Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Luca Pankratz

Lower Bounds für die Quadratic Form Distance

Gliederung

1. Einführung QFD
2. Einführung Indexing
3. Average Color Histogram
4. Multi-Step Similarity Query Processing
5. Qmap-Modell
6. Quellen

Einführung QFD

- Ziel: Schnelles Query Processing
- Lp-Distanzen gehen davon aus, dass alle Dimensionen des Vektorraumes nicht miteinander korrelieren
- Die Quadratic Form Distance (QFD) geht durch die Ähnlichkeitsmatrix A auf Korrelationen ein
- Ähnlichkeitsmatrix ist meist kein Query-Parameter und unabhängig von den Bilddaten
- Unterscheidung statische und dynamische QFD-Matrizen

$$\mathrm{L}_p(u,v) = \left(\sum_{i=1}^{n} |u_i - v_i|^p\right)^{1/p}, p \geq 1$$

Abb.1 Minkowski Distanzen

$$QFD_A(u,v) = \sqrt{(u-v)A(u-v)^T}$$

Abb.2 QFD-Formel

$$A = \begin{array}{c} \\ R \\ G \\ B \end{array} \begin{array}{c} \begin{array}{ccc} R & G & B \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{pmatrix} \end{array}$$

Abb.3 Ähnlichkeitsmatrix

$$A_{ij} = 1 - \frac{d_{ij}}{d_{max}}, \text{where } d_{max} = max_{i,j}(d_{ij})$$

Abb.4 Bsp. Matrixdefinition für RGB Bilder

Einführung QFD

Erstellung dynamischer QFD-Matrizen:

- MindReader berücksichtigt die Wahrnehmung der Nutzer und definiert darauf basierende wichtige Attribute. Die Attribute werden auf Korrelationen untersucht.
- Relevance Feedback Techniques
- Signature Quadratic Form Distance (SQFD)

$$SQFD(u,v) = \sqrt{(u| - v)A(u| - v)^T}$$

Abb.5 SQFD

Einführung Indexing

Spatial Access Methods (SAM):

- Vektorraum wird unabhängig von der Distanzfunktion betrachtet
- R-tree, X-tree, VA-file…
- Distanzfunktion kann zur query time als Parameter übergeben werden
- Eignet sich für dynamische A-Matrizen

Metric Access Methods (MAM):

- Nur Distanzen zwischen Vektoren können genutzt werden, nicht die Koordinaten
- M-tree, M-Index, Pivot Tables…
- In Clustern organisiert
- Vektor Dimensionalität ist irrelevant
- Eignet sich für statische A-Matrizen

Einführung Indexing

- QFD Berechnung ist aufwändig mit $O(n^2)$
- Sehr langsam bei Dimensionalität (n>100) Histogrammen
- Lower-bounding approaches sind notwendig

Average Color Histogram - 1995 Hafner et al.

- Für jedes color histogram x wird die average color x_{avg} berechnet
- Die Average-Color-Distanz repräsentiert einen lower bound für die Histogram-Distanz
- Dies garantiert keine false drops im filter step zu haben
- Verallgemeinerung von d_{avg} ist die k-dimensional distance function

$$\mathbf{x}_{avg} = \mathbf{C}\mathbf{x}, \quad \mathbf{y}_{avg} = \mathbf{C}\mathbf{y}.$$

$$d^2_{\text{avg}} = \left(\mathbf{x}_{avg} - \mathbf{y}_{avg}\right)^T \left(\mathbf{x}_{avg} - \mathbf{y}_{avg}\right) = \mathbf{z}^T \mathbf{C}'^T \mathbf{C}'\mathbf{z}.$$

$$\lambda_A d^2_{\text{avg}}(x, y) \le d^2_{\text{hist}}(x, y)$$

K-Index-Distanz - 1995 Hafner et al.

- Die k-index Einträge sind Resultat einer Reduzierung der Dimensionen, so dass d_k gleich zum Euklidischen Abstand ist
- Der Parameter K ist anpassbar und wirkt sich auf die Filterselektivität und die Rechenleistung aus
- Nachteil: Matrix A kann nach Indexierung nicht verändert werden

$$d_k^2(x, y) \leq d_{\text{hist}}^2(x, y)$$

K-dimensional Indexing

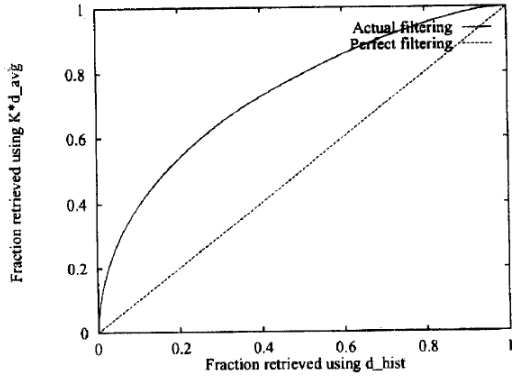


Fig. 1. d_{hist} vs. d_{avg} retrieval.

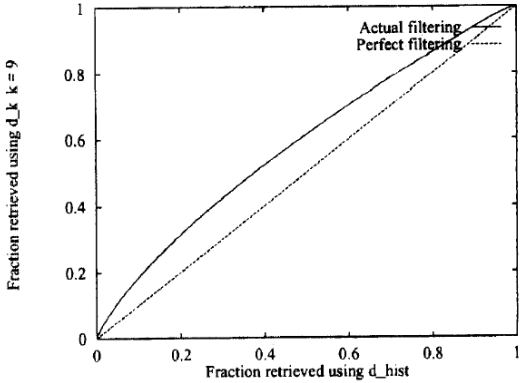


Fig. 4. d_{hist} vs. d_k retrieval for $k = 9$.

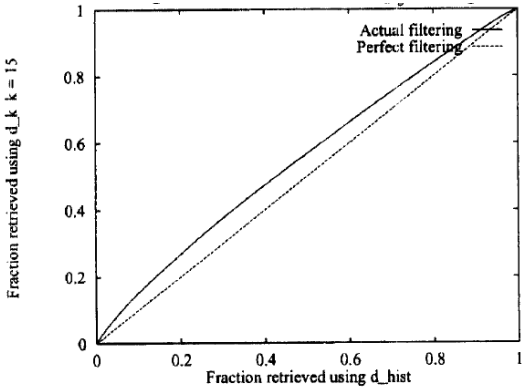
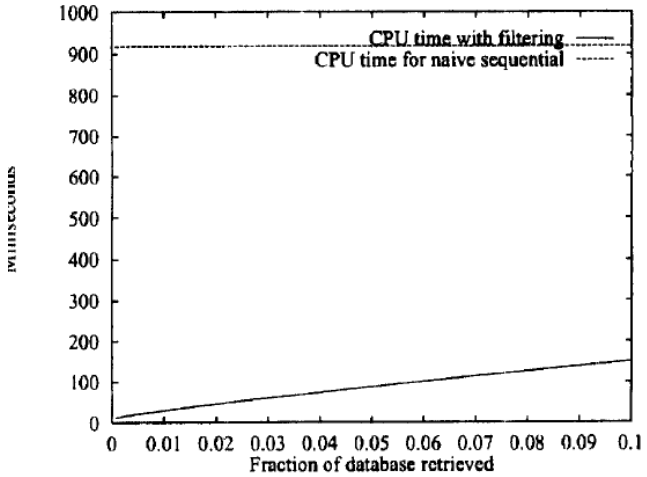


Fig. 5. d_{hist} vs. d_k retrieval for $k = 15$.

Abb.6 K-dimensional filtering

K-dimensional Indexing



ig. 8. CPU time for *naive* retrieval vs. filtered retrieval with d_k for $k = 15$.

Abb.7 Rechenleistung k-dimensional filtering

Multi-Step Similarity Query Processing – 1997 Seidl et al.

- Unterscheidung filter und refinement step
- N-Vektoren reduziert zu r-Vektoren sR
- Matrix A_N reduziert zu Query-Matrix A^R
- Komplette Datenbank wird als Matrix betrachtet die zerlegt werden muss
- Hohe Komplexität von $O(m^2+$

REDUCE_MATRIX $(A_N, (R^B)^{-1}) \longrightarrow A_r^R$

(1) *Distance-preserving rotation (cf. Lemma 3):*
Transform A_N to $A_N^R = (R^B)^{-1} \cdot A_N \cdot (R^{B^T})^{-1}$

(2) *Projection (cf. Lemma 4):*
For k from N down to $r+1$, reduce A_k^R to A_{k-1}^R

Algorithm SIM$_{\text{range}}$ $(A_N, A_r^R, q, \varepsilon)$

- *Preprocessing.* Reduce the query point q to qR
- *Filter Step.* Perform an ellipsoid range query on the SAM to obtain $\{s \mid d_{A_r^R}(sR, qR) \le \varepsilon\}$
- *Refinement Step.* From the candidates set, report the objects s that fulfill $d_{A_N}(s, q) \le \varepsilon$

Algorithm SIM$_{\text{k-nn}}$ (A_N, A_r^R, q, k)

- *Preprocessing.* Reduce the query point q to qR
- *Primary Candidates.* Perform an ellipsoid k-nn query around qR with respect to $d_{A_r^R}$ on the SAM
- *Range Determination.* For the primary candidates s, determine $d_{\max} = \max\{d_{A_N}(s, q)\}$
- *Final Candidates.* Perform an ellipsoid range query $\{s \mid d_{A_r^R}(sR, qR) \le d_{\max}\}$ on the SAM
- *Final Result.* Rank the final candidates s by $d_{A_N}(s, q)$, and report the top k objects

Abb.8 Algorithmen für das Multi-Step Query Processing

Qmap-Modell – 2011 Skopal et al.

- MAM-basierter Ansatz
- Homeomorphic Mapping des QFD-Space auf Euclidian-Space
- Matrix B wird genutzt um Query-Vektoren des QFD-Raumes zu Vektoren des Euclidean-Raumes umzuwandeln
- QFD-Matrix muss positiv-definit und symmetrisch sein

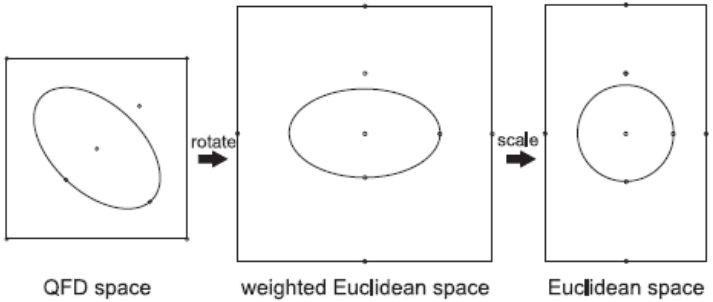


Abb.8 Idee der Qmap

Qmap-Modell – 2011 Skopal et al.

$$QFD(u,v)^2 = (u-v)A(u-v)^T$$

$$BB^T = A$$ ← Cholesky Decomposition

$$QFD(u,v)^2 = (u-v)BB^T(u-v)^T$$

$$L_2(u',v')^2 = (u'-v')(u'-v')^T$$

$$\begin{aligned}(u-v)BB^T(u-v)^T &= [(u-v)B][B^T(u-v)^T] \\ &= [(u-v)B][(u-v)B]^T \\ &= (uB-vB)(uB-vB)^T\end{aligned}$$

$(CD)E = C(DE)$, $D^TC^T = (CD)^T$, $(C-D)E = CE- DE$

Qmap-Modell – 2011 Skopal et al.

Table 1: Indexing Time Complexity Comparison

Method (model)	Indexing	Better
seq. file (QFD)	$O(mn)$	QFD
seq. file (QMap)	$O(mn^2)$	
Pivot tables (QFD)	$O(cn^2 + mn(pn))$	QMap
Pivot tables (QMap)	$O(cn + mn(p+n))$	
M-tree (QFD)	$O(mn^2 log(m))$	QMap
M-tree (QMap)	$O(mn^2 + mnlog(m)))$	

Table 2: Querying Time Complexity Comparison

Method (model)	Querying	Better
seq. file (QFD)	$O(mn^2)$	QMap
seq. file (QMap)	$O(mn)$	
Pivot tables (QFD)	$O(n(pn) + mp + xn^2)$	QMap
Pivot tables (QMap)	$O(n(p+n) + mp + xn)$	
M-tree (QFD)	$O(xn^2)$	QMap
M-tree (QMap)	$O(n^2 + xn)$	

Abb.9 Rechenaufwand Indexierung und Querying

Quellen

1. Hafner, J., H.S. Sawhney, W. Equitz, M. Flickner, und W. Niblack. „Efficient Color Histogram Indexing for Quadratic Form Distance Functions“. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17, Nr. 7 (Juli 1995): 729–36.
2. Seidl, Thomas, und Hans-Peter Kriegel. „Efficient User-Adaptable Similarity Search in Large Multimedia Databases“, o. J., 10.
3. Skopal, Tomáš, Tomáš Bartoš, und Jakub Lokoč. „On (Not) Indexing Quadratic Form Distance by Metric Access Methods“. In *Proceedings of the 14th International Conference on Extending Database Technology - EDBT/ICDT ’11*, 249. Uppsala, Sweden: ACM Press, 2011.