Übungen zu Multimedia-Retrieval
Aufgabenblatt 5 - Musterlösungen

Übung: Dipl.-Inform. Christian Feinen
Vorlesung: JProf. Dr.-Ing. Marcin Grzegorzek
Forschungsgruppe für Mustererkennung
Institut für Bildinformatik im Department ETI
Fakultät IV der Universität Siegen

Ausgabe:	13.01.2012
Abgabe:	27.01.2012 per E-Mail an **{christian.feinen, marcin.grzegorzek}@uni-siegen.de**
Format:	A1-Nachname1-Nachname2.pdf

Gebt bei allen Rechnungen sinnvolle Zwischenschritte an! Diesmal werden bis zu 60 Punkte vergeben.

1 Earth-Mover-Distanzfunktion (20 Punkte)

1. Es sind die folgenden Sequenzen gegeben:

$$p = \left\langle tuple\left(p_1 = \begin{pmatrix}1\\1\end{pmatrix}, w_{p_1} = 0,4\right), tuple\left(p_2 = \begin{pmatrix}2\\1\end{pmatrix}, w_{p_2} = 0,5\right)\right\rangle$$

und

$$q = \left\langle tuple\left(q_1 = \begin{pmatrix}1\\2\end{pmatrix}, w_{q_1} = 0,7\right)\right\rangle$$

Berechne den Distanzwert anhand der Transportkosten. Die f_{ij} Werte können dabei berechnet oder (mit Erklärung) geschätzt werden.

Musterlösung:

1. Berechnung der Grunddistanzwerte von p_1 und p_2 zu q_1

d_{L_2}	1
1	1
2	$\sqrt{2}$

2. Bestimmung der f_{ij} - Werte

f_{ij}	1	$\sum$
1	0.4	0.4
2	0.3	0.3
$\sum$	0.7	0.7

Die Werte können entweder durch die Lösung der folgenden Gleichung:

$$\min_{f_{ij}}(f_{11} + \sqrt{2} * f_{21})$$

berechnet werden.

Alternativ kann die Lösung aber auch logisch unter Berücksichtigung der folgenden Bedingungen hergeleitet werden:

$$f_{11} + f_{21} = 0.7$$

$$f_{11} \leq 0.4 \text{ und } f_{21} \leq 0.5$$

Da sich p_1 näher an q_1 befindet (als p_2) ist es sinnvoll, im ersten Schritt die gesamte Erde von p_1 nach q_1 zu transportieren. Anschließend wird die fehlende Erde von p_2 zum Auffüllen des verbleibenden Rests verwendet.

3. Berechnung der Kosten nach der Earth-Movers-Distanzfunktion:

$$d_{EM}(p, q) = \frac{min_{f_{ij}} \left(\sum_{i=1}^{m} \sum_{j=1}^{n} d(p_i, q_j) * f_{ij} \right)}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}}$$

Transportkosten:

$$Kosten(p, q, F) = 1 * 0.4 + \sqrt{2} * 0.3 \approx 0.824$$

Distanzwert:

$$d_{EM}(p, q) = \frac{(1 * 0.4 + \sqrt{2} * 0.3)}{0.4 + 0.3} = \frac{0.824264069}{0.7} = 1.177520099$$

---

2 Hochdimensionale Indexstrukturen und Suchbäume (20 Punkte)

1. Welche Anforderungen werden an hochdimensionale Indexstrukturen gestellt? Erklärt die jeweiligen Anforderungen.
2. Welche Arten von Suchanfragen gibt es für hochdimensionale Indexstrukturen? Erklärt die jeweiligen Anfragearten kurz.
3. Wie sind MIN-Distanz und $MINMAX$-Distanz definiert? Wie kann man sie interpretieren?
4. Welche Charakterisierungs-Kriterien gibt es für Suchbäume? Gebt auch jeweils eine kurze Erklärung an.

Musterlösung:

1. Lösungen Teilaufgabe 2.1

Korrektheit und Vollständigkeit Ergebnisse korrekt und vollständig bezüglich der Suchbedingung.

hochdimensionaler Suchraum Indexstruktur ohne Effizienzeinbußen skalierbar bezüglich der Anzahl der Dimensionen.

räumliche Ausdehnung der Objekte Unterstützung von Medienobjekten verschiedendimensionaler Ausdehnung.

Sucheffizienz Suchoperationen erfordern höchstens linearen Suchaufwand.

viele Anfragearten Es müssen mehrere Anfragearten unterstützt werden.

Update-Operationen Updates müssen effizient ausführbar sein.

verschiedene Distanzfunktionen Unterschiedliche Distanzfunktionen müssen unterstützt werden.

speicherplatzsparend Indexstrukturen müssen auf Basis eines möglichst geringen Speicherplatzbedarfs implementiert werden, damit sie realisierbar sind.

2. Lösungen Teilaufgabe 2.2

Nächste Nachbarn Suche Nächster Nachbar zum Anfrageobjekt bezüglich Distanz gesucht.

Approximative NN Suche Kleine, quantifizierbare Ergebnisungenauigkeiten werden in Kauf genommen.

Reverse NN Suche Es wird das *Objekt* gesucht, welches zu dem Anfrageobjekt der NN ist.

Bereichssuche Bereich wird spezifiziert und Objekte gesucht, die den Bereich schneiden.

Punktsuche Es wird ein Objekt gesucht, das dem Anfragepunkt enstpricht.

Partial-Match-Suche Ähnlich der zuvor genannten Punktsuche. Hier wird allerdings nur auf einer Teilmenge der Dimensionen operiert.

Ähnlichkeitsverbund Suche auf zwei Punktmengen. Es werden Paare, deren Abstand geringer als ein festgelegter Schwellwert ist, zurückgeliefert.

3. Lösungen Teilaufgabe 2.3

MINDIST Gibt die minimale Distanz zwischen einem Anfragepunkt und einem möglichen Objekt innerhalb eines MBR an.

$$MINDIST(q,(s,t)) = \sum_{i=1}^{n} |q[i] - r[i]|^2$$

$$r[i] = \begin{cases} s[i] & \text{wenn} \quad q[i] < s[i] \\ t[i] & \text{wenn} \quad q[i] > t[i] \\ q[i] & \text{sonst} \end{cases}$$

MINMAXDIST Gibt den maximalen Abstand an, den ein NN aus dem MBR zum Anfragepunkt haben kann.

$MINMAXDIST(q,(s,t)) = \min_{1\leq k\leq n}\left(|q[k]-rm[k]|^2 + \sum_{i\neq k, 1\leq i\leq n}|q[i]-rM[i]|^2\right)$

$rm[k] = \begin{cases} s[k] & \text{wenn} \quad q[k] < \frac{(s[k]+t[k])}{2} \\ t[k] & \text{sonst} \end{cases}$ $rM[i] = \begin{cases} s[i] & \text{wenn} \quad q[i] > \frac{(s[i]+t[i])}{2} \\ t[i] & \text{sonst} \end{cases}$

4. Lösungen Teilaufgabe 2.4

Cluster-Bildung Behandelt die Fragestellung: Wie werden die Elemente zu Clustern gruppiert?

- Global (teilend) bedeutet, dass mit der gesamten Menge angefangen wird, und diese dann anschließend auf Basis der inneren Distanz in weitere Cluster kleinerer Teilmengen aufgespaltet wird.
- Lokal (gruppierend) bedeutet, dass mit einzelnen Elementen angefangen wird und diese dann zu größeren Clustern gruppiert werden.

Cluster Überlappung Behandelt die Fragestellung: Dürfen sich Cluster überlappen?

Balance Ist der Suchbaum balanciert, d.h. stehen Breite und Höhe in einem Verhältnis bzw. haben die Blätter eine gemeinsame Distanz zur Wurzel.

Objektspeicherung Hier wird entschieden, ob die Objekte nur in den Blättern oder auch in den Knoten gespeichert werden.

Geometrie Die Geometrie der Cluster kann z.b. in Form von Hyperkugel, Hyperquader, u.v.m realisiert werden.

---

3 Nächste-Nachbarn-Suche (20 Punkte)

Gegeben seien die Punkte $P_1 = (1,6), P_2 = (2,5), P_3 = (6,7), P_4 = (7,6), P_5 = (7,4), P_6 = (1,1), P_7 = (3,2), P_8 = (6,3), P_9 = (7,3)$. Desweiteren gebe es die Rectangles $R_1,\dots,R_7$, mit $P_1,P_2 \in R_1; P_6,P_7 \in R_2; P_3,P_4 \in R_3; P_5,P_8,P_9 \in R_4; R_3,R_4 \in R_5; R_1,R_2 \in R_6$ und $R_5,R_6 \in R_7$.

1. Veranschaulicht den daraus resultierenden R-Baum graphisch und gebt eine Darstellung der Punkte und Rechtecke in der Ebene an.
2. Gebt die Minimum-Bounding-Rechtecke für $R_1,\dots,R_6$ an.
3. Wendet den RKV-Algorithmus an, um den nächsten Nachbarn für den Punkt $Q = (4,3)$ zu suchen.

Musterlösung:

1. Lösungen Teilaufgabe 3.1

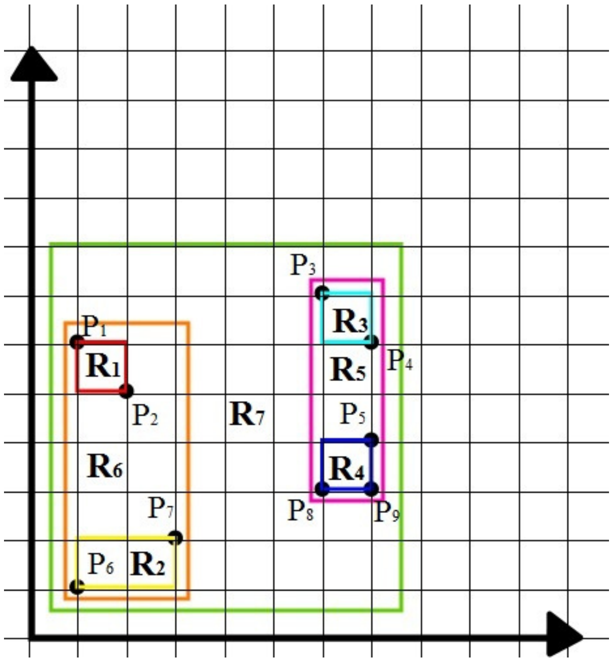

Abbildung 1: Suchbaum (Schaubild den Lösungen der Gruppe *Fitzke / Lubina* entnommen. Danke.)

2. Lösungen Teilaufgabe 3.2
   $MBR(R_1) = (1,5)^T, (2,6)^T$
   $MBR(R_2) = (1,1)^T, (3,2)^T$
   $MBR(R_3) = (6,6)^T, (7,7)^T$
   $MBR(R_4) = (6,3)^T, (7,4)^T$
   $MBR(R_5) = (6,3)^T, (7,7)^T$
   $MBR(R_6) = (1,1)^T, (3,6)^T$

3. Lösungen Teilaufgabe 3.3
   Start bei R_7, Berechnung der Branchlist und Distanzen für Kindknoten R_5, R_6.
   $MINDIST_{R_5}((4,3)^T, ((6,3)^T, (7,7)^T)) = |4-6|^2 + |3-3|^2 = 4$
   $MINDIST_{R_6}((4,3)^T, ((1,1)^T, (3,6)^T)) = |4-3|^2 + |3-3|^2 = 1$
   $MINMAXDIST_{R_5}((4,3)^T, ((6,3)^T, (7,7)^T)) = \min\{|4-6|^2 + |3-7|^2, |3-3|^2 + |4-7|^2\} = \min\{20, 9\} = 9$
   $MINMAXDIST_{R_6}((4,3)^T, ((1,1)^T, (3,6)^T)) = \min\{|4-3|^2 + |3-6|^2, |3-1|^2 + |4-1|^2\} = \min\{10, 13\} = 10$

   $RKV(R_7) : branchList = \{R_5, R_6\} \rightarrow^{\text{Sort } MINDIST} \{R_6, R_5\} \rightarrow^{\text{Strategie 1}} \{R_6, R_5\} \rightarrow^{\text{Strategie 2}} obereGrenze = 9 \rightarrow^{\text{Strategie 3}} \{R_6, R_5\}$. Es wird mit R_6 weitergearbeitet, da es sich an erster Position in der *branch list* befindet.

   $MINDIST_{R_1}((4,3)^T, ((1,5)^T, (2,6)^T)) = 8$

$MINDIST_{R_2}((4,3)^T, ((1,1)^T, (3,2)^T)) = 2$
$MINMAXDIST_{R_1}((4,3)^T, ((1,5)^T, (2,6)^T)) = 13$
$MINMAXDIST_{R_2}((4,3)^T, ((1,1)^T, (3,2)^T)) = 5$

$RKV(R_6) : branchList = \{R_1, R_2\} \rightarrow^{\text{Sort } MINDIST} \{R_2, R_1\} \rightarrow^{\text{Strategie 1}} \{R_2, R_1\} \rightarrow^{\text{Strategie 2}}$ $obereGrenze = 5 \rightarrow^{\text{Strategie 3}} \{R_2, R_1\}$. Berechnung der Distanz zu Blättern in R_2, z.B. euklidische:

$P_6 : d_{L_2}((4,3)^T, (1,1)^T) = \sqrt{|4-1|^2 + |3-1|^2} = \sqrt{13}$
$P_7 : d_{L_2}((4,3)^T, (3,2)^T) = \sqrt{|4-3|^2 + |3-2|^2} = \sqrt{2}$

Basierend darauf ist aktuell der nächste Nachbar $nn = P_7$ und $obereGrenze = \sqrt{2}$. Backtracking zu R_6, dort befindet sich nur noch R_1 in der *branch list*, die wird nach Strategie 3 gekürzt und ist leer. Backtracking zu R_7, dort ist R_5 noch Teil der *branch list*, nach Strategie 3 fällt es raus, der Algorithmus endet mit $nn = P_7$.