

Stefan Brecheisen, Hans-Peter Kriegel, Peter Kunath, Martin Pfeifle, Matthias Renz:

Der virtuelle Prototyp: Datenbankunterstützung für CAD-Anwendungen

1 Einleitung

Ein schneller und reibungsloser Entwicklungsprozess neuer Produkte ist ein wichtiger Faktor für den wirtschaftlichen Erfolg vieler Unternehmen insbesondere aus der Luft- und Raumfahrttechnik und der Automobilindustrie. Damit Ingenieure in immer kürzerer Zeit immer anspruchsvollere Produkte entwickeln können, werden effektive und effiziente Kollisions- und Ähnlichkeitsanfragen auf komplexen räumlichen Objekten benötigt. Um den hohen Anforderungen eines produktiven Einsatzes zu genügen, müssen entsprechend spezialisierte Zugriffsmethoden in vollwertige Datenbanksysteme integriert werden, so dass zentrale Datenbankdienste wie Transaktionen, kontrollierte Nebenläufigkeit und konsistente Wiederherstellung sichergestellt sind.

In diesem Artikel wird beschrieben, wie effektive und effiziente Kollisions- und Ähnlichkeitsanfragen, die die Grundlage moderner CAD-Anwendungen bilden, in kommerzielle objektrelationale Datenbanksysteme integriert werden können.

2 Anwendungsgebiete

In diesem Abschnitt werden unterschiedliche Typen von industriellen Anwendungen vorgestellt, die im Zusammenhang mit virtuellen Konstruktionsarbeiten stehen. Diese Applikationen profitieren maßgeblich von der Verwendung objektrelationaler Datenbanksysteme. Die dabei eingesetzten Methoden wurden in Kooperation mit Industriepartnern aus der Automobilindustrie sowie der Luft- und Raumfahrtindustrie analysiert und bewertet. Daran beteiligt waren die Firmen Daimler Chrysler AG, Stuttgart, Volkswagen AG, Wolfsburg, Deutsches Zentrum für Luft- und Raumfahrt DLR e.V., Oberpfaffenhofen und Boeing Company, Seattle.

2.1 Digitaler Zusammenbau (Digital Mockup)

Lange Zeit war es gängige Praxis in der Automobil-, Luftfahrt- und Maschinenbau-Industrie, physikalische Prototypen zu erstellen. Diese Prototypen wurden dann benutzt, um während des Entwicklungsprozesses sowohl das Produkt als auch dessen Herstellung auf der Basis des sogenannten Physical Mockup (PMU) zu testen. Die Erstellung und anschließende Testphase mit den realen Prototypen ist jedoch sehr zeitintensiv, die dabei erzielten Ergebnisse wirken sich deshalb erst zu einem späten Zeitpunkt auf den Konstruktionsprozess aus.

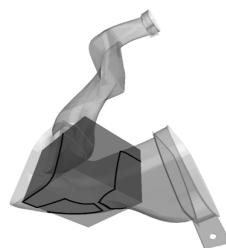
Heutzutage werden in der Industrie keine echten Testmodelle mehr hergestellt. Die Modelle, die man zum Beispiel im Fahrzeugbau benötigt, werden durch neue Methoden des Digital Mockup (DMU) abgelöst, indem man alle (digitalen) Daten des Computer Aided Design (CAD), Computer Aided Engineering (CAE) und Computer Aided Manufacturing (CAM) unter Hinzunahme von Simulationsergebnissen kombiniert. Die Hilfsmittel für das Digital Mockup von Industrieprodukten erlauben es unter anderem, effizient Kollisionsanfragen zu einem frühen Zeitpunkt des Entwicklungsprozesses zu stellen, die sich nur auf vorhandene digitale Daten stützen.

Diese Systeme arbeiten jedoch typischerweise nur im Hauptspeicher und sind deshalb nicht in der Lage, mehr als einige hundert Teile zu handhaben. Als Eingabe wird eine sorgfältig zusammengestellte Liste der zu untersuchenden

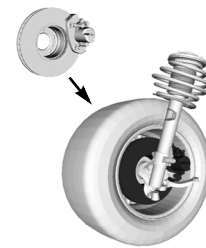
CAD-Bauteile benötigt. Beim traditionellen dateibasierten Ansatz muss jeder Benutzer die Dateien für diese Liste manuell zusammenstellen. Es kann Stunden oder gar Tage an Vorbereitungszeit benötigen, wenn die Bauteile auf unterschiedlichen CAD-Systemen erzeugt wurden oder von mehreren Benutzern auf diversen Dateiservern verwaltet werden. Typischerweise arbeiten in einem nebenläufigen Fertigungsprozess verschiedene Arbeitsgruppen zusammen. Jede dieser Arbeitsgruppen entwickelt dabei eigene Teile als Beitrag zum kompletten Produkt.

Für die Industrie nimmt zudem die Bedeutung des Internets beim Austausch von Dateien immer mehr zu. Ingenieure, die in den USA arbeiten, wollen eventuell den neuesten Bauteilentwurf in der Datenbank ihrer europäischen Teamkollegen ablegen, um Passungstests effizient durchzuführen.

Abbildung 1 zeigt zwei typische räumliche Anfragen für einen dreidimensionalen Konstruktionsraum. Die Boxanfrage liefert alle Teile, die ein gegebenes Hyperrechteck schneiden. Die Kollisionsanfrage hingegen ermittelt diejenigen Bauteile, die mit einem Anfrageteil kollidieren. Ein räumlicher Filter für DMU-bezogene Anfragen auf großen CAD-Datenbanken kann leicht mittels einer räumlichen Indexstruktur implementiert werden. Diese bestimmt eine möglichst kleine Obermenge der Teile, welche die Anfrage erfüllen. Der rechenintensive Verfeinerungsschritt der Kandidatenmenge, der die exakte Untersuchung der Schnittregionen durchführt, kann dann von einem geeigneten Hauptspeicherbasierten CAD-Werkzeug übernommen werden.



a) Boxanfrage



b) Kollisionsanfrage

Abb. 1: Räumliche Anfragen auf CAD-Daten.

2.2 Ähnlichkeitssuche

In den letzten 10 Jahren hat der Bedarf an Datenbankanwendungen zugenommen, für die eine effiziente und effektive Ähnlichkeitssuche wichtig ist. Die Ähnlichkeitssuche gewinnt in Anwendungsgebieten wie Multimedia, Computer Aided Engineering, Marketing, Bildverarbeitung im medizinischen Bereich und der virtuellen Einkaufsberatung immer mehr an Bedeutung. Besonderes Interesse kommt dabei der Aufgabenstellung zu, ähnlich geformte Teile im zwei- bzw. dreidimensionalen Raum zu finden.

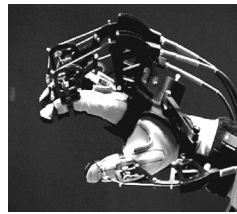
Kürzere Produktzyklen und eine größere Vielfalt an Modellen werden immer mehr entscheidende Faktoren im hart umkämpften Fahrzeug- und Flugzeugmarkt. Die Entwicklung, Konstruktion und Wartung von modernen Industrieprodukten sind sehr teure und schwierige Aufgaben. Diese können nur bewältigt werden, wenn Ingenieure einen Überblick über die schon vorhandenen CAD-Bauteile haben. Ein Konstrukteur sollte in der Lage sein, ähnliche Bauteile zu dem Teil zu finden, an das er gerade denkt. Selbst wenn ihm kein konkretes Bauteil vorschwebt, könnte es für einen Konstrukteur sehr nützlich sein, durch die bereits vorhandenen Teile zu navigieren, um sich einen Überblick zu verschaffen. Ein derartiges System spart Firmen Zeit und Geld, weil dadurch überflüssige Neuentwicklungen vermieden werden.

Zusammenfassend kann man sagen, dass die zunehmende Anzahl an bereits existierenden CAD-Bauteilen zu neuen Herausforderungen an deren Verwaltung führt. Die Frage ist nicht mehr, wie man neue CAD-Teile konstruieren kann, sondern wie man bereits entworfene und erprobte Bauteile auffinden kann.

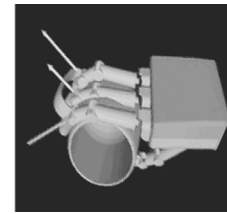
2.3 Weitere Anwendungsbereiche

Im folgenden erwähnen wir kurz weitere Anwendungsgebiete der virtuellen Konstruktion, wobei wir uns im weiteren Verlauf des Artikels auf den digitalen Zusammenbau und die Ähnlichkeitssuche konzentrieren.

Haptisches Rendering. Die bereits erwähnte Entwicklung vom physischen zum digitalen Zusammenbau hat die Nachfrage nach dem bekannten Problem



a) Haptisches Gerät



b) Virtuelle Umgebung

Abb. 2: Beispielszenario für haptisches Rendering.

der Simulation von Konstruktions- und Wartungsaufgaben angeregt. Deshalb wurden viele Ansätze verfolgt, um die physikalischen Eigenschaften von natürlichen Oberflächen nachzubilden, einschließlich der Berechnung von Kollisionskräften. Durch eine Rückkopplung dieser Kräfte an die kollidierenden Objekte kann der Kontakt der Objekte aufrecht erhalten werden und gleichzeitig kann verhindert werden, dass sich die Objekte (z.B. Teile und Werkzeuge) gegenseitig durchdringen. Abbildung 2a zeigt ein gebräuchliches haptisches Gerät, das mit einem virtuellen Abbild (Abbildung 2b) gekoppelt ist. Die berechnete Kollisionskraft zwischen dem virtuellen Modell und der virtuellen Umgebung wird auf das haptische Gerät (Datenhandschuh) übertragen. Die simulierte Umgebung, sowie die Vektoren der auftretenden Kräfte, sind in Abbildung 2b dargestellt. Durch die Kombination von haptischen Algorithmen und Hardware kann dem Benutzer ein realistisches Kraftgefühl der virtuellen Szene vermittelt werden.

Eine in Echtzeit durchgeführte haptische Berechnung erfordert, dass sich die beteiligten räumlichen Objekte im Hauptspeicher befinden. Um eine haptische Simulation einer großangelegten, aus Millionen von Objekten bestehenden Umgebung durchzuführen, ist eine sorgfältige Auswahl und ein effizientes Prefetching der räumlichen Objekte unverzichtbar. Die Kombination von echtzeitfähigem haptischem Rendering und index-basiertem Prefetching von persistenten räumlichen Objekten ermöglicht die Simulation und Evaluierung von Instandhaltungsaufgaben, die z.B. von autonomen Robotern ausgeführt werden.

Verwaltung von räumlichen Dokumenten. Während der Entwicklung, Dokumentationserstellung und Wartung von komplexen Industrieprodukten werden viele weitere Daten neben der geometrischen Beschreibung der Produktkomponenten erzeugt und aktualisiert. Die meisten dieser Daten können ebenfalls durch räumliche Schlüssel im dreidimensionalen Konstruktionsraum referenziert werden (siehe Abbildung 3). Dazu gehören kinematische Hüllkörper, die bewegliche Teile in jeder möglichen Lage darstellen, oder Abstandsanforderungen, um leere Bereiche zu reservieren, z.B. das minimale Volumen einer Passagierkabine oder Freiraum für die Luftzirkulation um Bereiche mit großer Hitzeentwicklung. Weiterhin können technische Zeichnungen, Prüfberichte und sogar die Geschäftsdaten von bestimmten Produktteilen räumlich indiziert werden. Die strukturelle Referenzierung derartiger Dokumente kann sehr aufwändig sein. Zum Beispiel können die für eine Konstruktion angesetzte Teambesprechungen, die sich mit einem bestimmten Detail eines Produkts beschäftigen, viele verschiedene Bauteile betreffen. Räumliche Organisation bietet hier eine Lösung, indem man die Sitzungen auf bequeme Weise mit einem bestimmten räumlichen Kennzeichen verknüpft, welches für die interessante Stelle angelegt wurde. Eine mögliche Anfrage könnte beispielsweise lauten: "Liefere mir alle Besprechungen des letzten Monats, die sich mit dem Bereich zwischen den Teilen A und B befassen". Räumliche Indexstrukturen können solche Anfragen effizient unterstützen.

3. CAD-Datenbanksysteme

Ein Datenbanksystem (DBS) dient zur Verwaltung und Analyse von sehr großen

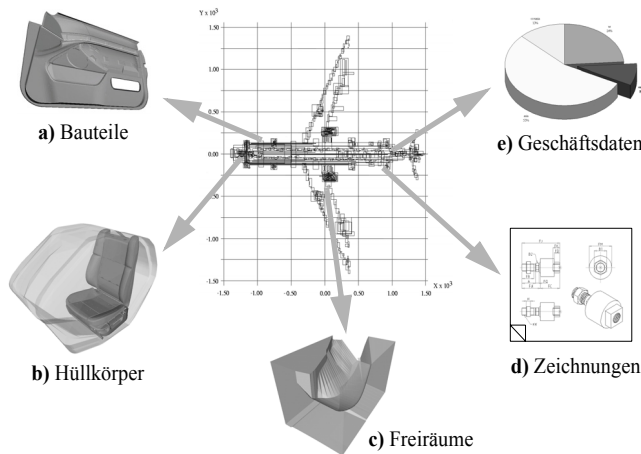


Abb. 3: Räumliches Referenzieren von Dokumenten.

Mengen an persistenten Daten. Im Vergleich zu einer dateibasierten Verwaltung bringt ein DBS wichtige Vorteile mit sich, wie z.B. logische und physische Datenunabhängigkeit, Kontrolle von Transaktionen und Mehrbenutzerbetrieb, Integritätsprüfung, konsistente Wiederherstellung bei Ausfall, Sicherheitskonzepte, Verwendung von Standards und Kontrolle bei verteilten Datenmengen [DAT 99].

Eines der vielversprechendsten Datenmodelle, die in Datenbanksystemen verwendet werden, ist das objektrelationale Modell. Zum einen sind objektrelationale Datenbanksysteme sehr praxisrelevant, da die objektrelationale Funktionalität in den meisten kommerziellen relationalen Datenbankservern realisiert ist, wie z.B. in Oracle [Doh 98], IBM DB2 [CCN+ 99] und Informix IDS/UDO [Bro 01]. Zum anderen ist eine Erweiterungsmöglichkeit eine notwendige Voraussetzung für die nahtlose Integration von benutzerdefinierten Datentypen und Prädikaten. Diese ist z.B. für die bei Konstruktionsarbeiten notwendige Verwaltung von komplexen ausgedehnten Objekten unerlässlich. Die Definition von räumlichen Datentypen und Prädikaten ermöglicht die Durchführung aller Arten von Ähnlichkeits- und Schnittanfragen. Darüber hinaus werden mittels Verwendung von geometrischen Prädikaten auch Kombinationen von strukturellen Anfragen unterstützt, wie z.B. "Ermittle alle Dokumente, die sich auf die diesjährige Version des Jet-Triebwerkes beziehen". Somit erlaubt ein objektrelationales Da-

tenbankmanagementsystem (ORDBMS) eine einfache Kombination von Engineering Data Management (EDM) Systemen und raumorganisierenden Datenbanksystemen. EDM-Systeme organisieren die riesigen zugrundeliegenden CAD-Dateien anhand von hierarchischen Produktstrukturen.

Ein raumorganisierendes Datenbanksystem ist gemäß Güting [Güt 94] ein vollwertiges Datenbanksystem mit zusätzlichen Modulen zur Bearbeitung von räumlichen Daten. Die meisten objektrelationalen Datenbanksysteme, inklusive Oracle [Ora 99, SMS+ 00], IBM DB2 [IBM 99, CCF+ 99], oder Informix IDS/UDO [Inf 98, BSSJ 99], bieten über eine Erweiterbarkeitsschnittstelle die Voraussetzung für die Integration der Verwaltung von räumlichen Daten.

3.1 Repräsentation von CAD-Objekten

Ein CAD-Produkt, wie etwa ein Fahrzeug, besteht üblicherweise aus einer Menge von individuellen Teilen mit

eventuell sehr komplexer geometrischer Form. Dabei werden die einzelnen Teile mit einer sehr großen Genauigkeit entwickelt. Eine exakte geometrische Modellierung erfordert den Einsatz von hochspezialisierten CAD-Anwendungen, die verschiedene Raumprimitive und Dateiformate verarbeiten können. Eine unternehmensweit eingeführte CAD-Datenbank sollte daher die gängigen Dateiformate unterstützen, wie z.B. VDAFS [VDA 87], IGES [IGES 96], STEP [STEP 94+] und VRML [CBM 97]. Um die verschiedenen Formate in Einklang mit der geometrischen Interpretation der Objekte zu bringen, werden universelle Datenmodelle verwendet. Sinnvolle Datenmodelle für die Beschreibung der Objektgeometrie sind triangulierte Oberflächen für die Visualisierung und Ermittlung von Objektkollisionen. Eine ebenfalls weitverbreitete Objektbeschreibung sind Voxelmengen zur konservativen Approximation der Geometrie des Objektkörpers. Für Anwendungen im Bereich der Ähnlichkeitssuche werden häufig auch Feature-Vektoren zur Repräsentation der Objektgeometrie verwendet.

Triangulierte Oberflächen

Typischerweise wird die exakte Oberfläche von CAD-Objekten durch parametrisierte bikubische Oberflächen beschrieben, wie z.B. Hermite-, Bézier-, oder B-Spline-Kurven. Diese Darstellungsformen sind jedoch sowohl für die graphische Ausgabe als auch für die effiziente Berechnung von Oberflächen-schnitten zu komplex [MH 99]. Für die gewünschten Operationen stellen Polygonnetze (z.B. triangulierte Oberflächen) eine bessere Alternative zur Beschreibung der Objektoberfläche dar (siehe Ab-

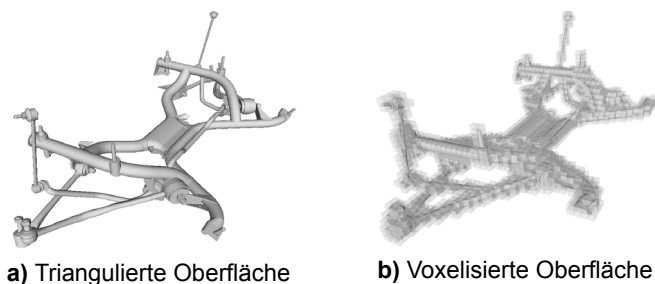


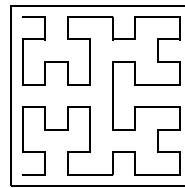
Abb. 4: 3D Scan Conversion einer triangulierten Oberfläche.

bildung 4a). Diese approximativen Repräsentationen können sehr einfach aus den exakten Oberflächenbeschreibungen der Objekte berechnet werden und ermöglichen sowohl interaktive und visuelle Darstellung der Objekte, als auch effiziente Berechnung von Kollisionsanfragen. Man unterscheidet drei Arten von Kollisionsanfragen [MH 99]: Die Kollisionserkennung überprüft, ob sich zwei Objekte schneiden oder berühren. Bei der Kollisionsermittlung berechnet man die Schnittregionen sich schneidender Objekte. Die Kollisionsreaktion bestimmt die Prozedur, die bei erkannter Kollision durchzuführen ist.

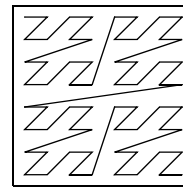
Voxelisierte Daten

Die Voxelisierung von Objekten, deren Geometrie durch Polygonnetze beschrieben wird, ist Hauptgegenstand der Forschung im Bereich Computergraphik und CAD. Techniken und Applikationen für die Voxelisierung haben zum Beispiel in der interaktiven Visualisierung von Volumen [HYFK 98] und dem haptischen Rendering [MPT 99] Verwendung gefunden. Ein Basisalgorithmus für die 3D-Konvertierung von Polygonen in eine voxelbasierte Volumenabbildung wurde von Kaufmann [Kau 87] vorgeschlagen, (siehe Abbildung 4).

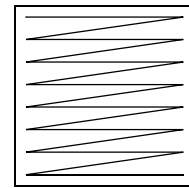
Repräsentation von massiven Körpern. Bei der Verwendung einer triangulierten Oberfläche zur Beschreibung eines massiven Objekts kann zur Unterscheidung zwischen Objektinneren und Objektäußeren jedes Dreieck mit einem in das Objektinnere zeigenden Normalenvektor versehen werden. Somit können nicht nur Objektoberflächen, sondern auch massive Objekte durch ein Triangulationsnetz beschrieben werden. Leider beinhalten die triangulierten Oberflächen, die von gängigen Triangulationsgeneratoren erzeugt wurden, geometrische und topologische Inkonsistenzen, wie z.B. sich überlappende Dreiecke, oder kleine Lücken auf der Objektoberfläche. Eine robuste Rekonstruktion des Objektinneren wird damit sehr umständlich. Als Maßnahme dafür wird für die Voxelisierung zunächst nur die triangulierte Objektoberfläche gerastert, um eine konsistente Repräsentation der Objektoberflä-



Hilbert-Ordnung



Z-Ordnung



lexikographische Ordnung

Abb. 5: Beispiele für raumfüllende Kurven im zweidimensionalen Raum.

che zu erzeugen. Als nächstes wird ein 3D-FloodFill-Algorithmus zur Berechnung der Voxel angewendet.

Raumfüllende Kurven. Die Voxel repräsentieren die Zellen eines Gitters, das den gesamten Datenraum überspannt. Anhand von raumfüllenden Kurven kann jede Zelle des Gitters durch eine einzige Integer-Zahl kodiert werden. Somit kann ein ausgedehntes voxelisiertes Objekt durch eine Menge von Integer-Zahlen beschrieben werden. Die meisten dieser raumfüllenden Kurven weisen eine sehr gute Cluster-Eigenschaft auf, so dass die Integer-Werte räumlich benachbarter Zellen sehr nahe beieinander liegen. Fortlaufende Integer-Werte entsprechen wiederum räumlich benachbarten Zellen. Die bekanntesten Beispiele für raumfüllende Kurven sind die Hilbert-, Z-, oder die lexikographische Ordnung, wie in Abbildung 5 dargestellt.

Indizierung räumlicher Objekte.

Die Voxel, die ein Objekt repräsentieren, können in Raumpartitionen, Intervalle oder Blöcke gruppiert werden, um sie durch entsprechende Indexstrukturen effizient zu verwalten. Die Basis für die Konvertierung eines ausgedehnten Objekts in eine der genannten räumlichen Grundbausteine ist die Granularität, d.h. die Auflösung des zugrundeliegenden Gitters. Wenn die Auflösung verfeinert wird, werden die Objekte genauer approximiert, die Redundanz steigt dagegen an. Der jeweilige Approximationsfehler entspricht dem Verhältnis des toten Raums zu dem vom Objekt belegten Raum. Die Redundanz ist näherungsweise proportional zu der Oberfläche des approximierten Objekts [FJM 97, MJFS 96]. Jedoch ist die durchschnittliche Anzahl der Intervalle, die durch die Z- oder Hilbert-Kurve gebildet werden, um einiges niedriger als die Anzahl der objektrepräsentierenden Raumpartitionen. Die

Hilbert-Ordnung erzielt die minimale Anzahl der Intervalle oder Raumpartitionen pro Objekt [Jag 90, FR 89], jedoch ist die Berechnung dieser Ordnung am aufwändigsten. Die Z-Ordnung ist ein guter Kompromiss zwischen Redundanz und Berechnungsaufwand.

Zerlegung der Objekte. In vielen Anwendungen ist für eine gute Balance zwischen Redundanz und Genauigkeit die Wahl einer geeigneten Datenraumauflösung oder raumfüllenden Kurve allein nicht ausreichend. Zum einen müsste die Granularität für jedes Objekt individuell gewählt werden, anstatt die gleiche Auflösung für alle Objekte zu verwenden. Zum anderen müsste die Auflösung beim Anlegen der Datenbank festgelegt werden, während die Objekte zum Einfüge- oder Anfragezeitpunkt unterschiedlich approximiert werden sollten. Ein Ansatz zur Kontrolle dieser Balance von Redundanz und Genauigkeit ist das Konzept der größen- und fehlerbeschränkten Approximation [Ore 89], die eine Erweiterung der bereits genannten granularitätsbeschränkten Approximation [Gae 95] darstellen. Die Zerlegung basiert auf einer rekursiven Aufteilung des Objekts und stoppt dann, wenn die erwünschte Redundanz (größenbeschränkt) oder der tolerierte maximale Approximationsfehler (fehlerbeschränkt) erreicht ist. In [KPS 01] wird gezeigt, wie der Algorithmus aus [Ore 89] so angepasst werden kann, dass Intervallsequenzen direkt erzeugt werden.

Repräsentation durch Feature-Vektoren

Distanzfunktionen bilden die Grundlage für Ähnlichkeitssuche. Triangulierte Oberflächen sowie Voxelmengen sind als Objektrepräsentationen für die Ähnlichkeitssuche ungeeignet, da es für diese

Darstellungen keine adäquaten Distanzfunktionen gibt. Deshalb werden Objektrepräsentationen benötigt, die die Definition eines effizient zu berechnenden und aussagekräftigen Distanzmaßes zulassen. Ein weit verbreiteter Ansatz ist die Repräsentation von Objekten durch numerische Vektoren, deren Distanz sich sehr einfach berechnen lässt. In diesem Fall werden bei einer Feature-Transformation aussagekräftige räumliche Charakteristiken eines Objekts extrahiert, die als numerische Werte zu einem Feature-Vektor zusammengefasst werden [KKM+ 03][KBK+03].

Unter der Annahme, dass die Ähnlichkeit der Feature-Distanz entspricht, kann man als Ähnlichkeitsmaß für je zwei Datenobjekte die Distanz zwischen den jeweiligen Feature-Vektoren definieren. Somit reduziert sich die Suche von ähnlichen Objekten zu einem gegebenen Anfrageobjekt auf eine Nachbarschaftssuche im Feature-Raum. Für die meisten Anwendungen wird die euklidische Metrik (L_2) zur Berechnung der Feature-Distanz benutzt, aber es gibt noch viele weitere häufig benutzten Metriken, wie z.B. die Manhattan-Metrik (L_1), oder die Maximums-Metrik (L_∞).

3.2 Anfragebearbeitung in CAD-Datenbanken

Auf der Grundlage der verschiedenen Arten der Datenrepräsentation aus dem vorherigen Abschnitt können die bei Konstruktionsarbeiten auftretenden Anfragetypen effektiv und effizient verarbeitet werden. Zwei dieser Anfragetypen sind grundlegend für Kollisionsanfragen, die übrigen für die Ähnlichkeitssuche. Im folgenden werden die Grundlagen effizienter Anfragebearbeitung angeschnitten, nämlich räumliche Indexstrukturen und das Konzept der mehrstufigen Anfragebearbeitung.

Ein CAD-Datenbankmanagementsystem sollte es den Ingenieuren ermöglichen, kollidierende Teile oder ähnliche Teile zu finden. Die dazu notwendigen Anfragetypen werden hier informell eingeführt, ohne die konkrete Datenrepräsentation der CAD-Objekte zu spezifizieren. Die Anfragen können auf triangulier-

ten Oberflächen oder auf davon abgeleiteten Darstellungen, wie z.B. Voxelmengen, durchgeführt werden.

Anfragetypen

Bei der Anfragebearbeitung in CAD-Datenbanken spielen folgende Anfragetypen eine Rolle.

Boolesche Schnitthanfrage. Eine Boolesche Schnitthanfrage liefert den Wert *true*, falls sich zwei gegebene Objekte schneiden, ansonsten den Wert *false*. Dabei ist zu beachten, dass sich die voxelisierten Objekte schneiden können, obwohl sich deren triangulierte Oberflächen nicht schneiden.

Gewichtete Schnitthanfrage. Im Gegensatz zur Booleschen Variante liefert die gewichtete Schnitthanfrage keinen Wahrheitswert, sondern einen numerischen Wert. Für die Berechnung des Schnittvolumens eignet sich die Voxeldarstellung, während sich triangulierte Oberflächen für die Berechnung der Schnittlänge eignen. Falls sich zwei Objekte nicht schneiden, liefert die Anfrage den Wert 0.

Bereichsanfrage. Bei einer Bereichsanfrage gibt der Benutzer ein Anfrageobjekt q und einen Anfrageradius $\epsilon \in \mathbb{R}_0^+$ vor. Das System ermittelt alle Objekte aus der Datenbank, deren Distanz von q nicht mehr als ϵ beträgt.

k -Nächste-Nachbarn-Anfrage. Bei einer k -Nächste-Nachbarn-Anfrage gibt der Benutzer ein Anfrageobjekt q und die Kardinalität k der Ergebnismenge vor. Das System liefert die k Objekte zurück, die q am nächsten liegen.

Distanz-Ranking. Beim Distanz-Ranking gibt der Benutzer ein Anfrageobjekt q vor und das System liefert alle Datenbankobjekte aufsteigend nach der Distanz zu q geordnet zurück. Aus Effizienzgründen sollte die Ranking-Anfrage nicht auf einmal vollständig berechnet werden. Stattdessen wird die Rangfolge inkrementell berechnet, denn meistens ist der Benutzer bereits mit der Ausgabe der ersten paar Elemente zufrieden.

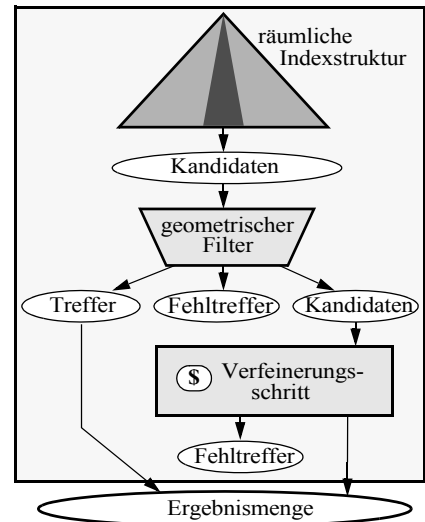


Abb. 6: Mehrstufige Anfragebearbeitung.

Strategie der mehrstufigen Anfragebearbeitung

Ein leistungsfähiges CAD-Datenbanksystem muss über einen effektiven und effizienten Anfrageprozessor verfügen. Grundlegende Konzepte dieser zentralen Systemkomponente sind räumliche Indexstrukturen und mehrstufige Anfragebearbeitung [BKSS94].

Räumliche Indexstrukturen.

Räumliche Indexstrukturen partitionieren den multidimensionalen Suchraum für räumliche Anfragen. Speziell für räumliche Selektionen und räumliche Ähnlichkeitssuche erlaubt die räumliche Indizierung den frühzeitigen Ausschluss vieler irrelevanter Objekte. Dadurch muss nur ein kleiner Teil der Datenbank nach der tatsächlichen Resultatmenge durchsucht werden. Da die genaue Repräsentation räumlicher Objekte beliebig komplex sein kann, verwenden räumliche Indexstrukturen gewöhnliche konservative Approximationen, um die Information über die Form und Position jedes Objekts zu erhalten.

Mehrstufige Anfragebearbeitung.

Die Benutzung von Approximationen führt zur Strategie der mehrstufigen Anfragebearbeitung (Abbildung 6). Zuerst wird ein Filterschritt durchgeführt, der eine Obermenge derjenigen Objekte liefert, die die Anfragebedingung erfüllen. Da der Filterschritt auf konservativen Approximationen beruht, werden keine tatsächlichen Treffer ausgeschlossen, d.h.

alle relevanten Objekte sind in der zurückgelieferten Kandidatenmenge enthalten. Eine Kaskade weiterer Filterschritte reduziert gegebenenfalls die Größe der Kandidatenmenge noch weiter, z.B. durch die Benutzung von genaueren Repräsentationen oder durch progressive Approximationen [BKS 93, BKSS 94]. Diese progressiven Approximationen sind auch geeignet, um Objekte zu identifizieren, die bereits zur Ergebnismenge gehören. Für hochselektive Anfragebedingungen sollte der erste Filterschritt auf einer geeigneten räumlichen Zugriffsmethode durchgeführt werden. Der mehrstufige Anfrageprozess wird mit dem Verfeinerungsschritt abgeschlossen, der die exakte Geometrie überprüft.

Schnittanfragen. Im Fall von Schnittanfragen basiert der Filterschritt auf voxelisierten Daten, während der Verfeinerungsschritt auf den triangulierten Oberflächen durchgeführt wird. Für viele Anwendungen sind hochaufgelöste voxelisierte Daten genau genug, so dass auf einen Verfeinerungsschritt verzichtet werden kann.

In den letzten Jahren wurden verschiedene räumliche Indexstrukturen zur effizienten Bearbeitung von Schnittanfragen entwickelt, wie z.B. der relationale Intervallbaum [KPS 00], lineare Quadtree [Sam 90] oder relationale R-Baum [RRSB 99]. Dabei stand die Integrationsfähigkeit in objektrelationale Datenbanksysteme im Vordergrund. Speziell für effiziente Schnittanfragen von hochaufgelösten Objekten werden zwei generische Verfahren verwendet. Das erste Verfahren benutzt statistische Informationen räumlicher Indexstrukturen, um eine gegebene Anfrage zu beschleunigen [KK-PR 04a]. Das zweite Verfahren beruht auf einer kostenbasierten Zerlegung komplexer räumlicher Datenbank-Objekte [KK-PR 04b]. Diese beiden Verfahren ergänzen sich gegenseitig und können unabhängig voneinander oder zusammen eingesetzt werden.

Ähnlichkeitsanfragen. Im Fall von Ähnlichkeitsanfragen wird der Filterschritt auf Featurevektoren oder einzelnen numerischen Werten durchgeführt, die das Objekt grob beschreiben. Manchmal gibt es eine genauere Distanzfunktion, die im Verfeinerungsschritt angewendet werden kann. Um zu garantieren, dass bei der mehrstufigen Anfragebearbeitung keine Antworten verloren gehen, sog. false drops, darf die im Filterschritt gewählte Distanzfunktion für kein Objektpaar einen größeren Distanzwert zurückliefern als die tatsächliche Objektdistanz.

4. Prototypen für CAD-Anwendungen

Im Maschinenbau wird dreidimensionales CAD im gesamten Entwicklungsprozess eingesetzt. Von den ersten Entwürfen bis zur Serienproduktion von Fahr- oder Flugzeugen werden Tausende bis Millionen von CAD-Objekten erzeugt, zuzüglich zur dazugehörigen Dokumentation. Im folgenden werden zwei CAD-Datenbankanwendungen vorgestellt, die für den Entwicklungsprozess nützlich sind.

4.1 DIVE

DIVE (Database Integration for Virtual Engineering) [KMPS 01] ermöglicht die Einbindung des digitalen Konstruktionsprozesses in die existierende Infrastruktur eines Unternehmens. Besonderer Wert wird auf die Integration interaktiver räumlicher Datenverwaltung in handelsübliche objektrelationale Datenbanksysteme gelegt. Im folgenden geben wir einen kurzen Überblick über die DIVE-Architektur.

Räumliche Datenverwaltung. Die Geometrie eines Bauteils belegt eine bestimmte Region im Konstruktionsraum. Wenn man diesen Bereich als räumlichen Schlüssel benutzt, können damit zugehörige Dokumente wie CAD-Dateien, VRML-Szenen oder Produktionspläne referenziert werden. Die größten Herausforderungen bei der Entwicklung einer robusten und dynamischen Datenbank-Schicht für die virtuelle Konstruktion sind: Verwaltung räumlicher Daten in einem konventionellen Datenbanksy-

stem, effiziente Auswertung der notwendigen räumlichen Anfrageprädikate und Erstellung leistungsfähiger Auswertungspläne für Anfragen, die sowohl aus geometrischen als auch aus strukturellen Prädikaten bestehen.

Bis jetzt wurden nur wenige räumliche Indexstrukturen entworfen, die dem objektrelationalen Modell folgen und ohne tiefgreifende Modifikationen oder Erweiterungen am physischen Kern des Datenbankservers auskommen. Der relationale Intervallbaum (RI-Baum) [KPS 00] ist eine schlanke Indexstruktur, die ausgedehnte Daten in jedem beliebigen relationalen Datenbanksystem effizient verwalten kann und dabei die eingebaute Transaktionsverwaltung und Recovery-Funktionen nutzt. Der RI-Baum hat einen größeren Nutzen und eine höhere Leistungsfähigkeit bei räumlichen Anwendungen als vergleichbare Techniken [KPS 01]. Deshalb wird der RI-Baum als räumliche Indexstruktur im DIVE-System eingesetzt.

Die originalen Bauteile, die in unterschiedlichsten CAD-Formaten vorliegen, werden in eine Voxeldarstellung überführt und mittels raumfüllender Kurve in eine Intervallsequenz umgewandelt. Diese werden dann unter der Verwendung des RI-Baum effizient verwaltet.

Anfragebearbeitung. Der DIVE Server bildet geometrische Anfrageprädikate auf Bereichsanfragen im indizierten Datenraum ab. Das zugrundeliegende mehrstufige Anfragesystem implementiert auf den abgespeicherten Intervallen einen hocheffizienten und selektiven Filterschritt. Es werden sowohl Bereichsanfragen, als auch Kollisions- und Abstandsanfragen unterstützt. Der nicht-räumliche Rest der Anfrage, z.B. strukturelle Prädikate, wird vom EDM-System verarbeitet.

Weiterhin ist in DIVE ein optionaler Verfeinerungsschritt für den digitalen Zusammenbau integriert, um Schnitte von hochpräzisen triangulierten Oberflächen zu erkennen. Außerdem kann ein Bauteil von der vorhergehenden Anfrage als Anfrageobjekt für die nächste geometrische Suche verwendet werden. Dadurch wird der Benutzer in die Lage versetzt, sich in einer sehr großen räumlichen Datenbank in Echtzeit umzusehen.

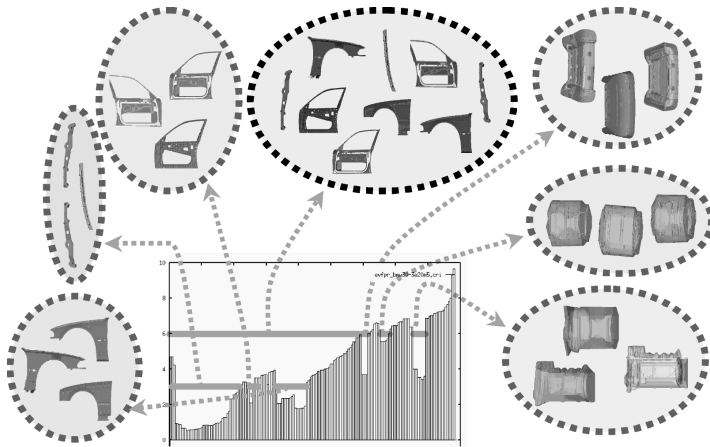


Abb. 7: Ähnlichkeitssuche mit BOSS.

4.2 BOSS

BOSS (Browsing Optics plots for Similarity Search) [BKK+ 04] ist ein interaktiver Browser, der effektive Ähnlichkeitssuche in sehr umfangreichen CAD-Datenbanken unterstützt. Dazu werden Erreichbarkeitsplots, die von dem hierarchischen Clustering-Algorithmus OPTICS [ABKS 99] berechnet werden, auf benutzerfreundliche Weise zusammen mit passenden Cluster-Repräsentanten angezeigt. Die Cluster-Struktur erschließt sich dem Benutzer auf einen Blick. Mit Hilfe effektiver Clustererkennungsalgorithmen wird aus dem Erreichbarkeitsplot eine hierarchische Struktur extrahiert [BKKP 04]. Ist der Benutzer nicht nur an der Art und Anzahl der Cluster interessiert, sondern auch an den einzelnen Objekten innerhalb der Cluster, ergibt sich bei großen Clustern das Problem, den Überblick über die Vielzahl an Objekten zu behalten (siehe Abbildung 7). Deshalb werden für jeden Cluster repräsentative Objekte bestimmt, mit deren Hilfe der Benutzer durch die Cluster-Struktur navigieren kann. Neben der Ähnlichkeitssuche hat BOSS noch Visuelles Data Mining und Evaluierung von Ähnlichkeitsmodellen als weitere Anwendungsgebiete.

Visuelles Data Mining. Der hierarchische Cluster-Algorithmus OPTICS wird auf die Datenmenge angewandt. Die resultierende Cluster-Struktur wird dem Benutzer als Zwischenergebnis präsentiert. Der Inhalt der einzelnen Cluster ist

dem Benutzer zunächst unbekannt. BOSS erlaubt dem Benutzer nun, sich visuell einen Überblick über die Cluster zu verschaffen und die Clusterstruktur zu analysieren. Auf diese Weise kann bisher unbekanntes Wissen aus der Datenmenge gewonnen werden, ohne ein konkretes Anfrageobjekt zu spezifizieren. Dies ist ein wesentlich allgemeinerer Ansatz als der der Ähnlichkeitssuche, da sich das gewonnene Wissen dort lediglich auf ähnliche Teile bezüglich eines vorgegebenen Anfrageobjekts beschränkt.

Evaluierung von Ähnlichkeitsmodellen. Hierarchisches Clustering ist eine effektive Technik zur Evaluation von Ähnlichkeitsmodellen [KKM+ 03], bei der die gesamte Datenmenge in die Beurteilung des Ähnlichkeitsmodells eingeht. In der Cluster-Struktur werden ähnliche Objekte bezüglich eines gegebenen Ähnlichkeitsmodells gruppiert und unähnliche Teile separiert. Der Benutzer kann nun entscheiden, ob das Ähnlichkeitsmodell für die Datenmenge tauglich ist, je nachdem, ob die Gruppierung seiner intuitiven Vorstellung von Ähnlichkeit entspricht oder nicht.

5 Zusammenfassung

Für eine effektive und effiziente Verwaltung von Konstruktionsdokumenten innerhalb eines EDM-Systems ist der Einsatz von räumlichen Datenbanklösungen unumgänglich. Der Grund dafür ist die sehr große Menge an CAD-Daten, die während eines Entwicklungsprozesses von komplexen Konstruktionen entsteht

und damit verwaltet werden muss. Unter der Verwendung von räumlichen Selektionen kann mit einer räumlichen Datenbank die benötigte Vorverarbeitungszeit von Stunden oder gar Tagen im Vergleich zu einem dateibasierten System auf einige Sekunden reduziert werden.

In diesem Artikel werden die wichtigsten Datenrepräsentationen von und Operationen auf CAD-Daten vorgestellt, die für eine effiziente Durchführung von räumlichen Anfragen auf CAD-Datenbanken notwendig sind. Es werden Methoden für die Konvertierung der ursprünglichen CAD-Objekte in räumliche Objektapproximationen vorgestellt, die notwendig sind, um die CAD-Objekte in relationalen Indexstrukturen verwalten zu können. Die relationale Indexstruktur wird entsprechend dem Paradigma der mehrstufigen Anfragebearbeitung für den Filterschritt eingesetzt. Diese Methode unterstützt die Durchführung von räumlichen Anfragen im Bereich der Ähnlichkeitssuche sowie im Bereich von digitalem Zusammenbau, haptischem Rendering und der Verwaltung von CAD-Dokumenten. Es werden zwei Prototypen, DIVE und BOSS vorgestellt, die sowohl Anfragen auf Objektkollisionen als auch die Suche nach ähnlichen Objektteilen unterstützen. Dabei wurden effiziente und effektive Anfragetechniken eingesetzt, die sehr kurze Antwortzeiten ermöglichen und damit die Interaktion zwischen Benutzer und System fördern.

Referenzen

- [ABKS 99] Ankerst M., Breunig M., Kriegel H.-P., Sander J.: OPTICS: Ordering Points To Identify the Clustering Structure. Proc. ACM SIGMOD Int. Conf. on Management of Data, 49-60, 1999.
- [BKK+ 04] Brecheisen S., Kriegel H.-P., Kröger P., Pfeifle M., Pötke M., Viermetz M.: BOSS: Browsing OPTICS-Plots for Similarity Search. Proc. 20th Int. Conf. on Data Engineering (ICDE), 2004.
- [BKKP 04] Brecheisen S., Kriegel H.-P., Kröger P., Pfeifle M.: Visually Mining Through Cluster Hierarchies. Proc. SIAM Int. Conf. on Data Mining (SIAMDM'04), 2004.
- [BKS 93] Brinkhoff T., Kriegel H.-P., Schneider R.: Comparison of Approximations of Complex Objects Used for Approximation-based Query Processing in Spatial Database Systems. Proc. 9th Int. Conf. on Data Engineering (ICDE), 40-49, 1993.
- [BKSS 94] Brinkhoff T., Kriegel H.-P.,

Schneider R., Seeger B.: Multi-Step Processing of Spatial Joins. Proc. ACM SIGMOD Int. Conf. on Management of Data, 197-208, 1994.

[Bro 01] Brown P.: Object-Relational Database Development – A Plumber’s Guide. Informix Press, Menlo Park, CA, 2001.

[BSSJ 99] Bliujute R., Saltenis S., Slivinskas G., Jensen C.S.: Developing a DataBlade for a New Index. Proc. 15th Int. Conf. on Data Engineering (ICDE), 314-323, 1999.

[CBM 97] Carey R., Bell G., Marrin C.: ISO/IEC 14772-1:1997 Virtual Reality Modeling Language. VRML Consortium, 1997.

[CCF+ 99] Chen W., Chow J.-H., Fuh Y.-C., Grandbois J., Jou M., Mattos N., Tran B., Wang Y.: High Level Indexing of User-Defined Types. Proc. 25th Int. Conf. on Very Large Databases (VLDB), 554-564, 1999.

[CCN+ 99] Carey M. J., Chamberlin D. D., Narayanan S., Vance B., Doole D., Rielau S., Swagerman R., Mattos N.: O-O, What Have They Done to DB2? Proc. 25th Int. Conf. on Very Large Databases (VLDB), 542-553, 1999.

[Dat 99] Date C. J.: An Introduction to Database Systems. Addison Wesley Longman, Boston, MA, 1999.

[Doh 98] Doherty C. G.: Database Systems Management and Oracle8. Proc. ACM SIGMOD Int. Conf. on Management of Data, 510-511, 1998.

[FJM 97] Faloutsos C., Jagadish H. V., Manolopoulos Y.: Analysis of the n-Dimensional Quadtree Decomposition for Arbitrary Hyperrectangles. IEEE Trans. on Knowledge and Data Engineering 9(3), 373-383, 1997.

[FR 89] Faloutsos C., Roseman S.: Fractals for Secondary Key Retrieval. Proc. ACM Symposium on Principles of Database Systems (PODS), 247-252, 1989.

[Gae 95] Gaede V.: Optimal Redundancy in Spatial Database Systems. Proc. 4th Int. Symp. on Large Spatial Databases (SSD), LNCS 951, 96-116, 1995.

[Güt 94] Güting R. H.: An Introduction to Spatial Database Systems. VLDB Journal, 3(4), 357-399, 1994.

[HYFK 98] Huang J., Yagel R., Filippov V., Kurzion Y.: An Accurate Method for Voxelize Polygon Meshes. Proc. IEEE Symp. on Volume Visualization, 119-126, 1998.

[IBM 99] IBM Corp.: IBM DB2 Universal Database Application Development Guide, Version 6. Armonk, NY, 1999.

[IGES 96] American National Standards Institute: ANSI/US PRO/IPO 100-1996 (Initial Graphics Exchange Specification IGES 5.3). New York, NY, 1996.

[Inf 98] Informix Software, Inc.: DataBlade Developers Kit User’s Guide, Version 3.4. Menlo Park, CA, 1998.

[Jag 90] Jagadish H. V.: Linear Clustering of Objects with Multiple Attributes. Proc. ACM SIGMOD Int. Conf. on Management of Data, 332-342, 1990.

[Kau 87] Kaufman A.: An Algorithm for 3D Scan-Conversion of Polygons. Proc. Eurographics, 197-208, 1987.

[KBM+ 03] Kriegel H.-P., Brecheisen S., Kröger

P., Pfeifle M., Schubert M.: Using Sets of Feature Vectors for Similarity Search on Voxelize CAD Objects. Proc. ACM SIGMOD Int. Conf. on Management of Data, 587-598, 2003.

[KKM+ 03] Kriegel H.-P., Kröger P., Mashael Z., Pfeifle M., Pötke M., Seidl T.: Effective Similarity Search on Voxelize CAD Objects. Proc. 8th Int. Conf. on Database Systems for Advanced Applications (DAS-FAA), 27-36, 2003.

[KKPR 04a] Kriegel H.-P., Kunath P., Pfeifle M., Renz M.: Statistic Driven Acceleration of Object-Relational Space-Partitioning Index Structures. Proc. 9th Int. Conf. on Database Systems for Advanced Applications (DAS-FAA), 2004.

[KKPR 04b] Kriegel H.-P., Kunath P., Pfeifle M., Renz M.: Effective Decomposition of Complex Spatial Objects into Intervals. Proc. International Conference on Databases and Applications (DBA), 2004.

[KMPS 01] Kriegel H.-P., Müller A., Pötke M., Seidl T.: DIVE: Database Integration for Virtual Engineering (Demo). Demo Proc. 17th Int. Conf. on Data Engineering (ICDE), 15-16, 2001.

[KPS 00] Kriegel H.-P., Pötke M., Seidl T.: Managing Intervals Efficiently in Object-Relational Databases. Proc. 26th Int. Conf. on Very Large Databases (VLDB), 407-418, 2000.

[KPS 01] Kriegel H.-P., Pötke M., Seidl T.: Interval Sequences: An Object-Relational Approach to Manage Spatial Data. Proc. 7th Int. Symposium on Spatial and Temporal Databases (SSTD), LNCS 2121, 481-501, 2001.

[MH 99] Möller T., Haines E.: Real-Time Rendering. A K Peters, Natick, MA, 1999.

[MJFS 96] Moon B., Jagadish H. V., Faloutsos C., Saltz J. H.: Analysis of the Clustering Properties of Hilbert Space-filling Curve. Techn. Report CS-TR-3611, University of Maryland, 1996.

[MPT 99] McNeely W. A., Puterbaugh K. D., Troy J. J.: Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling. Proc. ACM SIGGRAPH Int. Conf. on Computer Graphics and Interactive Techniques, 401-408, 1999.

[Ora 99] Oracle Corp.: Oracle8i Data Cartridge Developer’s Guide, Release 2 (8.1.6). Redwood Shores, CA, 1999.

[Ore 89] Orenstein J. A.: Redundancy in Spatial Databases. Proc. ACM SIGMOD Int. Conf. on Management of Data, 294-305, 1989.

[RRSB 99] Ravi Kanth K. V., Ravada S., Sharma J., Banerjee J.: Indexing Medium-dimensionality Data in Oracle. Proc. ACM SIGMOD Int. Conf. on Management of Data, 521-522, 1999.

[Sam 90] Samet H.: Applications of Spatial Data Structures. Addison Wesley Longman, Boston, MA, 1990.

[SMS+ 00] Srinivasan J., Murthy R., Sundara S., Agarwal N., DeFazio S.: Extensible Indexing: A Framework for Integrating Domain-Specific Indexing Schemes into Oracle8i. Proc. 16th Int. Conf. on Data En-

gineering (ICDE), 91-100, 2000.

[STEP 94+] International Organization for Standardization: ISO 10303, Industrial automation systems and integration – Product data representation and exchange (STEP, Parts 1-520). Geneva, 1994-2001.

[VDA 87] Verband der Automobilindustrie: VDA-Flächenschnittstelle VDAFS, Version 2.0. Frankfurt a. M., 1987 (in German)



Hans-Peter Kriegel ist seit 1991 Inhaber des Lehrstuhls für Datenbanksysteme an der Universität München und seit 2003 Direktor des Departments „Institut für Informatik“. Er hat über 200 Beiträge auf begutachteten Tagungen sowie in Fachzeitschriften veröffentlicht. Diplom (1973) und Promotion (1976) in Informatik erfolgten an der Universität Karlsruhe. Seine Forschungsschwerpunkte liegen in den Bereichen Datenbanksysteme für komplexe Objekte (Molekularbiologie, Medizin, Multimedia, CAD usw.), insbesondere Anfragebearbeitung, Ähnlichkeitssuche, hochdimensionale Indexstrukturen sowie im Bereich Knowledge Discovery in Databases und Data Mining.



Stefan Brecheisen arbeitet als wissenschaftlicher Mitarbeiter in der Gruppe von Prof. Hans-Peter Kriegel im Bereich Ähnlichkeitssuche von räumlichen Objekten



Peter Kunath arbeitet als wissenschaftlicher Mitarbeiter in der Gruppe von Prof. Hans-Peter Kriegel im Bereich Effiziente Verwaltung von räumlichen Objekten und Multimedia-daten.



Martin Pfeifle arbeitet als wissenschaftlicher Mitarbeiter in der Gruppe von Prof. Hans-Peter Kriegel. Er hat seine Dissertation mit dem Titel „Spatial Database Support for Virtual Engineering“ im Frühjahr 2004 eingereicht.



Matthias Renz arbeitet als wissenschaftlicher Mitarbeiter in der Gruppe von Prof. Hans-Peter Kriegel im Bereich Effiziente Verwaltung von räumlichen und mobilen Objekten.