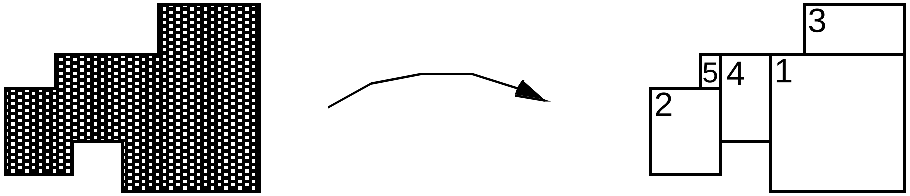
3.2.3 Überdeckungsmodell für 2D-Formen

- Idee [Jagadish. Proc. ACM Int. Conf. on Management of Data (SIGMOD) 1991]
  - Ähnlichkeitsmodell für 2D Objekte (leicht erweiterbar auf 3D)
  - Distanzfunktion: Flächeninhalt der symmetrischen Differenz zweier Formen.
  - Hier: Translations- und skalierungsinvariant, nicht jedoch rotationsinvariant.
  - Vorgehen: Repräsentation der Formen durch rechteckige Überdeckungen.
  - Speicherung der Rechtecksflächenmaßzahlen z.B. der Größe nach geordnet.



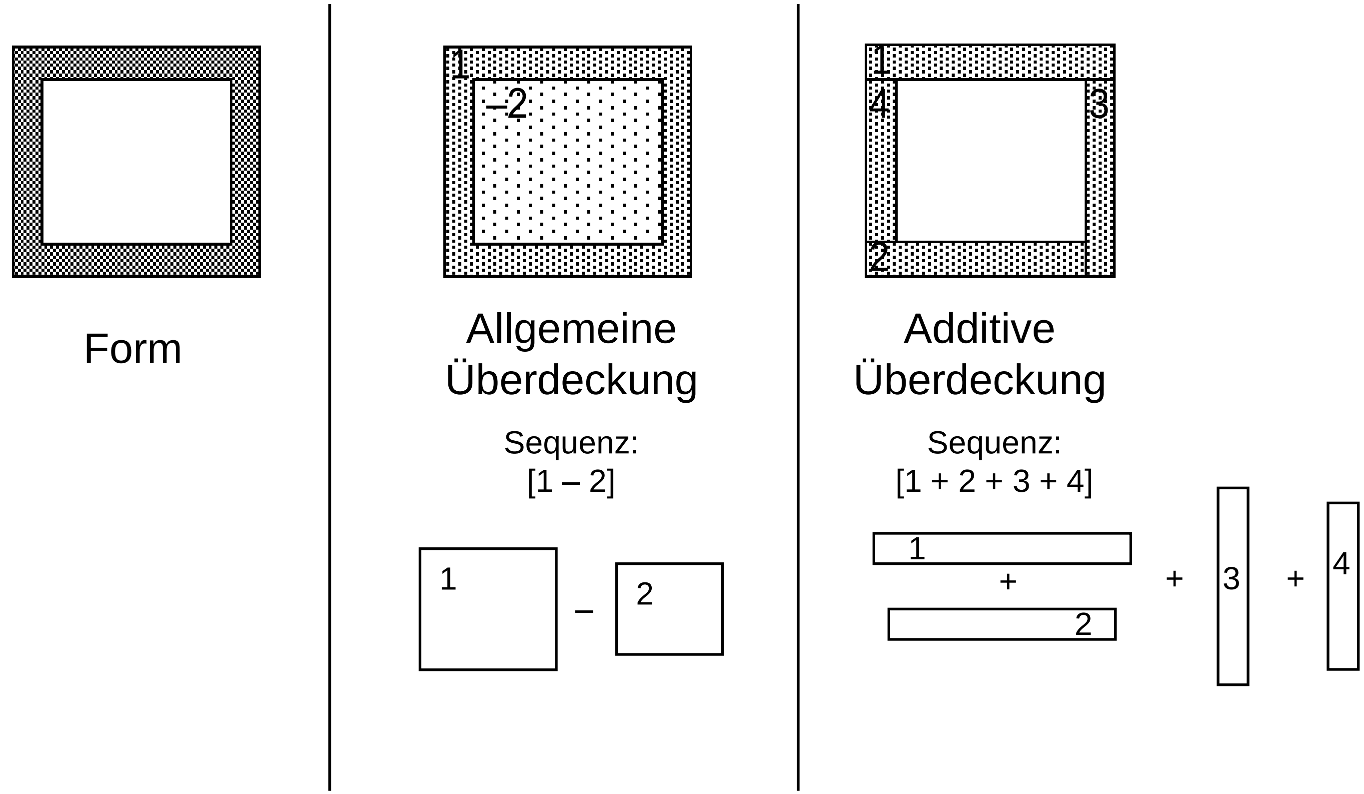
- Objektmodell
  - Formen sind als konturierte Objekte gegeben (d.h. Polygone)
  - Extraktion von Formen aus Grauwertbildern möglich, solange klare Konturen bestimmt werden können (Probleme z.B. bei teilweise verdeckten Objekten)
  - Polygone müssen nicht konvex sein (Einbuchtungen möglich)
- Rechtecksüberdeckung
  - Additive Überdeckung
    Durch eine Folge von Rechtecken $[R_1, R_2, \ldots, R_k]$ ist eine Folge von additiven Überdeckungen $[C_0, C_1, \ldots]$ wie folgt definiert:

    $$C_0 = \emptyset, \quad C_{i+1} = C_i \cup R_{i+1}$$

  - Allgemeine Überdeckung
    Neben dem Hinzufügen von Rechtecksflächen ($\cup$) ist auch das Entfernen von Rechtecksflächen (–) möglich:

    $$C_0 = \emptyset, \quad C_{i+1} = C_i \cup R_{i+1} \quad \text{oder} \quad C_{i+1} = C_i - R_{i+1}$$

- Für endliche Formen *S* konvergieren (additive) Überdeckungssequenzen schon im Endlichen, d.h. es gibt ein *K*, so dass $C_K = S$, und wir definieren $C_j = C_K$ für $j \geq K$.
- Überlappungen sind erlaubt, sollen aber möglichst gering ausfallen



- Approximative Rechtecksüberdeckungen
  - Statt *alle* Rechtecke einer Überdeckung nur *wenige* speichern
  - Entfernen kleiner Rechtecke entspricht dem Beseitigen hochfrequenter Fehler wie Schmutzflecken oder Diskretisierungsfehlern (z.B. bei eingescannten Bildern, Voxelisierung, etc.)

- Approximationsqualität
  - Die ersten Rechtecke einer Überdeckung sollen schon eine möglichst gute Approximation der ursprünglichen Form liefern
  - Kumulatives Fehlerkriterium: Die Approximationsfehler der Überdeckungssequenz $[C_0, C_1, \ldots, S]$ werden sukzessive aufsummiert, die Gesamtsumme zählt:

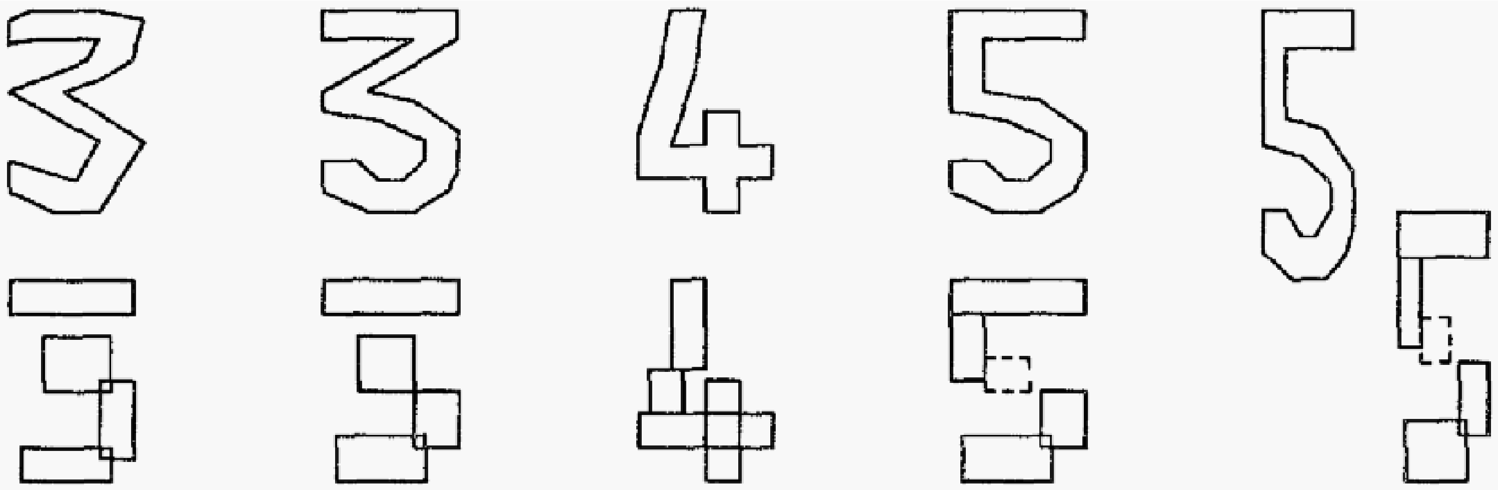
$$\text{kumulativer Fehler} = \sum_{i=1}^{n} |S - C_i|$$

  - Minimierung der Gesamtsumme:

    => Minimierung der "frühen" Fehler $|S - C_i|$ für kleine i, da diese mehrfach gewertet werden

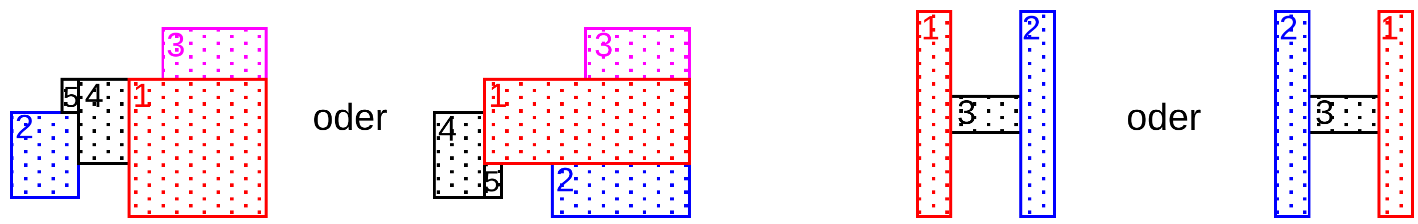
– Beispiel [Jagadish. Proc. ACM Int. Conf. on Management of Data (SIGMOD) 1991]

Formen

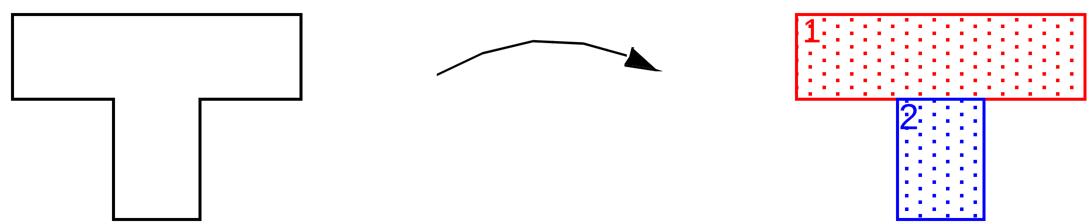


„erste" Rechtecke der Überdeckungssequenz

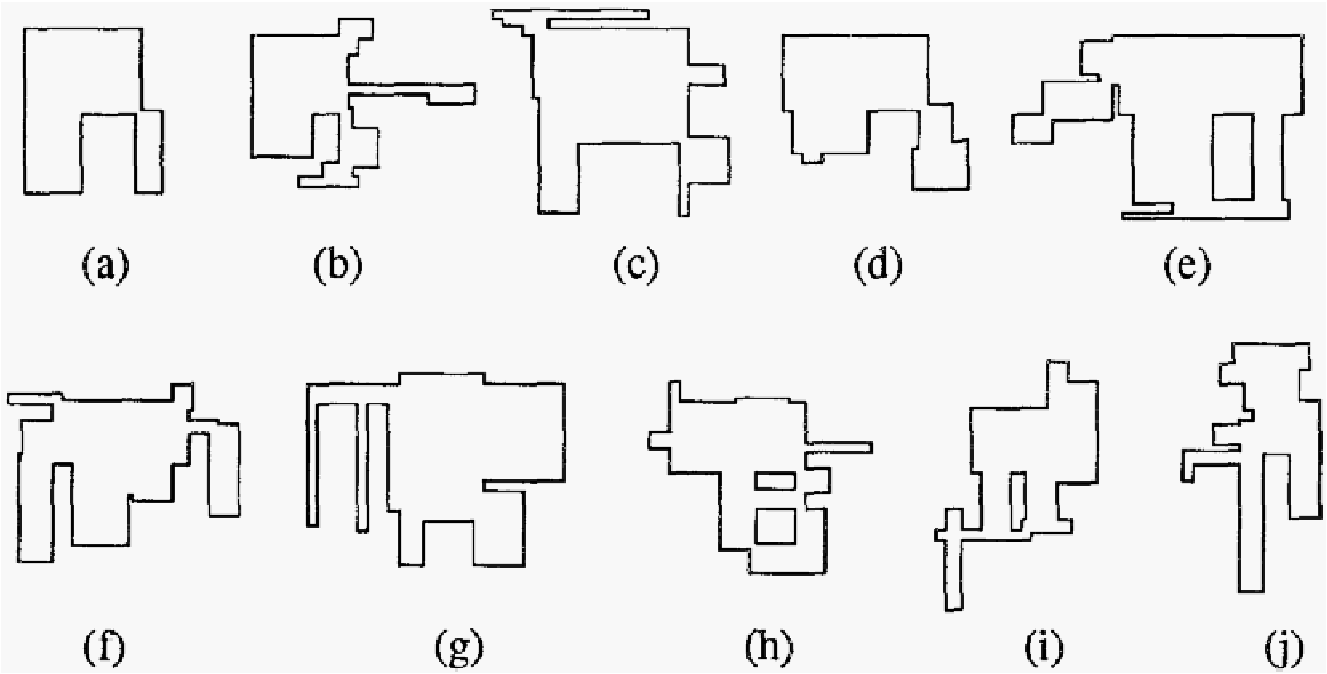
- Probleme der Rechtecksüberdeckung
  - Nicht-eindeutige Repräsentation
    - » Es kann unterschiedliche optimale Zerlegungen eines Objektes geben
    - » Insbesondere bei Symmetrie ist die Reihenfolge der Rechtecke nicht eindeutig
    - » Lösung: Objekt mehrfach speichern oder mehrfache Anfragen für eine Form



  - Rechteckige Formen
    - » Wird eine Form schon durch wenige Rechtecke exakt beschrieben, besteht die Überdeckungssequenz ggf. aus weniger Elementen, als bei anderen Objekten
    - » Lösung: speichere „dummy" Rechtecke (ohne Ausdehnung)



- Ähnlichkeitsanfragen
  - Testbed
    - » Datenbank: 16.000 synthetische Formen
    - » Form = Zusammensetzung von 10 zufällig erzeugten Rechtecken
    - » Additive Überdeckungen; jeweils die größten drei Rechtecke der Überdeckung in Index abgespeichert
    - » Anfragen: Bereichsanfragen um zufällig ausgewählte Formen der Datenbank
  - Beispiel für das Ergebnis einer Ähnlichkeitsanfrage:
    (a: Anfrageform; b – j: Ergebnisformen)

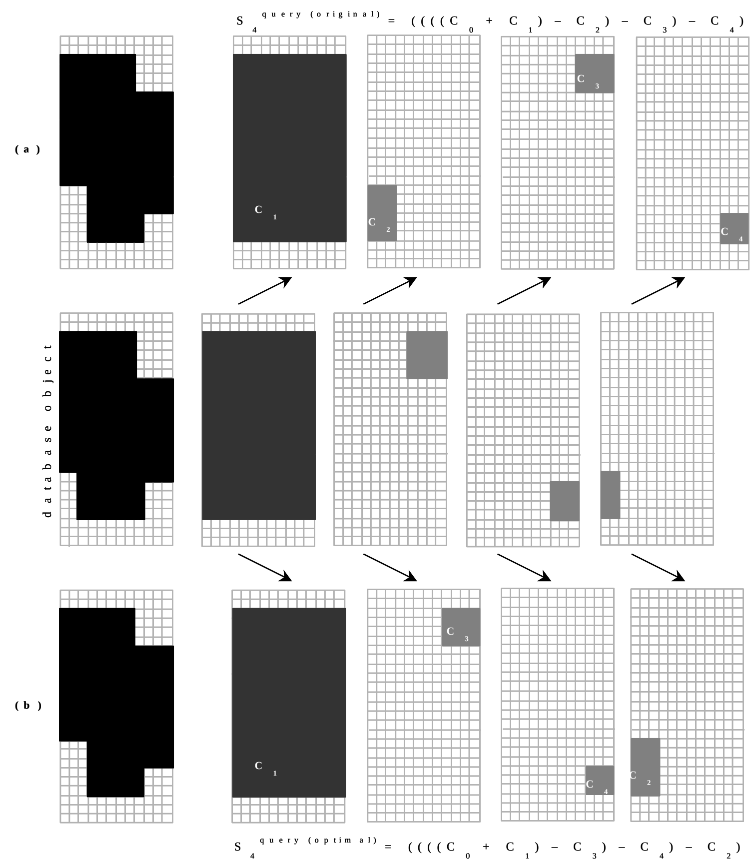


Quelle:
[Jagadish. Proc. ACM Int. Conf. on Management of Data (SIGMOD) 1991]

3.2.4 Erweiterung des Überdeckungsmodell für 3D-Objekte

[Kriegel, Brecheisen, Kröger, Pfeifle, Schubert. Proc. ACM Int. Conf. on Management of Data (SIGMOD) 2003]

– Motivation:
  - Ähnlichkeitsmodell für voxelisierte 3D-CAD Daten
  - Ziel: Größere Flexibilität beim Vergleich einzelner Überdeckungen innerhalb einer Überdeckungssequenz.
    – Löst das Problem der uneindeutigen Überdeckungssequenz ohne (Query-)Objekte mehrfach abzuspeichern

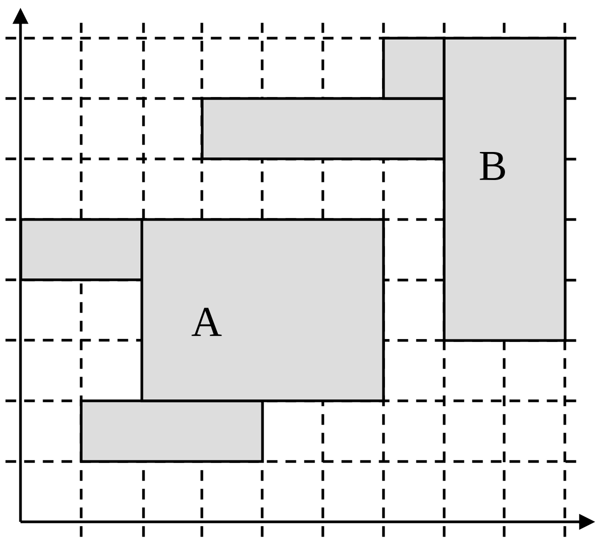


- Idee:
  - Objekt wird nicht mehr durch einen großen Feature-Vektor repräsentiert (Paramter der ersten *k* Überdeckungen)
  - Objekt wird nun durch eine Menge von Feature Vektoren repräsentiert
    - Jede Überdeckung wird zu einem 2·D-dimensionalen Feature-Vektor
      - » Koordinaten für den Eckpunkt der Überdeckung (D Werte)
      - » Ausdehnungen der Überdeckungen entlang der Raumachsen (D Werte)
    - Überdeckungssequenz wird zu einer Menge von Feature Vektoren
    - 2D Beispiel

Mengen!!!

A = {(1,1,3,1),(2,2,4,3),(0,4,2,1)}

B = {(7,3,2,4),(3,6,4,1),(6,7,1,1)}



– Abstand auf Punktmengen

- Mengen Aufzählung
  - Sei S eine endliche Menge
  - Abbildung $\pi(S)$ heißt Aufzählung von S, wenn jedem $s \in S$ eine eindeutige Nummer zuordnet, d.h. $\pi(s) = i \in \{1, \ldots, |S|\}$
  - $\Pi(S)$ bezeichnet die Menge aller möglichen Aufzählungen von S

- Minimal Matching Distance
  - Distanz zwischen Punktmengen X und Y
  - Formal („Minimal Weight Perfect Matching Distance"):

    Sei $X = (x_1, \ldots, x_{|X|})$, $Y = (y_1, \ldots, y_{|Y|})$, wobei oBdA $|X| \leq |Y|$

    Sei w eine Gewichtsfunktion für nicht zugeordnete Punkte

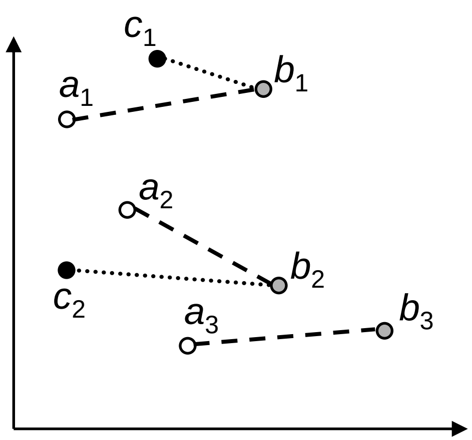
$$MinMatchDist(X, Y) = \min_{\pi \in \Pi(Y)} \Big(\underbrace{\sum_{i=1}^{|X|} dist(x_i, y_{\pi(i)})}_{\text{Jedem } x \text{ genau ein (unterschiedliches) } y \text{ zuordnen}} + \underbrace{\sum_{i=|X|+1}^{|Y|} w(y_{\pi(i)})}_{\text{Jedes noch nicht zugeordnete } y \text{ mit } w \text{ gewichten}} \Big)$$

  - Metrikeigenschaft hängt von der Gewichtsfunktion w ab

- Gewichstfunktion basierend auf „Dummy-Vektoren“
  - » $w(v)$ entspricht Distanz von v zum Null-Vektor, d.h.

$$w(v) = dist(v, \vec{0})$$

- Intuition und Beispiel
  - Punkte der Punktmengen X und Y sind Knoten in einem bipartiten Graphen
  - Kanten (x,y) zwischen den Punkten x und y sind mit $dist(x,y)$ gewichtet
  - Perfektes Matching:
    - » Jeder Knoten in X ist mit genau einem Knoten aus Y verbunden
  - Minimales (perfektes) Matching:
    - » Summe der Gewichte der Kanten des Matchings ist minimal



$$\text{MinMatchDist}(A,B) = dist(a_1,b_1) + dist(a_2,b_2) + dist(a_3,b_3)$$

$$\text{MinMatchDist}(C,B) = dist(c_1,b_1) + dist(c_2,b_2) + w(b_3)$$

$$\text{mit } w(b_3) = dist(\mathbf{0},b_3)$$

– Anfragebearbeitung

- Motivation:
  - Berechnung des Minimalen Matchings ist teuer („Kuhn-Munkres-Algorithmus“: O(k^3), k = Anzahl der Überdeckungen)
- Lösung:
  - Filter/Refinement
  - Gesucht: billigere Distanz, die Lower Bounding Eigenschaft erfüllt
- Centroid Filter
  - Centroid ist der Schwerpunkt/Mittelpunkt einer Punktmenge
  - Lemma:
    - » Seien X und Y Mengen mit k Vektoren und c_X, c_Y die entsprechenden Centroide
    - » Dann gilt

$$k \cdot L_2(c_X, c_Y) \leq \text{MinMatch}(X, Y)$$

    - » Verwalte Centroide in einem separaten Index
    - » Filter: auf Centroid-Index