EIN ALGORITHMUS FÜR DIE REKURSIVE ABSTANDBESTIMMUNG SPACETREE-CODIERTER RASTERGEOMETRIEN

André Borrmann[1] und Christoph van Treeck[1]

[1] Lehrstuhl für Bauinformatik, Technische Universität München

{ borrmann | treeck }@bv.tu-muenchen.de

Kurzfassung: *Dieses Paper stellt einen Algorithmus vor, mit dessen Hilfe die euklidische Distanz zweier Spacetree-codierter geometrischer Objekte im n-dimensionalen Raum bestimmt werden kann. Der Algorithmus bedient sich der Hierarchie der raumpartionierenden Struktur, indem durch rekursives Absteigen eine sukzessive Erhöhung der Genauigkeit erreicht wird. Dabei werden auf jeder Ebene diejenigen Zellpaare als Kandidaten für eine Untersuchung auf der nächstfeineren Ebene ausgewählt werden, deren Partner zu den potentiell am nächsten liegenden gehören. Auf diese Weise werden unnötige Berechnungen vermieden und mit jeder Rekursion die Genauigkeit schrittweise erhöht. Wesentlicher Vorteil gegenüber herkömmlichen Ansätzen ist, dass der Aufwand zur Berechnung des Abstands nicht abhängig von der Komplexität der Oberfläche, sondern von der benötigten Genauigkeit ist.*

1 Einführung

Zur besseren Unterstützung der räumlichen Analyse von Bauwerksmodellen und der Partialmodellbildung soll in einem neuen Forschungsprojekt eine Anfragesprache für Bauwerksmodelle entwickelt und implementiert werden, die es erlaubt, räumliche Operatoren für die Spezifikation der Resultatmenge zu benutzen. Dazu gehören neben topologischen, direktionalen und Booleschen auch metrische Operatoren [1]. Das von uns verfolgte Konzept zur Implementierung dieser „3D Spatial Query Language“ auf Basis von SQL:1999 sieht die Speicherung der expliziten Geometrie der einzelnen Bauteile eines Gebäudes innerhalb einer objekt-relationalen Datenbank vor.

Die Anfragebearbeitung soll auf Grundlage einer oktalbaum-codierten Voxelrepräsentation der Bauteilgeometrie durchgeführt werden (Abb. 1). Ein Voxel ist eine Zelle mit gleicher Kantenlänge im dreidimensionalen Raum und damit das Äquivalent zu einem Pixel in 2D. Die Voxelrepräsentation einer Geometrie erhält man durch Diskretisierung der Ausgangsrepräsentation (üblicherweise B-Rep) in einem uniformen kartesischen Gitter (= Raster). Man spricht daher auch von Rastergeometrien.

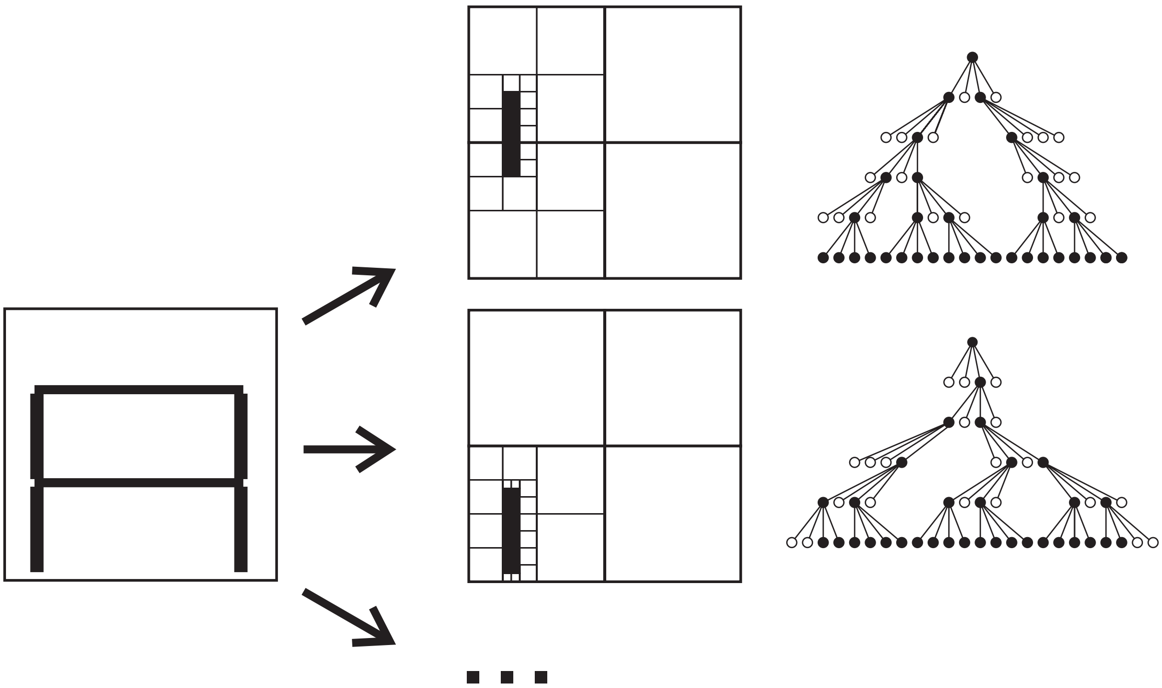

Abb. 1: Quadtree-Codierung der Bauteile eines Bauwerks in 2D. Für die Implementierung der räuml. Anfragesprache werden analog Oktalbäume zur Codierung der 3D-Geometrie verwendet.

Der Oktalbaum ist eine spezielle Ausprägung des Spacetree. Hierbei handelt es sich um eine raumpartitionierende, hierarchische Datenstruktur, bei der jedes Element (im weiteren wird ein Element als Zelle bezeichnet) einen Vorgänger (Elternzelle) und entweder *0* oder *2d* Nachfolger (Kindzellen) hat, wobei *d* der Anzahl der Raumdimensionen entspricht. Für *d=2* spricht man von einem Quadtree [3], für *d=3* von einem Octree [2] [4]. Die Verhältnis der Kantenlänge einer Kindzelle zur Elternzelle ist dabei stets 1:2. Die Vereinigung aller Kindzellen ergibt die Elternzelle. Betrachtet man nun eine Zelle an einer beliebigen Position im Baum isoliert mit ihrer kompletten nachfolgenden Struktur, so erhält man wieder einen Spacetree. Diese Eigenschaft erlaubt die Verwendung rekursiver Algorithmen.

Für den Aufbau des Spacetree gibt es verschiedene Vorgehensweisen, abhängig davon, ob bei einem zu diskretisierenden Element, das die gleiche Dimensionalität wie der der umgebende Raum hat (2D: Region, 3D: Körper), lediglich der Rand aufgelöst [7] oder auch das Innere entsprechende markiert werden soll [5]. Gemein ist allen Ansätzen, dass beginnend beim Wurzelelement nur diejenigen Kindzellen weiter verfeinert werden, die weder vollständig innerhalb, noch vollständig außerhalb des zu diskretisierenden Gebildes liegen, also seine Berandung schneiden. Im Falle eines zweifarbigen Baums werden die Randzellen als belegt (=schwarz) markiert, alle anderen als leer (=weiß). Beim dreifarbigen Baum werden hingegen innen liegende Zellen als schwarz, außen liegende als weiß und auf dem Rand liegende als grau markiert.

Der Einsatz der Oktalbaum-Codierung erlangt bei dem von uns verfolgten Ansatz zur Implementierung räumlicher Operatoren aus zwei Gründen besondere Bedeutung: Erstens muss für jedes einzelne Bauteil eine Voxelrepräsentation innerhalb eines Bereichs erzeugt werden muss, der das gesamte Bauwerk abdeckt (Abb.1). Dies führt zwangsläufig zu großen Bereichen innerhalb des jeweiligen Rasters mit unbelegten Zellen. Durch die Oktalbaum-Codierung ist es möglich, diese Bereiche sehr grob aufzulösen, Bereiche am Rand der Geometrie hingegen sehr fein. Zweitens birgt die hierarchische Struktur des Oktalbaums erhebliche algorithmische Vorteile beim Durchführen von Booleschen Operationen [5] und, wie im folgenden gezeigt wird, bei der Abstandsbestimmung.

2 Gegenstand

Gegenstand dieses Papers ist ein Algorithmus, mit dessen Hilfe der Euklidische Abstand zweier Objekte bestimmt werden kann, deren Geometrie mit Hilfe eines Spacetree codiert wurde. Wesentlicher Vorteil gegenüber herkömmlichen Algorithmen, die meist auf der Abstandsbestimmung zwischen den Dreiecken der facettierten Oberflächen beruhen, ist seine ausgezeichnete Skalierbarkeit. Im Gegensatz zu vorgenannten Algorithmen, ist der Aufwand zur Berechnung des Abstands nicht abhängig von der Komplexität der Oberfläche, sondern lediglich von der benötigten Genauigkeit.

Eingangsgrößen sind die beiden Oktalbäume der Objekte, deren Abstand bestimmt werden soll, sowie die benötigte Genauigkeit als Abbruchkriterium. Ausgabewert ist ein Intervall in IR^+, in dem der Abstand der Objekte liegt. Letzteres kann auch als Angabe des maximalen Fehlers interpretiert werden.

Wenngleich die praktische Verwertbarkeit auf 2D- und 3D-Räume beschränkt bleiben wird, ist der Algorithmus grundsätzlich für beliebig-dimensional Räume anwendbar. Entsprechend wird im folgenden das Wort „Zellen“ als Verallgemeinerung für Quadranten, Oktanten etc. verwendet.

Mit Hilfe des Algorithmus können verschiedenste metrische Operatoren implementiert werden wie:

double **distance**(GeoObject obj1, GeoObject obj2)

boolean **closerThan**(GeoObject obj1, GeoObject obj2, double dist)

boolean **fartherThan**(GeoObject obj1, GeoObject obj2, double dist)

Gerade bei den letzten beiden Operatoren kommt der iterative Charakter des Algorithmus zum Tragen, da er hier ggf. schon bei vergleichsweise grober Auflösung abbrechen kann.

3 Algorithmus

Ein naiver Algorithmus zur Distanzbestimmung sähe die Bestimmung des Abstands aller Zellen aus A zu allen Zellen aus B mit anschließender Minimum-Bildung vor. Bei einer sehr feinen Auflösung von beispielsweise 2^{12} x 2^{12} x 2^{12} Rasterpunkten, wie sie für eine ausreichend genaue Repräsentation der Bauteile eines Gebäude notwendig ist, wären hierzu im Extremfall 2^{36} x 2^{36} Abstandsbestimmungen notwendig.

Um diese hohe Zahl an Operationen zu vermeiden, beruht der hier vorzustellende Algorithmus auf einer sukzessiven Verfeinerung im Sinne der Ebenen eines Spacetree und dem Prinzip, dass auf einer beliebigen Ebene jeweils diejenigen Zellpaare (ein Paar besteht aus einer Zelle aus Baum A und einer aus Baum B) von einer weiteren Verfeinerung ausgeschlossen werden können, deren Abstand bereits über dem der übrigen Zellpaare liegt.

Hierbei ist zu bedenken, dass eine als „gefüllt" markierte Zelle lediglich eine unscharfe Aussage hinsichtlich der tatsächlichen Position der Berandung liefert (Abb. 2). Daher wird für jedes Zellpaar jeweils ein oberer und ein unterer Abstandswert bestimmt. Diese beiden Werte reflektieren den Bereich, in dem der tatsächliche Abstand liegt. Sie berechnen sich aus den Koordinaten der Zelle im Koordinatensystem der aktuellen Ebene des Spacetree.

Dabei wird in jeder Koordinatenrichtung wie folgt der minimale und der maximale Abstandwert berechnet: Ist der Koordinatenwert der beiden Zellen gleich, so ist der minimale Abstand 0 und der maximale Abstand 1. In allen anderen Fällen ergibt sich der minimale Abstand aus der Differenz der Koordinatenwerte minus 1 und der maximale Abstand aus der Differenz der Koordinatenwerte plus 1. Aus den drei so erhaltenen minimalen und maximalen Werten berechnen sich der minimale und der maximale Abstandswert als Summe der Quadrate dieser Werte. Die aufwändige Berechnung der Quadratwurzel ist hier nicht notwendig, da es zunächst lediglich festzustellen gilt, in welcher Ordnungsrelation (näher, ferner) die Zellpaare untereinander stehen.

Auf dieser Grundlage werden diejenigen Zellpaare ermittelt, die zu den Kandidaten für das Paar mit dem kürzesten Abstand gehören. Dabei wird zunächst der kleinste obere Abstandwert aller Paare bestimmt. Danach werden alle diejenigen Zellpaare ausgeschlossen, deren unterer Abstandwert größer ist als der kleinste obere Abstandswert.

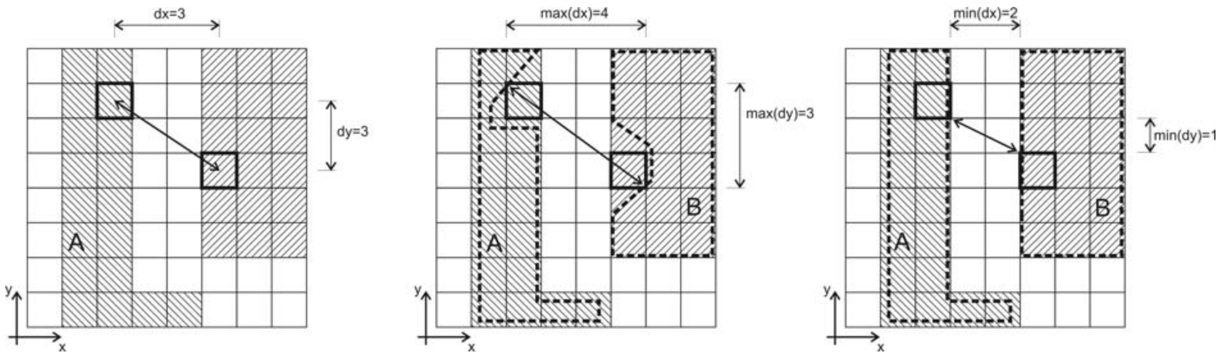

Abb. 2: Da die genaue zugrunde liegende Geometrie nicht bekannt ist, müssen für jedes Paar ausgehend von den richtungsweisen Zellabständen (links), obere Abstandswerte (potentielle Konstellation in der Mitte) und untere Abstandswerte (potentielle Konstellation rechts) bestimmt werden. Für das gezeigte Zellpaar ergibt sich der obere Abstandswert zu $4^2+3^2=9$ und der untere zu $1^2+2^2=5$.

Alle übrigen Zellpaare sind Kandidaten. Für sie wird der Algorithmus rekursiv durchgeführt, d.h. sie werden verfeinert und für die dabei entstehenden Subzellenpaare werden wiederum Abstandswerte bestimmt, auf deren Grundlage Kandidaten für die nächste Ebene ausgewählt werden. Abbruchkriterium für die Rekursion ist das Erreichen der geforderten Genauigkeit bei der Abstandsberechung bzw. das Erreichen der untersten Ebene der Spacetrees.

Der Bereich, in dem der tatsächliche Abstand liegt, wird nach Beendigung der Rekursion ermittelt. Er ergibt sich aus der Quadratwurzel des kleinsten unteren Abstandswertes und der Quadratwurzel des kleinsten oberen Abstandswertes, jeweils multipliziert mit der Kantenlänge einer Zelle auf der aktuellen Ebene des Oktalbaums.

4 Beispiel

Die Arbeitsweise des Algorithmus soll anhand eines Beispiels in 2D erläutert werden. Gegeben ist die Quadtree-codierte Rasterrepräsentation zweier Flächen A und B (Abb.3), deren exakte Hülle strichliert dargestellt ist. Zum besseren Verständnis werden in den Abbildungen A und B die Rasterrepräsentationen der beiden Körper auf der jeweiligen Ebene in einem gemeinsamen Raster gezeigt. Dabei werden durch Fläche A belegte Quadranten mit nach rechts unten laufender Schraffur dargestellt, durch Fläche B belegte Quadranten mit nach rechts oben laufender Schraffur.

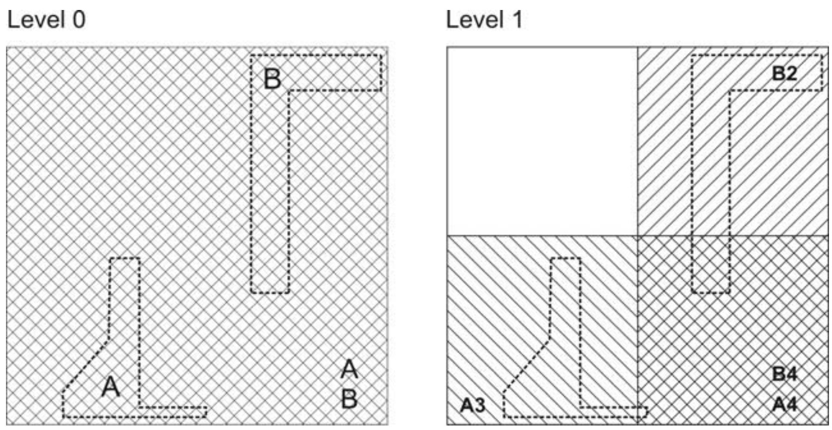

Abb. 3: Darstellung der Ebenen 0 und 1 der beiden Quadtrees in einem gemeinsamen Raster. Mit nach schräg unten verlaufender Schraffur sind Zellen gekennzeichnet, die zur Fläche A gehören, mit schräg nach oben verlaufender Schraffur zu Fläche B gehörende.

Der Algorithmus beginnt auf der obersten Ebene der Quadtrees (Ebene 0), auf der es nur einzelne Zelle, die Wurzelzelle gibt. Auf Ebene 0 wird sich immer eine Überlappung der beiden Körper ergeben (Abb. 3, links). Trotzdem gelten auch hier die oben beschriebenen Regeln zur Berechnung des unteren und des oberen Abstandswertes. Entsprechend kann auf dieser Ebene die Aussage getroffen werden, dass der Abstand der Körper zwischen 0 und $\sqrt{2}$ mal der Länge der Ursprungszelle liegt. Angenommen die Ursprungszelle habe eine Länge von 8m, so ergibt sich ein Abstandsbereich von 0 bis 11,3m. Eine derartig grobe Aussage wird aber in der Regel nicht von Nutzen sein. Daher ist eine weitere Verfeinerung notwendig.

Tabelle 1 zeigt die Berechnung der oberen und unteren Abstandswerte für die Ebenen 1 bis 3. Auf Ebene 1 ergeben sich zwei A-Quadranten und zwei B-Quadranten und somit 4 Paare. Der kleinste obere Abstandswert stellt sich beim Paar (A4,B4) ein und beträgt 2.

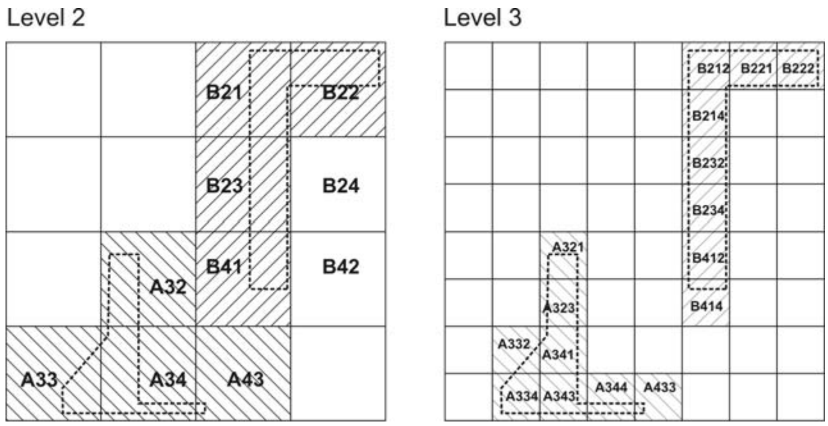

Abb. 4: Darstellung der Ebenen 2 und 3 der beiden Quadtrees. Die verwendete Nummerierung der Zellen ergibt sich aus ihrer „Geschichte", d.h. aus der Nummerierung der Vater- und Großvaterzellen

Ebene	Paar		Zellabstand		Unterer Abstandswert			Oberer Abstandswert			Kand.
	A	B	x	y	x	y	gesamt	x	y	gesamt	
0	1	1	0	0	0	0	0	1	1	2	ja
1	3	2	1	1	0	0	0	2	2	8	ja
	3	4	1	0	0	0	0	2	1	5	ja
	4	2	0	1	0	0	0	1	2	5	ja
	4	4	0	0	0	0	0	1	1	**2**	ja
2	32	22	2	2	1	1	2	3	3	18	ja
	32	23	1	1	0	0	0	2	2	8	ja
	32	24	2	1	1	0	1	3	2	13	ja
	32	41	1	0	0	0	0	2	1	**5**	ja
	32	42	2	0	1	0	1	3	1	10	ja
	33	21	2	3	1	2	5	3	4	25	ja
	33	22	3	3	2	2	**8**	4	4	32	**nein**
	33	23	2	2	1	1	2	3	3	18	ja
	33	24	3	2	2	1	5	4	3	25	ja
	33	41	2	1	1	0	1	3	2	13	ja
	33	42	3	1	2	0	4	4	2	20	ja
	34	21	1	3	0	2	4	2	4	20	ja
	34	22	2	3	1	2	5	3	4	25	ja
	34	23	1	2	0	1	1	2	3	13	ja
	34	24	2	2	1	1	2	3	3	18	ja
	34	41	1	1	0	0	0	2	2	8	ja
	34	42	2	1	1	0	1	3	2	13	ja
	43	21	0	3	0	2	4	1	4	17	ja
	43	22	1	3	0	2	4	2	4	20	ja
	43	23	0	2	0	1	1	1	3	10	ja
	43	24	1	2	0	1	1	2	3	13	ja
	43	41	0	1	0	0	0	1	2	5	ja
	43	42	1	1	0	0	0	2	2	8	ja
3	321	212	3	4	2	3	13	4	5	41	ja
	321	214	3	3	2	2	8	4	4	32	ja
	321	221	4	4	3	3	**18**	5	5	50	nein
	321	222	5	4	4	3	**25**	6	5	61	nein
	321	232	3	2	2	1	5	4	3	25	ja
	321	234	3	1	2	0	4	4	2	20	ja
	321	412	3	0	2	0	4	4	1	17	ja
	321	414	3	1	2	0	4	4	2	20	ja
	323	212	3	5	2	4	**20**	4	6	52	nein
	323	214	3	4	2	3	13	4	5	41	ja
	323	221	4	5	3	4	**25**	5	6	61	nein
	323	222	5	5	4	4	**32**	6	6	72	nein
	323	232	3	3	2	2	8	4	4	32	ja
	323	234	3	2	2	1	5	4	3	25	ja
	323	412	3	1	2	0	4	4	2	20	ja
	323	414	3	0	2	0	4	4	1	17	ja
	332	212	4	6	3	5	**34**	5	7	74	nein
	332	221									
	332	222									
	332	214	4	5	3	4	**25**	5	6	61	nein
	332	232	4	4	3	3	**18**	5	5	50	nein
	332	234	4	3	3	2	13	5	4	41	ja
	332	412	4	2	3	1	10	5	3	34	ja
	332	414	4	1	3	0	9	5	2	29	ja
	341	212	3	6	2	5	**29**	4	7	65	nein
	341	221	4	6	3	5	**34**	5	7	74	nein
	341	222	5	6	4	5	**41**	6	7	85	nein
	341	214	3	5	2	4	**20**	4	6	52	nein
	341	232	3	4	2	3	13	4	5	41	ja
	341	234	3	3	2	2	8	4	4	32	ja
	341	412	3	2	2	1	5	4	3	25	ja
	341	414	3	1	2	0	4	4	2	20	ja
	334	212	4	7	3	6	**45**	5	8	89	nein
	334	221	5	7	4	6	**52**	6	8	100	nein
	334	222	6	7	5	6	**61**	7	8	113	nein
	334	214	4	6	3	5	**34**	5	7	74	nein
	334	232	4	5	3	4	**25**	5	6	61	nein
	334	234	4	4	3	3	**18**	5	5	50	nein
	334	412	4	3	3	2	13	5	4	41	ja
	334	414	4	2	3	1	10	5	3	34	ja
	343	212	3	7	2	6	**40**	4	8	80	nein
	343	221	4	7	3	6	**45**	5	8	89	nein
	343	232	3	5	2	4	**20**	4	6	52	nein
	343	234	3	4	2	3	13	4	5	41	ja
	343	412	3	3	2	2	8	4	4	32	ja
	343	414	3	2	2	1	5	4	3	25	ja
	344	212	2	7	1	6	**37**	3	8	73	nein
	344	221	3	7	2	6	**40**	4	8	80	nein
	344	222	4	7	3	6	**45**	5	8	89	nein
	344	214	2	6	1	5	**26**	3	7	58	nein
	344	232	2	5	1	4	**17**	3	6	45	nein
	344	234	2	4	1	3	10	3	5	34	ja
	344	412	2	3	1	2	5	3	4	25	ja
	344	414	2	2	1	1	2	3	3	18	ja
	433	212	1	7	0	6	**36**	2	8	68	nein
	433	221	2	7	1	6	**37**	3	8	73	nein
	433	222	3	7	2	6	**40**	4	8	80	nein
	433	214	1	6	0	5	**25**	2	7	53	nein
	433	232	1	5	0	4	**16**	2	6	40	nein
	433	234	1	4	0	3	9	2	5	29	ja
	433	412	1	3	0	2	4	2	4	20	ja
	433	414	1	2	0	1	1	2	3	**13**	ja

Tabelle 1: Berechnung der oberen und unteren Abstandswerte für die Ebenen 1 bis 3 des Beispiels

Da es kein Paar gibt, dessen unterer Abstandswert oberhalb dieses Wertes liegt, müssen alle Paare den nächsten Rekursionsschritt durchlaufen, d.h. jedes der Paare wird verfeinert und für die sich ergebenden Kindzellen die entsprechenden Abstandswerte bestimmt.

Auf Ebene 2 ergeben sich vier von A belegte Quadranten und vier von B belegte Quadranten und damit insgesamt 16 Paare. Betrachtet man die sich ergebenden Abstandswerte in Tabelle 1, so wird man feststellen, dass genau ein Paar den Bedingungen für den Ausschluss aus der Rekursion (einer weiteren Verfeinerung) genügt: Das Paar (A33, B22) hat einen Abstandswert von 8 und liegt damit über dem kleinsten oberen Abstandswert von 5. Alle anderen Paare erfüllen dieses Kriterium nicht, und müssen daher weiter verfeinert werden.

Auf dieser Ebene lässt sich hinsichtlich des Abstands zwischen A und B die Aussage treffen, dass er zwischen 0 (= Wurzel aus kleinstem unteren Abstandswert) und $\sqrt{5}$ (Wurzel aus kleinstem oberen Abstandswert) mal Rasterabstand = Zellgröße = 2m liegt, also zwischen 0 und 4,47m.

Auf Ebene 3 ergeben sich sechs von A belegte Zellen und sechs von B belegte Zellen und damit insgesamt 36 Paare. Da die Kinder von A33 und B22 jedoch bereits ausgeschlossen wurden, entfallen die Paare (A332, B221) und (A332, B222), und die Anzahl der zu berechnenden Paare erniedrigt sich auf 34. Wichtig ist, dass nicht alle Kombinationen, in denen einer der beiden Väter A33 oder B22 ist, ausgeschlossen werden, sondern nur die Kombinationen, bei denen sowohl A33 als auch B22 Väter sind. Der kleinste obere Abstandswert auf Ebene 5 beträgt 13 und tritt beim Paar (A433, B414) auf. Es können 33 Paare von der weiteren Verfeinerung ausgeschlossen werden, da ihr größter unterer Abstandswert größer als 5 ist. Lediglich 29 Paare müssen weiter verfeinert werden. Würde der Algorithmus hier abbrechen, ließe sich der Abstand der Körper A und B als zwischen 1 mal Zellänge=1m und $\sqrt{13}$ * 1m = 3,61m gelegen angeben.

5 Diskussion

Der vorgestellte Algorithmus zur Abstandsbestimmung lässt sich für unterschiedliche Spacetree-Arten und Generierungstechniken einsetzen. So ist er sowohl für zweifarbige als auch für dreifarbige Spacetrees einsetzbar. Weiterhin kann man entweder mit bereits vollständig generierten Spacetrees arbeiten, oder entsprechend der von Mundani vorgeschlagenen Vorgehensweise die Spacetrees erst während der Distanzmessung erzeugen. Hierbei ist es möglich, schon bei der Generierung nur diejenigen Zellen zu verfeinern, die zu den Kandidaten der Abstandsmessung gehören. Dies bringt mehrere Vorteile: Zum einen wird die Tiefe des Baums nicht im vornherein festgelegt, sondern durch die benötigte Genauigkeit bestimmt. Zum anderen wird der Baum nur an den „interessanten“ Stellen in der notwendigen Tiefe aufgebaut.

Schließlich lässt sich noch anmerken, dass der Algorithmus mit leichten Anpassungen auch zur Feststellung einer Durchdringungstiefe zweier Körper geeignet ist. Dazu muss zunächst der Boolesche Schnitt beider Körpern bestimmt werden, beispielsweise mit dem in [6] beschriebenen Verfahren. Für den resultierenden Spacetree wird dann der maximale Abstand zu sich selbst bestimmt. Dazu werden auf jeder Ebene nicht wie bei der oben beschriebenen Abstandsbestimmung die Paare mit einem minimalen Abstandswert für eine weitere Verfeinerung ausgewählt, sondern die mit einem maximalen.

Literatur

[1] A. Borrmann, C. van Treeck and E. Rank: "Towards a 3D Spatial Query Language for Building Information Models," *Proc. of Joint Int. Conf. for Computing and Decision Making in Civil and Building Engineering*, 2006.

[2] G. M. Hunter: “Efficient computation and data structures for graphics.” PhD thesis, Department of Electrical Engineering and Computer Science, Princeton University, Princeton, NJ, USA, 1978.

[3] A. Klinger: “Patterns and Search Statistics,” in Rustagi, J. (ed.): *Optimizing Methods in Statistics*, Academic Press, New York, 1971, 303-337.

[4] D. Meagher: "Geometric modeling using octree encoding," *IEEE Computer Graphics and Image Processing* vol. 19, no. 2, pp. 129-147, 1982.

[5] R.-P. Mundani "Hierarchische Geometriemodelle zur Einbettung verteilter Simulationsaufgaben," Dissertation, Institut für Parallele und Verteilte Systeme, Technische Universität Stuttgart, 2005.

[6] R.-P. Mundani, H.-J. Bungartz, E. Rank, R. Romberg and E. Niggl "Efficient Algorithms for Octree-Based Geometric Modelling," *Proc. of 9th Int. Conf. on Civil and Structural Engineering Computing*, 2003.

[7] P. Wenisch and O. Wenisch "Fast octree-based Voxelization of 3D Boundary Representation-Objects," Technical Report, Lehrstuhl für Bauinformatik, Technische Universität München, 2004.