

Verfahren zur Ähnlichkeitssuche auf 3D-Objekten^{*}

Martin Heczko¹, Daniel A. Keim², Dietmar Saupe³, and Dejan V. Vranic³

¹ Institut für Informatik, Universität Halle heczko@informatik.uni-halle.de

² AT&T Shannon Research Labs, USA and Universität Konstanz
keim@informatik.uni-konstanz.de

³ Institut für Informatik, Universität Leipzig
{saupe|vranic}@informatik.uni-leipzig.de

Zusammenfassung In diesem Papier wird die inhaltsbasierte Ähnlichkeitssuche auf Datenbanken von 3D-Modellen behandelt. Auf Objekte aus 3D-Datenbanken wird traditionell durch angehängte Strukturinformationen sowie Textanmerkungen zugegriffen, was jedoch für viele Anwendungen unzureichend ist und durch eine inhaltsbasierte Suche ergänzt werden muß. Das hier vorgestellte inhaltsbasierte 3D-Modell-Suchsystem sucht ähnliche Modelle anhand eines gegebenen Modells, dessen Formbeschreibung automatisch generiert wird. Die vorgeschlagenen Merkmalsvektoren erfassen die 3D-Form und sind invariant gegenüber Translation, Rotation, Skalierung und Modifikation der Detailgenauigkeit. Geplant ist die Anwendung auf großen verteilten Datenbeständen der Computergrafik (VRML-Daten).

Schlüsselworte: 3D - Modell - inhaltsbasiert - Suchsystem - Merkmalsvektoren - Invarianzen

1 Einleitung

Die Menge audiovisueller Informationen, die in digitaler Form verfügbar ist, nimmt stark zu. Deshalb werden Systeme entwickelt, die eine Suche nach den entsprechenden Inhalten ermöglichen. Für die Suche können entweder Meta-Informationen benutzt werden, oder aber die Daten selbst. Inhaltsbasierte Suchsysteme wurden hauptsächlich für Bilder [32, 35, 37, 38] und Audio- und Videodaten [12] entwickelt, jedoch fehlt es an inhaltsbasierten 3D-Modell-Suchsystemen. Durch die in diesem Papier vorgestellten Forschungsarbeiten soll diese Lücke geschlossen werden.

Auf Gebieten des Designs, der Herstellung, der Forschung und der Unterhaltung werden oft 3D-Modelle eingesetzt. Der Bedarf an detaillierten 3D-Modellen

^{*} Diese Arbeit wurde gefördert durch das DFG Graduiertenkolleg Wissensrepräsentation an der Universität Leipzig. Wir danken Jörg Richter für die Programmierung der bildbasierten Merkmalsvektoren.

von Gebäuden, Werkzeugen, Flugzeugen, Autos, Buchstaben, Tieren und Menschen ist in den vergangenen Jahren stark angestiegen und auch für die kommenden Jahre ist noch mit einem weiteren kräftigen Anstieg zu rechnen. Im Rahmen des bevorstehenden MPEG-7-Standards [20, 19, 18] (Multimedia Content Description Interface) wird deshalb auch versucht, Werkzeuge zur Beschreibung von Multimediainhalten zu standardisieren. Bisher ist jedoch das 3D-Modell-Schema im Rahmen von MPEG-7 noch weitgehend unspezifiziert und weitere Forschungsanstrengungen sind erforderlich, um geeignete Vorschläge zu entwickeln. Das Problem der effizienten Suche von Multimedia-Dokumenten wird auch diskutiert, wobei der Schwerpunkt aber eher auf zweidimensionalen Strukturen liegt.

Bekannterweise ist die 3D-Ähnlichkeitssuche ein schwieriges Problem, da die Daten wesentlich komplexer als 2D-Daten sind. Die Ähnlichkeit wird mittels verschiedener Maße definiert. Für gewöhnlich wird dabei ein merkmalsbasierter Ansatz verfolgt [13, 33], der im wesentlichen als Filter zur Einschränkung des Suchraumes fungiert. Abbildung 1 zeigt die Grundidee dieses Ansatzes. Mittels einer Merkmalstransformation werden das Datenobjekt beschreibende Merkmale extrahiert, deren Werte zusammen einen Merkmalsvektor ergeben. Diese hochdimensionalen Merkmalsvektoren werden in eine geeignete Indexstruktur abgelegt, mittels derer die Nächste-Nachbarn-Suche effizient unterstützt wird.

Im Bereich von 2D-Pixelbildern werden Merkmale wie Farbhistogramme [10, 36], die Form von Objekten in den Bildern [3, 29, 31] oder die Textur [46, 43] zur Suche herangezogen. Verschiedene Systeme wie z.B. QBIC [4, 13] oder CHARLOT [45] unterstützen die Kombination von Merkmalen. Andere ermöglichen die Suche nach partieller Ähnlichkeit (WALRUS [34]). Viele auf 2D-CAD-Objekten arbeitende Algorithmen verwenden als Ausgangsbasis das umschreibende Polygon [5, 33]. Es werden in regelmäßigen Abständen Punkte herausgegriffen und aus ihnen wird der Merkmalsvektor erzeugt.

Grundsätzlich gibt es verschiedene Ähnlichkeitsbegriffe, gerade für 3D-Objekte. Man kann Projektionen betrachten, Texturen, Formen, Oberflächeneigenschaften u.s.w. Ähnlichkeit kann global gesucht werden wie im Geometriebasierten Ansatz von [24], welcher hierarchische Approximationen des Volumens betrachtet, oder sie kann partiell auf Oberflächensegmenten, wie in [26, 27, 1], wo potentielle Andockstellen im Rahmen einer molekularen Andockvorhersage gesucht werden. Weitere Form-basierte Ansätze finden sich in [2] (3D-Histogramm) oder in [25] (morphologische granulometrische Distanz).

Entscheidend für die Effizienz der Suche in hochdimensionalen Räumen ist die Verwendung einer geeigneten Indexstruktur. Da die herkömmlichen multidimensionalen Indexstrukturen (*R-Baum* [21], *R*-Baum* [6]) für höhere Dimensionen nicht geeignet sind [8, 44], wurden spezielle Zugriffsstrukturen für den hochdimensionalen Fall entwickelt. Hier seien nur einige genannt: *M-Baum* [9], *IQ-Baum* [7], *X-Baum* [8], *VA-File* [44], *TV-Baum* [30]. Andere Ansätze nutzen die starke Korrelationen oder Gruppierungen im Raum und versuchen die Dimensionalität zu reduzieren: *FastMap* [14], *Multidimensional Scaling* [28], *Hauptkomponentenanalyse* [11] und *Faktor-Analyse* [22].

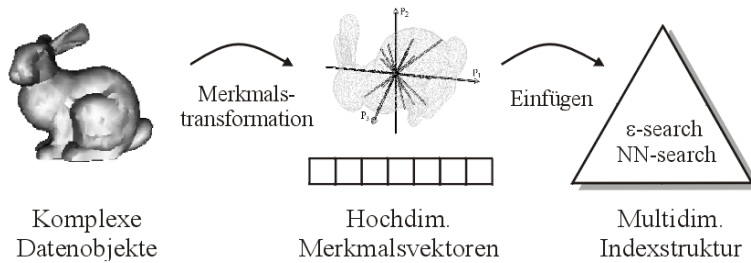


Abbildung 1. Grundidee der Merkmal-basierten Ähnlichkeitssuche

Das vorliegende Papier beschreibt die Implementation eines 3D-Suchsystems, das Suchanfragen mittels gegebener 3D-Modelle unterstützt. Das größte Problem beim Entwurf eines solchen Systems ist die Generierung geeigneter Merkmalsvektoren, die die 3D-Modelle unter den erforderlichen Invarianzen (Translation, Rotation, Skalierung, Detailgenauigkeitsgrad) beschreiben. Weiterhin muß bei der Suche eine geeignete Kombination von Merkmalsvektoren ausgewählt werden, um eine effiziente und effektive Suche zu ermöglichen.

Das Papier ist wie folgt aufgebaut: Abschnitt 2 betrachtet die Probleme der 3D-Ähnlichkeitssuche und präsentiert die prinzipielle Vorgehensweise unseres Lösungsansatzes. Abschnitt 3 erläutert dann die von uns verwendeten Merkmals-transformationen. In Abschnitt 4 wird eine Auswahl der ersten experimentellen Ergebnisse im Hinblick auf Effizienz und Effektivität dargestellt und Möglichkeiten zur Erweiterung des Ansatzes aufgezeigt. Eine abschließende Zusammenfassung befindet sich in Abschnitt 5.

2 Ähnlichkeitssuche auf 3D-Objekten

2.1 Probleme

Die wichtige Teilaufgabe ist die Erstellung eines Satzes von Merkmalen, die eine umfassende Beschreibung der 3D-Modelle ermöglichen. Einige einfache Merkmale wie z.B. Anzahl der Punkte und Polygone, Fläche, umgebendes Rechteck, Geschlossenheit, Volumen u.s.w., sind zwar für spezielle Anfragen hilfreich, sind jedoch nicht für eine effektive inhaltsbasierte Ähnlichkeitssuche geeignet. Mehr Nutzen bringen Merkmale, die sich aus der 3D-Form der Objekte ergeben, insbesondere wenn sie invariant gegenüber Rotation, Translation, Skalierung und Detailauflösung sind. Die Merkmale sollten zudem möglichst robust gegenüber kleinen Veränderungen der Geometrie und Topologie des Modells sein.

Die Wahl der Gestaltmerkmalsvektoren ist für die Ähnlichkeitssuche von zentraler Bedeutung; ihre Auswahl hat entscheidenden Einfluß auf Möglichkeiten der Suchverfahren und deren Effektivität und Effizienz. Zur Auswahl stehen verschiedene geometrie- und bildbasierte Gestaltmerkmalsvektoren. Geometrie-basiert sind z.B. die Multiskalenapproximation des Volumenmodells, die aus Po-

lygonmodellen gewonnen werden kann. Ein bildbasierter Gestaltmerkmalsvektor entsteht zum Beispiel durch das Betrachten der Eigenschaften von 2D-Bildern (Flächenverhältnis, Form etc.), die durch das Abbilden der 3D-Modelle aus mehreren Blickwinkeln entstehen. Einzelne Gestaltmerkmalsvektoren betrachten jeweils isoliert einzelne Eigenschaften. Durch eine Kombination von Merkmalsvektoren ist eine flexiblere Ähnlichkeitssuche möglich, die die Nachteile der einzelnen Merkmalstransformationen vermeiden hilft.

Abhängig von Anfrageobjekt sind in der Regel verschiedene Kombinationen von Merkmalsvektoren sinnvoll. Die Komplexität bei der Verwendung aller Merkmale gleichzeitig ist im allgemeinen zu hoch. Es ist ein schwieriges Problem, eine geeignete Kombination von Merkmalsvektoren für eine effektive Ähnlichkeitssuche zu bestimmen. Nicht jeder Merkmalsvektor muß für die angestrebte Ähnlichkeit relevante Informationen enthalten. Die Verwendung von nicht-relevanten Merkmalsvektoren führt sogar zu einer negativen Beeinträchtigung der Effektivität. Bei der Auswahl der Merkmalsvektoren gilt es zusätzlich, einen Kompromiß zwischen der Genauigkeit der Repräsentation und der Komplexität der Speicherung und Suche zu finden. Eine zu hohe Dimensionalität führt bekanntermaßen zu einer erheblichen Beeinträchtigung der Effektivität und Effizienz der Nächste-Nachbarn-Suche (Fluch der Dimensionalität)[8, 44].

2.2 Lösungsansatz

Unser Ansatz zur Lösung des 3D-Ähnlichkeitssuchproblems ist ein merkmalsbasierter Ansatz (vergleiche Abbildung 1). Die Idee ist, ein möglichst breites Spektrum verschiedener Merkmalsvektoren, die Teilaspekte der 3D-Modelle beschreiben, zu betrachten. Die Verwendung von vielen verschiedenen Merkmalsvektoren ist im Fall der 3D-Ähnlichkeitssuche dringend erforderlich, da einzelne Merkmalsvektoren immer nur einen kleinen Teil der komplexen 3D-Modelle beschreiben können. Zur Suchzeit muß dann allerdings - in Abhängigkeit vom aktuellen Suchobjekt - eine geeignete Kombination der vorhandenen Merkmalsvektoren ermittelt und die Nächste-Nachbarn bezüglich dieser Kombination bestimmt werden. Bevor wir die verwendeten Merkmalsvektoren sowie die Ermittlung geeigneter Kombinationen von Merkmalsvektoren bei der Suche genauer beschreiben, sollen zunächst kurz die Verfahren zur Erreichung der erforderlichen Invarianzen (Translation, Rotation, Skalierung und Detailgenauigkeitsgrad) diskutiert werden.

Einige der **Invarianzeigenschaften** können durch die Anwendung einer 3D-Karhunen-Loeve-Transformation (KLT) erreicht werden. Die Eingabe für die KLT besteht grundsätzlich aus einer Menge von (hier dreidimensionalen) Vektoren, die aus gewählten Punkten des 3D-Modells gewonnen werden¹. Die sich aus der KLT ergebende Transformation des Raumes wird auf das Objekt angewendet, welches sich dann in einer kanonischen Lage und Orientierung befindet. Die Invarianz bezüglich Skalierung der Objekte wird durch Normierung

¹ Im Fall der 3D-Modelle bietet es sich z.B. an, die Eckpunkte der Polygone zu wählen.

der geometrischen Merkmalsvektoren bzw. für den Fall der bild-basierten Merkmalsvektoren durch eine weitere Transformation der Struktur in ein Einheitsvolumen gewährleistet. Die Invarianz bezüglich der Detailgenauigkeit (“level-of-detail”) wird dadurch erreicht, daß in der KLT die Eingangsvektoren (Ecken des 3D-Modells) proportional zu der Fläche der angrenzenden Polygone gewichtet werden. Erwartungsgemäß liefern feinere Abtastungen bessere Ergebnisse bei den Merkmalsvektoren. Daher ist es von Vorteil, vor der Extraktion der Eckpunkte für die KLT vergleichsweise große Polygone in kleinere zu unterteilen, was allerdings eine erhöhte Komplexität zur Folge hat. Nach der geometrischen Transformation zur Gewährleistung der Invarianzen kann dann als nächstes die Merkmalsextraktion durchgeführt werden.

Wir benutzen drei verschiedene Arten von **geometriebasierten Merkmalsvektoren**: auf Abtastung basierend, Volumen-basiert und Voxel-basiert. Für die auf Abtastung basierenden Merkmale schneiden wir die vom Objektzentrum in bestimmte Richtungen gerichteten Strahlen mit den Objektpolygonen. Der Abstand der Schnittpunkte zum Zentrum ergibt die Komponenten des Merkmalsvektors. Anschließend werden die Vektoren auf Einheitslänge skaliert, um die Skalierungsinvarianz zu erhalten. Der zweite Merkmalsvektor basiert auf der Geometrie und wird erhalten durch die Erweiterung eines jeden Polygons des Objektgitters zu einem Polyeder. Der Polyeder entsteht durch Hinzufügen des Objektzentrums als weiteren Punkt. Die resultierenden Volumina der Polyeder ergeben den Merkmalsvektor. Der dritte Merkmalsvektor ergibt sich aus der Rasterung der Polygone in ein Volumen (in “Voxels”), d.h. das 3D-Modell wird durch eine Menge binärer Voxel dargestellt.

Bildbasierte Merkmalsvektoren ergeben sich aus 2D-Bildern des Modells, die man durch das Abbilden der 3D-Modelle aus verschiedenen Richtungen erhält. Im derzeitigen System verwenden wir 3 oder 6 Bilder – abhängig von der Art der betrachteten Merkmale und den Richtungen (positive und negative Richtung der Hauptachsen). Ein Merkmalsvektor ergibt sich aus den Fourier-Koeffizienten der drei Fourier-transformierten Silhouetten. Die Anzahl der genommenen Koeffizienten bestimmt die Größe des Vektors. Ein anderer bildbasierter Merkmalsvektor wird aus 6 Grauwert-Bildern generiert. Jeder Bildpunkt wird durch ein Byte repräsentiert und der Wert kodiert den Abstand zur entsprechenden Seite des umgebenden Rechtecks. In anderen Worten, es werden sechs Z-Puffer durch Grauwert-Bilder repräsentiert. Die Details und Beispiele der verwendeten Merkmalstransformationen werden in Abschnitt 3 beschrieben.

Auswahl der Merkmale. Die Verwendung vieler verschiedener Merkmalsvektoren erfordert bei der Suche eine Auswahl und Kombination der Merkmalsvektoren. Bei den dabei verwendeten Verfahren muß zu einem darauf geachtet werden, daß bei der Suche die Effektivität gewährleistet bleibt, d.h. daß die Suche die gewünschten Ergebnisse liefert; zum anderen muß aber auch die Effizienz gewährleistet bleiben, d.h. die Suchzeit muß in gewissen Grenzen bleiben. Beide Aspekte sind insbesondere bei hochdimensionalen Merkmalsvektoren zu beachten, da durch den sogenannten “Curse of Dimensionality” in hochdimensionalen

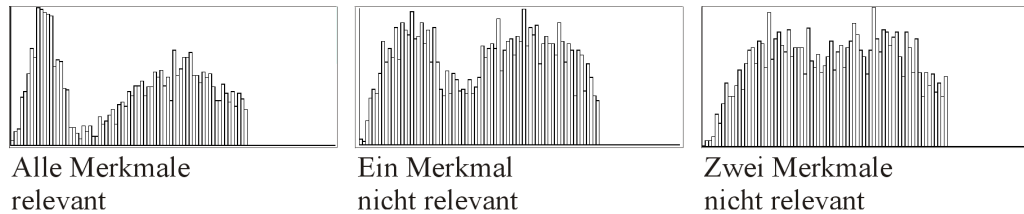


Abbildung 2. Die Distanz-Verteilung bei unterschiedlicher Anzahl relevanter Merkmale

Räumen die Effizienz und Effektivität der Nächste-Nachbarn Suche degeneriert [8, 44].

Ein naiver Ansatz zur Suche auf beliebigen Kombinationen von Merkmalsvektoren ist die getrennte Suche auf den einzelnen Merkmalsvektoren und die Kombination der Ergebnisse. Dieses Verfahren ist aber weder von der Effizienz noch von der Effektivität her sinnvoll. Die Effizienz wird durch die unnötige Suche auf den nicht-relevanten Merkmalsvektoren sowie die potentiell sehr aufwendige Kombination der Einzelergebnisse beeinträchtigt und die Effektivität kann durch Verwendung von nicht-relevanten Merkmalsvektoren sehr stark beeinträchtigt werden.

Die Idee zur Lösung des Problems ist die Ermittlung der relevanten Merkmalsvektoren aufgrund der Verteilung der Distanzen der Merkmalsvektoren vom Anfragepunkt. Die Grundidee ist ähnlich zu der in [23] beschriebenen generalisierten Nächsten-Nachbarn-Suche. Dabei werden die Gesamtdistanzen einer Kombination von Merkmalsvektoren zum Anfragepunkt berechnet, zum Beispiel mit Hilfe einer (gewichteten) Linearkombination und einem euklidischen Abstandsmaß im kombinierten Merkmalsraum. Aufgrund der Verteilung der Distanzen dann entschieden, ob die Kombination für das Anfrageobjekt relevant ist oder nicht. Dem Ansatz liegt die Beobachtung zu Grunde, daß die Distanzverteilung sich in Abhängigkeit von der Anzahl der relevanten Merkmale deutlich ändert. In Abbildung 2 sind drei Distanzverteilungen für Daten mit 10 Merkmalen abgebildet. Im ersten Bild sind alle 10 Merkmale auch relevant, wohingegen in zweiten Bild nur 9 Merkmale relevant sind und im dritten Bild nur 8 Merkmale. Die Veränderung der Distanzverteilung ist deutlich zu erkennen.

Ziel der automatischen Erkennung relevanter Merkmale ist es nun, Distanzverteilungen zu ermitteln, bei denen möglichst viele Merkmale relevant sind (siehe Abbildung 2). Um das zu erreichen, muß man den Fall der Abbildung 2 (links) mit mehreren Maxima von dem in Abbildung 2 (rechts) mit einem Maximum unterscheiden. Formal kann die Bewertung der Distanzverteilung mit Hilfe von Glättungsfaktoren (σ) der Kurve vorgenommen werden. Um kleinere Schwankungen der Distanzverteilung auszugleichen, wird die Distanzverteilung zunächst solange geglättet, bis die Kurve nur noch zwei Maxima enthält. Der entsprechende Glättungsfaktor sei σ_{min} (siehe Abbildung 3). Dann wird weiter geglättet bis die Kurve nur noch ein Maximum enthält. Der entsprechende

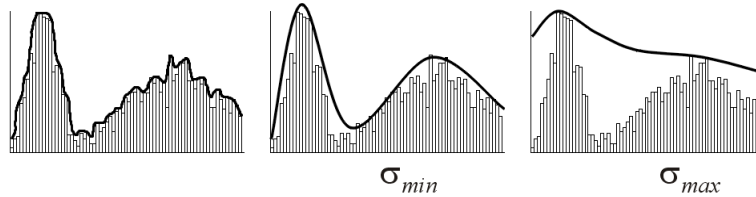


Abbildung 3. Glättung bei verschiedenen Werten für σ

Glättungsfaktor sei σ_{max} (siehe Abbildung 3). Die Güte der aktuell betrachteten Kombination von Merkmalsvektoren kann nun durch

$$K = \sigma_{max} - \sigma_{min}$$

definiert werden. Bei großem K enthält die Kurve mehrere Maxima und damit sind alle betrachteten Merkmale relevant, wohingegen bei kleinem K die betrachteten Merkmale nur zum Teil relevant sind. Folglich: je größer K ist, desto höher ist die Güte der betrachteten Merkmale. Eine Bewertung durch dieses Qualitätsmaß erlaubt nun eine Auswahl geeigneter Merkmalskombinationen. Als Algorithmus zur Bestimmung dieser Merkmalskombination kann ein heuristischer Optimierungsalgorithmus – ähnlich zu dem in [23] beschriebenen Algorithmus, der eine auf der Güte basierende Greedy-Auswahlstrategie mit einem genetischen Suchverfahren kombiniert – verwendet werden.

3 3D Gestaltmerkmalsvektoren

3.1 Kanonische Lage und Orientierung

Um die Invarianz bezüglich Translation, Rotation und Spiegelung zu erlangen, wird — wie im vorherigen Abschnitt schon beschrieben — eine Hauptachsentransformation (KLT) durchgeführt, bei der die Eingabevektoren geeignet gewichtet werden, um auch die Invarianz bei Polygonsimplifizierung bzw. Verfeinerung des Modells zu erhalten. In diesem Abschnitt geben wir dazu Details an, wobei wir unserer vorherigen Arbeit [39] folgen.

Zur Vereinfachung der Darstellung beziehen wir uns im folgenden nur auf den Fall von 3D-Modellen, die in Form von Triangulierungen gegeben sind, d.h. unsere Modelle bestehen aus einer Menge von Dreiecken in 3D, den zugehörigen Kanten und Ecken. Verallgemeinerungen auf allgemeine Polygonmodelle sind möglich.

Sei $P = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}$ die Menge der Eckpunkte des Modells und sei $D = \{D_1, \dots, D_F\}$ die Menge der Dreiecke. Die Eckpunkte gehen als Eingabe in die KLT ein, die eine affine Transformation τ ergibt, welche, angewendet auf die Eckpunkte die transformierte Punktmenge $P' = \{\mathbf{p}'_1, \dots, \mathbf{p}'_N\}$ produziert. Die Transformation τ muß so konstruiert sein, daß für eine beliebige Konkatenation

σ von Rotationen, Translationen und Spiegelungen an beliebigen Ebenen die Gleichung

$$P' = \tau(P) = \tau(\sigma(P))$$

gilt und somit die gewünschte Invarianz bezüglich aller Transformationen vom Typ σ gegeben ist.

Wir stellen uns vor, daß die Dreiecke aus infinitesimal dünnem, aber Masse-behaftetem Material beschaffen sind. Die Masse jedes Dreiecks ist also proportional zur Fläche und wird zu gleichen Teilen auf deren drei Eckpunkte verteilt. Der Massenschwerpunkt \mathbf{c} des Punktmassenmodells ergibt sich somit zu:

$$\mathbf{c} = \frac{1}{N} \sum_{i=1}^N w_i \mathbf{p}_i, \quad w_i = \frac{N \cdot S_i}{3S}, \quad i = 1, \dots, N, \quad (1)$$

wobei w_i das der Ecke \mathbf{p}_i zugeordnete Gewicht ist. S bezeichnet die Masse (bzw. Fläche) aller Dreiecke zusammen und S_i ist die Summe der Flächen der Dreiecke mit Eckpunkt \mathbf{p}_i . Somit ist das Gewicht w_i gegeben durch die der i -ten Ecke zugeordnete Masse normalisiert durch die mittlere Masse pro Eckpunkt, S/N . Es gilt $\sum_{i=1}^N w_i = N$.

Man kann sich leicht davon überzeugen, daß eine Zerlegung eines der Objektdreiecke in mehrere, d.h. eine Verfeinerung der Triangulierung das oben definierte Massenzentrum invariant läßt.

Die Kovarianzmatrix C für die Punktmenge P wird ebenfalls unter Berücksichtigung der Eckengewichte berechnet,

$$C = \frac{1}{\sum w_i} \sum_{i=1}^N w_i (\mathbf{p}_i - \mathbf{c}) \cdot (\mathbf{p}_i - \mathbf{c})^T.$$

Die Gewichte bewirken, daß die Kovarianzmatrix bei Verfeinerungen oder Simplifizierungen des geometrischen Modells ungefähr gleich bleibt. Die Invarianz ist in diesem Fall also nur approximativ gegeben. Erwartungsgemäß liefern feinere Abtastungen bessere Ergebnisse bei den Merkmalsvektoren. Daher ist es von Vorteil, vor der Extraktion der Eckpunkte für die KLT vergleichsweise große Polygone in kleinere zu unterteilen, was allerdings eine erhöhte Komplexität bedingt. Eine geeignete Strategie, die wir implementiert haben, verfeinert das Modell so, daß die entstehenden Dreiecke etwa gleich groß sind. Noch weitergehend ist es möglich, infinitesimal kleine Dreiecke zu betrachten, bzw. zu einer Integraldarstellung einer "kontinuierlichen" KLT überzugehen [42]. Diese Integrale sind explizit berechenbar und führen zu einer weiteren Verbesserung.

Nach der Berechnung der 3×3 Kovarianzmatrix werden deren Eigenwerte und normalisierte Eigenvektoren bestimmt. Die Karhunen-Loeve-Transformation, angewendet auf die Eckpunkte des Modells, ergibt sich zu

$$\mathbf{p}'_i = A(\mathbf{p}_i - \mathbf{c}), \quad i = 1, \dots, N.$$

wobei die Zeilen der Matrix A aus den Eigenvektoren bestehen, die gemäß der absteigenden Beträge der Eigenwerte sortiert sind. Diese Transformation ist ein-

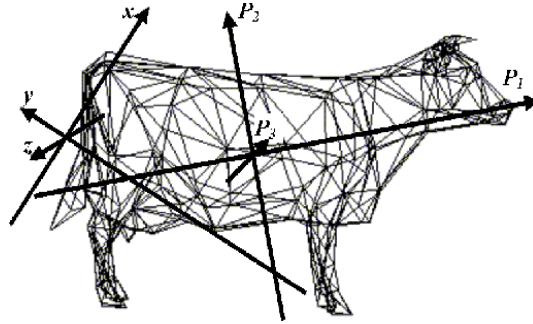


Abbildung 4. Die modifizierte Karhunen-Loeve-Transformation angewendet für ein Modell.

deutig bis auf das Vorzeichen der Eigenvektoren. Um diese Ambiguität zu entfernen, setzen wir die Vorzeichen der Eigenvektoren \mathbf{P}_j , $j = 1, 2, 3$, so, daß

$$\sum_{i=1}^N \text{sign}(p'_{ij}) \cdot (p'_{ij})^2 \geq 0$$

wobei wir die Notation $\mathbf{p}'_i = (p'_{i1}, p'_{i2}, p'_{i3})$, $i = 1, \dots, N$ benutzt haben. Durch diesen Trick wird auch erreicht, daß die KLT und die dann daraus abgeleiteten Merkmalsvektoren invariant bezüglich Spiegelungen ist.

Abbildung 4 illustriert den Effekt der KLT an einem Beispiel. Die Achsen des gegebenen Koordinatensystems für das Objekt sind mit \mathbf{x} , \mathbf{y} , \mathbf{z} bezeichnet und die neuen Achsen nach der KLT heißen \mathbf{P}_1 , \mathbf{P}_2 , und \mathbf{P}_3 .

3.2 Die Merkmalsvektoren

In diesem Papier sind drei geometrische und zwei bildbasierte Merkmalsvektoren vorgeschlagen.

1. Der erste geometriebasierte Merkmalsvektor, siehe Abbildung 5, beschreibt 3D-Objekte durch ihre Ausdehnung in verschiedene Richtungen. Nach der KLT interpretieren wir den Ursprung des neuen Koordinatensystems als Zentrum des 3D-Objektes und legen um das Zentrum eine das Objekt einschließende Kugel. Die Kugelfläche wird nach einer festen Vorschrift abgetastet und Strahlen von den Abtastpunkten in Richtung Zentrum verfolgt. Der erste Schnittpunkt jedes Strahls mit einem der Objektpolygone liefert eine Komponente des Merkmalsvektors. Diese ist definiert als der Abstand des Schnittpunktes zum Zentrum. Wenn kein Schnittpunkt des Strahls mit dem Objekt existiert, dann ist die zugehörige Komponente des Merkmalsvektors gleich Null gesetzt. Die Dimension des Merkmalsvektors ist ein Parameter des Verfahrens.

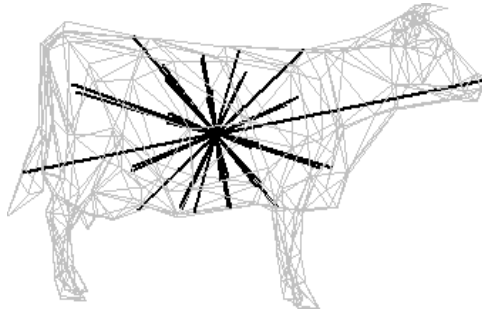


Abbildung 5. Illustration eines einfachen geometrischen Merkmalsvektors.

- Der zweite Merkmalsvektor zielt darauf hin ab, eine bessere Robustheit zu erhalten. Kleine Störungen, z.B. in den Parametern der KLT, können in besonderen Fällen nämlich große Änderungen in den Komponenten des obigen Merkmalsvektors bewirken. Hierzu betrachten wir zu jedem Polygon das Polyedervolumen, das durch Hinzunahme des Zentrums entsteht. Dieses berücksichtigt die Orientierung des Polygons, so daß auch negative Volumen möglich sind. Der das Objekt einschließende Würfel wird in etwa gleich große Pyramiden-ähnliche Segmente unterteilt (die Spitze jeweils im Objektzentrum). Jedes Segment liefert eine Komponente des Merkmalsvektors, die sich aus dem Volumen der Schnittmenge des Kugelsegmentes mit der Vereinigung der Polyedervolumen ergibt. Die Dimension des Merkmalsvektors ist durch die Anzahl der Segmente gegeben und ein Parameter des Verfahrens. Die Abbildung 6 illustriert die Idee in einer 2D-Skizze.

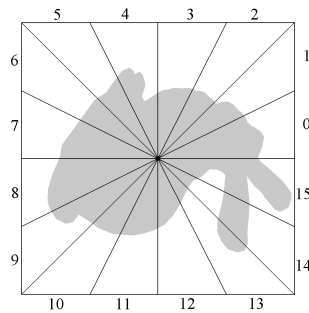


Abbildung 6. Illustration des zweiten Merkmalsvektors: die Volumen in den Segmenten ergeben Komponenten des Vektors.

- Die Polygone werden in ein Volumen gerastert. Hier kann man die anteilige Fläche in jeder Volumenzelle (Voxel) als Skalar akkumulieren oder nur die

Zellbelegung als binären Wert abspeichern. Das entstehende skalare Volumenmodell bzw. die Volumen-Bitmap ist hierarchisch als Octree organisiert. Die daraus entstehenden Vektoren verschiedener Auflösungsstufen fungieren als Merkmalsvektoren. Die Volumen-Bitmap läßt sich auch in den Ortsfrequenzraum transformieren wonach man alternativ Merkmalsvektoren aus Fourierkoeffizienten oder deren Absolutbeträgen zusammensetzen kann [41].

4. Der erste Bild-basierte Ansatz benutzt Abbildungen der Objekte unter Parallelprojektion auf Ebenen orthogonal zu den drei Raumachsen (nach der KLT). Zuvor wird das Objekt noch auf ein Einheitsvolumen (Würfel) skaliert. Aus diesen drei quadratischen Objektbildern werden die Silhouetten berechnet und bei einer beliebigen Stelle beginnend gegen den Uhrzeigersinn abgetastet. Die drei so entstehenden geschlossenen diskreten 2D-Kurven werden Fourier-transformiert und das Power-Spektrum wird berechnet, siehe Abbildung 7. Merkmalsvektoren entstehen durch Extraktion der ersten k Komponenten der drei Powerspektren. Die Dimension $3k$ der Merkmalsvektoren ist wiederum ein Parameter des Verfahrens.

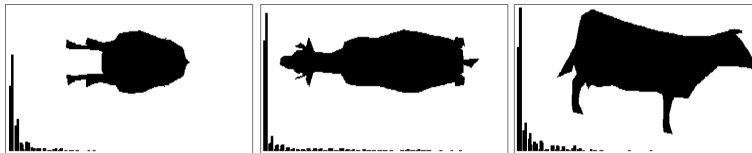


Abbildung 7. Drei Silhouetten und zugehörige Fourierkoeffizienten.

5. Die Abbildungen aus dem letzten Punkt werden ergänzt durch entsprechende Abbildungen aus den genau entgegengesetzten Richtungen und somit ergeben sich insgesamt sechs Ansichten des Objektes welche anhand eines Beispiels in Abbildung 8 dargestellt sind. Diese Grauwertbilder ergeben sich jeweils aus einem Tiefenpuffer (“depth map”) mit geringer Auflösung von je n^2 , wobei Hintergrundpixel den größten Tiefenwert zugewiesen bekommen (255 bei einer Tiefenauflösung von 8 Bit). Die entstehenden Werte können zu einem Merkmalsvektor der Dimension $6n^2$ zusammengesetzt werden. Damit hat dieser Merkmalsvektor eine enge Beziehung zu dem ersten in unserer Liste, der interpretiert werden kann als eine grobe Abtastung der Tiefenpuffer. Alternativ kann anstelle des Tiefenpuffers ein normal gerendertes Bild des Objektes mit normierter Beleuchtung und standardisierten Oberflächeneigenschaften verwendet werden.

Für den Vergleich der Merkmalsvektoren des Anfragemodells mit denen der Modelle im Suchraum ist ein Abstandsmaß erforderlich. Grundsätzlich verwenden wir den l_2 -Abstand (Summe der quadratischen Differenzen). Im Fall der binarisierten Volumendarstellung ist der Hausdorff-Abstand der Voxelmengen, der weniger sensitiv bezüglich kleinen Störungen ist, vorzuziehen.

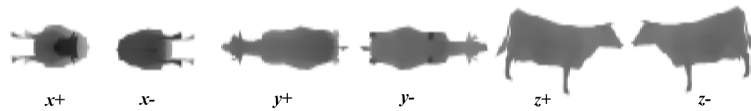


Abbildung 8. Sechs Tiefenpuffer.

Die Merkmalsvektoren haben unterschiedliche Differenzierungseigenschaften. Während die bildbasierten Vektoren und der erste geometriebasierte Merkmalsvektor unserer Liste sich ganz auf die von außen sichtbaren Teile der 3D-Modelle konzentrieren, bieten die anderen geometriebasierten Merkmalsvektoren auch Differenzierungsmöglichkeiten für Objekte, die Polygone in mehreren Schichten enthalten, wie z.B. in Maschinen- oder Architekturmodellen. In dem Fall sind die inneren Schichten nämlich den äußeren Ansichten verborgen und daher irrelevant für die genannten Merkmalsvektoren. Je nach Anfrageobjekt sind also verschiedene Merkmalsvektoren unterschiedlich gut geeignet für die Ähnlichkeitssuche. Zum Beispiel bei Anfrageobjekten, die viele relevante Konkavitäten aufweisen, sind die bildbasierten Merkmalsvektoren eher ungeeignet und die Voxel-basierte Methode ist wahrscheinlich die beste.

4 Ergebnisse und Erweiterungen

4.1 Experimentelle Ergebnisse

Unsere Sammlung von 3D-Modellen umfaßt zur Zeit etwa 2000 Objekte, wovon wir in dieser Studie 536 Stück verwendet haben. Die Modelle stammen größtenteils aus VRML-Dateien, die im Internet verfügbar sind. Einige der Modelle haben wir durch Anwendung des Mesh-Simplifizierungsverfahren von QSlim Simplification Software (Version 2.0), siehe [15, 16], in verschiedenen Auflösungsstufen erzeugt und in die Modellbibliothek eingefügt. Weiterhin sind einige Modelle prozedural durch Gaußsches Rauschen verschiedener Stärke perturbiert sowie durch Skalierungen und affine Transformationen modifiziert worden und somit in verschiedenen Versionen in der Sammlung enthalten. Für unsere Implementierung haben wir uns auf vorerst nur ein Dateiformat festgelegt.

Die Oberfläche des Prototyps unseres 3D-Ähnlichkeits-Suchsystems ist in Abbildung 9 zusammen mit einer Beispielanfrage zu sehen. In diesem Papier berichten wir über erste Ergebnisse bezüglich der Komplexität der Verfahren zur Berechnung der kanonischen Lage und Orientierung (modifizierte Karhunen-Loeve-Transformation), der Berechnung der Merkmalsvektoren bei verschiedenen Wahlen von Dimensionen sowie der Nächste-Nachbar-Suche im Merkmalsraum. Untersuchungen zur Evaluierung der Suchergebnisse werden in einem zukünftigen Papier behandelt (siehe nächster Abschnitt).

Zunächst illustrieren wir in unserem ersten Graphen die Verteilung der Komplexität der 536 3D-Modelle unserer Studie. Die Abbildung 10 zeigt ein Histo-

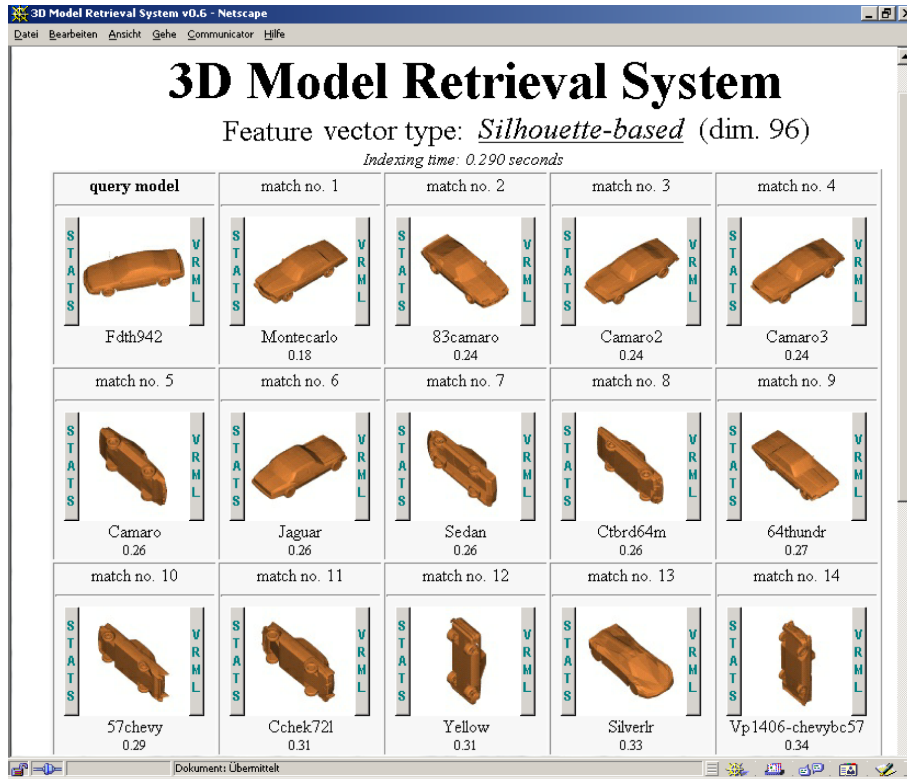


Abbildung 9. Oberfläche des 3D-Ähnlichkeits-Suchsystems.

gramm der Anzahl der Eckpunkte und Dreiecke in den Modellen. Die durchschnittliche Anzahl der Eckpunkte liegt bei 3511, die der Dreiecke 5731.

Die Rechenzeit für die Bestimmung der KLT setzt sich zusammen aus der Bestimmung der Eckpunktgewichte, der Ermittlung des Massenzentrums, der Akkumulation der Elemente der Kovarianzmatrix sowie der Berechnung der Eigenvektoren der Kovarianzmatrix. Die zuerst genannten Anteile skalieren im wesentlichen linear mit der Anzahl der Eckpunkte. Die Kovarianzmatrix hat unabhängig von der Anzahl der Eckpunkte stets die Dimension 3×3 und daher erfordert die Eigenwert- und Eigenvektorberechnung nur einen konstanten und sehr kleinen Rechenaufwand. Somit erwarten wir eine lineare Zeitkomplexität $O(N)$ für die Bestimmung der KLT. Das Diagramm in Abbildung 11 bestätigt diese Überlegungen. Sämtliche Simulationsrechnungen sind auf einem PC mit einem 450 MHz Pentium III Prozessor durchgeführt worden.

Im derzeitigen Suchsystem mit nur 536 3D-Modellen entspricht die Suchzeit im wesentlichen der Zeit für die Merkmalstransformation, da die Nächste-Nachbarn-Suche im Verhältnis dazu vernachlässigbar klein ist. Die Zeit für die

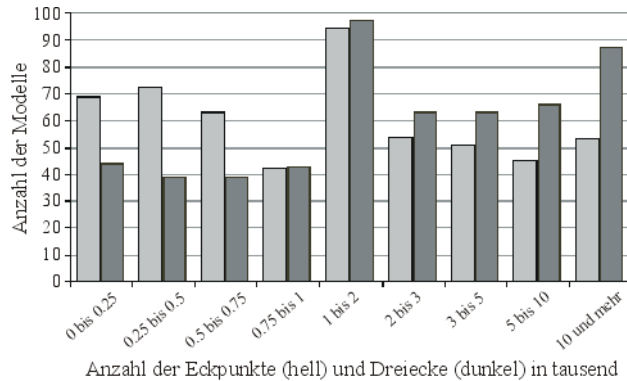


Abbildung 10. Histogramm für die Modellkomplexität.

Merkmalstransformation ist für die verschiedenen Transformationen unterschiedlich (vgl. Tabelle 1).

In diesem Papier sollen die geometriebasierten Merkmalsvektoren in verschiedenen Dimensionen betrachtet werden. Der erste Merkmalsvektor wird wie beschrieben durch Strahlverfolgung bestimmt. Der Rechenaufwand hängt ab sowohl von der Modellkomplexität als auch von der Anzahl der eingesetzten Strahlen, also der Dimension des Merkmalsvektors. Das Problem, Strahlen mit einer Kollektion von Polygonen zu schneiden, ist ein wohlbekanntes Problem der Computergrafik ("ray tracing") und es sind mannigfache Methoden zur effizienten Schnittpunktberechnung entwickelt worden [17], die wir in unserer Implementierung aber noch nicht eingesetzt haben. In der Tabelle geben wir hier nur die mittlere Berechnungszeit pro Objekt an, die im wesentlichen der Suchzeit entspricht. Die Dimensionen des Merkmalsvektors sind 42 (klein), 162 (mittel) und 642 (groß).

Typ	Dimension klein	Dimension mittel	Dimension groß
Strahlenbasiert	0.7s	2.6s	10.4s
Volumenbasiert	1.7s	1.7s	1.7s
3D-Rasterung	1.5s	1.5s	1.5s

Tabelle 1. Mittlere Berechnungszeit in Sekunden für verschiedene Merkmalsvektoren.

Die Dimensionen des Volumenbasierten Merkmalsvektors sind 96 (klein), 384 (mittel) und 1536 (groß) und die aus der 3D-Rasterung in Octrees sind $4^3 = 64$, $8^3 = 512$ und $16^3 = 4096$. Bei den strahlenbasierten Merkmalsvektoren ist die Rechenzeit linear proportional zur Größe des Vektors. Für die anderen Merkmale liegt die konstante Zeit in der Methode begründet.

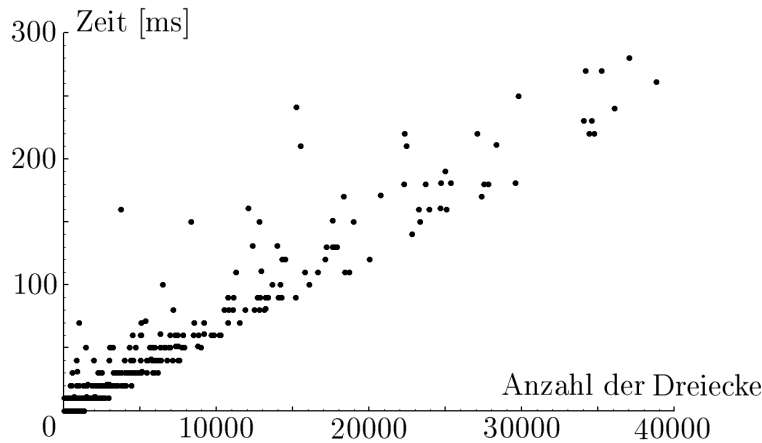


Abbildung 11. Scatter Plot für die Zeitkomplexität der Berechnung der Parameter der modifizierten Karhunen-Loeve Transformation zur Berechnung der kanonischen Lage und Orientierung.

4.2 Erweiterungen und Ausblick

Die nächsten Arbeitsschritte in unserem Forschungsvorhaben bestehen in der Erweiterung der Anzahl der Modelle und der weiteren Verfeinerung der Implementierung zur Auswertung der hier schon teilweise gemachten Komplexitätsstudien. Es besteht weiterhin Forschungsbedarf für eine Methode zur Evaluierung der erzielbaren Ergebnisse. Liefert der nächste Nachbar im Merkmalsraum wirklich das beste zum Anfragemodell passende Objekt? Sind die ersten 5 oder 10 3D-Modelle aus der Suchoperation die besten Objekte und stimmt deren Reihenfolge? Zu diesen Fragen werden wir objektive, d.h. numerisch quantifizierte und reproduzierbare Methoden entwickeln. Der Lösungsansatz hierfür besteht in der Definition einer Referenzlösung ("ground truth") für jedes gegebene Anfrageobjekt, die auf einer Kombination von besonders hochdimensionalen Merkmalsvektoren aufbauen kann. Anhand dieser aufwendig zu berechnenden Musterlösung können Suchergebnisse beurteilt und verglichen werden. Diesen "objektiven" Methoden sollen auch subjektive Evaluationen folgen in denen Juroren Anfrageergebnisse in Testserien am Bildschirm auf ihre Relevanz hin einschätzen.

Eine wichtige Anwendung der Evaluation besteht in der Herausforderung, diejenige Kombination von Merkmalsvektoren und deren Dimensionen zu bestimmen, die unter Einhaltung von Komplexitätsschranken bei Speicher und Rechenzeit die besten Suchergebnisse liefern.

Mit Ausnahme des Voxel-basierten Verfahrens und dem Ansatz über Silhouetten lassen sich die hier vorgestellten Merkmalsvektoren interpretieren als Abtastung einer auf der Einheitssphäre definierten reellen Funktion. Es besteht die Möglichkeit, diese Funktionen mittels sphärischen Fourier- oder Wavelettrans-

formationen in eine Multiresolutiondarstellung zu überführen. Dies wird einen einheitlichen und in der Dimension adaptiven Zugang zu den Merkmalsvektoren ermöglichen [42, 40]. Weitere, hier nicht besprochene Merkmalsvektoren sind denkbar. Zum Beispiel kann die Medial Axis Transformation erwogen werden, um Skelette der Objekte zu erzeugen, welche anstelle geometrischer eher strukturelle Informationen über 3D-Objekte kodieren. Es ergibt sich die Frage, welche Abstandsmaße in Zusammenhang mit diesem Ansatz bei Anwendungen zur Ähnlichkeitssuche sinnvoll sind.

In typischen VRML-Dateien sind 3D-Modelle häufig auch durch Farb- oder Texturinformation (Texturabbildungen mit zweidimensionalen Texturkoordinaten an den Polygoneckpunkten) und durch Normalenvektoren für Illuminationsimulationen ergänzt. In zukünftigen Anwendungen zur Ähnlichkeitssuche werden solchen Daten auch in Merkmalsvektoren einzubeziehen sein.

Eine besondere Herausforderung für 3D-Deskriptoren entsteht durch Anfrage nach partiellen Ähnlichkeiten. Solche Fragestellungen tauchen zum Beispiel in der Molekularbiologie bei Proteinandockproblemen auf.

5 Zusammenfassung

Im vorliegenden Papier haben wir ein System zur Ähnlichkeitssuche auf 3D-Modellen vorgestellt. Grundidee ist die Verwendung einer Reihe verschiedener Merkmalsvektoren, die verschiedene Aspekte der 3D-Modelle repräsentieren und weitgehend invariant gegenüber Transformation, Rotation, Skalierung und Detailgenauigkeitsgrad sind. Erste experimentelle Ergebnisse mit der Kombination von Merkmalsvektoren zeigen vielversprechende Resultate. In zukünftigen Arbeiten soll das System um weitere Merkmalsvektoren ergänzt werden sowie die Effizienz der Verfahren optimiert werden. Weiterer Forschungsbedarf ist bei der Kombination von Merkmalen notwendig, um ein automatisches distanzabhängiges Verfahren zu entwickeln, welches in Abhängigkeit vom Anfrageobjekt die bestmögliche Kombination von Merkmalsvektoren ermittelt.

Literatur

- [1] Mihael Ankerst, Bernhard Braunmüller, Hans-Peter Kriegel, and Thomas Seidl. Improving adaptable similarity query processing by using approximations. In *VLDB'98, Proceedings of 24rd International Conference on Very Large Data Bases, New York City, New York, USA*, pages 206–217. Morgan Kaufmann, 1998.
- [2] Mihael Ankerst, Gabi Kastenmüller, Hans-Peter Kriegel, and Thomas Seidl. 3D shape histograms for similarity search and classification in spatial databases. In Ralf Hartmut Güting, Dimitris Papadias, and Frederick H. Lochovsky, editors, *Lecture Notes in Computer Science*, volume 1651, pages 207–226. Springer, 1999.
- [3] Mihael Ankerst, Hans-Peter Kriegel, and Thomas Seidl. A multistep approach for shape similarity search in image databases. *TKDE*, 10(6):996–1004, 1998.
- [4] Jonathan Ashley, Myron Flickner, James L. Hafner, Denis Lee, Wayne Niblack, and Dragutin Petkovic. The query by image content (QBIC) system. In Michael J. Carey and Donovan A. Schneider, editors, *Proceedings of the 1995 ACM SIGMOD*, page 475. ACM Press, 1995.

- [5] A. Badel, J.P. Mornon, and S. Hazout. Searching for geometric molecular shape complementary using bidimensional surface profiles. *Journal of Molecular Biology*, 10:205–211, December 1992.
- [6] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The R*-tree: An efficient and robust access method for points and rectangles. In *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, Atlantic City, NJ*, pages 322–331. ACM Press, 1990.
- [7] Stefan Berchtold, Christian Böhm, H. V. Jagadish, Hans-Peter Kriegel, and Jörg Sander. Independent quantization: An index compression technique for high-dimensional data spaces. In *Proceedings of the 16. Int. Conference on Data Engineering*, pages 577–588. IEEE Computer Society, 2000.
- [8] Stefan Berchtold, Daniel A. Keim, and Hans-Peter Kriegel. The X-tree: An index structure for high-dimensional data. In T. M. Vijayaraman, Alejandro P. Buchmann, C. Mohan, and Nandlal L. Sarda, editors, *VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases, September 3-6, 1996, Mumbai (Bombay), India*, pages 28–39. Morgan Kaufmann, 1996.
- [9] Paolo Ciaccia, Marco Patella, and Pavel Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece*, pages 426–435. Morgan Kaufmann, 1997.
- [10] Luigi Cinque, Stefano Levaldi, Kai A. Olsen, and A. Pellicanò. Color-based image retrieval using spatial-chromatic histograms. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, volume II, pages 969–973. IEEE Computer Society, 1999.
- [11] G. Dunn and B. Everitt. *An Introduction to Mathematical Taxonomy*. Cambridge University Press, Cambridge, MA, 1982.
- [12] J.-R. Ohm et al. A multi-feature description scheme for image and video database retrieval. In *IEEE Multimedia Signal Processing Workshop, Copenhagen*. IEEE Computer Society, September 1999.
- [13] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3:231–262, 1994.
- [14] Christos Faloutsos and King-Ip Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In Michael J. Carey and Donovan A. Schneider, editors, *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, California, May 22-25, 1995*, pages 163–174. ACM Press, 1995.
- [15] Michael Garland. Quadric-based polygonal surface simplification. Technical Report CMU-CS-99-105, Carnegie Mellon School of Computer Science, 1999. Ph.D. Dissertation.
- [16] Michael Garland and Paul S. Heckbert. QSlim simplification software. <http://www.cs.cmu.edu/afs/cs/user/garland/www/quadrics/qslim20.html>.
- [17] Andrew S. Glassner. *An Introduction to Ray Tracing*. Academic Press, 1989.
- [18] MPEG DDL Group. Mpeg-7 description. Definition Language Document V2. Doc.ISO/MPEG N2997, Melbourne, October 1999.
- [19] MPEG Requirements Group. Mpeg-7. Requirements Document V.10. Doc. ISO/MPEG N2996, Melbourne, October 1999.
- [20] MPEG Requirements Group. Overview of the mpeg-7 standard. Technical Report Doc. ISO/MPEG N3158, Maui, Hawaii, December 1999.

- [21] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. In Beatrice Yorrmak, editor, *SIGMOD'84, Proceedings of Annual Meeting, Boston, Massachusetts, June 18-21, 1984*, pages 47–57. ACM Press, 1984.
- [22] H. Harman 'H. *Modern Factor Analysis*. University of Chicago Press, 1967.
- [23] Alexander Hinneburg, Charu C. Aggarwal, and Daniel A. Keim. What is the nearest neighbor in high dimensional spaces? In *VLDB'2000, Proceedings of 26th International Conference on Very Large Data Bases, September 10-14, Cairo, Egypt*, pages 506–515. Morgan Kaufmann, 2000.
- [24] Daniel A. Keim. Efficient geometry-based similarity search of 3d spatial data-bases. In *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, Philadelphia, Pennsylvania, USA*, pages 419–430. ACM Press, 1999.
- [25] Flip Korn, Nikolaos Sidiropoulos, Christos Faloutsos, Eliot Siegel, and Zenon Protopapas. Fast nearest neighbor search in medical image databases. In *VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases, Mumbai (Bombay), India*, pages 215–226. Morgan Kaufmann, 1996.
- [26] H.-P. Kriegel, T. Schmidt, and T. Seidl. 3d similarity search by shape approximation. In *Lecture Notes in Computer Science*, volume 1262, pages 1–28. Springer, 1997.
- [27] H.-P. Kriegel and T. Seidl. Approximation-based similarity search for 3-d surface segments. *GeoInformatica Journal*, 2(2):113–147, 1998.
- [28] J. B. Kruskal and M. Wish. *Multidimensional Scaling*. SAGE publications, Beverly Hills, 1978.
- [29] Longin Latecki and Rolf Lakämper. Contour-based shape similarity. In Dionysius P. Huijsmans and Arnold W. M. Smeulders, editors, *Lecture Notes in Computer Science*, volume 1614, pages 617–624. Springer, 1999.
- [30] King-Ip Lin, H. V. Jagadish, and Christos Faloutsos. The tv-tree: An index structure for high-dimensional data. *VLDB Journal*, 3(4):517–542, 1994.
- [31] Guojun Lu and Atul Sajjanhar. Region-based shape representation and similarity measure suitable for content-based image retrieval. In P. Venkat Rangan, editor, *Multimedia Systems*, volume 7(2), pages 165–174. ACM/Springer, 1999.
- [32] J. Malik, C. Carson, and S. Belongie. Region-based image retrieval. In *DAGM'99, Mustererkennung*, pages 152–154. Springer Verlag, 1999.
- [33] Rajiv Mehrotra and James E. Gary. Feature-based retrieval of similar shapes. In *Proceedings of the Ninth International Conference on Data Engineering, April 19-23, 1993, Vienna, Austria*, pages 108–115. IEEE Computer Society, 1993.
- [34] Apostol Natsev, Rajeev Rastogi, and Kyuseok Shim. WALRUS: A similarity retrieval algorithm for image databases. In Alex Delis, Christos Faloutsos, and Shahram Ghandeharizadeh, editors, *SIGMOD 1999*, pages 395–406. ACM Press, 1999.
- [35] S. Ravela and R. Manmahta. On computing global similarity in images. In *IEEE Workshop on Applications of Computer Vision (WACV98), Princeton*, pages 82–87. IEEE Computer Society, 1998.
- [36] Seidl T. and Kriegel H.-P. Efficient user-adaptable similarity search in large multimedia databases. In *Proceedings of the 23rd International Conference on Very Large Data Bases, Athens, Greece, 1997*, pages 506–515, 1997.
- [37] Digital Library Project University of California, Berkeley. Image retrieval by image content. <http://galaxy.cs.berkeley.edu/photos/blobworld/>.
- [38] Center for Intelligent Information Retrieval University of Massachusetts. Image retrieval demo. <http://cowarie.cs.umass.edu/~demo/Demo.html>.

- [39] Dejan V. Vranić and Dietmar Saupe. 3d model retrieval. In Bianca Falcidieno, editor, *Proc. Spring Conference on Computer Graphics and its Applications (SC-CG2000)*, May 3-6, 2000, Budmerice Manor, Slovakia, pages 89–93. Comenius University, 2000.
- [40] Dejan V. Vranić and Dietmar Saupe. 3d model retrieval with spherical harmonics and moments. In B. Radig and S. Florczyk, editors, *Proceedings of the DAGM 2001, Munich, Germany, September 2001*, pages 392–397. Springer-Verlag, 2001.
- [41] Dejan V. Vranić and Dietmar Saupe. 3d shape descriptor based on 3d fourier transform. In K. Fazekas, editor, *Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services (ECMCS 2001, Budapest, Hungary, Sept. 2001)*, pages 271–274. Comenius University, 2001.
- [42] Dejan V. Vranić, Dietmar Saupe, and Jörg Richter. Tools for 3d-object retrieval: Karhunen-loeve transform and spherical harmonics. In J.-L. Dugelay and K. Rose, editors, *Proceedings of the IEEE 2001 Workshop Multimedia Signal Processing Cannes, France, October 2001.*, pages 293–298. IEEE, 2001.
- [43] N. Vujovic and D. Brazakovic. Evaluation of an algorithm for finding a match of a distorted texture pattern in a large image database. In *ACM Transactions on Information Systems*, volume 16(1), pages 31–60. ACM, 1998.
- [44] R. Weber, H.-J. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB*, volume 24, New York, USA, August 1998.
- [45] Roger Weber. Project chariot. <http://zuelle.ethz.ch/Chariot/>.
- [46] Aidong Zhang, Biao Cheng, and Raj Acharya. Texture-based image retrieval in image database systems. In Norman Revell and A. Min Tjoa, editors, *DEXA '95 - Workshop*, pages 349–356, San Mateo, California, 1995. ONMIPRESS.