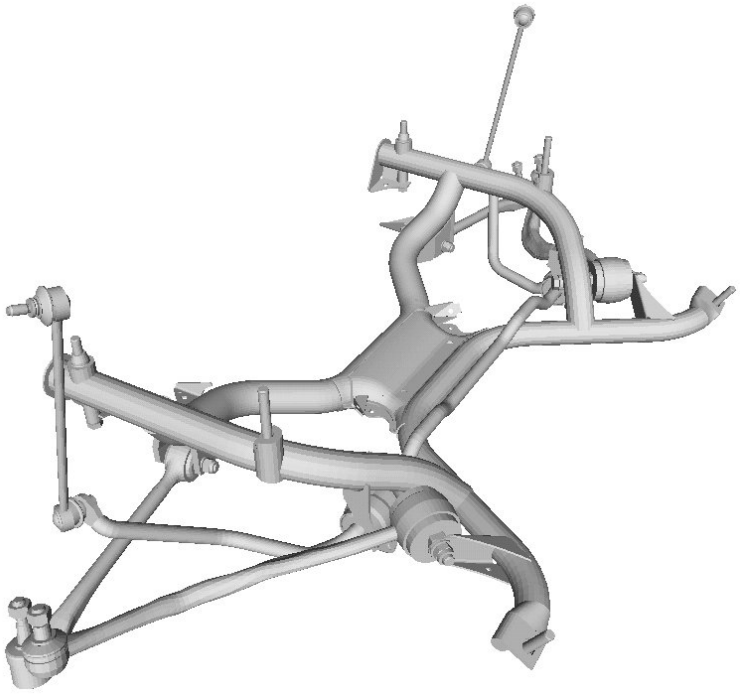
Diplomarbeit

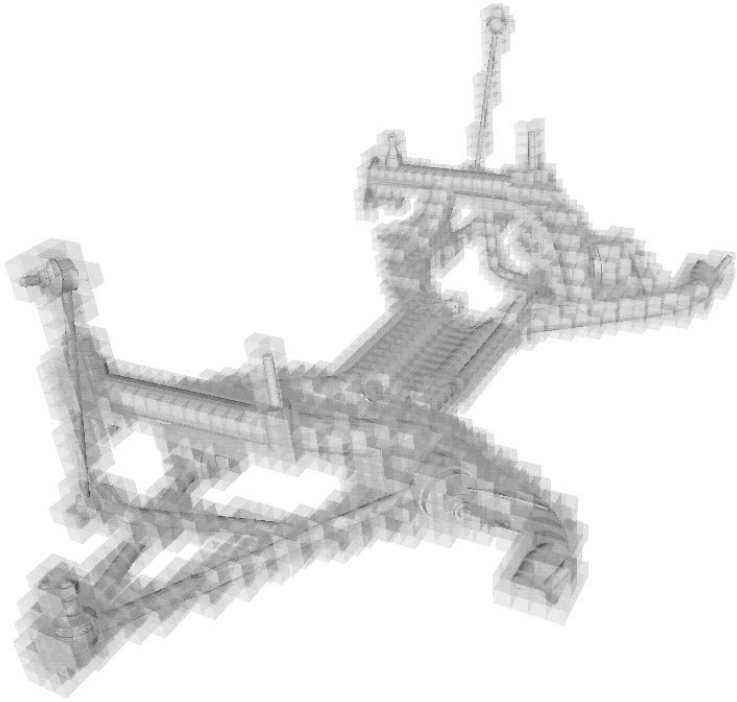
# Using Sets of Feature Vectors for Similarity Search on Voxelized CAD Data

Stefan Brecheisen

| | |
|---|---|
| Aufgabensteller: | Prof. Dr. Hans-Peter Kriegel |
| Betreuer: | Martin Pfeifle |
| Dank an: | Peer Kröger, Matthias Schubert |



CAD-Teil



Voxelmenge

# Gliederung

- Ähnlichkeitsmodelle für voxelisierte CAD-Daten
  - Distanzbasiertes Ähnlichkeitsmaß
  - Normalisierung
  - Das Volumenmodell
  - Das Solid-Angle-Modell
  - Das Cover-Sequence-Modell
- Das Vektormengen-Modell
- Effiziente Anfragebearbeitung
- Experimentelle Evaluierung
- Ausblick

# Distanzbasiertes Ähnlichkeitsmaß

- Ziel: Repräsentation der Ähnlichkeit zweier CAD-Teile durch eine reelle Zahl
- Menge von Objekten $O$
- Featuretransformation $F : O \rightarrow \mathbb{R}^d$
- Distanzfunktion $dist : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, z.B. euklidische Distanz

$$d_{euclid}(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$$

- Definition des distanzbasierten Ähnlichkeitsmaßes $simdist : O \times O \rightarrow \mathbb{R}$:

$$simdist(o_1, o_2) = dist(F(o_1), F(o_2))$$

# Normalisierung

- Anforderung: Invarianz gegenüber Skalierung, Translation, $90°$-Rotation, Spiegelung
- Invarianz bzgl. einer Klasse $K$ von Transformationen bedeutet:
  für alle Objekte $o_1, o_2 \in O$ und Transformationen $T \in K$:

$$simdist(o_1, o_2) = simdist(T(o_1), o_2) = simdist(o_1, T(o_2))$$

- Erweiterte Definition des distanzbasierten Ähnlichkeitsmaßes $simdist : O \times O \rightarrow \mathbb{R}$:

$$simdist(o_1, o_2) = \min_{T \in K} \{dist(F(o_1), F(T(o_2)))\}$$

- Invarianz durch Normalisierung der Daten und Erzeugen von Varianten zur Laufzeit

$$\left(\begin{array}{ccc|c} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right), \left(\begin{array}{ccc|c} 1 & 0 & 0 & -t_x \\ 0 & 1 & 0 & -t_y \\ 0 & 0 & 1 & -t_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right), \left(\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right), \left(\begin{array}{ccc|c} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right)$$

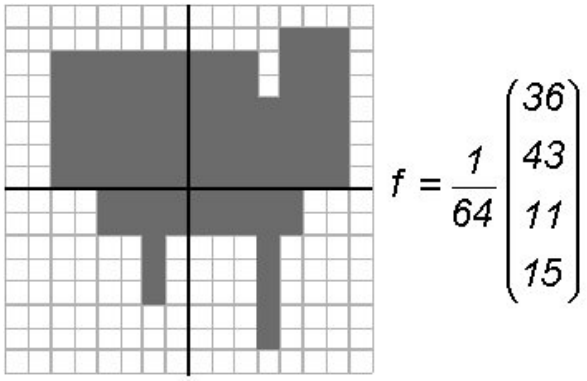
# Das Volumenmodell

Grundidee:

- Partitionierung des Datenraums in disjunkte Zellen
- Extrahieren von einer oder mehreren Kennzahlen aus jeder Zelle
- Zusammenfassen der Kennzahlen in Feature-Vektoren

Beim Volumenmodell:

- Bestimme die Anzahl der Voxel in jeder Zelle
- jede Zelle entspricht einer Dimension im Feature-Vektor



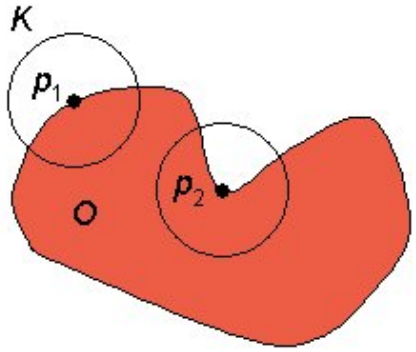
$$f = \frac{1}{64}\begin{pmatrix} 36 \\ 43 \\ 11 \\ 15 \end{pmatrix}$$

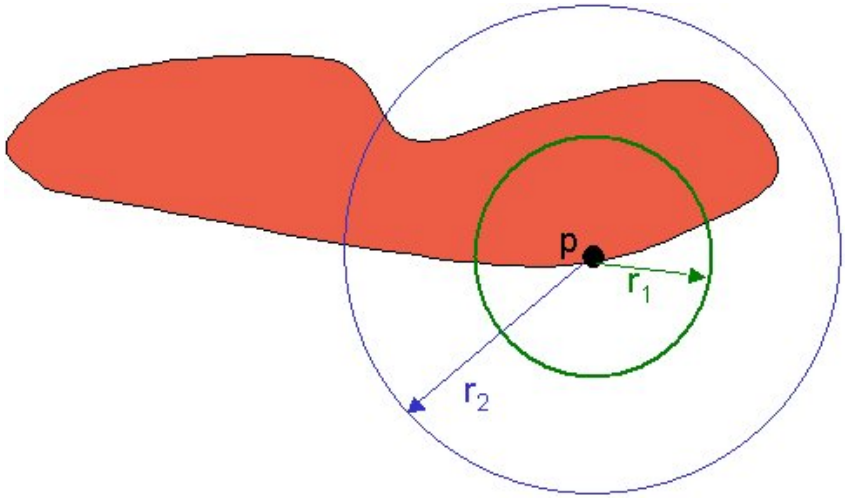
# Das Solid-Angle-Modell

- Idee: Berücksichtigung von geometrischen Eigenschaften innerhalb der Zellen (Konvexität, Konkavität)
- Wähle geeigneten Radius $r$ und bestimme für jeden Randvoxel $v$ den Solid-Angle-Wert

$$\mathrm{Sa}(v,r) = \frac{|K_{v,r} \cap o|}{|K_{v,r}|}$$

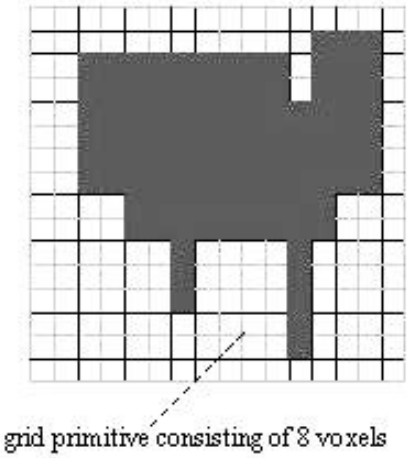
- Bestimme für jede Zelle den Durchschnitt der jeweiligen Solid-Angle-Werte

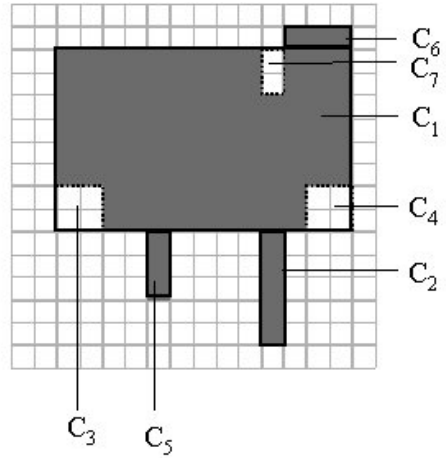


# Das Cover-Sequence-Modell

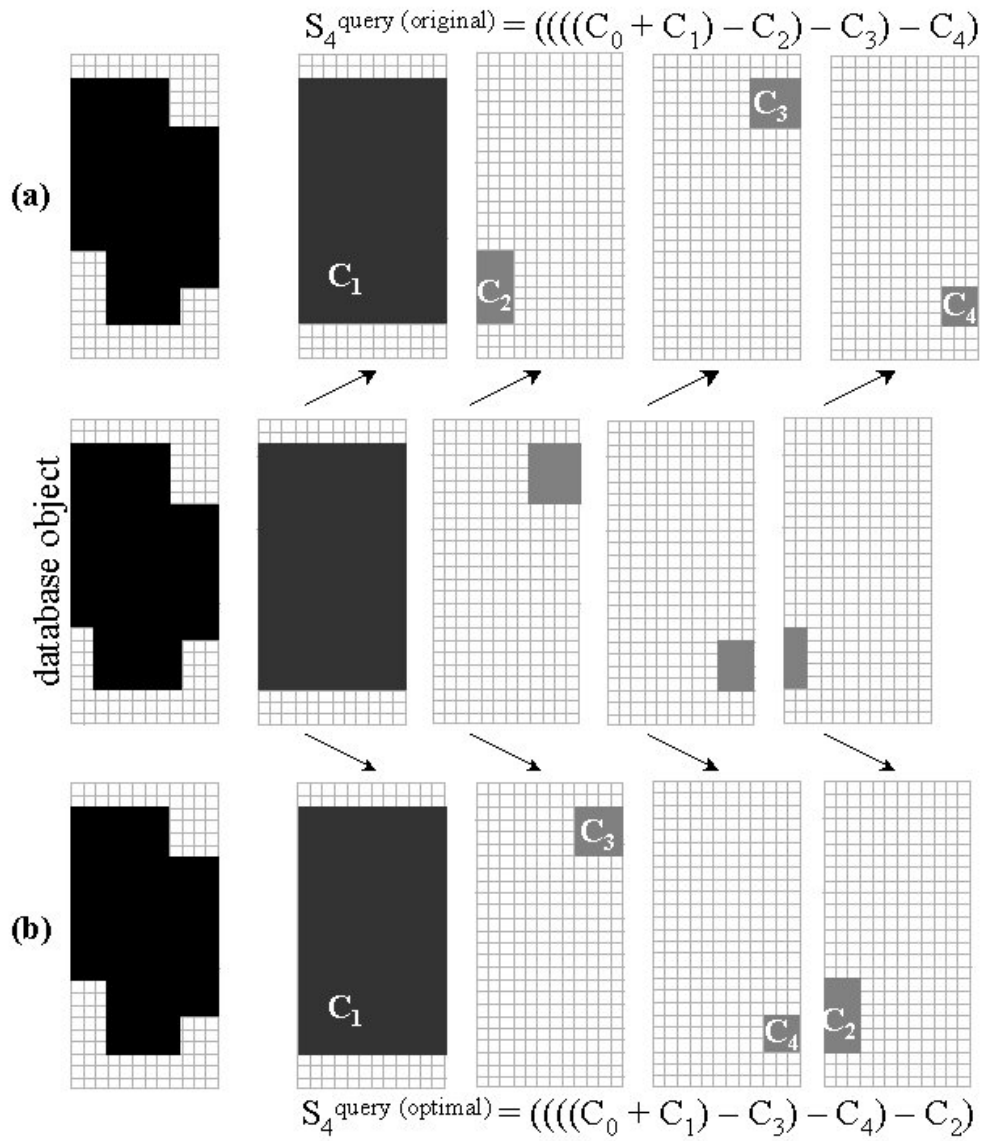
- Überdeckungssequenz $S_k = (((C_0\,\sigma_1\,C_1)\,\sigma_2\,C_2)\,\ldots\,\sigma_k\,C_k)$, $\sigma_i \in \{+,-\}$, bestehend aus achsenparallelen (Hyper-)Rechtecken
- Qualitätsmaß: symmetrische Volumendifferenz $Err_k = |\,o\,\mathrm{XOR}\,S_k\,|$
- Greedy Algorithmus: minimiere $Err_i$ in jedem Schritt $i$, polynomielle Laufzeit




$S_1=(C_0+C_1)$ $\quad Err_1=21$
$S_2=((C_0+C_1)+C_2)$ $\quad Err_2=16$
$S_3=(((C_0+C_1)+C_2)-C_3)$ $\quad Err_3=12$
...

$f_o^{6i+1}$ = $x$-position of $C_i$
$f_o^{6i+2}$ = $y$-position of $C_i$
$f_o^{6i+3}$ = $z$-position of $C_i$
$f_o^{6i+4}$ = $x$-extension of $C_i$
$f_o^{6i+5}$ = $y$-extension of $C_i$
$f_o^{6i+6}$ = $z$-extension of $C_i$

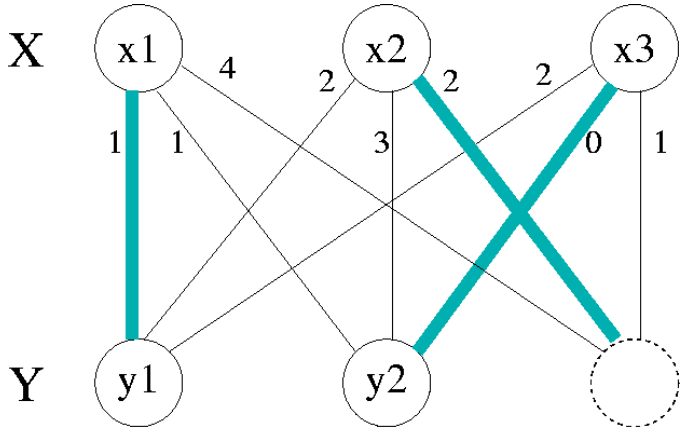

$S_4^{\text{query (original)}} = ((((C_0 + C_1) - C_2) - C_3) - C_4)$
(a)
$C_1$
$C_2$
$C_3$
$C_4$
database object
(b)
$C_1$
$C_3$
$C_4$
$C_2$
$S_4^{\text{query (optimal)}} = ((((C_0 + C_1) - C_3) - C_4) - C_2)$

# Das Vektormengen-Modell

Repräsentiere Überdeckungssequenz $S_k$ durch Vektormenge $X \subset \mathbb{R}^6$, $|X| \leq k$

Distanzmaß zwischen zwei Vektormengen $X$ und $Y$:

- Konstruiere gewichteten vollständigen bipartiten Graph $G = (X \cup Y, X \times Y)$
- Kantengewicht für $(\vec{x}, \vec{y}) \in X \times Y$ ist $d_{euclid}(\vec{x}, \vec{y})$
- Gewichtsfunktion $w : X \rightarrow \mathbb{R}$, falls $|X| > |Y|$
- Bestimme maximales Matching mit minimalem Gewicht:
  Algorithmus von Kuhn und Munkres, Laufzeit $O(k^3)$



# Effiziente Anfragebearbeitung

- Problem: Effiziente Beantwortung von Bereichs- und $k$-NN-Anfragen
- Mehrstufige Anfragebearbeitung: Filterschritt, Verfeinerungsschritt
- Kriterium für Korrektheit des Filterschritts:
  Untere-Schranken-Eigenschaft für Filterdistanz $d_f$ und Objektdistanz $d_o$

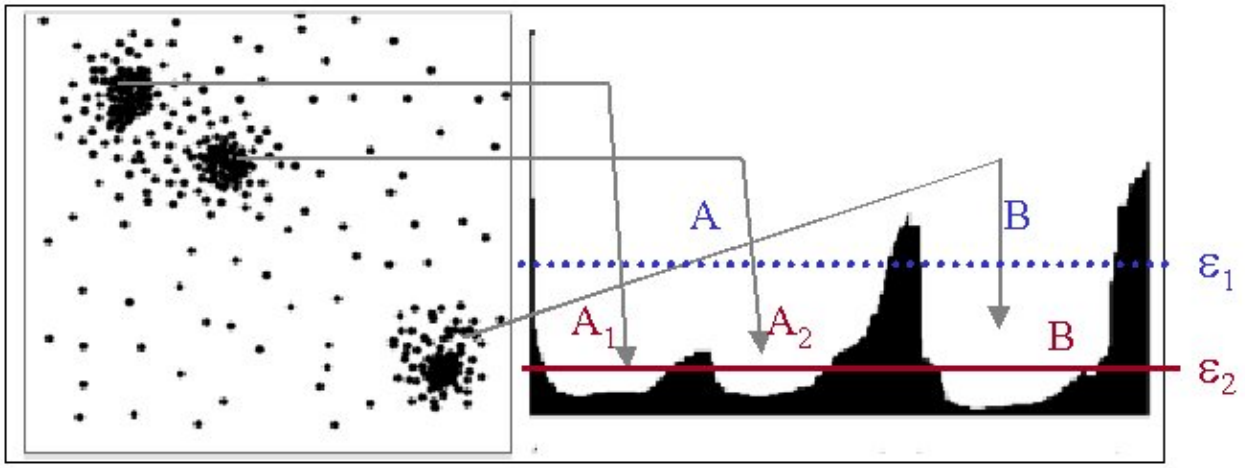
$$\forall o_1, o_2 \in O : d_f(o_1, o_2) \leq d_o(o_1, o_2)$$

Filterschritt für das Vektormengen-Modell:

- Fülle ggf. Vektormenge mit Dummy-Überdeckungen auf
- Bestimme Schwerpunkt der Vektoren in jeder Vektormenge
- $k$-fache euklidische Distanz zwischen Schwerpunkten ist untere Schranke für die Minimal-Matching-Distanz

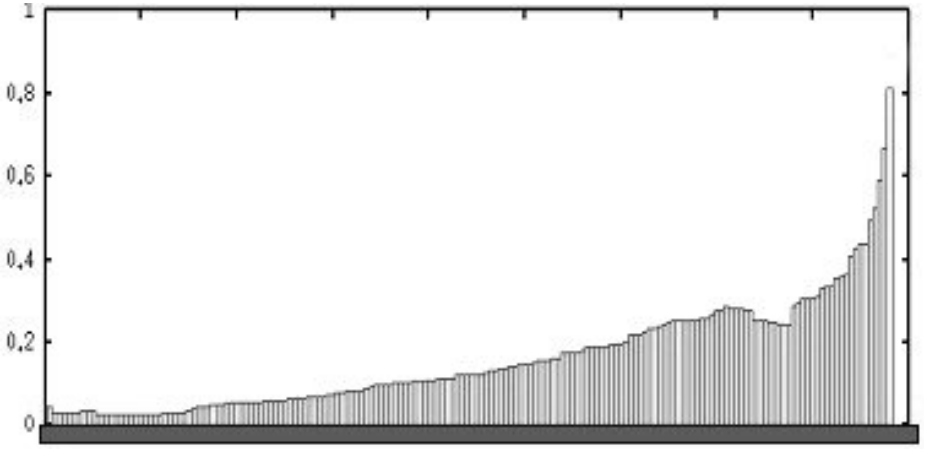
# Evaluierung: OPTICS

- Dichtebasiertes hierarchisches Clustering mit Parametern $\varepsilon$ und $MinPts$
- Clusterordnung basierend auf Kerndistanz und Erreichbarkeitsdistanz
- Cluster sind Täler im Reachability-Plot
- Erwünschtes Ergebnis: Objekte in einem Cluster möglichst ähnlich, Objekte in verschiedenen Clustern möglichst unähnlich
- Gesamte Datenmenge geht in die Evaluierung ein

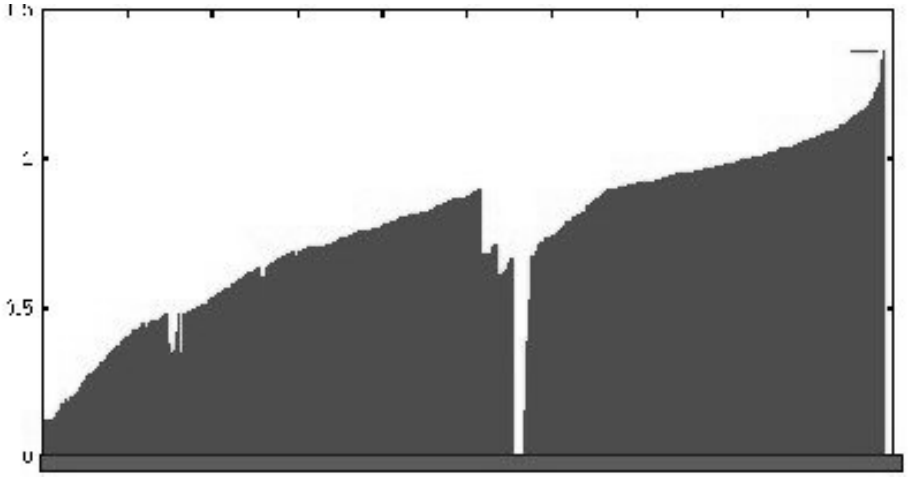


# Evaluierung: Volumenmodell

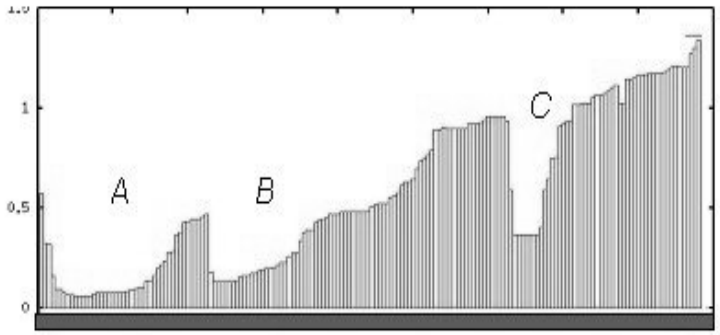
Autodaten



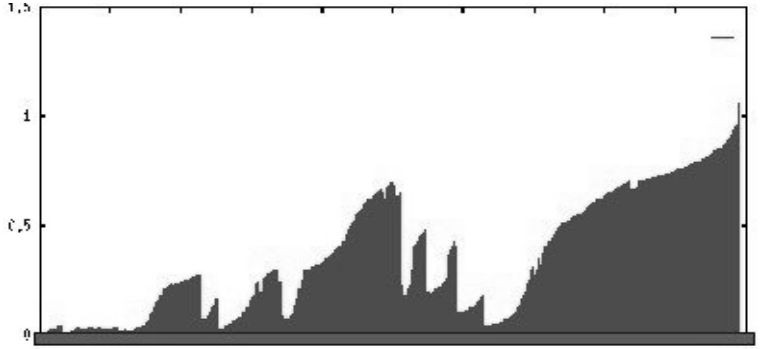

Flugzeugdaten



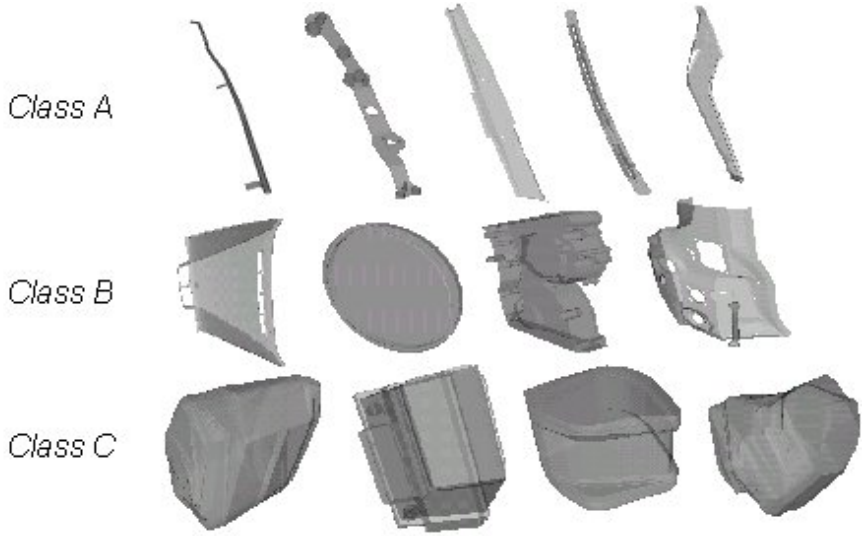
# Evaluierung: Solid-Angle-Modell



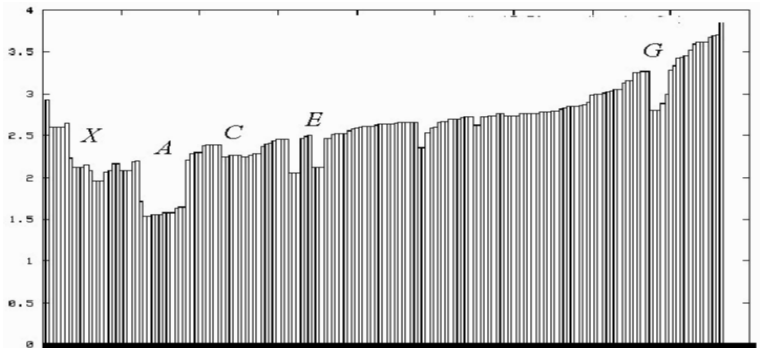

Autodaten



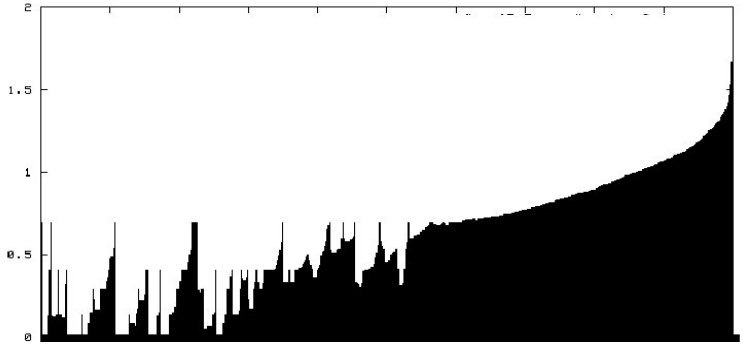
Flugzeugdaten



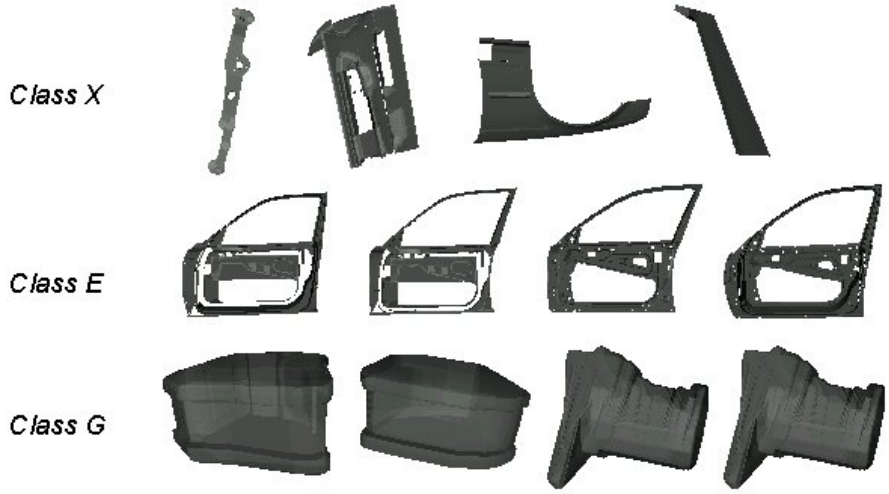
# Evaluierung: Cover-Sequence-Modell




Autodaten (7 Überdeckungen)

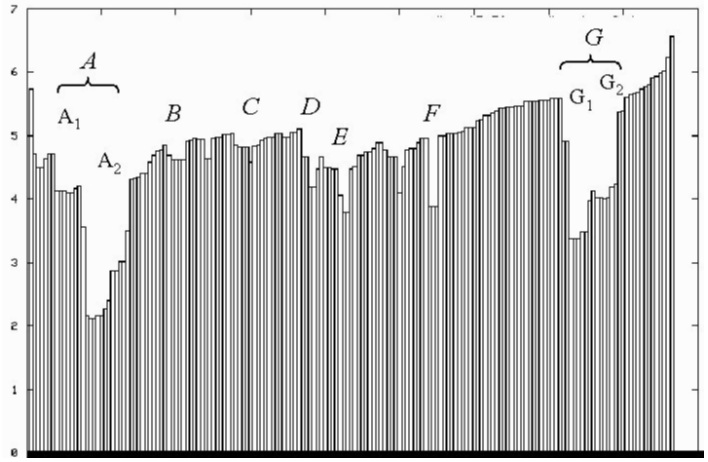


Flugzeugdaten (7 Überdeckungen)

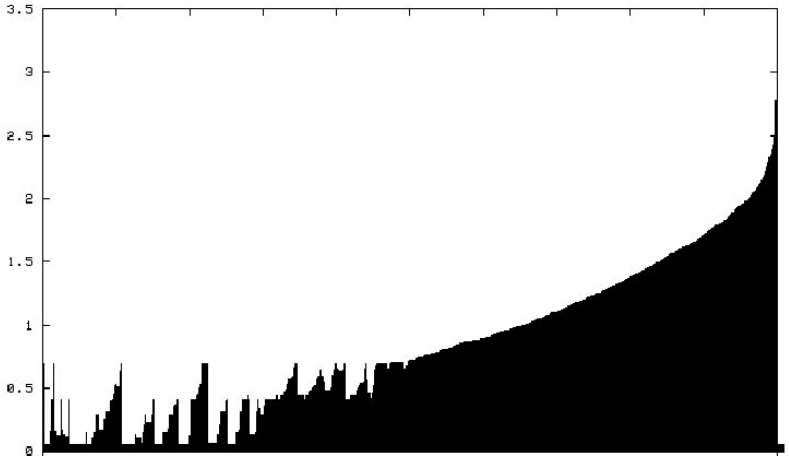


# Evaluierung: Vektormengen-Modell

Autodaten (7 Überdeckungen)




Flugzeugdaten (7 Überdeckungen)



# Evaluierung: Vektormengen-Modell

# Evaluierung: Effizienz

- 100 10-NN-Anfragen auf der Flugzeugdatenbank
- Indexstruktur: X-tree
- Rechner: Intel Xeon 1.7 GHz, 2 GByte RAM
- Berechnete I/O-Kosten: 8 ms pro Seitenzugriff, 200 $\mu$s pro gelesenes Byte
- Filterschritt bringt Beschleunigung um Faktor 2
- Selektivität des Filterkriteriums: ca. 20%

Laufzeit für 10-NN-Anfragen in Sekunden:

| Modell | CPU-Zeit | I/O-Zeit | Gesamt |
|---|---|---|---|
| Cover Sequence | 142.82 | 2632.06 | 2774.88 |
| Vektormengen mit Filter | 105.88 | 932.80 | 1038.68 |
| Vektormengen seq. scan | 1025.32 | 806.40 | 1831.72 |

# Ausblick

*BOSS* (*B*rowsing *O*PTICS-Plots for *S*imilarity *S*earch)

- Interaktiver Browser für OPTICS-Plots
- Schneller Überblick über die Cluster-Hierarchie
- Anzeige der einzelnen Teile in einem Cluster
- Anzeige von Cluster-Repräsentanten