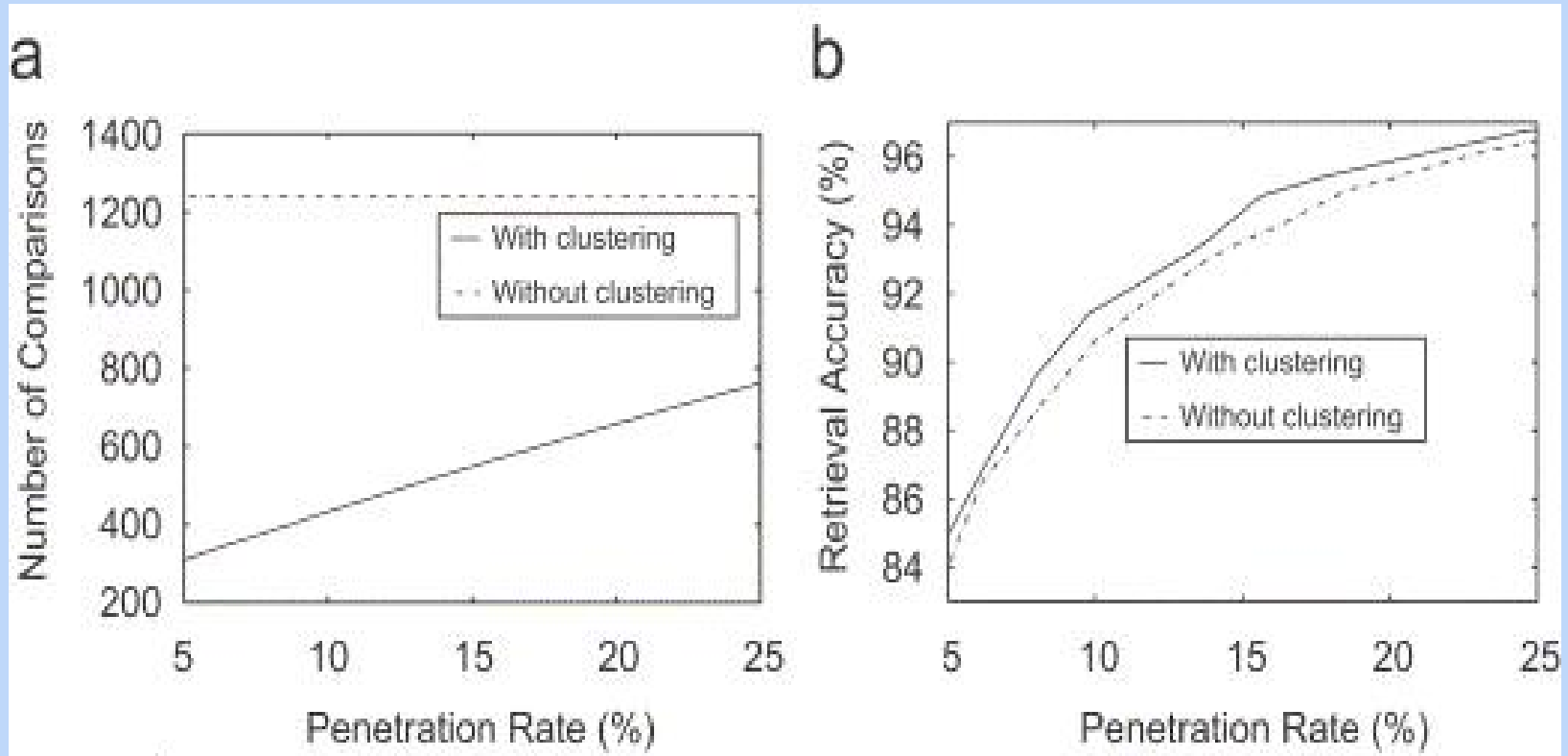
2. Phase: Online Query Processing

1. Vergleichen des Orientierungsvektors mit den Repräsentanten der Cluster
2. Durchsuchen der Cluster mit Hilfe der Average-Ridge-Distance
3. Vergleichen der Ergebnisse mit Hilfe des Orientierungsvektors



Ergebnisse (1)





Viele Dank für die Aufmerksamkeit!

Fragen???