
Modellierung, Implementierung und Visualisierung unscharfer räumlicher Objekte

Diplomarbeit

Thomas Behr

2002

Betreuer:



Dr. M. Schneider

Fernuniversität Hagen

Fachbereich: Informatik

Lehrgebiet: Praktische Informatik IV

Inhaltsverzeichnis

1	Einleitung	6
1.1	Problemstellung	6
1.2	Anwendungen	7
1.3	Ziel und Aufbau der Arbeit	9
2	Frühere Arbeiten und Einordnung der eigenen Arbeit	12
2.1	Unschärfe Objekte	12
2.2	Topologische Beziehungen	14
3	Abstraktes geometrisches Datenmodell	17
3.1	Definition scharfer geometrischer Objekte mit Hilfe von Punktmengen und Punkt- mengentopologie	17
3.2	Definition unscharfer geometrischer Objekte mittels Fuzzy-Mengen	21
3.2.1	Formale Grundlagen von unscharfen Mengen	22
3.2.2	Eine Abgrenzung zur Wahrscheinlichkeitstheorie	23
3.2.3	Der Skalierungsfaktor	24
3.2.4	Unstetigkeiten in der Zugehörigkeitsfunktion	25
3.2.5	Der Zugehörigkeitsgrad 0	26
3.2.6	Darstellung der Objekte	27
4	Operationen auf unscharfen geometrischen Objekten	29
5	Verallgemeinerung des 9-Intersection-Modells auf komplexe geometrische Ob- jekte	41
5.1	Topologische Beziehungen zwischen Punkten	42
5.2	Topologische Beziehungen zwischen Linien	42
5.3	Topologische Beziehungen zwischen Regionen	50
5.4	Topologische Beziehungen zwischen unterschiedlichen Objekten	55
5.4.1	Beziehungen zwischen Punkt und Linie	55
5.4.2	Beziehungen zwischen Punkt und Region	57
5.4.3	Beziehungen zwischen Linie und Region	58
5.5	Cluster und topologische Prädikate	62
6	Topologische Beziehungen auf komplexen unscharfen geometrischen Objek- ten	65
6.1	Beziehungen zwischen zwei unscharfen Punkten	67
6.2	Topologische Beziehungen zwischen unscharfen Linien	68
6.3	Topologische Beziehungen zwischen unscharfen Regionen	69
6.4	Topologische Beziehungen zwischen unscharfen Punkten und Linien	70
6.5	Topologische Beziehungen zwischen unscharfen Punkten und Regionen	70
6.6	Topologischen Beziehungen zwischen unscharfen Linien und Regionen	71

6.7	Clustering der topologischen Beziehungen unscharfer Objekte	71
7	Realisierung des abstrakten durch ein diskretes Modell	74
7.1	Das Repräsentationsproblem	74
7.2	Die geometrische Basis	75
7.2.1	Partitionierung	75
7.2.2	Wahl der Gitterpunkte	76
7.2.3	Elementardatentypen der Basis	77
7.2.4	Operatoren und Prädikate auf den Elementardatentypen	79
7.3	Repräsentation der Datentypen	81
7.3.1	Unschärfe Regionen	81
7.3.2	Unschärfe Linien	82
7.3.3	Unschärfe Punkte	83
7.4	Realisierung der Operationen	83
7.5	Berechnung von topologischen Beziehungen	110
7.5.1	Beziehungen in der Basis	110
7.5.2	Beziehungen zwischen den unscharfen Objekten	115
8	Implementierung des diskreten Modells	125
8.1	Wahl der Programmiersprache	125
8.2	Datenmodellierung	125
8.3	Algorithmen	126
8.3.1	Mengenoperatoren	126
8.3.2	Normierung	129
8.3.3	Berechnung von Zugehörigkeitsgraden	129
8.3.4	Gleichheit unscharfer Objekte	130
8.3.5	Alpha-Cuts	130
8.3.6	Der sharp-Operator	130
8.3.7	Grenze von Linie oder Region	131
8.3.8	Gemeinsame Linien von Regionen	132
8.3.9	Gemeinsame Punkte von Linien und Regionen	132
8.3.10	Größenangaben	132
8.3.11	Der Mittelwert	132
8.3.12	Basis von unscharfen Objekten	134
8.3.13	Faces von Linien und Regionen	134
8.3.14	Löcher in Regionen	135
8.3.15	Kontur einer Region	137
8.3.16	Ähnlichkeit zweier Objekte	137
8.3.17	Berechnung topologischer Beziehungen	138
8.4	Pakete	139
8.5	Klassen	140
9	Visualisierung	142
9.1	Darstellungsmöglichkeiten	142
9.1.1	Erzeugung von Farben	142
9.1.2	Die einfachste Darstellung	143
9.1.3	Eine verbesserte Darstellung	143
9.1.4	3D - Gitterdarstellung	144
9.1.5	3D - solide Darstellung	144
9.1.6	3D Darstellung mit Verlauf	145
9.2	Berechnung der 3D-Darstellung	145

10 Vorschlag für eine Benutzeroberfläche	148
10.1 Darstellung von unscharfen Objekten	148
10.1.1 Allgemeine Parameter	149
10.1.2 Spezielle Parameter der 2D-Darstellung	149
10.1.3 Spezielle Parameter der 3D-Darstellung	150
10.2 Erzeugung unscharfer Objekte	150
10.2.1 Erstellen von unscharfen Punkten	150
10.2.2 Erstellen unscharfer Linien	152
10.2.3 Erstellung unscharfer Regionen	153
10.3 Manipulation von unscharfen Objekten	154
10.4 Operationen auf unscharfen Objekten	154
11 Einbindung in Datenbanken	157
12 Schlußfolgerungen und Ausblick	159

Abbildungsverzeichnis

Europawetter am 25.6.2001	8
Von der realen Welt zur Implementierung	10
Linien mit gleicher 4-intersection-Matrix	15
Die 9-Intersection-Matrix	15
Beispiele gültiger Kurven	18
Kein Abschluß des Schnitts auf regulären Mengen	21
Eine unscharfe Menge	22
Alpha-cut für unscharfe Mengen	23
Notwendigkeit eines Skalierungsfaktors	24
Schlechte Anwendung eines interpretierten Schnittes	25
Übertragung von Unstetigkeiten	26
Kein Abschluß unter Differenz	26
Keine Löcher	34
Topologische Beziehungen zwischen Punkten	42
Topologische Beziehungen zwischen zwei Linien	50
Beziehungen zwischen zwei Regionen	55
Topologische Beziehungen zwischen Linie und Punkt	57
Beziehungen zwischen Punkt und Region	58
Beziehungen zwischen Linie und Region	62
Schnittpunkt zweier Linien in einem Raster	74
Unterschiedliche Triangulierungen	76
Verschiebung einer Region	76
Vorgeschlagene Triangulierung	77
Alternative Dreiecksdarstellung	78
Fläche eines Basisdreiecks	79
Berechnung der Zugehörigkeitsfunktion eines fET	85
Durch ein Elementardreieck bestimmter Körper	89
Dreieckige Pyramide	89
Grundfläche der Pyramide	90
Höhe der Pyramide	91
Vergleich von abstrakter und diskreter Vereinigung	97
Vergleich des Schnitts im diskreten und abstrakten Modell	99
Vergleich von abstrakter und diskreter Differenz	101
Gemeinsame Punkte zweier Linien	107
Verlauf von Zugehörigkeitsfunktionen über einem Segment	118
Bedeutung der Variablen	119
Verwendete Pakete	140

Einfachste Darstellung einer Region	143
Darstellung mit Erhalt der Stetigkeitsinformation	144
Gitterdarstellung einer unscharfen Region	144
3D-Darstellung einer unscharfen Region	145
Verbesserte 3D-Darstellung einer unscharfen Region	145
Festlegen der Sicht	146
Die Sichtpyramide	146
Parameter in der 2D-Darstellung	148
Vom \mathbb{R}^2 auf den Monitor	149
Parameter der 3D-Darstellung	150
Einteilung in Quadranten	152
Manipulation der Operandenliste	154
Anwendung binärer Operationen	155

Kapitel 1

Einleitung

1.1 Problemstellung

Heutige Datenbanksysteme(DBS) sind nicht nur in der Lage Zahlen und Texte zu verwalten, sondern auch sehr komplexe Objekte. Solche Objekte können Bilder, Videos oder andere Gebilde sein. Erweiterbaren DBS können neue Objekttypen hinzugefügt werden, wobei auch Manipulationsmöglichkeiten für diese Objekte (Operationen) in die jeweilige Abfragesprache aufgenommen werden. Um beliebige Objekte der realen Welt in solche DBS einzugliedern, ist es notwendig, sie formal zu definieren und in elektronischer Form abzuspeichern. Weiterhin muß es Manipulations- und ggf. Visualisierungsmöglichkeiten geben. Mit solchen Themen befaßt sich diese Diplomarbeit. Hierbei sollen solche Objekte betrachtet werden, die vorwiegend in Geodatenbanken und Geographischen Informationssystemen (GIS) Anwendung finden (geometrische Objekte). Die dort verwalteten realen Objekte sind z.B. Länder, Grundstücke, Seen, Straßen, Flüsse oder einzelne Koordinaten, wie Kreuzungen von Elektroleitungen. Dabei lassen sich drei Objekttypen unterscheiden. Dieses sind Regionen für die Darstellung flächiger Gebilde, Linien und Punkte.

Heutzutage sind diese Typen sehr einfach definiert, so daß nur wenige reale Objekte ausreichend genau dargestellt werden können. So sind bei einigen Anwendungen Löcher in Regionen nicht erlaubt. Dies verhindert jedoch z.B. die Darstellung von Inseln innerhalb eines Sees. Es wird manchmal gefordert, daß Regionen zusammenhängend sein müssen, wodurch eine Darstellung von Italien mit seinen vorgelagerten Inseln nicht möglich ist. Linien werden oft so eingeschränkt, daß sie keine Selbstschnitte besitzen dürfen und zusammenhängend sind. In der Realität gibt es jedoch Linien, bei denen dies nicht der Fall ist. So ist beispielsweise die Landesgrenze von Italien nicht durch eine zusammenhängende Linie darstellbar. Punkte werden meist durch eine einzige Koordinate dargestellt. Jedoch kann der Schnittpunkt zweier Linien eine ganze Menge solcher Koordinaten ergeben. Ein weiteres Problem solcher einfachen Definitionen sind Abschlußeigenschaften unter bestimmten Operationen. So ergibt die Vereinigung zweier disjunkter Regionen eine nicht zusammenhängende Region. Die Differenz einer Region und einer anderen Region ergibt eine Region mit Loch, wenn sie Teil dieser anderen Region ist. Somit sind die Ergebnisse solcher Operationen nicht darstellbar.

Ein anderes Problem bei existierenden Darstellungen ist, daß Regionen, Linien und Punkte als scharf begrenzt dargestellt werden. Das heißt, daß für jeden Punkt das Enthaltensein in einem geometrischen Objekt durch einen booleschen Wert (ja/nein) angegeben werden kann. Für einige Objekte, z.B. Häuser, Landesgrenzen oder Straßen, ist dies auch völlig ausreichend, bei anderen ist dies nicht der Fall. Dies kann unterschiedliche Ursachen haben, welche sich in zwei große Kategorien einteilen lassen.

Die erste Gruppe von Ursachen entsteht durch Messfehler oder ungenügende Darstellungsmöglichkeiten der Objekte. Dabei hängen die Ursachen stark von der verwendeten Technik und von der benötigten Genauigkeit der Messungen ab. So kann die Darstellung von geometrischen Objekten in einem Computer dadurch verbessert werden, indem ein genaueres Zahlenformat

gewählt wird und/oder mehr Messpunkte(Stützstellen) verwendet werden. Hier sollen solche Ungenauigkeiten nicht weiter betrachtet werden.

Die zweite Ursache ist das darzustellende Objekt selbst. Betrachtet man z.B. eine Region „Lebensraum der Sumfdotterblume“, so gibt es Stellen, an denen diese Pflanze sehr häufig vorkommt und Orte, an denen sie nur vereinzelt zu finden ist. Der Versuch, eine solche Region in eine scharfe Grenze zu pressen, ist sehr subjektiv, was folgendes Beispiel aus [27] zeigt :

Mehrere Menschengruppen sollten aus einem Orthophoto (Maßstab 1:10000) der Gemeinde Oslip im Burgenland folgende Inhalte digital abgrenzen und erfassen:

- Siedlung (begrenzt durch den äußeren Siedlungsrand)
- Wald
- vielfältig, reich strukturierte Kulturlandschaft

Als Ergebnis dieses Tests konnte man feststellen, daß es sogar bei der ersten (wahrscheinlich schärfsten) Kategorie Abweichungen in der Fläche um den Faktor 2 gab. (45 ha und 90 ha). Bei der dritten Kategorie lag die ausgewiesene Fläche zwischen 5 und 300 ha.

In diesem Experiment kann man beobachten, daß es tatsächlich Objekte gibt, die sich schlecht abrupt abgrenzen lassen. Meßfehler kann man bei diesem Experiment ausschließen, da alle Gruppen das gleiche Luftbild für ihre Auswertungen benutzten. Auch die Genauigkeit (Anzahl der verwendeten Stützstellen), mit der die Flächen letztendlich dargestellt wurden, hat zumindest bei der letzten Abweichung kaum eine Bedeutung.

In [46] heißt es: „[...] wenn die Komplexität eines Systems ansteigt, verlieren präzise Aussagen ihren Sinn und sinnvolle Aussagen ihre Präzision“. Dies kann durch das eben angesprochene Beispiel bestätigt werden. Dazu betrachte man die Aussage: „Die Fläche der vielfältig, reich strukturierten Kulturlandschaft der Gemeinde Oslip beträgt 5 ha“. Hier handelt es sich um eine präzise Aussage, die jedoch durch ihre starke Subjektivität wenig bedeutungsvoll ist. Beim Versuch, solche Subjektivität möglichst auszuschalten, erhält man die folgende Aussage: „ Die Fläche der vielfältig, reich strukturierten Kulturlandschaft der Gemeinde Oslip liegt zwischen 5 und 300 ha“. Dies ist natürlich sehr unpräzise. Im Endeffekt kann man sagen, daß durch die aufgezwungene Präzision des verwendeten Modells wenig sinnvolle Aussagen möglich sind.

Geometrische Objekte, bei denen eine einfache ja/nein-Aussage für eine Koordinate nicht genügt, um die Zugehörigkeit zu einem Objekt auszudrücken, werden im Folgenden als unscharfe Objekte bezeichnet.

1.2 Anwendungen

Es gibt eine Reihe von Anwendungen, die mit Hilfe der heutigen GIS nur unzureichend gelöst werden können. Solche Anwendungen sind z.B. die Darstellung von Gebieten der Luftverschmutzung, der Strahlenbelastung und der Schallemission.

Als Beispiel sei eine Wetterkarte für Europa am 25.6.2001 angegeben:

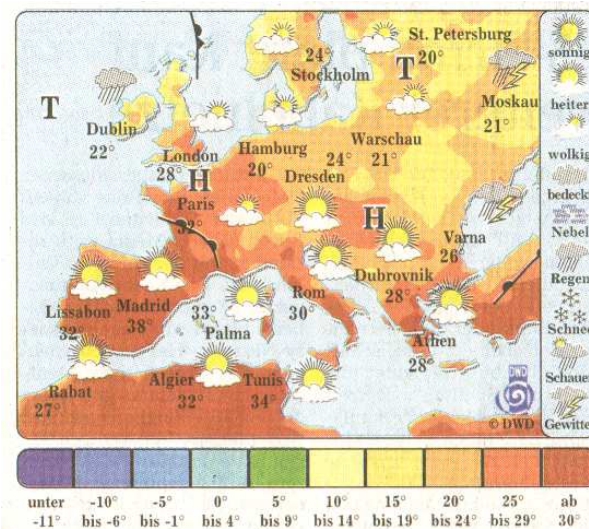


Abb. 1 Europawetter am 25.6.2001

Hier betrachte man die Darstellung der erwarteten Temperaturen. Zwischen den verschiedenen Temperaturbereichen treten stets Sprünge auf, was in der Realität selbstverständlich nie vorkommt. Weiterhin fällt auf, daß eine einfarbige Fläche einen gesamten Temperaturbereich abdeckt. So kann die Temperaturverteilung innerhalb solch einer Fläche nicht abgelesen werden. Hier handelt es sich im Endeffekt auch nicht um die Darstellung einer einzigen Region „Temperaturen in Europa“, sondern um mehrere Regionen „Temperatur X in Europa“. Mit dem Modell, welches in dieser Arbeit entwickelt wird, ist es möglich, die gesamte Temperaturverteilung innerhalb einer einzigen Region darzustellen, wobei auch die Temperaturen einzelner Koordinaten dargestellt werden. Aufgrund von Abschlußigenschaften kann solch eine Region z.B. aus der Vereinigung der Regionen „Temperaturen in Land xy“ hervorgehen. Ebenfalls wird es möglich sein, die „Temperaturen in Deutschland“ durch einen Schnitt mit der (scharfen) Region „Deutschland“ zu extrahieren.

Auch die meisten natürlichen Gebiete, wie Lebensräume bestimmter Tier- und Pflanzenarten, lassen sich nur schlecht in feste Grenzen pressen. Aber nicht nur bei flächigen Gebilden gibt es Unzulänglichkeiten in den bisherigen Darstellungsmöglichkeiten der Objekte. So lassen sich beispielsweise die Staubbelastung auf Autobahnen oder Unfallschwerpunkte nicht modellieren. Die bisherigen Ausführungen beschränkten sich allein auf die Repräsentation von unscharfen Objekten. Diese können jedoch nur dann sinnvoll angewendet werden, wenn man in der Lage ist, solche Objekte auch zu verarbeiten. Da die klassischen scharfen Objekte eine Teilmenge der Objekttypen des hier entwickelten Modells darstellen, lassen sich die Fragestellungen für diese Objekte auch hier lösen.

Solche Fragen können sein:

- Wieviel Meter Zaun benötigt Grundstücksbesitzer A, um sein gesamtes Grundstück einzuzäunen?
- Wieviel Zaun benötigt Grundstücksbesitzer A, um sein Grundstück gegen seinen Nachbarn B abzusichern?
- Wie groß ist das größte zusammenhängende Waldstück in Niedersachsen?
- Welches Land gehört B, wenn er/sie zusätzlich Grundstück A erwirbt?
- Welches Land gehört B nach dem Verkauf von Grundstück C?

- Welche Grundstücke sind zu Grundstück A benachbart?
- usw.

Solche und ähnlich Aufgaben können bereits mit Modellen für scharfe Objekte gelöst werden. Da es sich jedoch um eine Erweiterung dieser Modelle handelt, können auch solche Fragestellungen beantwortet werden, für die scharfe Modelle nicht geeignet sind. Dies können z.B. sein:

- Wie sieht die Gesamtluftverschmutzung aller Kraftwerke in einem bestimmten Gebiet aus?
- Welche Teile des Naturschutzgebietes X werden durch Kraftwerk Y beeinflusst?
In einem scharfen Modell läßt sich dies zwar ebenfalls berechnen, jedoch kann keine Aussage darüber getroffen werden, wie stark diese Belastung in bestimmten Bereichen ist.
- In welchen Gebieten ist das Vorkommen der Tierart X nicht durch Tierart Y gefährdet?
Hier läßt sich im Gegensatz zu herkömmlichen Modellen auch der Grad der Gefährdung darstellen.
- Wie hoch ist die Bevölkerungsdichte an einem bestimmten Punkt?
- Wo sollte der Fischfang unterbleiben, da die Belastung des Wassers durch bestimmte Chemikalien zu hoch ist?

Die Visualisierung unscharfer Objekte führt zu weiteren Anwendungen. So lassen sich z.B. aussagekräftigere Wetterkarten darstellen, als oben angegeben wurde. Durch entsprechende Darstellungsmöglichkeiten ist man auch in der Lage, die Schadstoffbelastung eines Gebiets durch eine entsprechende Grafik darzustellen, in der z.B. hohe Belastungen in rot hervorgehoben sind, wobei ungefährliche Bereiche in grüner Farbe zu sehen sind. Eine reine Zahlendarstellung wäre hierbei nicht sehr anschaulich.

Als weiteres Beispiel gewinnt der Satz „Die durchschnittliche Arbeitslosenquote der Bundesrepublik liegt bei 19%.“ an Bedeutung, wenn in einer Karte die Verteilung dieses Durchschnitts veranschaulicht werden kann. Ganz ähnlich verhält es sich mit der Kriminalitätsrate oder den unterschiedlichen Umweltbelastungen.

Für Biologen und Umweltschützer ist das Vorkommen gefährdeter Tierarten interessant. Mittels einer Karte, die das Vorkommen solcher Tiere beschreibt, sowie weiteren Karten, welche bestimmte Umweltbedingungen darstellen, lassen sich Rückschlüsse ziehen, in welchen Gebieten das Auswildern solcher Tiere den größten Erfolg haben könnte. Auch die Ursachen des Rückgangs der Besiedlung durch bestimmte Tierarten könnte durch Auswertung solcher Karten ermittelt werden.

Investoren könnten durch Auswertung von Karten der Arbeitslosenquote, Klimabedingungen und der Infrastruktur Rückschlüsse auf geeignete Standorte ziehen.

Bei Bebauungsplänen kann man die unterschiedliche Beschaffenheit des Bodens einbeziehen. Durch solche Untersuchungen wären die Kosten besser abzuschätzen.

1.3 Ziel und Aufbau der Arbeit

Ein Hauptziel dieser Arbeit ist die Implementierung eines Modells, welches mit unscharfen Objekten umgehen kann. Zu diesem Ziel gelangt man in mehreren Schritten:

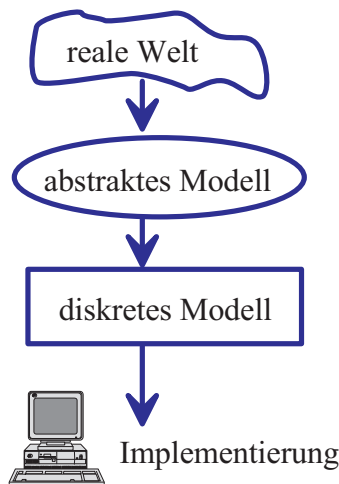


Abb. 2 Von der realen Welt zur Implementierung

Zunächst muß überhaupt ein Modell entworfen werden, welches unscharfe und scharfe geometrische Objekte der realen Welt darstellen und verarbeiten kann. Dieses abstrakte Modell befindet sich auf einer rein mathematischen Ebene, d.h. die Darstellbarkeit auf Computern wird vernachlässigt. Hier werden die formalen Grundlagen geschaffen, wie unscharfe Objekte definiert und durch Operatoren manipuliert werden können. Zunächst werden die scharfen Grundmengen von unscharfen Punkten, Linien und Regionen definiert. Als Grundmenge (oder Basis) eines Objekts werden Punktmenge bezeichnet, die zu diesem Objekt gehören. Anschließend werden diese Punktmenge derart erweitert, daß es möglich wird, die Unschärfe real existierender Objekte zu repräsentieren. Es wird untersucht, welche Operationen auf den jeweiligen Objekten benötigt werden, um bestimmte Aufgaben lösen zu können. Dann erfolgt eine formale Definition dieser Operationen. Danach beschäftigt sich diese Arbeit mit der Lage zweier Objekte im \mathbb{R}^2 zueinander. Dazu werden zunächst die topologischen Beziehungen der Grundmengen betrachtet. Das ist notwendig, weil die Grundmengen der entworfenen Objekte sehr komplex sein können und frühere Untersuchungen entweder von einfacheren Objekten ausgehen oder nicht alle Objekttypen abdecken (vgl. Abschnitt 2.2). Schließlich müssen noch die topologischen Beziehungen zwischen unscharfen Objekten untersucht werden.

In einem zweiten Schritt wird aus dem abstrakten Modell ein mögliches diskretes Modell abgeleitet. Dies erklärt auch den Zwischenschritt über das abstrakte Modell. Es ist nämlich durchaus möglich, weitere diskrete Modelle abzuleiten. Das diskrete Modell befindet sich ebenfalls noch auf einer mathematischen Ebene, berücksichtigt jedoch die begrenzte Darstellbarkeit von Objekten innerhalb von Computern. Ausgelassen werden hierbei noch Fragestellungen von Datenstrukturen und effizienten Algorithmen. Es werden hier zuerst Objektrepräsentationen angegeben, die eine endliche Darstellbarkeit gewährleisten. Dann werden die Operationen des abstrakten Modells so definiert, daß sie auf diesen Repräsentationen arbeiten. Hierbei werden ebenfalls die topologischen Beziehungen berücksichtigt. Beim Entwurf wurde besonders auf numerische Robustheit geachtet. So sollten sich z.B. die topologischen Beziehungen zwischen zwei Objekten nicht durch Rundungsfehler (wie sie zwangsläufig durch die endliche Darstellung von Zahlen in Computern entstehen) ändern können.

Im dritten Schritt, der Implementierung, werden Datenstrukturen und Algorithmen angegeben, welche einen effizienten Umgang mit unscharfen Objekten ermöglichen. Wichtiges Thema hierbei sind Laufzeituntersuchungen der angegebenen Algorithmen.

Mit graphischen Darstellungsmöglichkeiten beschäftigt sich das Kapitel der Visualisierung. Zuerst werden hier die Möglichkeiten betrachtet, unscharfe Objekte zu zeichnen. Anschließend wird gezeigt, wie man solche Bilder aus den Objekten berechnen kann.

Wie unscharfe Objekte in Datenbanken gespeichert und verarbeitet werden können, ist Thema des darauffolgenden Kapitels. Hier werden Anfragen an ein erweiterbares Datenbanksystem angegeben, die bestimmte Aufgabenstellungen erfüllen.

Kapitel 2

Frühere Arbeiten und Einordnung der eigenen Arbeit

2.1 Unscharfe Objekte

Es gibt bereits einige Ansätze, unscharfe Objekte darzustellen. Ein sehr einfacher (und daher leicht zu realisierender Ansatz) ist es, die minimale und maximale Ausdehnung der unscharfen Region anzugeben. Das minimale Ausmaß bildet den sogenannten Kern der Region, wogegen die Differenz der maximalen und minimalen Ausdehnung als breite Grenze betrachtet wird. Dies führt zum Konzept der vagen Regionen, wie es in [18] beschrieben ist. Auch in [5] wird die eigentliche Region durch zwei scharfe Regionen dargestellt, um Ungewissheit im Grenzverlauf beschreiben zu können. Außerdem werden dort die topologischen Beziehungen solcher unscharfen Objekte untersucht. Ein weiterer Ansatz, eine Region mit unbestimmter Grenze durch zwei Regionen mit scharfen Grenzen darzustellen ist die „Egg-Yolk-Repräsentation“ [7],[37],[38],[1]. Der allgemeinste dieser drei Ansätze ist in [18] beschrieben. In den anderen Fällen werden Einschränkungen gemacht, wie die beiden Teile einer Region zueinander liegen können. Dieser Ansatz soll kurz erläutert werden. Eine vage Region wird als Tupel (K, G) von zwei disjunkten scharfen Regionen dargestellt. K stellt den Kern der Region dar und G die breite Grenze. Diese scharfen Regionen müssen unter Operatoren wie Vereinigung, Schnitt und Differenz abgeschlossen sein, damit es keine Probleme beim Abschluß der vagen Regionen unter diesen Operatoren gibt. So läßt sich eine Region „Ölvorkommen“ durch eine vage Region darstellen. Der Kern der Region stellt dabei die Teile dar, an denen viel Öl vorhanden ist, im Bereich der Grenze ist hingegen weniger Öl zu finden. Man kann feststellen, daß diese beiden Teile in der Realität nicht verbunden sein müssen. Dies wird jedoch in den anderen beiden oben genannten Ansätzen gefordert, wodurch ihre Anwendbarkeit eingeschränkt wird. Die Operationen auf vagen Regionen werden auf Operationen ihrer Komponenten zurückgeführt. Z.B. wird die Vereinigung zweier vagen Regionen in [18] wie folgt dargestellt:

$$\text{union}((K_1, G_1), (K_2, G_2)) \stackrel{\text{def}}{=} ((\text{union}(K_1, K_2), \text{difference}(\text{union}(G_1, G_2), \text{union}(K_1, K_2))))$$

Man beachte, daß auf der rechten Seite nur Operatoren auf herkömmlichen scharfen Objekten verwendet werden. Es ist ebenfalls zu erkennen, daß solche Modelle unabhängig von einer speziellen Darstellung der scharfen Komponenten sind. Es spielt also keine Rolle, ob zur Repräsentation der beiden Objektteile eine Vektor- oder Rasterdarstellung genommen wird. Solche Ansätze lassen sich relativ leicht auf einem Computer realisieren. Hat man bereits eine Darstellungsmöglichkeit für herkömmliche scharfe Objekte, genügt ein Tupel, um solche unscharfen Objekte darzustellen. Da alle Operatoren in diesen Modellen auf Operatoren von herkömmlichen Objekten beruhen, stellt auch deren Implementierung kein Problem dar.

Der größte Vorteil solcher Modelle, nämlich ihre Einfachheit, ist zugleich ihr größter Nachteil. So lassen sich viele Problemstellungen nur unbefriedigend lösen. Z.B. ist eine kontinuierliche Ände-

zung von „voll zur Region gehörend“ zu „nicht zur Region gehörend“ mit diesen Ansätzen nicht darstellbar. Solche kontinuierlichen Übergänge sind jedoch häufig zu beobachten. Als Beispiel seien Regionen genannt, die Gebiete mit Luftverschmutzung darstellen. Im Modell, welches hier entwickelt wird, sind stetige Übergänge problemlos darstellbar.

Ein weiterer Nachteil ist der, daß sich diese Ansätze nur auf Regionen beziehen. So ist eine Darstellung unscharfer Linien und Punkte nicht in diesen Ansätzen enthalten. Das in dieser Arbeit entwickelte Modell schließt dagegen alle drei Objekttypen ein.

Solch ein Ansatz wird ebenfalls in [29] verfolgt. Als formale Grundlage zur Darstellung der scharfen Teile dient hier die ROSE(RObust Spatial Extension)-Algebra [28]. Aufgrund der numerischen Robustheit dieses Modells ergeben sich ähnliche Eigenschaften für die daraus entwickelten vagen Regionen. Auch in der vorliegenden Arbeit wird ein numerisch robustes Modell zur Darstellung der scharfen Grundmenge unscharfer Objekte verwendet, um topologische Anomalien zu vermeiden.

Eine Erweiterung dieses Ansatzes sind die sogenannten α -Level, wie sie in [43],[41],[42] beschrieben sind. Hier wird die Region partitioniert und jeder Partition wird ein α aus einer endlichen Menge von Werten zugewiesen. In [30] wird gezeigt, welche Probleme bei der Umwandlung von Vektor-Daten in eine Gitterdarstellung auftreten können, wenn die Anzahl der Gitterpunkte begrenzt ist. Es wird z.B. auf das Problem eingegangen, daß Linien trotz Entfernen innerer Punkte im Gittermodell verbunden bleiben können, ein Phänomen, welches es in der Vektordarstellung nicht gibt. Auch werden Beispiele angegeben, bei denen Regionen mit unterschiedlichen Grenzen die gleichen Gitterpunkte abdecken. Als ein Ergebnis dieser Betrachtungen kann man festhalten, daß die enthaltenen Gitterpunkte eines Objekts in einer Gitterdarstellung allein nicht genügen, um dieses Objekt eindeutig darzustellen. Weiterhin wird dort gezeigt, wie solchen Problemen durch die α -Level-Regionen begegnet werden kann. Zudem werden Operationen auf solchen Objekten eingeführt und ihre topologischen Beziehungen betrachtet. Da solche α -Level-Regionen jedoch nur aus einer begrenzten Anzahl von Partitionen bestehen und jeder dieser Partitionen ein fester Wert zugewiesen wird, treten zwischen den einzelnen Teilen stets Sprünge auf, die in der Realität nicht notwendig vorkommen müssen. Die Darstellung einer kontinuierlichen Änderung der Zugehörigkeit kann somit nur approximiert werden. Eine gute Annäherung an eine solche Abnahme der Zugehörigkeit erfordert eine sehr feine Partitionierung der zugrundeliegenden Region und daher entsteht bei der Realisierung auf einem Computer ein sehr großes Datenvolumen. Man kann feststellen, daß das hier entwickelte Modell auch in der Lage ist, solche α -Level-Regionen darzustellen. Da in dieser Arbeit jedoch eine lineare statt einer konstanten Approximation der Zugehörigkeit verwendet wird, können auch stetige Übergänge modelliert werden.

Andere Ansätze verwenden fuzzy-Mengen, um solche Regionen darzustellen. Hier wird jedem Punkt innerhalb einer scharfen Grenze ein Wert zugeordnet, der aussagt, mit welchem Grad dieser Punkt zur Region gehört. Dieser Zugehörigkeitsgrad stammt aus dem Intervall $[0,1]$, es sind jedoch auch beliebige andere Skalierungen denkbar.

So werden in [8] unscharfe Regionen definiert. Eine Region wird als Menge von Koordinatenpaaren mit je einem Zugehörigkeitsgrad dargestellt. Da die Koordinaten aus dem \mathbb{N}^2 stammen, handelt es sich um eine Rasterrepräsentation solcher Regionen. Die Definition ist jedoch recht unpräzise. So ist z.B. nicht zu erkennen, ob mit einer solchen Koordinate eine ganze Fläche des \mathbb{R}^2 gemeint ist. Wie aus darauffolgenden Abbildungen hervorgeht, ist dies aber offensichtlich gemeint. Es können hier, wie bei α -Level-Regionen, keine kontinuierlichen Übergänge dargestellt werden. Zwischen solchen Regionen werden noch einige Beziehungen wie Abstand und Richtung definiert, deren Anwendung jedoch durch die unpräzise Objektdefinition fragwürdig ist. Auf unscharfe Linien oder Punkte wird in dieser Arbeit nicht eingegangen. Hier wird im Gegensatz dazu versucht, eine genaue Definition der drei räumlichen Objekttypen anzugeben. Im diskreten Modell wird daher neben der Darstellung solcher Objekte auch angegeben, welche Punktmengen des \mathbb{R}^2 zu ihnen gehören.

In [31] und [32] wird der \mathbb{R}^2 mittels eines planaren Graphen partitioniert. Die hieraus entstehen-

den Partitionen werden als Zellkomplexe interpretiert, aus denen räumliche Objekte aufgebaut werden können. Ein (scharfes) räumliches Objekt wird hier durch eine Funktion beschrieben, die angibt, ob ein bestimmter Zellkomplex Teil dieses Objekts ist oder nicht. Die Erweiterung auf unscharfe Objekte entspricht der Erweiterung der Funktion. Kann diese nicht nur zu 0 und 1 ausgewertet werden, sondern auch Werte dazwischen liefern, entsteht ein unscharfes Objekt. Jedoch treten auch hier zwischen den einzelnen Zellkomplexen stets Sprünge auf. Ebenfalls fehlt eine Betrachtung unscharfer Linien und Punkte.

Auch in [4] wird auf das Problem unscharfer Objekte eingegangen. Es wird auf die Notwendigkeit hingewiesen, für nicht-exakte Objekte entsprechende Modelle zu verwenden. Weiterhin wird auf Unterschiede zwischen scharfen und unscharfen Mengen eingegangen. Jedoch fehlt eine Definition unscharfer räumlicher Objekte. Auch auf die Problematik der Anwendung von Operatoren (und deren Realisierung) wird im Gegensatz zur vorliegenden Arbeit nicht eingegangen.

2.2 Topologische Beziehungen

Die Topologischen Beziehungen zwischen den einzelnen Objekten, also ihre gegenseitige Lage im \mathbb{R}^2 stellt eine wichtige Fragestellung bei geometrischen Objekten dar. Wichtig ist dies z.B. bei Fragen wie:

- Ist Grundstück A durch die Verschmutzung von Kraftwerk B beeinflusst?
- Welche Grundstücke sind durch einen Bebauungsplan betroffen?
- Gibt es ein Grundstück, welches die Grundstücke A und B verbindet?
- Welche Grundstücke liegen innerhalb des Landkreises C?

Es gibt eine Reihe von Modellen, um topologische Beziehungen zwischen scharfen Objekten darzustellen. Einige Modelle wurden jedoch bisher nur auf einfachen Objekten angewendet. Andere Modelle können zwar die topologischen Beziehungen zwischen komplexen Regionen beschreiben (RCC-8 [35, 36]), sind jedoch nicht fähig, Beziehungen zwischen Linien oder Punkten zu beschreiben.

Um topologische Beziehungen genauer beschreiben zu können, werden die zu untersuchenden Objekte meist in ihre Bestandteile „Inneres“ und „Grenze“ zerlegt, wobei die aus der Mathematik bekannten Definitionen verwendet werden (vgl. auch Abschnitt 3.1). In [13] demonstriert M.Egenhofer, wie durch den Test auf Leere der Schnitte von Innerem und Grenze der beiden Objekte (daher der Name 4-Intersection-Modell) ihre relative Lage beschrieben werden kann. Die Ergebnisse dieser Tests werden üblicherweise in einer Matrix eingetragen, welche dann die Beziehung beschreibt. In [14] wird dieses Modell verwendet, um topologische Beziehungen zwischen Regionen mit Löchern zu beschreiben. Dazu werden die einzelnen topologischen Beziehungen der „Komponenten“, d.h. Regionsteile und Löcher der Regionen, in einer Matrix zusammengefaßt, um die Gesamtrelation darzustellen. Hierbei hat man das Problem, daß die Größe der Matrix und somit auch die Anzahl möglicher topologischer Beziehungen von der Anzahl der Komponenten der betrachteten Regionen abhängt. In dieser Arbeit wird gezeigt, daß topologische Beziehungen zwischen komplexen Objekten auch ohne solche Abhängigkeiten dargestellt werden können.

Eine Erweiterung dieses Modells ist die DEM (dimension extended method [17]), bei der ebenfalls die Schnitte von Grenze und Innerem der Objekte betrachtet werden. Hier werden diese jedoch nicht nur darauf getestet, ob sie leer sind oder nicht, sondern es wird zusätzlich die Dimension des Schnittes untersucht.

Durch diese beiden Modelle lassen sich die topologischen Beziehungen zwischen zwei einfachen Regionen recht gut beschreiben. Leider ist dies bei zwei einfachen Linien (verbunden, genau zwei Endpunkte) nicht mehr so. Natürlich lassen sich die topol. Beziehungen zweier Linien auch durch

dieses Modell beschreiben, jedoch ist die entstehende Clusterung dabei sehr grob. So werden z.B. Beziehungen der Linien des folgenden Bildes alle durch die gleiche Matrix beschrieben:

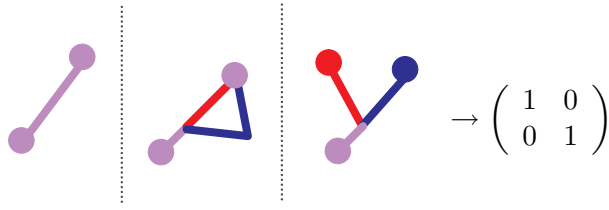


Abb. 3 Linien mit gleicher 4-intersection-Matrix

Da in dieser Arbeit jedoch ein einheitliches Modell für alle Objekttypen verwendet werden soll, sind beide Modelle für diesen Zweck ungeeignet.

In [34] wird das 4-Intersection-Modell um die Dimension der Schnitte sowie um die Anzahl von Berührungspunkten erweitert. Eine weitere Verfeinerung ist [21] zu finden. Hier werden neben der Leere von Schnitten auch Dimension und „Art“ (berühren, kreuzen) betrachtet. Diese Erweiterungen erlauben zwar eine detailliertere Beschreibung der topologischen Beziehung, lösen jedoch nicht das eben angesprochene Problem.

Im RCC-Modell [38] können Beziehungen zwischen Linien und Punkten nicht betrachtet werden, da dieses Modell von 5 bzw. 8 Basisrelationen zwischen einfachen Regionen ausgeht, um Relationen zwischen komplexeren Regionen zu beschreiben.

Ein weiterer Ansatz ist die Calculus Based Method (CBM) [19]. Wie im RCC-Modell werden mehrere Basisrelationen beschrieben. In diesem Modell gibt es jedoch Basisrelationen für Linien, Punkte und Regionen, wobei nicht jede Relation für alle Typen definiert ist. Z.B. ist die *overlap*-Beziehung nur für Linien und Regionen definiert. Desweiteren verfügt dieses Modell über einen *boundary*-Operator, welcher für Linien und Regionen definiert ist. Durch logische Verknüpfungen der Basisrelationen und des *boundary*-Operators lassen sich sehr viele Relationen unterscheiden. In [11] wurde dieses Modell verwendet, um Beziehungen zwischen nicht notwendig zusammenhängenden Regionen zu beschreiben. Weiterhin wurde dieses Modell in [16] angewendet, um die topologischen Beziehungen zwischen komplexen Regionen, Linien und Punkten anzugeben.

Das durch Egenhofer eingeführte 9-Intersection-Modell [23], beschreibt die Beziehung zwischen zwei Objekten durch Untersuchung der Schnitte von Innerem, Grenze und Äußerem des Objekts. Es handelt sich dabei um eine Erweiterung des 4-Intersection-Modells. Diese Schnitte werden auf Leere getestet und die Ergebnisse in einer Matrix festgehalten.

$$T(A, B) = \begin{pmatrix} A^\circ \cap B^\circ \neq \emptyset & A^\circ \cap \partial B \neq \emptyset & A^\circ \cap B^{-1} \neq \emptyset \\ \partial A \cap B^\circ \neq \emptyset & \partial A \cap \partial B \neq \emptyset & \partial A \cap B^{-1} \neq \emptyset \\ A^{-1} \cap B^\circ \neq \emptyset & A^{-1} \cap \partial B \neq \emptyset & A^{-1} \cap B^{-1} \neq \emptyset \end{pmatrix}$$

Abb. 4 Die 9-Intersection-Matrix

Durch die Hinzunahme des Äußeren zum 4-intersection-Modell lassen sich Beziehungen der Linien der Abb. 3 unterscheiden. Dieses Modell wurde von Elisio Clementini und Paolo Di Felice angewendet, um die Beziehungen von Regionen mit breiten Grenzen zu beschreiben [5],[6]. In [12] untersuchen Egenhofer und Sharma die realisierbaren Matrizen von Regionen in Rasterrepräsentation. Weitere Untersuchungen zu diesem Modell innerhalb von Rastern sind in [20] zu finden. In der vorliegenden Arbeit wird untersucht, wie dieses Modell benutzt werden kann, um die topologischen Beziehungen zwischen komplexen scharfen geometrischen Objekten darzustellen. Eine Matrix zur Beschreibung einer topologischen Beziehung ist jedoch für Datenbankanfragen ungeeignet. Daher müßte man den realisierbaren Matrizen sinnvolle Namen zuordnen. Hierbei gibt es das Problem, daß die Anzahl dieser Matrizen sehr groß ist. Zum Beispiel lassen sich zwischen zwei komplexen Linien über 80 Matrizen realisieren und deshalb ebensoviele topologische

Beziehungen unterscheiden. Würde man jeder dieser Matrizen einen Namen geben, wäre der Benutzer eines Datenbanksystems gezwungen, alle diese Namen und ihre Bedeutung zu kennen. Um dieses Problem zu lösen, wird in dieser Arbeit eine sinnvolle Clusterung dieser Matrizen herausgearbeitet. Dabei werden ähnliche topologische Beziehungen in einem Cluster zusammengefaßt. In [15] wird untersucht, wie Berechnungen topologischer Beziehungen basierend auf dem 9-Intersection-Modell beschleunigt werden können. Es wird herausgestellt, daß durch eine bestimmte Reihenfolge der Berechnung (aufgrund geometrischer Bedingungen) nicht mehr alle Schnitte ausgewertet werden müssen. Durch die in dieser Arbeit entwickelte Clusterung muß auch nur ein Teil der möglichen Einträge berechnet werden, um zwei Objekte eine topologische Beziehung zuzuordnen.

Durch eine Erweiterung des 9-Intersection-Modells, die in der vorliegenden Arbeit entwickelt wird, ist es möglich, auch die Beziehungen zwischen zwei komplexen unscharfen Objekten zu beschreiben. Da die scharfen Objekte eine Teilmenge dieser Objekte bilden, ist klar, daß noch weitere Beziehungen unterschieden werden können. Um die riesige Anzahl der dadurch entstehenden Matrizen in den Griff zu bekommen, wird eine mögliche Clusterung herausgearbeitet.

Kapitel 3

Abstraktes geometrisches Datenmodell

3.1 Definition scharfer geometrischer Objekte mit Hilfe von Punkt- mengen und Punktmengentopologie

Die Grundlage des hier vorgestellten abstrakten Modells bildet das sogenannte Punktmengenmodell. Dieses dient zur Beschreibung der scharfen Abgrenzung der geometrischen Objekte. Die Idee des Punktmengenmodells ist recht einfach. Das gesamte Universum wird als eine (zweidimensionale) unendliche Punktmenge angesehen. Alle Objekte, die sich in diesem Universum befinden, werden als Teilmengen dieser Punktmenge beschrieben. Da es sich um gewöhnliche Mengen handelt, können auch die üblichen Mengenoperatoren (Vereinigung, Schnitt und Differenz) auf ihnen ausgeführt werden. Beziehungen zwischen zwei Objekten können ebenfalls auf Mengenbeziehungen zurückgeführt werden, z.B.:

$$\begin{aligned} \text{equal}(A, B) &\Leftrightarrow \text{points}(A) = \text{points}(B) \\ \text{inside}(A, B) &\Leftrightarrow \text{points}(A) \subset \text{points}(B) \\ \text{disjoint}(A, B) &\Leftrightarrow \text{points}(A) \cap \text{points}(B) = \emptyset \\ \text{overlap}(A, B) &\Leftrightarrow \text{points}(A) \cap \text{points}(B) \neq \emptyset \wedge \\ &\quad \text{points}(A) - \text{points}(B) \neq \emptyset \wedge \\ &\quad \text{points}(B) - \text{points}(B) \neq \emptyset \end{aligned}$$

usw.

Wobei mit $\text{points}(A)$ die Punktmenge des Objekts A bezeichnet wurde. Leider lassen sich hierdurch nur recht grobe Beziehungen unterscheiden. So kann nicht unterschieden werden, ob sich zwei Regionen nur berühren, oder ob sie gemeinsame Bereiche besitzen. Um dies zu ermöglichen, werden die Objekte in zwei Bestandteile zerlegt. Ein Objekt A wird dabei in Grenze (∂A) und Inneres (A°) unterteilt. Alle Punkte des Universums, die nicht in einem dieser Teile enthalten sind, werden dem Äußeren von A (A^{-1}) zugeordnet.

Da allgemeine Punktmengen sehr komplexe Formen annehmen können, z.B. nicht begrenzt sein müssen, ist es für einen sinnvollen Umgang mit den geometrischen Objekten notwendig, diese einzuschränken. So besteht z.B. eine Linie nur aus eindimensionalen Komponenten. D.h. einer Linie können weder einzelne Punkte fehlen, noch kann sie zusätzliche Punkte enthalten.

Als Punkte werden im hier entwickelten Modell endliche Mengen von Koordinaten angesehen:

Point $\stackrel{\text{def}}{=} \{P \subset \mathbb{R}^2 : P \text{ ist endlich}\}$

Dies entspricht nicht der klassischen Darstellung von Punkten, bei denen ein Punkt nur aus einer einzelnen Koordinate besteht. Diese Erweiterung der Punktdefinition hat einige Vorteile. So können z.B. alle Schnittpunkte zweier Linien durch einen einzigen Punkt beschrieben werden.

Auch lassen sich alle Endkoordinaten einer Linie durch einen solchen Punkt beschreiben. Man beachte, daß hierbei auch Kreise eingeschlossen sind, da auch eine leere Punktmenge erlaubt ist. Weiterhin hat diese Darstellung den Vorteil, daß Punkte unter Operationen wie Vereinigung und Schnitt abgeschlossen sind, wogegen bei der klassischen Punktdefinition solche Operatoren nicht sinnvoll definierbar sind.

Zum Inneren eines Punktes gehören alle seine Koordinaten, und seine Grenze ist leer. D.h. Für alle $P \in \mathbf{Point}$ gilt :

$$P^\circ = P$$

$$\partial P = \emptyset$$

Linien können sehr komplex sein. Sie können Selbstschnitte besitzen, Kreise bilden und müssen nicht zusammenhängend sein. Um ein solch komplexes Gebilde beschreiben zu können, werden zunächst zusammenhängende Teile von Linien definiert.

curve $\stackrel{\text{def}}{=} \{f(c) \subset \mathbb{R}^2 : f \text{ erfüllt untere Bedingungen}\}$

(Zur Darstellung von Kurven als Funktion vgl. [25].)

Die Bedingungen, die an f zu stellen sind, sind die folgenden:

Bed. 1 $f : [a, b] \rightarrow \mathbb{R}^2$ mit $a < b$

D.h. f ist eine Abbildung eines Intervalls in die euklidische Ebene. Die Forderung $a < b$ (zusammen mit Bed. 3) sichert dabei, daß der Definitions- und damit der Wertebereich nicht nur aus einem einzigen Punkt besteht.

Bed. 2 f ist stetig

Bed. 3 $\forall (x, y) \in \mathbb{R}^2 : |\{c \in [a, b] : f(c) = (x, y)\}| < \infty$

Damit besitzt die Kurve nur endlich viele Selbstschnitte.

Bed. 4 $f(c)$ ist rektifizierbar. Das heißt, daß die Kurve eine endliche (meßbare) Länge besitzt, die später noch definiert wird.

Von einer solchen Kurve läßt sich die Grenze (Endpunkte) berechnen. Die Kurve kann höchstens zwei Endpunkte, nämlich $f(a)$ und $f(b)$ besitzen. Andere Endpunkte kann die Kurve nicht enthalten, da f als stetig vorausgesetzt war. Gibt es einen Wert in (a, b) , der einem dieser Kandidaten entspricht, so handelt es sich nicht mehr um einen Endpunkt der Linie. Gilt $f(a) = f(b)$, so bildet die Kurve einen Kreis und besitzt keinen Endpunkt. Formal ergibt sich:

$$\partial K \stackrel{\text{def}}{=} \begin{cases} \{f(a), f(b)\} - \{f(c) : c \in (a, b)\} & \text{wenn } f(a) \neq f(b) \\ \emptyset & \text{sonst} \end{cases}$$

Das Innere einer solchen Kurve ergibt sich nun durch:

$$K^\circ \stackrel{\text{def}}{=} \{f(c) : c \in [a, b]\} - \partial K$$

Einige Beispiele für Kurven sind :

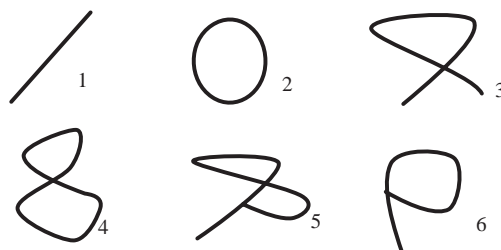


Abb. 5 Beispiele gültiger Kurven

Die Kurven haben die folgenden Eigenschaften:

- 1 Einfache Kurve mit 2 Punkten in der Grenze
- 2 Kurve mit gleichem End- und Anfangspunkt (keine Grenze)
- 3 Kurve mit Selbstschnitt (2 Grenzpunkte)
- 4 geschlossene Kurve mit Selbstschnitt (keine Grenze)
- 5 sich selbst berührende und schneidende Kurve (1 Grenzpunkt)
- 6 sich selbst berührende Kurve (1 Grenzpunkt)

Die Länge einer solchen Kurve läßt sich über ihre Bogenlänge bestimmen. Dazu wird in die Kurve ein Polygonzug einbeschrieben, von dem die Länge bestimmt werden kann.

Sei t_0, \dots, t_n eine Zerlegung des Intervalls $[a, b]$, so läßt sich die Bogenlänge der Kurve K wie folgt berechnen:

$$BLen_n(K) = \sum_{i=0}^{n-1} \sqrt{(f(t_i).x - f(t_{i+1}).x)^2 + (f(t_i).y - f(t_{i+1}).y)^2}$$

Dabei bezeichne $f(t).x$ die x-Komponente von $f(t)$ ($f(t).y$ analog). Die tatsächliche Länge ist dann durch $\lim_{n \rightarrow \infty} BLen_n(K)$ gegeben. Sie ist stets endlich, da eine solche Kurve als rektifizierbar vorausgesetzt war.

Eine Linie wird nun über Kurven definiert.

$$\mathbf{Line} \stackrel{\text{def}}{=} \left\{ \bigcup_{i=0}^{n < \infty} L_i \mid L_i \text{ ist eine Kurve und für } i \neq j \text{ gilt : } L_i \cap L_j \text{ ist endlich} \right\}$$

Im Folgenden soll gezeigt werden, daß die Endlichkeit der Schnitte die Menge der möglichen Linien nicht einschränkt. Man betrachte zwei Kurven mit einer unendlichen gemeinsamen Punktmenge. Das Ergebnis kann auch auf mehr als zwei Linien erweitert werden. Da beide Kurven über die endlichen Intervalle $[a_1, b_1]$ und $[a_2, b_2]$ definiert und stetig sind, können sie nur dann eine unendliche gemeinsame Punktmenge besitzen, wenn sie ein zusammenhängendes Stück gemeinsam haben. Unendlich viele Schnittpunkte ohne ein gemeinsames Teilstück kann es nicht geben. Man betrachte dazu die Folge der Schnittpunkte, die entlang einer der beiden Linien auftreten. Da die Linie stetig ist, sind zwei benachbarte Elemente dieser Folge durch ein Liniensegment verbunden. Somit müßte die Linie selbst aber eine unendliche Länge besitzen *widerspruch*. Formal gilt also:

$$\exists [a_1^*, b_1^*] \subseteq [a_1, b_1], [a_2^*, b_2^*] \subseteq [a_2, b_2] : f_1(a_1^* + \delta(b_1^* - a_1^*)) = f_2(a_2^* + \delta(b_2^* - a_2^*)); \delta \in [0, 1]$$

Falls mehrere solcher Stücke existieren, so können diese analog zum nun beschriebenen Verfahren eliminiert werden. Die gleiche Linie ohne diesen Abschnitt läßt sich wie folgt berechnen. Suche in beiden Kurven das Teilintervall, welches auf die gemeinsame Punktmenge abgebildet wird, also $[a_1^*, b_1^*]$ und $[a_2^*, b_2^*]$. Dann lassen sich mehrere Fälle unterscheiden.

Fall 1:

Im einfachsten Fall ergibt sich das gesamte Intervall einer Kurve, z.B. $a_1^* = a_1$ und $b_1^* = b_1$. Dann kann diese Kurve aus der Menge aller Kurven der Linie entfernt werden, ohne daß sich die Punktmenge ändert.

Fall 2:

Das Teilintervall einer Kurve beginnt mit a oder endet mit b. Dann wird diese Kurve durch entsprechende Änderung von $[a, b]$ verkürzt. Es ist klar, daß eine solche Änderung des Intervalls

die Kurveigenschaft nicht verändert. Die Punktmenge der Gesamtlinie ändert sich nicht, da die ‚abgeschnittenen‘ Punkte durch die andere Kurve geliefert werden.

Fall 3:

Das jeweils ermittelte Teilintervall liegt im Inneren von $[a_1, b_1]$ bzw. $[a_2, b_2]$. Dann kann eine der Kurven (hier z.B. K_1) in drei Kurven zerlegt werden, indem das zugrundeliegende Intervall aufgeteilt wird. $[a_1, b_1]$ läßt sich aufgliedern in $[a_1, a_1^*], [a_1^*, b_1^*]$ und $[b_1^*, b_1]$. Das Entfernen der Originalkurve und das Einfügen der Kurven, die durch die Funktion des Originals und der Intervalle $[a_1, a_1^*]$ und $[b_1^*, b_1]$ entstehen, löst das Problem.

Die Länge der Linie ergibt sich aus der Summe der Längen der einzelnen Kurven.

$$\text{len}\left(\bigcup K_i\right) = \sum_i \text{len}(K_i)$$

Als Grenzpunkte einer solchen Linie kommen nur die Grenzpunkte der enthaltenen Kurven in Betracht. Diese sind jedoch ein innerer Punkt der Linie, wenn eine andere Kurve diesen Punkt ebenfalls enthält.

$$\text{Für } L = \bigcup K_k \text{ sei } G(K_i) = \partial K_i - \bigcup_{j \neq i} K_j$$

Die Menge aller Grenzpunkte von K_i , die nicht in einer der anderen Kurven enthalten sind. Dann gilt:

$$\partial(L) = \bigcup G(K_i)$$

Das Innere wird (wie bei Kurven) durch die restlichen Punkte gebildet:

$$L^\circ = \bigcup K_i - \partial L$$

Nachdem nun Punkte und Linien dargestellt werden können, fehlt noch eine Definition für flächige Gebilde. Um diese zu definieren, wird das Konzept der regulären abgeschlossenen Punktmen- gen verwendet.

Region $\stackrel{\text{def}}{=} \{R \subset \mathbb{R}^2 : R \text{ ist regulär abgeschlossen und begrenzt} \wedge \partial R \text{ ist rektifizierbar}\}$

Zunächst einige Definitionen (vgl.[25] S. 9.18ff):

Randpunkt einer Menge $A \subseteq \mathbb{R}^2$:

$(x, y) \in \mathbb{R}^2$ ist Randpunkt von $A \Leftrightarrow$ In jeder Umgebung von (x, y) sind sowohl Elemente von A wie auch von $\mathbb{R}^2 - A$ enthalten.

Rand einer Menge $A \subseteq \mathbb{R}^2$:

$$\partial A = \{ (x, y) : (x, y) \text{ ist Randpunkt von } A \}$$

Man beachte, daß der Rand einer Region, der Grenze der Region entspricht, dies jedoch bei Punkten und Linien nicht der Fall ist.

Innerer Punkt einer Menge $A \subseteq \mathbb{R}^2$:

$(x, y) \in \mathbb{R}^2$ heißt innerer Punkt von $A \Leftrightarrow$ Es existiert eine Umgebung von (x, y) , die in A enthalten ist.

Inneres einer Menge $A \subseteq \mathbb{R}^2$:

$$A^\circ = \{ (x, y) : (x, y) \text{ ist innerer Punkt von } A \}$$

$A \subseteq \mathbb{R}^2$ ist offen $\Leftrightarrow \partial A \cap A = \emptyset$
 $A \subseteq \mathbb{R}^2$ ist abgeschlossen $\Leftrightarrow \partial A \subseteq A$

Der Abschluss einer Region R wird definiert als:

$$\overline{R} = R \cup \partial R$$

Eine abgeschlossene, reguläre Punktmenge S ist eine Punktmenge für die gilt :

$$S = \overline{S^\circ}$$

Ist eine Punktmenge nicht abgeschlossen und regulär, so kann ein Regularisierungsoperator reg verwendet werden.

$$reg : R \subset \mathbb{R}^2 \rightarrow \mathbf{Region}$$

$$reg(R) \rightarrow \overline{R^\circ}$$

Dieser Operator arbeitet folgendermaßen: Im ersten Schritt wird das Innere der übergebenen Punktmenge gebildet. Daher werden zusätzliche Punkte und Linien eliminiert, da diese keine inneren Punkte darstellen. Im zweiten Schritt wird die gesamte Grenze der Punktmenge hinzugefügt, sowie evtl. in der Menge selbst fehlende Punkte oder Linien ausgefüllt.

Die regulären Punktmenge haben den Nachteil, daß einige klassische Mengenoperatoren auf ihnen nicht abgeschlossen sind.

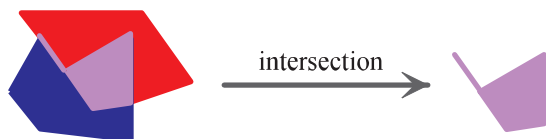


Abb. 6 Kein Abschluß des Schnitts auf regulären Mengen

Dadurch ist man gezwungen, neue Operationen zu definieren, um den Abschluß zu gewährleisten. Man kann feststellen, daß die Vereinigung von regulären abgeschlossenen Mengen wieder eine reguläre abgeschlossene Menge ergibt. Bei der Differenz ist es möglich, daß das Ergebnis nicht mehr abgeschlossen ist (betrachte z.B. die Differenz zweier Regionen mit einem gemeinsamen Grenzabschnitt). Beim Schnitt kann die Regularität verletzt werden (vgl. Abb. 6).

$$A \cap_{reg} B \rightarrow reg(A \cap B)$$

$$A -_{reg} B \rightarrow reg(A - B)$$

Es läßt sich leicht feststellen, daß die restlichen Eigenschaften von Regionen, d.h. Begrenztheit und Linieneigenschaft der Grenze durch die Mengenoperatoren nicht geändert werden.

3.2 Definition unscharfer geometrischer Objekte mittels Fuzzy-Mengen

Bei einer klassischen Menge kann zu jedem Element des zugrundeliegenden Universums bestimmt werden, ob es zu dieser Menge gehört oder nicht. In vielen Bereichen, ist eine solche strenge Zuordnung jedoch nicht ausreichend. So wird beispielsweise in der KI und der Linguistik

eine Darstellung benötigt, welche eine gestufte Zuordnung gestattet. Oft ist dies der Fall, wenn Mengen beschrieben werden sollen, welche natürlichsprachige Ausdrücke verwenden, wie z.B. die Menge aller jungen Erwachsenen (hier nur durch das Alter charakterisiert). Sowohl jung als auch erwachsen sind hier nicht eindeutig. Man kann mit Sicherheit sagen, daß ein 1-jähriger nicht zu dieser Menge gehört, da er nicht erwachsen ist. Ebenso würde man eine 80-jährige nicht mit dazuzählen, da das Prädikat „jung“ nicht erfüllt ist. Einen Menschen von 20 Jahren könnte man zu 100% zu dieser Menge dazurechnen. Schwieriger wird die Entscheidung bei 28-40 jährigen. Es lassen sich zahlreiche solcher Beispiele finden, so eben auch „das von Kraftwerk xy verschmutzte Gebiet“.

Um solche Mengen darzustellen, werden unscharfe Mengen verwendet. Hier wird jedem Element ein Zugehörigkeitsgrad (membership value) zugeordnet. Dieser Wert liegt im Intervall $[0, 1]$. Ist die Grundmenge endlich (wie z.B. bei der Menge junger Erwachsener), können die Werte durch Aufzählen zugeordnet werden. Bei einer unendlichen Menge ist dies natürlich nicht möglich. Dann wird eine Funktion verwendet, die jedem Element den Grad zuordnet. Selbstverständlich kann man diese Darstellung auch für endliche Mengen verwenden. Die Menge der jungen Erwachsenen könnte dann wie folgt dargestellt werden:

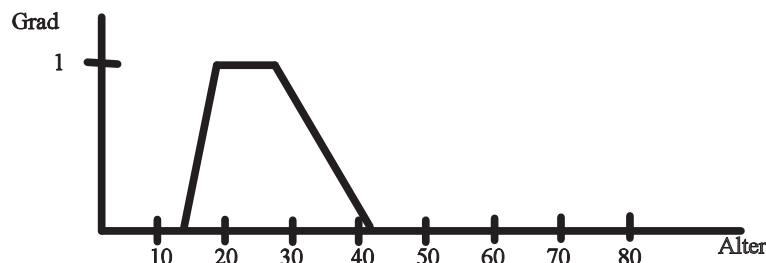


Abb. 7 Eine unscharfe Menge

3.2.1 Formale Grundlagen von unscharfen Mengen

Die folgenden Ausführungen sind in vielen Einführungen von unscharfen Mengen enthalten, z.B. [9]. Eine unscharfe Menge μ ist eine Abbildung des betrachteten Universums X in das Einheitsintervall $[0, 1]$.

$$\mu : X \rightarrow [0, 1]$$

Damit wird jedem Element des Universums ein Zugehörigkeitsgrad zugewiesen. Mit $F(X)$ werden gewöhnlich alle unscharfen Mengen über dem Universum X bezeichnet.

Klassische Operationen auf unscharfen Mengen

Wie auch für scharfe Mengen, gibt es für unscharfe Mengen einige Standardoperatoren, die im Folgenden kurz vorgestellt werden sollen.

Ein wichtiger Operator stellt einen Filter für unscharfe Mengen dar. Es handelt sich hierbei um den α -cut-Operator, der diejenigen Elemente des Universums bestimmt, deren Zugehörigkeitsgrad mindestens so groß wie α ist. Bei einer weiteren Version dieses Operators (*strong- α -cut*) muß der Grad größer als α sein. Formal ergibt sich:

$$\alpha\text{-cut} : F(X) \rightarrow M \subset X$$

$$\alpha\text{-cut}(\mu) \stackrel{\text{def}}{=} \{x \mid \mu(x) \geq \alpha\}$$

$$\text{strong-}\alpha\text{-cut} : F(X) \rightarrow M \subset X$$

$$\text{strong-}\alpha\text{-cut}(\mu) \stackrel{\text{def}}{=} \{x \mid \mu(x) > \alpha\}$$

Somit stellen diese beiden Operatoren eine Abbildung einer unscharfen Menge auf eine scharfe Menge dar.

Beispiel:

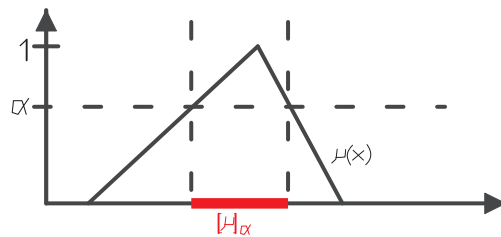


Abb. 8 Alpha-cut für unscharfe Mengen

Auch Mengenoperatoren sind für unscharfe Mengen definiert. Hierzu benötigt man die Definitionen einer t-Norm und einer t-Conorm (auch s-Norm genannt).

Definition Eine Funktion $T : [0, 1]^2 \rightarrow [0, 1]$ heißt t-Norm, wenn die folgenden Bedingungen gelten:

- (1) $T(a, 1) = a$ 1 ist neutrales Element
- (2) $a \leq b \Rightarrow T(a, c) \leq T(b, c)$ Monotonie
- (3) $T(a, b) = T(b, a)$ Symmetrie
- (4) $T(a, T(b, c)) = T(T(a, b), c)$ Assoziativität

Die meist verwendete t-Norm ist der *min*-Operator.

Definition Eine Funktion $S : [0, 1]^2 \rightarrow [0, 1]$ heißt s-Norm (oder t-Conorm), wenn folgende Eigenschaften erfüllt sind:

- (1) $T(a, 0) = a$ 0 ist neutrales Element
- (2) $a \geq b \Rightarrow T(a, c) \geq T(b, c)$ Monotonie
- (3) $T(a, b) = T(b, a)$ Symmetrie
- (4) $T(a, T(b, c)) = T(T(a, b), c)$ Assoziativität

Die meist verwendete s-Norm ist der *max*-Operator.

Mit diesen beiden Definitionen ausgerüstet, können nun einige Mengenoperatoren definiert werden:

$$(\mu \cap \mu')(x) \stackrel{\text{def}}{=} T(\mu(x), \mu'(x))$$

$$(\mu \cup \mu')(x) \stackrel{\text{def}}{=} S(\mu(x), \mu'(x))$$

$$\bar{\mu}(x) \stackrel{\text{def}}{=} 1 - \mu(x)$$

Hierbei muß S eine s-Norm und T eine t-Norm darstellen.

3.2.2 Eine Abgrenzung zur Wahrscheinlichkeitstheorie

Oft ist keine klare Abgrenzung zwischen den Ansätzen der Unschärfe und wahrscheinlichkeitstheoretischen Betrachtungen zu erkennen. Dies liegt daran, daß der Grad an Zugehörigkeit auch als Wahrscheinlichkeitsgrad interpretiert werden kann. Es gibt jedoch einen sehr wichtigen Unterschied zwischen Unschärfe und Wahrscheinlichkeit. Die Wahrscheinlichkeit drückt einen Mangel an Wissen aus. Hieraus folgt, daß die Wahrscheinlichkeit bei vollständigen Wissen stets den Wert 1 annimmt. Die Unschärfe dagegen ist dem betrachteten Objekt eigen und besteht auch bei vollständigen Wissen weiter. Ein einfaches Beispiel ist ein „dicht besiedeltes Gebiet“. Sind die Bevölkerungszahlen in diesem Gebiet nicht genau bekannt, sondern können nur geschätzt

werden, kann z.B. die folgende Aussage getroffen werden: „Das Gebiet ist mit einer Wahrscheinlichkeit von 70% ein mit dem Grad 0,5 dicht besiedeltes Gebiet“. Die Wahrscheinlichkeit ergibt sich aus der Genauigkeit der Schätzung. Der Grad der Besiedlung ergibt sich aus dem Wert der Schätzung der Bevölkerungszahl und einem Referenzwert, ab welcher Zahl an Bewohnern ein Gebiet der Größe X als dicht besiedelt gilt. Nach einer Zählung der Einwohner in diesem Gebiet (also bei völliger Sicherheit) kann nun diese Aussage getroffen werden: „Das Gebiet ist mit dem Grad 0,6 ein dicht besiedeltes Gebiet“. Man erkennt, daß die Unschärfe auch bei völliger Sicherheit erhalten bleibt.

3.2.3 Der Skalierungsfaktor

Bei geographischen Systemen kann es vorkommen, daß es Objekte gibt, deren Grad an Zugehörigkeit auf dem gleichen Merkmal beruht, denen jedoch ein unterschiedlicher Referenzwert für dessen Berechnung zugrundeliegt. Als Beispiel seien die Regionen „Luftverschmutzung des Kraftwerkes X“ und „Luftverschmutzung des Kraftwerkes Y“ genannt. In beiden Fällen beruht der Zugehörigkeitsgrad auf der gemessenen Luftverschmutzung am betreffenden Punkt. Daher ist dieser Wert für jede Region im Bereich des Kraftwerks 1, da die Luftverschmutzung der entsprechenden Anlage dort am höchsten ist. Handelt es sich um Kraftwerke mit unterschiedlichem Schadstoffausstoß, ist es bei den klassischen unscharfen Mengen nicht möglich, dies zu berücksichtigen. Dies ist jedoch z.B. notwendig, wenn die Gesamtverschmutzung beider Kraftwerke berechnet werden soll.

Ein weiteres Beispiel seien einige Berge, die jeweils durch eine unscharfe Region dargestellt sind. Der höchste Punkt eines Berges gehört zu 100% (also mit Zugehörigkeitsgrad 1) zu dem entsprechenden Berg. Die Vereinigung der Berge kann z.B. ein ganzes Gebirge sein. Bei Verwendung der klassischen unscharfen Mengen hätten in dem entstehenden Gebirge sämtliche Berge die gleiche Höhe, wie durch die nachfolgende Abbildung verdeutlicht wird:

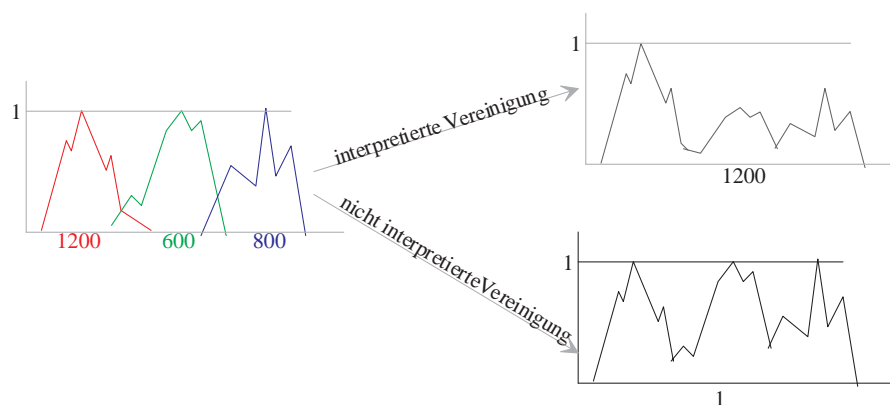


Abb. 9 Notwendigkeit eines Skalierungsfaktors

Auf der linken Seite des Bildes eine Menge von Bergen zu sehen, deren Zugehörigkeitsgrad durch die jeweilige Höhe bestimmt ist. Aus Gründen der Übersichtlichkeit wurde hier nur eine 2-dimensionale Darstellung gewählt. Der höchste Punkt jedes Berges hat den Zugehörigkeitsgrad 1. Die jeweiligen Skalierungsfaktoren wurden ebenfalls notiert. Auf der rechten Seite des Bildes ist die Vereinigung dieser Berge dargestellt. Im oberen Teil wurden (im Gegensatz zu unten) die Referenzwerte mit einbezogen, was zum gewünschten Ergebnis führt.

Um dem Problem der unterschiedlichen Referenzwerte zu begegnen, werden hier die unscharfen Mengen um einen „Skalierungsfaktor“ erweitert, der Informationen über den Referenzwert aufnehmen kann. Operationen, die den Skalierungsfaktor mit einbeziehen, werden im Folgenden als interpretierte Operationen bezeichnet.

Manchmal ist es nicht erwünscht, Operationen wie Schnitt oder Vereinigung interpretiert durchzuführen. Möchte z.B. ein Bauer wissen, wo die Luftverschmutzung von Kraftwerk X auf seinen Grundstücken am größten ist, so ist eine interpretierte Operation, also eine, die den Skalierungsfaktor mit einschließt, nicht sinnvoll. Da ein solches Grundstück eine scharfe Region darstellt, ist auch nicht klar, welchen Skalierungsfaktor die Ergebnisregion erhalten soll. Dieser könnte z.B. mit 1 festgelegt werden. Bei der obigen Fragestellung ergäbe sich bei einer interpretierten Ausführung des Schnittes jedoch folgendes Bild.

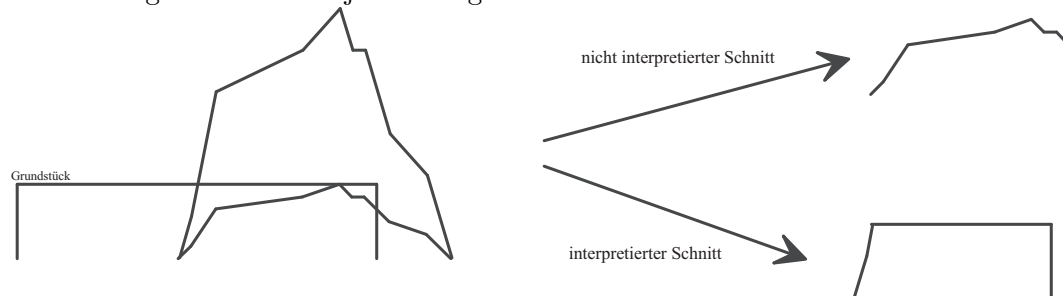


Abb. 10 Schlechte Anwendung eines interpretierten Schnittes

Im linken Teil des Bildes erkennt man die Darstellung des Grundstücks, sowie die skalierte und nicht skalierte Darstellung der Region der Luftverschmutzung. Im rechten Teil des Bildes ist leicht zu sehen, daß durch die interpretierte Anwendung des Schnittes die gewünschte Information verlorengeht. Der „künstliche“ Skalierungsfaktor für das Grundstück müßte also soweit angehoben werden, bis der Funktionsverlauf oberhalb der Zugehörigkeitsfunktion der anderen Region liegt. Somit ist dieser aber abhängig von einem anderen Skalierungsfaktor zu wählen. Dies ist jedoch nicht möglich, da die Skalierungsfaktoren eine feste Größe für die einzelnen Objekte darstellen. Solche Probleme treten immer dann auf, wenn ein scharfes geometrisches Objekt in Berechnungen verwendet wird. Ebenfalls ist ein interpretierter Operator dann nicht sinnvoll, wenn die Referenzwerte der beteiligten Operanden auf unterschiedlichen Merkmalen beruhen. Als Lösung hierzu werden die meisten Operatoren doppelt angeboten. Es gibt dann sowohl eine interpretierte als auch eine nicht-interpretierte Version für jeden Operator. Je nach Anwendungsfall muß dann entschieden werden, welche Version genommen werden soll.

3.2.4 Unstetigkeiten in der Zugehörigkeitsfunktion

Es stellt sich die Frage, ob es sinnvoll ist, Unstetigkeiten in der Zugehörigkeitsfunktion zuzulassen. Ein allgemeines Verbot solcher Unstetigkeiten würde bedeuten, daß mit diesem Modell keine scharfen Regionen dargestellt werden könnten, da diese an den Rändern unstetig sind. Solche Unstetigkeiten in den Rändern können durch Operationen wie z.B. Vereinigung und Differenz in „zentrale Bereiche“ der entstehenden Region überführt werden:

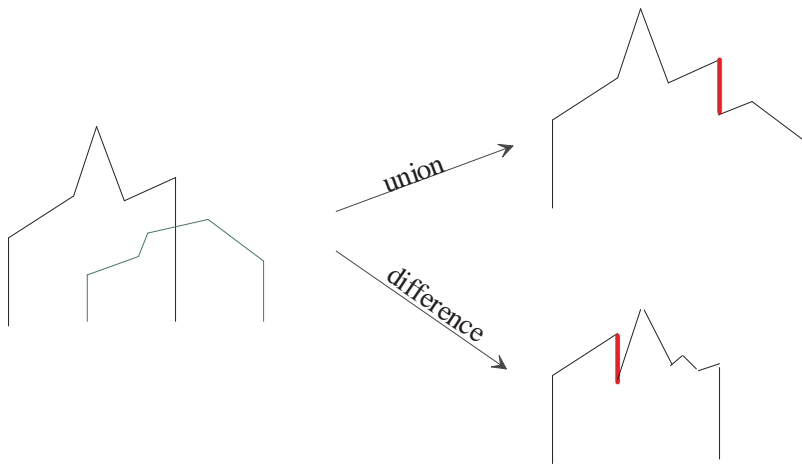


Abb. 11 Übertragung von Unstetigkeiten

Würde man nun Unstetigkeiten im Inneren von Objekten ausschließen, so wären die genannten Operationen auf solchen Objekten nicht abgeschlossen. Es ist zudem möglich, daß z.B. auch neu definierte Regionen in ihrem Inneren Unstetigkeiten haben. Man denke z.B. an eine Region „Schallemmission“. Befindet sich innerhalb dieser Region eine Mauer, so ändert sich die Schallbelastung abrupt und die Region hat an dieser Stelle eine Unstetigkeit. Aus den oben genannten Gründen müssen Unstetigkeiten der Zugehörigkeitsfunktion zugelassen werden. Hierbei kann jedoch die Einschränkung gemacht werden, daß die Anzahl dieser Unstetigkeiten endlich ist.

3.2.5 Der Zugehörigkeitsgrad 0

Eine weitere Frage ist, ob man Zugehörigkeitsgrade zulassen soll, welche Null sind oder ob man verlangen sollte, daß Zugehörigkeitsgrade stets im Intervall $(0, 1]$ liegen. Hier wird gefordert, daß nur solche zusammenhängenden Mengen, die mind. eine Dimension weniger haben, als das betrachtete Objekt, den Grad Null zugewiesen bekommen dürfen. Dies bedeutet, daß innerhalb der Grundmenge einer Region nur Linien oder einzelne Punkte den Wert 0 haben können. Innerhalb der Grundmenge einer Linie darf nur eine endliche Menge einzelner Koordinaten den Wert 0 besitzen. Koordinaten von Punkten dürfen nie einen Zugehörigkeitsgrad von 0 aufweisen. Diese Entscheidung soll anhand von Regionen beschrieben werden und kann problemlos auf Linien oder Punkte übertragen werden.

Um die eindeutige Repräsentation der Regionen zu sichern, wird ausgeschlossen, daß flächige Teile innerhalb der Grundmenge den Wert Null erhalten dürfen. Das Hinzufügen oder Entfernen dieser Flächen ändert die Region selbst nicht, so daß im anderen Fall mehrere Darstellungen der gleichen Region existieren würden. Verbietet man den Zugehörigkeitsgrad 0 vollständig, haben alle Regionen Unstetigkeiten bei der Grenze. Solche Unstetigkeiten sind jedoch in der Realität nicht zwingend. Weiterhin sind die unscharfen Regionen unter diesen Bedingungen nicht unter Differenz abgeschlossen.

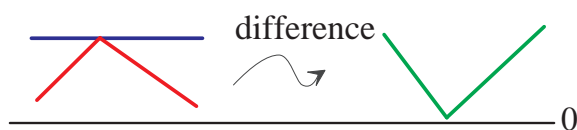


Abb. 12 Kein Abschluß unter Differenz

Durch diese Entscheidung lassen sich jedoch Regionen erzeugen, welche nicht mehr regulär und abgeschlossen sind (vgl. Abschn. 3.1). Es ist möglich, daß einzelne Punkte oder Linien in der Region fehlen und daß die Grenze nur teilweise oder gar nicht zur Region gehört. Weiterhin können Zusammenhangseigenschaften durch eine solche Entscheidung geändert werden. Eine zusammenhängende Region kann nämlich durch eine Null-Linie geteilt werden und ist so nicht mehr zusammenhängend. Dies ist aber nicht so entscheidend, da die Grundmenge ja stets eine reguläre abgeschlossene Menge ist und sich auch der Zusammenhangsbegriff über die Grundmenge definieren läßt.

3.2.6 Darstellung der Objekte

Ein unscharfes räumliches Objekt kann nun als Triple dargestellt werden:

$$\mathbf{FSO} = (B, F, SF)$$

Hierbei bedeuten:

- B : die zugrundeliegende Punktmenge, eine Teilmenge des \mathbb{R}^2
- F : die Zugehörigkeitsfunktion
- SF > 0 : der Skalierungsfaktor

Dabei müssen einige Forderungen erfüllt werden.

Die zugrundeliegende Punktmenge muß einem der im Abschnitt 3.1 definierten Objekte entsprechen.

Die Werte der Zugehörigkeitsfunktion liegen im Intervall $[0,1]$.

Die Funktion muß an jeder Koordinate $(x, y) \in \mathbb{R}$ definiert sein. Außerhalb von B ist die Funktion stets Null. Im Inneren von B können nur solche Teile den Wert 0 annehmen, deren Dimension kleiner als die von B ist (vgl. Abschn.3.2.5).

Formal ergibt sich also für F:

$$F : \mathbb{R} \times \mathbb{R} \rightarrow [0, 1]$$

$$\forall (x, y) \notin B : F(x, y) \stackrel{\text{def}}{=} 0$$

$$\exists (B', F', SF') : B' \subseteq B \wedge F' \equiv 0 \wedge \text{type}(B') = \text{type}(B)$$

Die Menge aller Funktionen, die diese Eigenschaften erfüllen sei mit Θ bezeichnet.

Bei einigen Operationen kann es vorkommen, daß das Ergebnis von diesen Vorgaben abweicht. Speziell könnte das Ergebnis auch Punkt Mengen enthalten, deren Grad stets 0 ist und deren Dimension der Dimension des Ergebnisses entspricht. Um diesem Umstand zu begegnen, wird ein Regularisierungsoperator für unscharfe Objekte eingeführt.

$$\text{reg}(B, F, SF) \stackrel{\text{def}}{=} (\text{reg}(B - \{(x, y) \in B | F(x, y) = 0\}), F, SF)$$

Dieser entfernt zunächst alle Koordinaten aus B, deren Grad Null beträgt. Die Regularisierung dieses Ergebnisses sorgt dann dafür, daß evtl. fehlende Grenzen oder niedriger-dimensionale Teile wieder eingefügt werden.

Um einige Operationen definieren zu können, müssen Funktionen bzw. die Funktionsterme verbunden werden können:

Seien $F_1, F_2 \in \Theta$:

$$\begin{aligned} \min(F_1, F_2) &\stackrel{\text{def}}{=} F_3 : \mathbb{R} \times \mathbb{R} \rightarrow [0, 1] \\ F_3(x, y) &\stackrel{\text{def}}{=} \min(F_1(x, y), F_2(x, y)) \end{aligned}$$

Analog kann \max definiert werden.

$$F_1 + F_2 \stackrel{\text{def}}{=} F_3 : \mathbb{R} \times \mathbb{R} \rightarrow [0, 2]$$
$$F_3(x, y) \stackrel{\text{def}}{=} F_1(x, y) + F_2(x, y)$$

Der Operator $-$ kann ebenfalls so definiert werden, jedoch ist der Wertebereich der sich ergebenden Funktion $[-1, 1]$.

Kapitel 4

Operationen auf unscharfen geometrischen Objekten

In diesem Abschnitt geht es darum, welche Operatoren und Prädikate nötig sind, um ein benutzbares Modell zu erhalten. Es werden Anwendungsfälle betrachtet, aus denen die Operatoren abgeleitet werden, welche zur Lösung einer vorliegenden Problemstellung geeignet sind. Zunächst soll eine Übersicht gegeben werden, welche Operatoren es gibt und welche Semantik diese besitzen.

<i>Operator</i>	<i>Semantik</i>
binäre Operatoren mit unscharfem Objekt als Ergebnis	
<i>union</i>	entspricht der klassischen Vereinigung zweier unscharfer Mengen (hier Punktmengen) der Skalierungsfaktor wird nicht berücksichtigt und daher im Ergebnis auf 1 gesetzt
<i>scaled_union</i>	entspricht der Vereinigung zweier Objekte unter Berücksichtigung des Skalierungsfaktors
<i>intersection</i>	hier werden die gemeinsamen Teile (Schnitt) zweier Objekte berechnet der Skalierungsfaktor bleibt unberücksichtigt und wird im Ergebnis auf 1 gesetzt
<i>scaled_intersection</i>	der Schnitt zweier Objekte mit Berücksichtigung des Skalierungsfaktors
<i>difference</i>	es wird berechnet, welche Teile zur einen Region, jedoch nicht zur anderen Region gehören keine Berücksichtigung des Skalierungsfaktors
<i>scaled_difference</i>	wie <i>difference</i> , jedoch mit Beachtung des Skalierungsfaktors
<i>add</i>	entspricht der Summierung der Zugehörigkeitsgrade von zwei Objekten keine Beachtung des Skalierungsfaktors
<i>scaled_add</i>	wie <i>add</i> , jedoch mit Berücksichtigung des Skalierungsfaktors
<i>common_lines</i>	berechnet gemeinsame Linien zweier Regionen, die nicht Teil eines gemeinsamen zweidimensionalen Teils dieser Regionen sind
<i>common_points</i>	berechnet gemeinsame Koordinaten zweier Linien oder Regionen, die nicht Teil einer gemeinsamen Linie sind

unäre Operatoren mit unscharfem Objekt als Ergebnis

<i>α-cut</i>	berechnet den regulären Teil des unscharfen Objekts, bei dem alle Zugehörigkeitsgrade (bis auf evtl. unreguläre Teile) größer oder gleich α sind
<i>strong-α-cut</i>	wie <i>α-cut</i> , nur daß die Gleichheit mit α nicht zugelassen wird
<i>sharp</i>	setzt alle Zugehörigkeitsgrade eines Objekts auf 1
<i>faces</i>	berechnet die zusammenhängenden Teile einer Region
<i>holes</i>	berechnet Löcher innerhalb einer Region
<i>boundary</i>	berechnet die Grenze einer Region
<i>endpoint</i>	berechnet den Endpunkt (Grenze) einer Linie
<i>contour</i>	berechnet die Grenze einer Region ohne Berücksichtigung von Grenzteilen, die durch Löcher entstehen

Berechnung von Zugehörigkeitsgraden

<i>Zfkt</i>	berechnet den Grad eines Objekts an einer gegebenen Koordinate
<i>maxZ</i>	berechnet den größten Grad eines Objekts insgesamt

Größenangaben von Objekten

<i>basic_card</i>	Anzahl der durch einen unscharfen Punkt abgedeckten Koordinaten ohne Berücksichtigung ihrer Grade
<i>card</i>	Summe der durch die Grade gewichteten Koordinaten eines unscharfen Punktes
<i>basic_len</i>	Länge der von einer unscharfen Linie abgedeckten Linie des \mathbb{R}^2
<i>len</i>	Länge einer Linie mit Berücksichtigung der Zugehörigkeitsgrade
<i>3D_len</i>	Länge der durch eine unscharfe Linie definierten 3D-Struktur
<i>basic_area</i>	Fläche des von einer unscharfen Region abgedeckten Gebiets
<i>area</i>	Fläche mit Berücksichtigung der Zugehörigkeitsgrade
<i>surface</i>	Größe der Oberfläche der durch eine unscharfe Region definierten 3D-Struktur

Operatoren auf Mengen von unscharfen Objekten

<i>mid</i>	Mittelwertberechnung mehrerer unscharfer Objekte ohne Berücksichtigung der Skalierungsfaktoren
<i>scaled_mid</i>	wie <i>mid</i> jedoch mit Berücksichtigung der Skalierungsfaktoren

Ähnlichkeit zweier Objekte

<i>basic_similar</i>	berechnet die Ähnlichkeit zweier durch unscharfe Objekte überdeckten Grundmengen
<i>similar</i>	berechnet die Ähnlichkeit zweier unscharfer Objekte insgesamt

Testen auf Leere

<i>isempty</i>	prüft, ob ein unscharfes Objekt Koordinaten des \mathbb{R}^2 abdeckt
----------------	--

Hilfsoperatoren

<i>basics</i>	berechnet die durch ein unscharfes Objekt überdeckte Grundmenge
<i>norm</i>	normiert ein unscharfes Objekt dazu wird Information aus den Graden in den Skalierungsfaktor übertragen der maximale Grad des Ergebnisobjekts ist stets 1, wenn das Ausgangsobjekt nicht leer war

Im Folgenden wird beschrieben, welche Aufgaben mit den oben aufgeführten Operationen gelöst werden können. Weiterhin werden die Operationen formal definiert.

Zunächst kann festgehalten werden, daß die Operatoren der klassischen scharfen Objekte erhalten bleiben müssen. Da solchen Objekten schwer ein Skalierungsfaktor zuzuordnen ist, müssen hier stets nicht-interpretierte Operatoren verwendet werden. Speziell bedeutet dies, daß für alle drei Typen nicht-interpretierte Versionen von Vereinigung, Schnitt und Differenz angeboten werden müssen. Einfache Anwendungsbeispiele sind:

Jemand erwirbt das Grundstück eines anderen. Dann muß beim Käufer die Vereinigung seines alten Grundstücks mit dem neuen berechnet werden, wogegen beim Verkäufer die Differenz des alten und des verkauften Grundstücks zu ermitteln ist. Der Schnitt muß z.B. dann ermittelt werden, wenn jemand den Teil seines Grundstücks wissen möchte, der innerhalb eines Naturschutzgebietes liegt. Die gleichen Operationen werden auch für die anderen beiden Typen angeboten. Man erhält aus diesen Überlegungen die folgenden Signaturen:

union : **FObjekt_i** × **FObjekt_i** → **FObjekt_i**

intersection : **FObjekt_i** × **FObjekt_i** → **FObjekt_i**

difference : **FObjekt_i** × **FObjekt_i** → **FObjekt_i**

Der Index i bedeutet, daß hier stets Objekte des gleichen Typs verwendet werden. Diese Operationen können wie folgt definiert werden:

$$\textit{union}((B_1, F_1, SF_1), (B_2, F_2, SF_2)) \stackrel{\text{def}}{=} (B_1 \cup B_2, \max(F_1, F_2), 1)$$

Hier handelt es sich um die klassische Vereinigung zweier unscharfer Mengen. Jedoch können die beiden Objekte hier eine unterschiedliche Grundmenge haben. Die Grundmenge wird vereinigt und als Zugehörigkeitsgrad wird das Maximum der Zugehörigkeitsgrade der Operanden gewählt. Da der Skalierungsfaktor bei der „einfachen“ Vereinigung nicht interpretiert wird, ist er im Ergebnis auf 1 gesetzt worden. Man kann sich leicht überlegen, daß hier ein unscharfes Objekt entsteht, welches alle Forderungen für ein gültiges Objekt erfüllt. Insbesondere ist die Grundmenge regulär, die Zugehörigkeitsgrade liegen in [0,1] und nur solche Teilmengen der Grundmenge, welche eine geringere Dimension als diese aufweisen, können überall den Grad Null haben.

$$\textit{intersection}((B_1, F_1, SF_1), (B_2, F_2, SF_2)) \stackrel{\text{def}}{=} (B_1 \cap_{\text{reg}} B_2, \min(F_1, F_2), 1)$$

Hier wurde der klassische Schnitt zweier unscharfer Mengen so angepaßt, daß der Schnitt der Basismengen regulär ist.

$$\textit{difference}((B_1, F_1, SF_1), (B_2, F_2, SF_2)) \stackrel{\text{def}}{=} \textit{reg}(B_1 - B_2, \max(0, F_1 - F_2), 1)$$

Dieser Operator berechnet den neuen Zugehörigkeitsgrad als Differenz der beiden ‚alten‘, wobei Werte unter Null ‚abgeschnitten‘ werden. Der Regularisierungsoperator sorgt dafür, daß sich ein gültiges unscharfes Objekt ergibt, d.h. es werden Teile, die komplett den Grad Null aufweisen aus der Grundmenge entfernt.

Man sieht an den Signaturen, daß es sich um Operatoren handelt, die auf dem jeweiligen Typ abgeschlossen sind. So kann man z.B. durch den Schnittoperator nur gemeinsame eindimensionale Teile von Linien, jedoch nicht deren Schnittpunkte ermitteln. Da Schnittpunkte von Linien oder gemeinsame Grenzteile oder Berührungspunkte von Regionen jedoch benötigt werden, müssen hierfür entsprechende Operatoren angeboten werden. Z.B. benötigt man den gemeinsamen Schnittpunkt bei der Beantwortung der Frage „Wo kreuzen sich die Straßen A und B“.

$$\mathit{commonLines} : \mathbf{FRegion} \times \mathbf{FRegion} \rightarrow \mathbf{FLine}$$

$$\mathit{commonPoints} : \mathbf{FRegion} \times \mathbf{FRegion} \rightarrow \mathbf{FPoint}$$

$$\mathit{commonPoints} : \mathbf{FLine} \times \mathbf{FLine} \rightarrow \mathbf{FPoint}$$

Die zweifache Signatur des *commonPoints*-Operators bedeutet, daß er auf Linien oder Regionen angewendet werden kann, wobei diese Objekte jedoch nicht gemischt auftreten dürfen. Innerhalb des abstrakten Modells können diese Operatoren wie folgt definiert werden:

$$\mathit{commonPoints}((B_1, F_1, SF_1), (B_2, F_2, SF_2)) \stackrel{\text{def}}{=} (\{P \in B_1 \cap B_2 : \exists L \in \mathbf{Line} : L \subseteq B_1 \cap B_2 \wedge P \subseteq L\}, \mathbf{1}, \mathbf{1})$$

Es werden hier also nur solche gemeinsamen Punkte betrachtet, die nicht Teil einer gemeinsamen Linie (und damit auch nicht Teil einer gemeinsamen Fläche) sind.

$$\mathit{commonLines}((B_1, F_1, SF_1), (B_2, F_2, SF_2)) \stackrel{\text{def}}{=} (\{L \in B_1 \cap B_2 : \exists R \in \mathbf{Region} : R \in B_1 \cap B_2 \wedge L \subseteq R\}, \mathbf{1}, \mathbf{1})$$

Dieser Operator berechnet solche Linien, die beide Regionen gemeinsam besitzen und nicht Teil einer gemeinsamen Fläche sind. Es ist klar, daß dieser Operator nur für Regionen Sinn macht, da Punkte keine gemeinsame Linie besitzen können und bei Linien der Schnitt-Operator diese Aufgabe erfüllt.

Für jeden der klassischen Mengenoperatoren wird zusätzlich eine interpretierte Version angeboten. Diese heißt entsprechend *scaled_union*, *scaled_intersection* etc. und besitzt die gleiche Signatur wie die nicht-interpretierte Version. Um diese Operatoren definieren zu können, werden einige Hilfsoperatoren verwendet:

$$\mathit{maxZ} : \mathbf{FObject}^* \rightarrow \mathbb{R}$$

$$\mathit{maxZ}(B, F, SF) \stackrel{\text{def}}{=} \text{MAX}(\{F(x, y) \mid (x, y) \in B\})$$

Dieser Operator bestimmt den maximalen Zugehörigkeitsgrad des gesamten Objekts. **FObject*** bedeutet, daß dieser Operator auch auf „ungültige“ unscharfe Objekte angewendet werden kann und zwar solche, deren Grundmenge nicht regulär ist oder deren Zugehörigkeitsgrade das Intervall [0,1] verlassen haben. Zudem können diese Objekte auch Teile in ihrer Grundmenge besitzen, deren Zugehörigkeitsgrad Null ist und deren Dimension mit der Dimension der Grundmenge übereinstimmt.

$$\mathit{minZ} : \mathbf{FObject}^* \rightarrow \mathbb{R}$$

$$\mathit{minZ}(B, F, SF) \stackrel{\text{def}}{=} \text{MIN}(\{F(x, y) \mid (x, y) \in B\} \cup \{0\})$$

Dieser Operator arbeitet ähnlich wie der $maxZ$ -Operator. Man kann erkennen, daß bei gültigen Objekten stets der Wert 0 geliefert wird.

$norm : \mathbf{FObject}_i^* \rightarrow \mathbf{FObject}_i$

$$norm(O = (B, F, SF)) \stackrel{\text{def}}{=} reg((B, \frac{F - minZ(O)}{maxZ(O) - minZ(O)}, SF \cdot (maxZ(O) - minZ(O))))$$

Dieser Operator wandelt ungültige unscharfe Objekte in äquivalente gültige Objekte um. Gleichzeitig findet eine Normierung der Zugehörigkeitsgrade statt. D.h. der maximale Grad eines solchen Objekts beträgt 1 und der minimale Grad beträgt 0. Die Information der ‚alten‘ Extrema wird in den Zugehörigkeitsgrad übertragen. Probleme mit der Grundmenge (Regularität) werden durch den reg -Operator entfernt.

Mit Hilfe dieses Operators lassen sich nun die interpretierten Mengenoperatoren definieren:

$$scaled_union((B_1, F_1, SF_1), (B_2, F_2, SF_2)) \stackrel{\text{def}}{=} norm((B_1 \cup B_2, max(SF_1 \cdot F_1, SF_2 \cdot F_2), 1))$$

Hier wird zunächst die Information von Grad und Skalierungsfaktor in den Zugehörigkeitsgrad aufgenommen. Diese Gesamtinformation wird anschließend verarbeitet. Durch den $norm$ -Operator wird dann ein Teil der Information wieder in den Skalierungsfaktor verschoben und dadurch ein gültiges Objekt erzeugt.

$$scaled_intersection((B_1, F_1, SF_1), (B_2, F_2, SF_2)) \stackrel{\text{def}}{=} norm((B_1 \cup B_2, min(SF_1 \cdot F_1, SF_2 \cdot F_2), 1))$$

Auch bei diesem Operator wird die Information wie bei $scaled_union$ zweimal verschoben.

$$scaled_difference((B_1, F_1, SF_1), (B_2, F_2, SF_2)) \stackrel{\text{def}}{=} norm((B_1 \cup B_2, SF_1 \cdot F_1 - SF_2 \cdot F_2, 1))$$

Innerhalb des $norm$ -Operators können hier Objekte entstehen, die Grade kleiner als Null besitzen (Differenzbildung), die Grade größer als 1 haben (Informationsverschiebung vom Skalierungsfaktor) oder die Teile mit dem Zugehörigkeitsgrad Null besitzen. Durch den $norm$ -Operator werden diese „Zwischenprodukte“ in äquivalente gültige unscharfe Objekte transformiert.

Das hier verwendete Modell hat eine recht unstrukturierte Sicht auf die räumlichen Objekte. Das heißt, daß man es einem Objekt allein nicht „ansieht“, ob es aus mehreren Komponenten besteht oder ob z.B. eine Region Löcher enthält. Da Objekte sehr komplex sein können, insbesondere aus mehreren Komponenten bestehen können, ist es jedoch manchmal wünschenswert, die einzelnen Komponenten, die im Folgenden als Faces bezeichnet werden, trennen zu können. Ein Beispiel, wo eine solche Trennung benötigt wird, ist es herauszufinden, ob jemand ein zusammenhängendes Grundstück der Größe n hat, um darauf einen Betrieb errichten zu können. Es ergibt sich:

$faces : \mathbf{FRegion} \cup \mathbf{FLine} \rightarrow \mathbf{Set\ of\ FObject}$

$$faces((B, F, SF)) \stackrel{\text{def}}{=} \{(B_{new}, F, SF) \mid \bigcup B_{new} = B \wedge (i \neq j \Rightarrow Dimension(B_{new_i} \cap B_{new_j}) \leq Dimension(B) - 2)\}$$

wobei gilt :

$$Dimension(B) < 0 \Leftrightarrow B = \emptyset$$

Bei einer Region können zusätzlich die in ihr enthaltenen Löcher interessant sein.

$holes : \mathbf{FRegion} \rightarrow \mathbf{FRegion}$

Das Ergebnis dieses Operators stellt alle Löcher dieser Region gemeinsam dar. Um einzelne Löcher zu erhalten, werden die Faces dieser „Lochregion“ ermittelt.

In der Literatur werden Regionen mit Löchern meist aus Mengen von einfachen Regionen konstruiert. Dabei wird die Differenz einer Region mit der Vereinigung aller anderen Regionen (welche die Löcher darstellen) gebildet. Die einzelnen Regionen müssen dabei bestimmte Eigenschaften erfüllen, auf die hier jedoch nicht näher eingegangen werden soll.

In dieser Arbeit wird genau der umgekehrte Weg gegangen. Der Ausgangspunkt wird von einer komplexen Region gebildet, und es soll die Frage geklärt werden, welche Eigenschaften eine Punktmenge haben muß, um als Loch dieser Region zu gelten.

Um einen Operator anzugeben, der die Löcher einer Region berechnet, muß zunächst geklärt werden, was in dieser Arbeit als Loch einer Region angesehen wird. Löcher besitzen die folgenden Eigenschaften:

Eigenschaft 1: Ein Loch ist Teil des Äußeren der Region.

Eigenschaft 2: Ein Loch ist vollständig von einem Face der Region eingeschlossen. Dies bedeutet, daß Teile des Äußeren, die sich zwischen zwei Faces befinden, nicht unter den Begriff „Loch“ fallen.

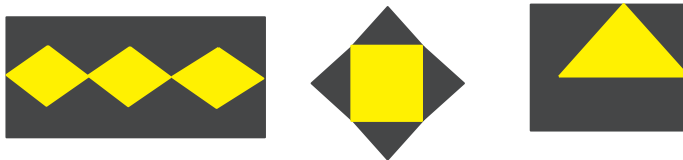


Abb. 13 Keine Löcher

Hieraus ergibt sich, daß man die Löcher einer Region berechnen kann, wenn man in der Lage ist, ein Loch eines Faces zu berechnen. Die Löcher der Region ergeben sich dann aus der Vereinigung aller dieser Löcher, von der noch einmal die „scharfe“ Region abgezogen werden muß, um eventuelle Inseln zu berücksichtigen.

$$\text{holes}(R) \stackrel{\text{def}}{=} \text{reg}\left(\bigcup_{F \in \text{faces}(R)} \text{holes}(F) - \text{sharp}(R)\right)$$

Der *sharp*-Operator setzt alle Grade innerhalb von B auf 1 und wird später genauer erläutert. Es ergibt sich die Frage, wie man nun die Löcher eines einzelnen Faces F berechnen kann. Ein Loch könnte als eine Punktmenge angesehen werden, die eine echte Teilmenge des Äußeren von F ist. Bei konsequenter Anwendung dieser Definition ist ein Loch jedoch eine offene Punktmenge. Um ein Loch als Region darstellen zu können, ist diese Menge um ihre Grenze zu erweitern. In den folgenden Ausführungen soll nun geklärt werden, welche Eigenschaften eine Punktmenge H erfüllen muß, um Loch eines Face F zu sein.

Da H ein einzelnes Loch darstellen soll, muß diese Menge selbst ein Face darstellen. Das bedeutet, daß H stark verbunden sein muß. Im Einzelnen lassen sich je zwei Punkte in H durch eine vollständig in H enthaltene Jordankurve verbinden. Auch wenn aus H eine endliche Punktmenge entfernt wird, darf diese Eigenschaft nicht verloren gehen.

Es macht keinen großen Sinn, eine leere Menge als Loch zu betrachten, daher muß H Elemente enthalten.

Löcher stellen „inneres Äußeres“ einer Region dar. Hieraus ergibt sich, daß der Schnitt der Inneren von F und H leer sein muß.

Es bleibt noch zu klären, wie gesichert werden kann, daß zum einen H nicht zum „äußeren Äußeren“ gehört und nicht nur ein „Teilloch“ von F darstellt. Man betrachte hierzu den Verlauf

der Grenze von H .

Angenommen, die Grenze von H schneidet das Innere von F . In jeder Umgebung eines Grenzpunktes von H gibt es sowohl Punkte, die zu H gehören, als auch Punkte des Äußeren von H . Weiterhin gibt es eine Umgebung für jeden Punkt des Inneren von F , die vollständig in F enthalten ist. Betrachtet man nun einen gemeinsamen Punkt der Grenze von H und des Inneren von F , so gibt es auch einen Schnitt des Inneren von F mit dem Inneren von H *widerspruch*.

Es muß nun noch der Schnitt der Grenze von H mit dem Äußeren von F geprüft werden. Hierbei werden zwei Fälle unterschieden. Angenommen, die Grenze schneidet das äußere Äußere von F . Mit einer ähnlichen Begründung wie oben, läßt sich auf einen Schnitt des äußeren Äußeren von F mit dem Inneren von H schließen *widerspruch*.

Beim Schnitt der Grenze mit dem inneren Äußeren, würde nicht das gesamte Loch von F durch H ausgefüllt werden, da ein Teil im Äußeren von H liegen würde *widerspruch*.

Somit darf die Grenze von H nur die Grenze F schneiden.

Umgekehrt liegt jedes Face H , welches diese Eigenschaften erfüllt, im inneren Äußeren von F , wobei jeweils ein ganzes Loch ausgefüllt wird.

Es ist klar, daß solch ein Face H im Äußeren von F liegt, da es nicht leer ist und ein Schnitt mit dem Inneren von F nicht erlaubt ist.

Angenommen, so ein Face liegt im äußeren Äußeren von F . Da H begrenzt ist, muß es eine Grenze im Äußeren von F geben *widerspruch*.

Angenommen, so ein Face füllt nur einen Teil eines Loches von F aus, dann muß es jedoch innerhalb eines solchen Loches eine Grenze von H geben. Dies bedeutet einen Schnitt von Grenze und Äußeren von F *widerspruch*.

Insgesamt ergeben die obigen Ausführungen die folgende Aussage: „Ein nichtleeres Face H ist genau dann ein Loch eines Face F , wenn sich die Inneren von F und H nicht schneiden und die Grenze von H eine echte Teilmenge der Grenze von F ist.“

Unter Verwendung des 9-Intersection-Modells (vgl. Kapitel 5), läßt sich ein entsprechendes Prädikat einführen, welches die Locheigenschaft eines Face prüft:

$isHole : \mathbf{Face} \times \mathbf{Face} \rightarrow \mathbf{boolean}$

$$isHole(H, F) \stackrel{\text{def}}{=} T[H, F] = \begin{array}{ccc|ccc} & & & F^\circ & \partial F & F^{-1} \\ & & & H^\circ & 0 & 0 & 1 \\ & & & \partial H & 0 & 1 & 0 \\ & & & H^{-1} & 1 & 1 & 1 \end{array}$$

Der Lochoperator liefert nun zu einem Face die Menge aller Löcher als eine einzige Region:

$holes : \mathbf{Face} \rightarrow \mathbf{FRegion}$

$$holes(F) \stackrel{\text{def}}{=} \bigcup_{H:isHole(H,F)} H$$

Um die Gesamtverschmutzung von mehreren Kraftwerken bestimmen zu können, ist die normale (auf dem Maximum-Operator beruhende) Vereinigung nicht angebracht. Es ist vielmehr eine additive Verknüpfung der einzelnen Zugehörigkeitsgrade der betreffenden Regionen notwendig.

$add : \mathbf{FObjekt}_i \times \mathbf{FObjekt}_i \rightarrow \mathbf{FObjekt}_i$

$$add((B_1, F_1, SF_1), (B_2, F_2, SF_2)) \stackrel{\text{def}}{=} (B_1 \cup B_2, \min(1, F_1 + F_2), 1)$$

Die Zugehörigkeitsgrade könnten hier Werte erreichen, die größer als 1 sind. Da es sich hier um einen nicht-interpretierten Operator handelt, kann diese Information nicht in den Skalierungsfaktor verschoben werden. Es sind vielmehr solche Werte „abzuschneiden“, was in der obigen Formel durch die Minimumsbildung erreicht wird.

$$\textit{scaled_add} : \mathbf{FObjekt}_i \times \mathbf{FObjekt}_i \rightarrow \mathbf{FObjekt}_i$$

$$\textit{scaled_add}((B_1, F_1, SF_1), (B_2, F_2, SF_2)) \stackrel{\text{def}}{=} \textit{norm}((B_1 \cup B_2, SF_1 \cdot F_1 + SF_2 \cdot F_2, 1))$$

Im Gegensatz zur nicht-interpretierten Version wird hier die Information über zu große Grade in den Skalierungsfaktor geschoben.

Wie bei scharfen Punktmenge auch, möchte man über das Enthaltensein eines Punktes in einem Objekt informiert werden. Da jeder Punkt hier jedoch zu einem bestimmten Grad dieser Struktur angehört, kann es sich hierbei nicht um ein Prädikat handeln.

$$\textit{Zfkt} : \mathbf{FObjekt} \times (\mathbb{R}, \mathbb{R}) \rightarrow [0, 1]$$

$$\textit{Zfkt}((B, F, SF), (x, y)) \stackrel{\text{def}}{=} F(x, y)$$

Möchte ein Landwirt eine bestimmte Pflanze anbauen, kann es entsprechend gesetzlichen oder ökologischen Anforderungen möglich sein, daß diese nur dort angebaut werden darf, wo die Grenzwerte der Belastung durch Chemikalien nicht überschritten werden. Hierfür werden Filteroperatoren benötigt, die geometrische Objekte auftrennen können. Diese werden vom *alpha_cut* bzw. dem *strong_alpha_cut* der unscharfen Mengen bereitgestellt. Da in dieser Arbeit (im Gegensatz zu klassischen unscharfen Mengen) jedes Objekt eine eigene Grundmenge hat, muß das Ergebnis dieser Filter keine scharfe Menge darstellen. Es wird vielmehr die Grundmenge des Objekts geändert und die Grade beibehalten.

$$\alpha_cut : \mathbf{FObjekt}_i \times [0, 1] \rightarrow \mathbf{FObjekt}_i$$

$$\alpha_cut((B, F, SF), \alpha) \stackrel{\text{def}}{=} \textit{reg}(\{(x, y) \in B : F(x, y) \geq \alpha\}, F, SF)$$

$$\textit{strong_alpha_cut} : \mathbf{FObjekt}_i \times [0, 1] \rightarrow \mathbf{FObjekt}_i$$

$$\textit{strong_alpha_cut}((B, F, SF), \alpha) \stackrel{\text{def}}{=} \textit{reg}(\{(x, y) \in B : F(x, y) > \alpha\}, F, SF)$$

Daß solch ein Filter die Zugehörigkeitsgrade unberührt läßt, kann manchmal unerwünscht sein. Ist z.B. der Bau von Wohnsiedlungen dort verboten, wo das Vorkommen der Pflanze XY einen bestimmten Grad übersteigt, ist mit den übernommenen Graden der Region „Vorkommen von XY“ nicht viel anzufangen. Die durch den entsprechenden Filter entstandene Region muß vielmehr in eine scharfe Region umgewandelt werden, da sich das Verbot zu 100% über die gesamte Region erstreckt.

$$\textit{sharp} : \mathbf{FObjekt}_i \rightarrow \mathbf{FObjekt}_i$$

$$\textit{sharp}(B, F, SF) \stackrel{\text{def}}{=} (B, \mathbf{1}, SF)$$

Dieser Operator macht nichts weiter, als alle Grade innerhalb von B auf 1 zu setzen.

Da durch Operatoren wie Schnitt, Differenz oder Alpha-Cut leere Objekte entstehen können, muß ein Prädikat angeboten werden, der ein Objekt daraufhin überprüfen kann, ob er überhaupt eine Koordinate enthält.

$isempty : \mathbf{FObjekt} \rightarrow \mathbf{boolean}$

$$isempty((B, F, SF)) \stackrel{\text{def}}{=} B = \emptyset$$

Die Länge einer Basislinie ist z.B. wichtig, wenn eine neue Straße gebaut werden soll. Aus dieser Länge (und natürlich anderen Faktoren) können Baukosten und Bauzeit abgeleitet werden. Führt diese Straße nun über unterschiedlichen Untergrund, so kann der Weg, auf dem die Straße gebaut werden soll, als unscharfe Linie modelliert werden. Die unterschiedliche Beschaffenheit des Untergrundes ist dabei als Zugehörigkeitsgrad zu modellieren. Man modelliert also eine unscharfe Linie mit Namen ‚Schwieriger Untergrund‘. Die unscharfe Länge dieser Linie betrachtet also zusätzlich die Beschaffenheit des Bodens und kann daher genauere Auskünfte über Kosten geben. Es gibt also zwei Längenoperatoren, die auf Linien definiert werden. Weiterhin kann man eine solche Linie als 3D-Struktur betrachten und die Länge dieses Gebildes berechnen wollen. Daraus ergeben sich folgende Operatoren.

$basic_len : \mathbf{FLine} \rightarrow \mathbb{R}$

$$basic_len((B, F, SF)) \stackrel{\text{def}}{=} len(B)$$

$len : \mathbf{FLine} \rightarrow \mathbb{R}$

Um diesen Operator definieren zu können, benötigt man die Darstellung der Basis der Linie als Menge von Kurven. Die Länge der Linie ergibt sich durch Addition der Längen der enthaltenen Kurven.

$$len(\mathbf{K}, F, SF) \stackrel{\text{def}}{=} \sum_{K \in \mathbf{K}} len(K, F)$$

Die Länge einer mit einer Zugehörigkeitsfunktion versehenen Kurve berechnet sich wie folgt:

$$len(f(c), F) \stackrel{\text{def}}{=} \int_a^b F(f(x)) dx$$

Hierbei sei $f(c)$ die Parameterdarstellung der Kurve und $[a, b]$ das der Kurve zugrundeliegende Intervall.

$3D_len : \mathbf{FLine} \rightarrow \mathbb{R}$

Auch bei der Berechnung der durch eine unscharfe Linie definierten 3D-Struktur, muß man auf die Darstellung der Basis der Linie zurückgreifen.

$$3D_len(\mathbf{K}, F, SF) \stackrel{\text{def}}{=} \sum_{K \in \mathbf{K}} 3D_len(K, F)$$

Um die 3D-Länge einer Kurve berechnen zu können, wird in diese ein Polygonzug eingeschrieben, dessen Länge berechnet werden kann. Es sei mit $t_0 \dots t_n$ eine Zerlegung des Intervalls $[a, b]$ gegeben.

$$3D_BLen_n(K, F) \stackrel{\text{def}}{=} \sum_{i=0}^{n-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2 + (z_i - z_{i+1})^2}$$

mit:

$$\begin{aligned}x_i &= f(t_i).x \\y_i &= f(t_i).y \\z_i &= F(f(t_i))\end{aligned}$$

Die 3D-Länge der Kurve ist dann durch $\lim_{n \rightarrow \infty} 3D_BLen(K, F)$ gegeben.

Was für eine Linie die Länge ist, ist für ein in diesem Modell verwendetes Punktobjekt die Anzahl der enthaltenen Punkte. Es kann hierbei wieder zwischen der Anzahl enthaltener Basispunkte und gewichteter Summe der Basispunkte unterschieden werden:

$$basic_card : \mathbf{FPoint} \rightarrow \mathbb{N}$$

$$basic_card((B, F, SF)) \stackrel{\text{def}}{=} |B|$$

$$card : \mathbf{FPoint} \rightarrow \mathbb{R}$$

$$card((B, F, SF)) \stackrel{\text{def}}{=} \sum_{b \in B} Zfkt(P, b)$$

Nachdem nun für Punkte und Linien ein Ausmaß berechnet werden kann, fehlen noch entsprechende Operatoren für Regionen. Wie bei Linien kann die Größe der Basisfläche, die Größe der bewerteten Fläche und die Oberfläche der 3D-Struktur berechnet werden:

$$basic_area : \mathbf{FRegion} \rightarrow \mathbb{R}$$

$$basic_area((B, F, SF)) \stackrel{\text{def}}{=} \int_{(x,y) \in B} 1$$

$$area : \mathbf{FRegion} \rightarrow \mathbb{R}$$

$$area((B, F, SF)) \stackrel{\text{def}}{=} \int_{(x,y) \in B} F(x, y)$$

$$3D_area : \mathbf{FRegion} \rightarrow \mathbb{R}$$

Die Oberfläche einer Region läßt sich wie folgt durch eine Parameterdarstellung angeben:

$$\kappa(x, y) = \begin{pmatrix} x \\ y \\ F(x, y) \end{pmatrix}$$

Dann läßt sich die Oberfläche wie folgt berechnen (vgl. [25])

$$\int_B \sqrt{1 + f_x^2(x, y) + f_y^2(x, y)} db$$

Diese Formel kann dann angewendet werden, wenn F differenzierbar ist. Da die Funktion als stückweise stetig vorausgesetzt wird, kann hiermit bereits eine große Anzahl der möglichen Funktionen (i.E. alle Funktionen, die bei einer praktischen Realisierung in Betracht kommen) berechnet werden.

$$3D_area((B, F, SF)) \stackrel{\text{def}}{=} \int_B \sqrt{1 + F_x^2(x, y) + F_y^2(x, y)} db$$

Hat man die Umweltverschmutzung eines Kraftwerks zu mehreren Zeitpunkten aufgenommen, so kann man sich dafür interessieren, wie ähnlich sich diese beiden Regionen sind. Dazu kann folgender Operator verwendet werden:

$$\mathit{similar} : \mathbf{FObjekt}_i \times \mathbf{FObjekt}_i \rightarrow [0, 1]$$

$$\mathit{similar}(O_1, O_2) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{wenn } \mathit{isempty}(O_1) \wedge \mathit{isempty}(O_2) \\ \frac{\mathit{size}(\mathit{intersection}(O_1, O_2))}{\mathit{size}(\mathit{union}(O_1, O_2))}, & \text{sonst} \end{cases}$$

Hier sind für den Operator *size* jeweils *len*, *card* oder *area* einzusetzen. Sind beide Objekte gleich, so ergeben Vereinigung und Schnitt das Objekt selbst. Demnach sind die entsprechenden Ausmaße auch gleich groß und der Operator liefert den Wert 1. Bei disjunkten Objekten ist der Schnitt gleich der leeren Menge und der Operator liefert Null.

Ist man nur an der Ähnlichkeit der Basis des entsprechenden Objekts interessiert, ist der *basic_similar* - Operator zu verwenden :

$$\mathit{basic_similar} : \mathbf{FObjekt}_i \times \mathbf{FObjekt}_i \rightarrow [0, 1]$$

$$\mathit{basic_similar}(O_1, O_2) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{wenn } \mathit{isempty}(O_1) \wedge \mathit{isempty}(O_2) \\ \frac{\mathit{basic_size}(\mathit{intersection}(O_1, O_2))}{\mathit{basic_size}(\mathit{union}(O_1, O_2))}, & \text{sonst} \end{cases}$$

wobei *basic_size* für *basic_len*, *basic_card* bzw. *basic_area* steht (in Abhängigkeit von den betrachteten Objekten).

Möchte jemand sein Grundstück einzäunen, so muß er die Länge der Grenze dieses Grundstücks berechnen. Dazu muß er die Grenze seines Grundstücks bestimmen können. Da es i.a. nicht erforderlich ist, dazu Löcher einer Region zu betrachten, gibt es einen Operator, der nur die äußere Grenze zurückgibt. Um die Gesamtgrenze bestimmen zu können, wird ein zweiter Operator verwendet.

$$\mathit{boundary} : \mathbf{FRegion} \rightarrow \mathbf{FLine}$$

$$\mathit{boundary}((B, F, SF)) \stackrel{\text{def}}{=} (\partial B, \mathbf{1}, 1)$$

Da als Grenze nur die Grenze der scharfen Begrenzung angesehen werden kann, muß diese nur noch in ein unscharfes Objekt ‚verpackt‘ werden.

$$\mathit{contour} : \mathbf{FRegion} \rightarrow \mathbf{FLine}$$

$$\mathit{contour}(O = (B, F, SF)) \stackrel{\text{def}}{=} \mathit{reg}((\partial B - \partial(\mathit{holes}(O))), \mathbf{1}, 1)$$

Der *contour*-Operator liefert die Grenze ohne Betrachtung der Löcher.

Ebenso kann es interessant sein, die Endpunkte (oder vielmehr den Endpunkt) einer Linie zu berechnen.

$$\mathit{endpoint} : \mathbf{FLine} \rightarrow \mathbf{FPoint}$$

$$\mathit{endpoint}((B, F, SF)) \stackrel{\text{def}}{=} (\partial B, \mathbf{1}, 1)$$

In einem Gebiet seien die Verschmutzungen durch verschiedene chemische Stoffe durch Regionen beschrieben. Interessiert man sich nun für die durchschnittliche Verschmutzung durch mehrere Stoffe, kann dies mit den bisherigen Mitteln nicht berechnet werden. Man benötigt daher einen

zusätzlichen Operator, der auf Mengen von Objekten arbeitet.

$mid : \text{set of Objekt}_i \rightarrow \text{Objekt}_i$

$$mid(SoFO) \stackrel{\text{def}}{=} \left(\bigcup_{(B_i, F_i, SF_i) \in SoFO} B_i, \frac{\sum_{(B_i, F_i, SF_i) \in SoFO} F_i}{|SoFO|}, 1 \right)$$

Das Ergebnis enthält hier die mittleren Zugehörigkeitsgrade aller beteiligten Operanden.

$scaled_mid : \text{set of Objekt}_i \rightarrow \text{Objekt}_i$

$$scaled_mid(SoFO) \stackrel{\text{def}}{=} norm\left(\left(\bigcup_{(B_i, F_i, SF_i) \in SoFO} B_i, \frac{\sum_{(B_i, F_i, SF_i) \in SoFO} F_i \cdot SF_i}{|SoFO|}, 1 \right)\right)$$

Auch hier werden die mittleren Zugehörigkeitsgrade berechnet, wobei hier zusätzlich die Skalierungsfaktoren der Operanden einbezogen werden.

Kapitel 5

Verallgemeinerung des 9-Intersection-Modells auf komplexe geometrische Objekte

In den folgenden Abschnitten wird untersucht, welche Matrizen des 9-Intersection-Modells bedeutungsvolle topologische Beziehungen zwischen den jeweils betrachteten Objekten beschreiben können. Dazu wird eine Menge von Bedingungen definiert, die solche Matrizen ausschließen, die eine unmögliche Konstellation der Objekte ausdrücken.

Am Beispiel der topologischen Beziehungen zwischen Regionen soll nun erläutert werden, wie solche Bedingungen gefunden werden können. Man beginnt zuerst mit der Menge aller möglichen (512) Matrizen. Man versucht, zu einer Matrix eine mögliche Realisierung zu finden, also eine Lagebeziehung zwischen zwei Regionen, auf die diese Matrix zutrifft. Es geht speziell darum, ein Bild zweier Regionen zu zeichnen, deren topologische Beziehung von dieser Matrix beschrieben wird. Findet man keine solche Zuordnung, untersucht man, warum dies der Fall ist. Dies kann meist weiter verallgemeinert werden. So entsteht eine neue Bedingung, welche die Anzahl realisierbarer Matrizen weiter eindämmt. Die Begründungen sichern dabei, daß keine Matrizen eliminiert werden, welche doch noch eine bedeutungsvolle Lagebeziehung darstellen können. Dies führt man solange fort, bis jede verbleibende Matrix eine Lagebeziehung zugeordnet bekommen hat. Im Falle der Regionen habe ich so insgesamt 9 Bedingungen gefunden. Da den verbleibenden Matrizen jeweils ein Paar von Regionen mit der entsprechenden Beziehung zugeordnet werden konnte, sind diese Bedingungen ausreichend. Es ist jedoch möglich, daß sich innerhalb dieser Menge redundante Bedingungen befinden. Dies ist dann der Fall, wenn später hinzugekommene Bedingungen die ‚alten‘ überdecken. Die Untersuchung der 9 Matrizen für Regionen ergab, daß die Bedingung 2 nur Matrizen ausschließt, welche bereits durch andere entfernt werden. Jede der anderen Bedingungen schließt mindestens eine Matrix allein aus. Somit bilden die Bedingungen 1,3,4,5,6,7,8 und 9 eine minimale und vollständige Menge von Einschränkungen, um alle nicht realisierbaren Matrizen auszuschließen.

Hier werden nur solche Objekte untersucht, die nicht leer sind, d.h. im Inneren und/oder der Grenze Punkte enthalten. Eine Beziehung zwischen zwei Objekten zu beschreiben, von denen mind. eins leer ist, ist nicht sehr sinnvoll. Die 9-Intersection-Matrix ist dort trivial zu ermitteln. Aufgrund der Leere von Grenze und Innerem, können diese Teile keine Schnitte besitzen. Demnach sind Zeile oder Spalte (je nach Position des leeren Objekts) mit 0-Einträgen besetzt. Die Zeile/Spalte des Äußeren des leeren Objekts hat genau an den Positionen eine 1, an denen das andere betrachtete Objekt nicht leer ist.

5.1 Topologische Beziehungen zwischen Punkten

Die topologischen Beziehungen von Punkten wurden in der Literatur bisher wenig beachtet. Dies ist auch verständlich, da Punkte nur durch eine Koordinate dargestellt wurden und daher sich die Beziehungen auf Gleichheit und Disjunktheit beschränkten. Da im vorliegenden Modell Punkte durch Mengen von Koordinaten dargestellt sind, ergibt sich ein anderes Bild.

Die geometrischen Bedingungen, die alle Matrizen ausschließen, welche nichtrealisierbare Beziehungen zwischen zwei Punkten beschreiben, sind die folgenden (vgl. [44]):

- Aufgrund der leeren Grenze, haben sowohl Zeile als auch Spalte der Grenze stets nur 0-Einträge.
- Da eine endliche Punktmenge nicht die gesamte Grundmenge abdecken kann, müssen sich die beiden Äußeren der Punkte schneiden.
- Da nur nichtleere Punktobjekte betrachtet werden und durch die Menge der Koordinaten und des Äußeren die gesamte Grundmenge abgedeckt wird, muß das Innere eines Punktes einen Schnitt mit dem Inneren des anderen Punktes oder mit dem Äußeren des anderen Punktes besitzen.

Von den 512 möglichen Matrizen erfüllen die folgenden die oben genannten Bedingungen:

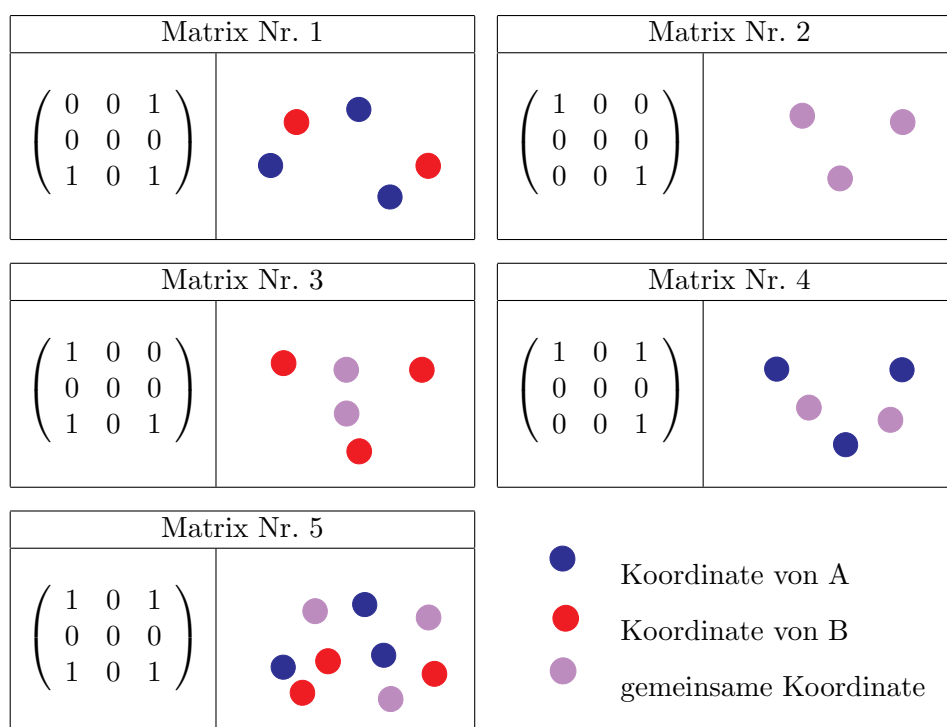


Abb. 14 Topologische Beziehungen zwischen Punkten

5.2 Topologische Beziehungen zwischen Linien

Auch die topologischen Beziehungen zwischen Linien können durch das 9-Intersection-Modell beschrieben werden. In [23] beschreibt Egenhofer die Beziehungen zweier „komplexer“ Linien. Der Begriff „komplex“ wird dort jedoch anders verwendet, als in der hier vorliegenden Arbeit. Es wird von Egenhofer gefordert, daß die Linien zusammenhängend sind, und keine Kreise bilden

dürfen. Solchen Einschränkungen unterliegen Linien im hier vorgestellten Modell nicht. Aufgrund der Tatsache, daß es sich hier um allgemeinere Linien handelt, gibt es zusätzliche (zur Arbeit von Egenhofer) topologische Beziehungen zwischen den Linien. Um aus den 512 Matrizen diejenigen herauszufiltern, die realisierbare topologische Beziehungen beschreiben, können folgende Bedingungen angegeben werden.

Bed. 1 Es gibt einen Schnitt der beiden Äußeren.

Begründung:

Dies resultiert aus der Tatsache, daß die 2-dimensionale Grundfläche nie durch endlich viele eindimensionale Gebilde endlicher Länge abgedeckt werden kann.

Bed. 2 Die Grenze kann nie Inneres oder Äußeres abdecken, d.h. es gibt einen Schnitt Inneres/Inneres oder Inneres/Äußeres und umgekehrt.

Begründung:

Dies läßt sich durch die unterschiedliche Dimension von Grenze und Innerem/Äußeren begründen.


Bed. 3 Schneidet die Grenze einer Linie (also ein Endpunkt der Linie) das Innere einer anderen Linie, so schneidet auch ihr Äußeres das Innere der anderen Linie.


Begründung:


Am Endpunkt einer Linie startet/endet genau eine Kurve der Linie. Das Innere einer Kurve kann als Startpunkt zweier unterschiedlicher Kurven angesehen werden, indem das Intervall der Kurve geteilt wird.


Man betrachte nun den Schnitt $\partial L1 \cap L2^\circ$. Der Schnittpunkt sei mit S bezeichnet. Dann startet in P genau eine Kurve aus $L1$ und mindestens 2 Kurven aus $L2$. Eine dieser Kurven von $L2$ muß daher im Äußeren von $L1$ liegen.

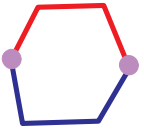
Bed. 4 Schneidet die Grenze das Äußere, so schneidet auch das Innere das Äußere. Für jeden Punkt des Äußeren, läßt sich eine ϵ -Umgebung finden, die sich vollständig im Äußeren befindet. In jeder Umgebung eines Endpunkts hingegen müssen sich innere Punkte der Linie befinden, da diese dort startet. Diese schneiden somit das Äußere der anderen Linie.

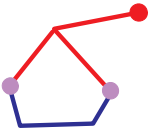
Matrix Nr. 1	
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	

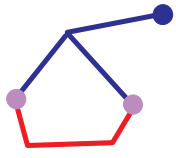
Matrix Nr. 2	
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	

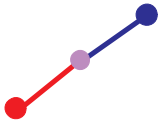
Matrix Nr. 3	
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	

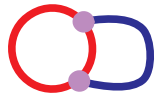
Matrix Nr. 4	
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	

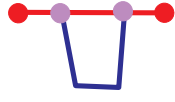
Matrix Nr. 5	
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	


Matrix Nr. 6	
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	

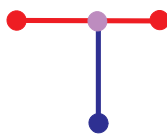
Matrix Nr. 7	
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	

Matrix Nr. 8	
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	

Matrix Nr. 9	
$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	

Matrix Nr. 10	
$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	

Matrix Nr. 11	
$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	

Matrix Nr. 12	
$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	

Matrix Nr. 13

$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 14

$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 15

$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 16

$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 17

$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 18

$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 19

$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 20

$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 21

$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 22

$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 23

$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 24

$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 25

$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 26

$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 27

$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 28

$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 29

$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 30

$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 31

$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 32

$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 33

$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 34


$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	
---	--


Matrix Nr. 35


$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
---	--


Matrix Nr. 36


$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	
---	--


Matrix Nr. 37	
$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	


Matrix Nr. 38	
$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	


Matrix Nr. 39	
$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	

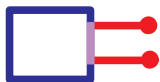
Matrix Nr. 40	
$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	


Matrix Nr. 41	
$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	


Matrix Nr. 42	
$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	

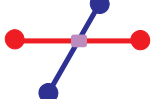
Matrix Nr. 43	
$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	


Matrix Nr. 44	
$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	


Matrix Nr. 45	
$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	

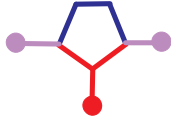
Matrix Nr. 46	
$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$	


Matrix Nr. 47	
$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	

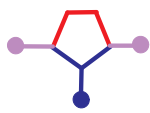
Matrix Nr. 48	
$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	

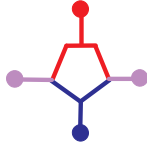
Matrix Nr. 49	
$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	

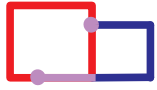
Matrix Nr. 50	
$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	

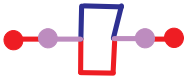
Matrix Nr. 51	
$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	

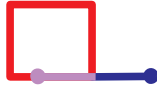
Matrix Nr. 52	
$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$	

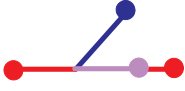
Matrix Nr. 53	
$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	

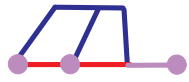
Matrix Nr. 54	
$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	

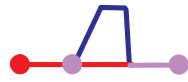
Matrix Nr. 55	
$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	

Matrix Nr. 56	
$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	

Matrix Nr. 57	
$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	

Matrix Nr. 58	
$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	

Matrix Nr. 59	
$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	

Matrix Nr. 60	
$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	

Matrix Nr. 61

$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 62

$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 63

$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 64

$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 65

$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 66

$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 67

$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 68

$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 69

$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 70

$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 71

$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 72

$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$	
---	--

Matrix Nr. 73	
$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	
Matrix Nr. 74	
$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 75	
$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	
Matrix Nr. 76	
$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 77	
$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	
Matrix Nr. 78	
$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 79	
$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	
Matrix Nr. 80	
$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 81	
$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	
Matrix Nr. 82	
$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	

Abb. 15 Topologische Beziehungen zwischen zwei Linien

Es verbleibt auch noch die Möglichkeit, die Dimension der Schnittmenge mit in die Betrachtungen einzubeziehen. Dadurch könnte man eine „echte“ *overlap*-Beziehung von einer *cross*-Beziehung unterscheiden, bei der sich die Inneren nur in einzelnen Punkten schneiden. In dieser Arbeit wird darauf jedoch nicht näher eingegangen.

5.3 Topologische Beziehungen zwischen Regionen

Als Bedingungen für Regionen können angegeben werden (vgl.[44]):

Bed.1 Jeder Teil einer Region hat einen Schnitt mit mindestens einem Teil der anderen Region.

Begründung:

Im Gegensatz zu Punkten und Linien kann bei einer Region keiner dieser Teile leer sein: Da nur nichtleere Regionen betrachtet werden und eine reguläre Punktmenge nicht nur aus einer Grenze bestehen kann, ist das Innere nicht leer. Jedes Innere hat eine eindeutige

Abgrenzung, die nichtleere Grenze (Jordankurventheorem). Da das betrachtete Universum unendlich ist, Regionen jedoch beschränkt sind, ist auch das Äußere einer Region nicht leer. Die Vereinigung der drei Teile ergibt die gesamte Grundmenge. Da jeder Teil in der Grundmenge liegt, gibt es auch einen Schnitt.

Bed.2 Das Innere einer Region hat einen Schnitt mit dem Inneren oder dem Äußeren der anderen Region.

Begründung:

Nach Bedingung 1 hat das Innere einer Region einen Schnitt mit mind. 1 Teil der anderen Region. Würde Bed. 2 nicht gelten, so gäbe es nur einen Schnitt des Inneren mit der Grenze. Dies würde jedoch bedeuten, daß die Grenze der anderen Region das Innere vollständig abdeckt. Dies führt jedoch aufgrund des Dimensionsunterschiedes zu einem Widerspruch.

Bed.3 Der Schnitt der beiden Äußeren ist nicht leer.

Begründung:

Die Grundmenge wird hier als unbegrenzt angenommen, wogegen die Regionen selbst begrenzt sind.

Bed.4 Sind die Grenzen beider Regionen gleich, so sind die Regionen gleich.

Begründung:

Scharfe Regionen werden eindeutig durch ihre Grenzen beschrieben (Jordankurventheorem).

Bed.5 Schneidet die Grenze einer Region das Innere der anderen Region, so schneidet sowohl ihr Inneres als auch ihr Äußeres das Innere der anderen Region.

Begründung:

In jeder ϵ -Umgebung der Grenze von A befinden sich Punkte des Inneren und des Äußeren von A. Für jeden Punkt des Inneren einer Menge B gibt es eine ϵ -Umgebung, in der nur Punkte des Inneren von B existieren. Betrachtet man also einen Schnittpunkt der Grenze von A mit dem Inneren von B, läßt sich eine Umgebung finden, in der sich nur Punkte des Inneren von B und Punkte des Inneren und des Äußeren von A befinden. Daraus ergibt sich die Behauptung.

Bed.6 Schneidet die Grenze einer Region das Äußere der anderen Region, so schneidet auch ihr Inneres sowie ihr Äußeres das Äußere der anderen Region. (Begründung analog zu Bed. 5)

Bed.7 Es gibt stets einen Schnitt beider Grenzen oder einen Schnitt Grenze/Äußeres.

Begründung:

Annahme dies gilt nicht, dann schneidet jede Grenze das Innere der anderen Region (vgl. Bed. 1). Man wählt nun eine Koordinate aus diesem Schnitt. (Annahme $K \in A^\circ \cap \partial B$) Dann konstruiert man von dieser Koordinate K aus einen Strahl S in beliebige Richtung. Dieser Strahl stößt auf die Grenze von A, da ansonsten wäre die Komponente in der sich S befindet unbegrenzt *widerspruch*. Der Schnittpunkt von Strahl und Grenze liegt nun entweder im Äußeren, auf der Grenze oder im Inneren von B. Die ersten beiden Fälle können nach Vorbedingung nicht gelten, demnach liegt der gefundene Schnittpunkt im Inneren von B. Dies ist nun wieder die gleiche Situation wie oben, nur daß die Objekte A und B ihre Rollen getauscht haben. S wird nun weiter verlängert und trifft auf die Grenze von B, die im Inneren von A liegen muß. Der Strahl kann beliebig verlängert werden und trifft alternierend auf einen Punkte der Grenze oder des Inneren von A (bzw. B). Das bedeutet jedoch, daß A (B) unbegrenzt ist *widerspruch*.

Bed.8 Schneiden sich die beiden Inneren, so gibt es auch einen Schnitt Inneres/Grenze oder Grenze/Grenze :

Begründung :

Man betrachte die beiden Faces, bei denen dieser Schnitt auftritt. Wenn es keinen Schnitt beider Grenzen gibt, so liegt ein Face vollständig im Inneren des anderen Face. Die Grenze des Inneren Faces muß dann aber zwangsläufig das Innere des anderen Face schneiden.

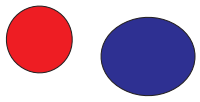
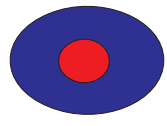
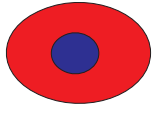
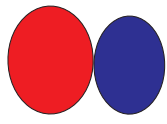
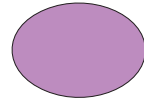
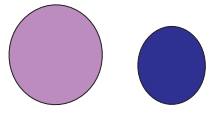
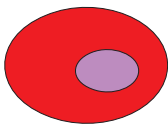
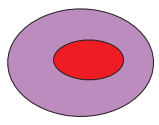
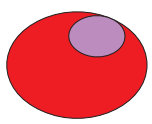
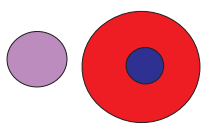
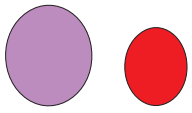
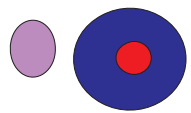
Bed.9 Schneidet das Innere einer Region das Äußere einer anderen Region, so schneiden sich beide Grenzen oder das Innere schneidet die Grenze oder die Grenze schneidet das Äußere der anderen Region.

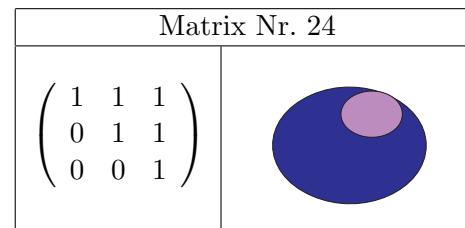
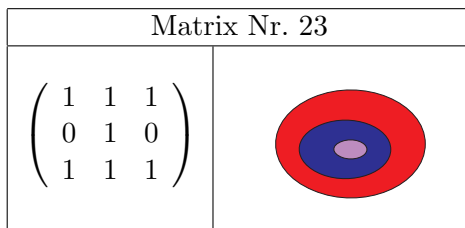
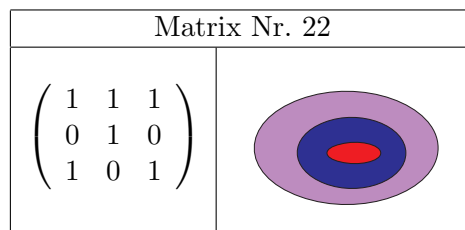
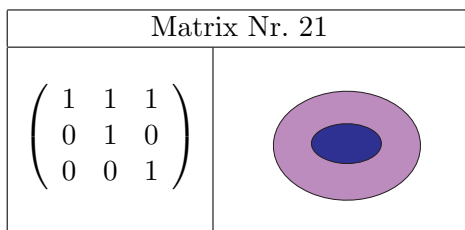
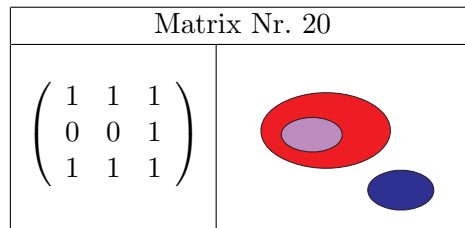
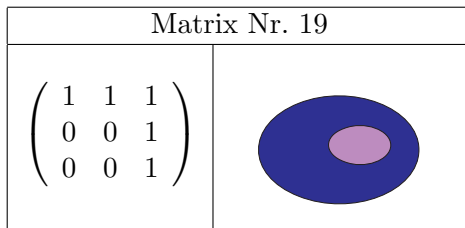
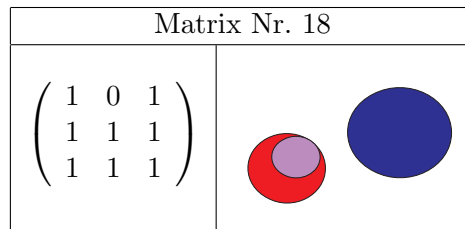
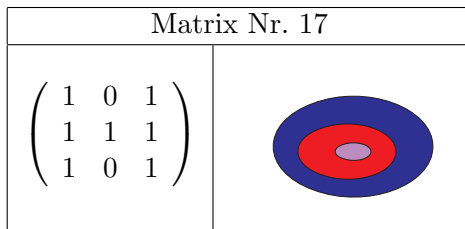
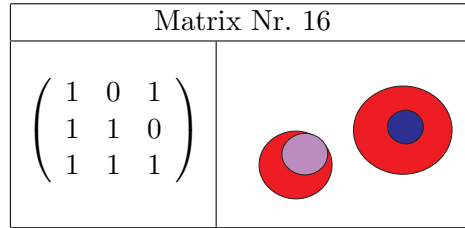
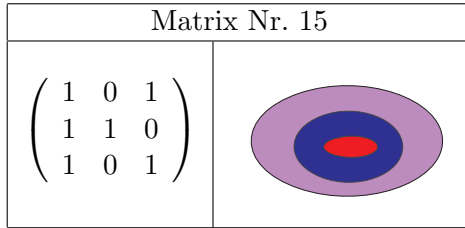
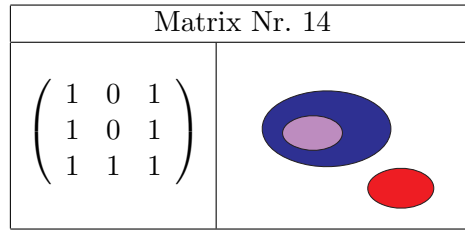
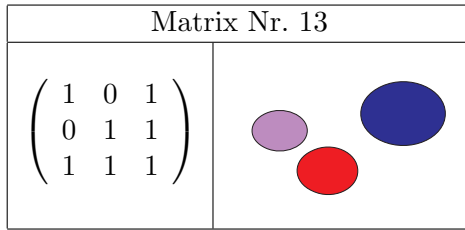
Begründung:

Ist der Schnitt des Inneren mit dem Äußeren der anderen Region nicht leer, so gibt es folgende Möglichkeiten für jede Komponente, welche den Schnitt verursacht :

- Das Innere der Komponente liegt teilweise im Inneren und teilweise im Äußeren der anderen Region. Dann verläuft jedoch die Grenze der anderen Region durch das Innere. \Rightarrow Schnitt Inneres/Grenze.
- Das Innere der Komponente liegt vollständig im Äußeren der anderen Region.
 - Die Grenze der Komponente liegt (teilweise) im Äußeren der anderen Region. \Rightarrow Schnitt Grenze/Äußeres
 - Die Grenze der Komponente liegt auf der Grenze der anderen Region \Rightarrow Schnitt Grenze/Grenze
 - die Grenze kann nicht durch das Innere der anderen Region verlaufen, da sich sonst das Innere nicht vollständig im Äußeren befindet.
- Die Komponente liegt vollständig (incl. Grenze) im Äußeren der anderen Region. Dann liegt auch die Grenze der Komponente im Äußeren und diese schneidet somit das Äußere der anderen Region.

Durch diese Bedingungen bleiben von den $2^9 = 512$ insgesamt möglichen Matrizen 33 übrig, welche bedeutungsvolle Lagebeziehungen beschreiben:

Matrix Nr. 1	$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 2	$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 3	$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	
Matrix Nr. 4	$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 5	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	
Matrix Nr. 6	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 7	$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 8	$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	
Matrix Nr. 9	$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 10	$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 11	$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$	
Matrix Nr. 12	$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	



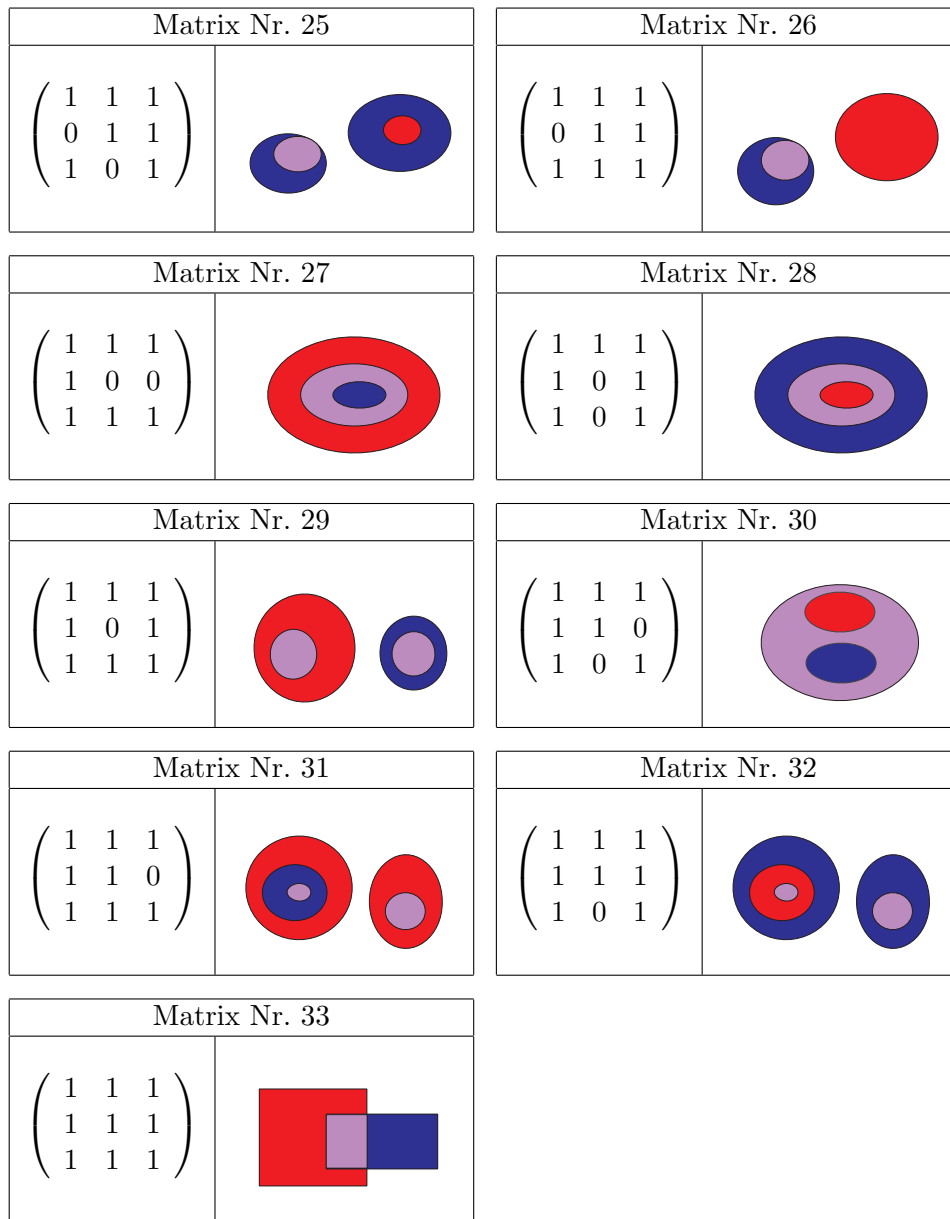


Abb. 16 Beziehungen zwischen zwei Regionen

5.4 Topologische Beziehungen zwischen unterschiedlichen Objekten

5.4.1 Beziehungen zwischen Punkt und Linie

Auch die Beziehung zwischen einem Punkt und einer Linie kann durch das 9-Intersection-Modell dargestellt werden. Es ist klar, daß einige Beziehungen nicht auftreten können, so ist es z.B. nicht möglich, daß eine Linie in einem Punkt enthalten ist. Der umgekehrte Fall ist selbstverständlich realisierbar. So ergibt sich je nach Stellung der „Matrixparameter“ eine unterschiedliche Menge realisierbarer Matrizen. Diesem Problem wird hier damit begegnet, daß der Punkt stets an erster und die Linie an zweiter Stelle stehen muß. Auch hier können Bedingungen gefunden werden, die nichtrealisierbare Matrizen ausschließen:

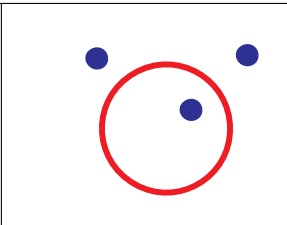
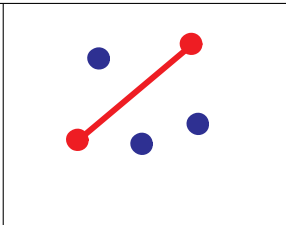
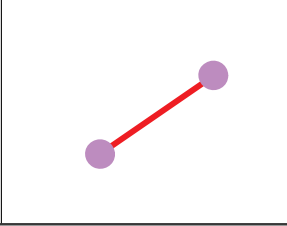
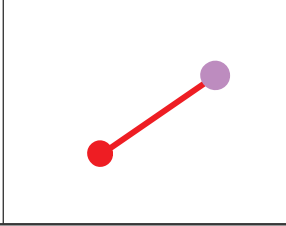
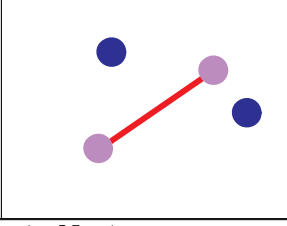
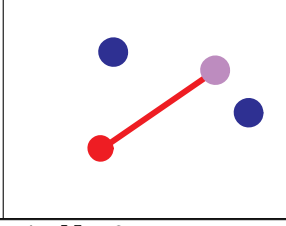
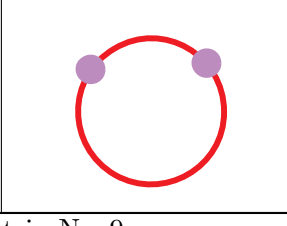
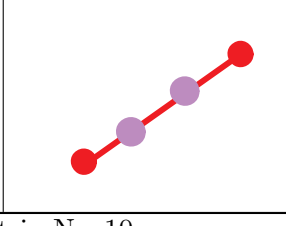
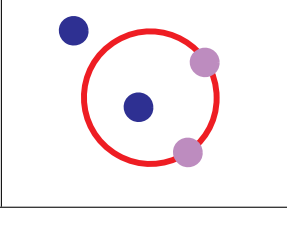
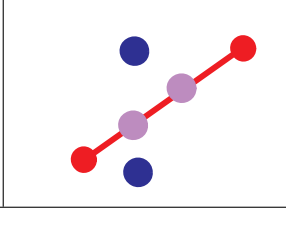
Bed.1 Da die Grenze eines Punktes leer ist, befinden sich in der mittleren Zeile der Matrix nur 0-Einträge.

Bed.2 Der Schnitt des Inneren der Linien und des Äußeren des Punktes kann nicht leer sein, da sonst der Punkt die gesamte Linie abdecken würde, was jedoch aufgrund der geringeren Dimension des Punktes nicht möglich ist.

Bed. 3 Der Schnitt beider Äußeren ist nicht leer, da nicht das gesamte Universum von einer Linie oder einem Punkt abgedeckt werden kann.

Bed.4 Da das Innere eines Punktes nicht leer ist, wird von diesem mindestens ein Teil der Linie geschnitten.

Es verbleiben die folgenden Matrizen, für welche Bilder als Beispiel für eine mögliche Realisierung angegeben sind:

Matrix Nr. 1		Matrix Nr. 2	
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$		$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 3		Matrix Nr. 4	
$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$		$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 5		Matrix Nr. 6	
$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$		$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 7		Matrix Nr. 8	
$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 9		Matrix Nr. 10	
$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	

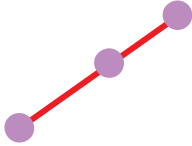
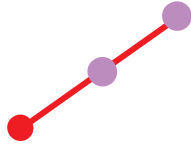
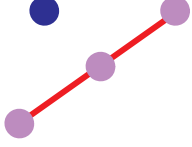
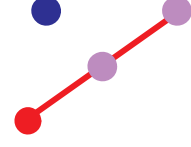
Matrix Nr. 11		Matrix Nr. 12	
$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 13		Matrix Nr. 14	
$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	

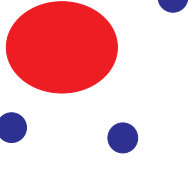
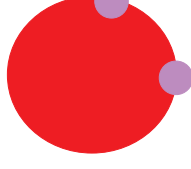
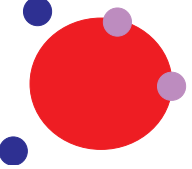
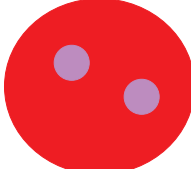
Abb. 17 Topologische Beziehungen zwischen Linie und Punkt

5.4.2 Beziehungen zwischen Punkt und Region

Um die Eindeutigkeit der Menge der gültigen Matrizen bei topologischen Beziehungen von Punkt und Region zu gewährleisten, wird wieder der Parameter mit der geringeren Dimension an erster Stelle vorausgesetzt. Hier lassen sich die folgenden Bedingungen formulieren:

- Bed. 1** Der Schnitt der Äußeren ist nicht leer (mit der gleichen Begründung wie in den anderen Fällen).
- Bed. 2** Da die Grenze des Punktes leer ist, kann diese keinen Schnitt mit einem der Regionsteile besitzen.
- Bed. 3** Grenze und Inneres der Region schneiden das Äußere des Punktes, da der Punkt diese Teile aufgrund der niedrigeren Dimension nicht abdecken kann.
- Bed. 4** Das Innere des Punktes schneidet wenigstens einen Teil der Region. Das ergibt sich daraus, daß der Punkt nicht leer ist und die Vereinigung der drei Regionsteile das gesamte Universum bildet.

Es sind die folgenden Beziehungen möglich:

Matrix Nr. 1		Matrix Nr. 2	
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$		$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 3		Matrix Nr. 4	
$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	

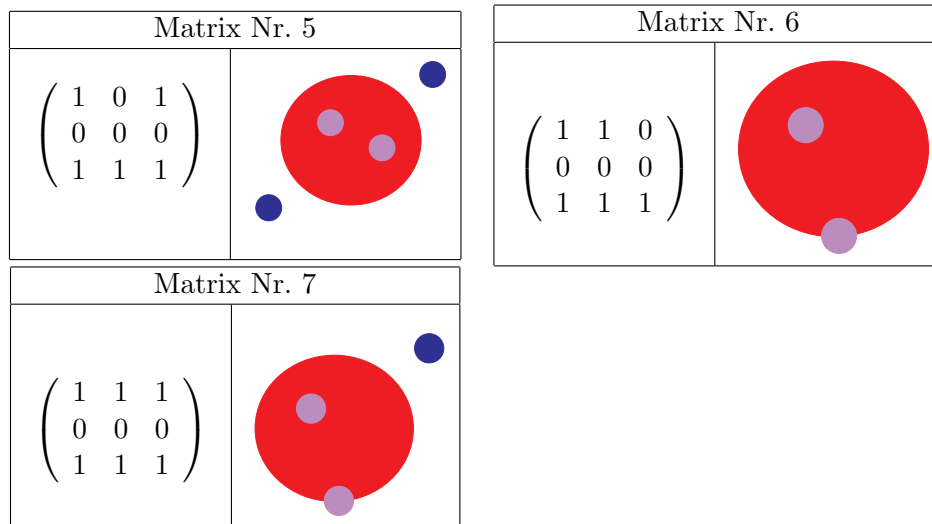


Abb. 18 Beziehungen zwischen Punkt und Region

5.4.3 Beziehungen zwischen Linie und Region

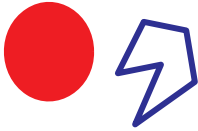
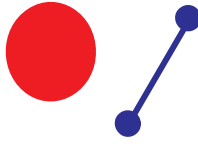
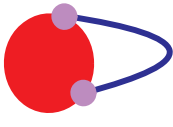
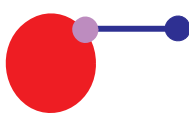

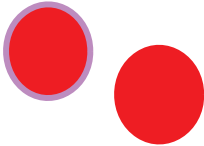
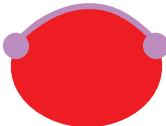
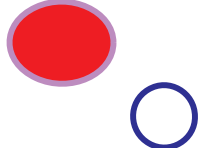
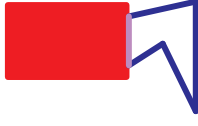
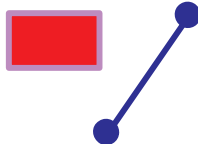
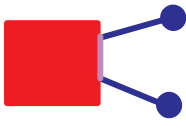
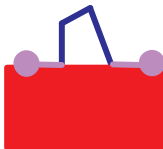
Auch bei den topologischen Beziehungen zwischen Linien und Regionen hängt die Menge der realisierbaren Matrizen von der Stellung der „Matrixparameter“ ab. Es wird hierbei gefordert, daß die Region als zweiter Parameter verwendet wird.

Wie sonst auch, kann die Menge der Matrizen, für die es eine topologische Beziehung zwischen den Objekten gibt, durch eine Reihe von geometrischen Bedingungen eingeschränkt werden.



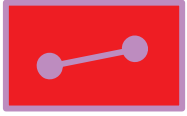


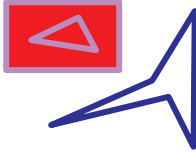

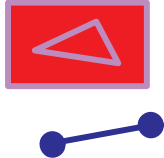
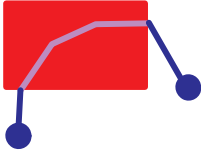
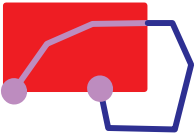
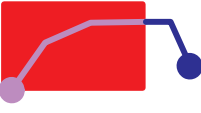
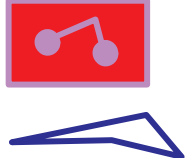
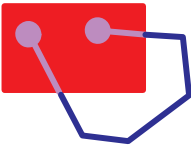
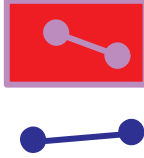
- Bed. 1** Der Schnitt beider Äußeren ist nicht leer. Dies folgt aus der Unendlichkeit des Universums und der Voraussetzung, daß Regionen begrenzt sind sowie der Tatsache, daß durch eine Linie nicht das Universum abgedeckt werden kann.
- Bed. 2** Das Innere der Region schneidet das Äußere der Linie, weil eine Linie durch den Dimensionsunterschied nicht das gesamte Innere der Region abdecken kann.
- Bed. 3** Die Grenze der Region wird vom Äußeren oder vom Inneren der Linie geschnitten. Da die Grenze der Region nicht leer ist, muß sie einen der drei Teile der Linie schneiden. Aufgrund des Dimensionsunterschiedes reicht die Grenze der Linie allein nicht aus, um die gesamte Grenze der Region abzudecken. Daher muß noch einer der beiden anderen Teile geschnitten werden.
- Bed. 4** Das Innere der Linie schneidet mindestens einen Teil der Region, da die Linie nicht leer ist und im Universum liegt, welches die Vereinigung der drei Regionsteile darstellt.
- Bed. 5** Schneidet die Grenze einer Linie das Innere einer Region, so schneidet auch ihr Inneres diese Region. Da die Grenze einer Linie den Startpunkt einer stetigen Kurve darstellt, liegen in jeder ϵ -Umgebung dieser Grenze unendlich viele Punkte des Inneren der Linie. Im Inneren einer Region jedoch läßt sich eine ϵ -Umgebung finden, in der nur Punkte des Inneren der Region liegen. Einige dieser Punkte müssen so zwangsweise das Innere der Linie schneiden.
- Bed. 6** Schneidet die Grenze einer Linie das Äußere einer Region, so schneidet auch ihr Inneres das Äußere der Region. Die Begründung ist analog zu der von Bed. 5 zu führen.

Bed. 7 Schneidet die Grenze einer Linie die Grenze einer Region, so schneidet auch ihr Äußeres die Grenze der Region. Die Grenze einer Region ist eine Linie, die nur aus Punkten des Inneren besteht. Daher liegt der gleiche Fall wie in Bed. 3 bei den topologischen Beziehungen zwischen 2 Linien vor.

Es verbleiben die folgenden Matrizen:

<p>Matrix Nr. 1</p> $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$		<p>Matrix Nr. 2</p> $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	
<p>Matrix Nr. 3</p> $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$		<p>Matrix Nr. 4</p> $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	
<p>Matrix Nr. 5</p> $\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$		<p>Matrix Nr. 6</p> $\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
<p>Matrix Nr. 7</p> $\begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$		<p>Matrix Nr. 8</p> $\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	
<p>Matrix Nr. 9</p> $\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$		<p>Matrix Nr. 10</p> $\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	
<p>Matrix Nr. 11</p> $\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$		<p>Matrix Nr. 12</p> $\begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	

Matrix Nr. 13		Matrix Nr. 14	
$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 15		Matrix Nr. 16	
$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 17		Matrix Nr. 18	
$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 19		Matrix Nr. 20	
$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 21		Matrix Nr. 22	
$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 23		Matrix Nr. 24	
$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 25		Matrix Nr. 26	
$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	

Matrix Nr. 27		Matrix Nr. 28	
$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 29		Matrix Nr. 30	
$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 31		Matrix Nr. 32	
$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	
Matrix Nr. 33		Matrix Nr. 34	
$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	
Matrix Nr. 35		Matrix Nr. 36	
$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 37		Matrix Nr. 38	
$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	
Matrix Nr. 39		Matrix Nr. 40	
$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	

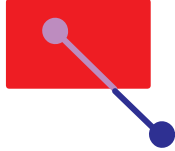
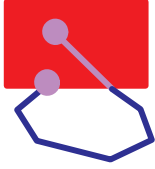
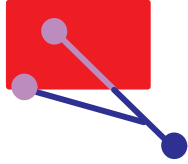
Matrix Nr. 41		Matrix Nr. 42	
$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$		$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	
Matrix Nr. 43			
$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$			

Abb. 19 Beziehungen zwischen Linie und Region

5.5 Cluster und topologische Prädikate

Durch die oben angeführten Matrizen lassen sich einige topologische Beziehungen unterscheiden. Leider ist die Anzahl der realisierbaren Matrizen zwischen einigen Objekten sehr groß. Um eine benutzbare Anzahl topologischer Prädikate zu erhalten, ist man daher gezwungen, Cluster zu bilden (vgl. [17]). Hier wird eine Clusterung vorgeschlagen, die insgesamt 8 mögliche topologische Beziehungen unterscheiden kann. Da z.B. nur 5 Matrizen die topologischen Beziehungen von Punkten unterscheiden, ist klar, daß einige topologische Prädikate zwischen bestimmten Objekten nicht auftreten können. Um die einzelnen Cluster zu beschreiben, wird festgelegt, welche Eigenschaften gelten müssen.

Zwei geometrische Objekte gelten als disjunkt, wenn sich die Teile eines Objekts jeweils höchstens mit dem Äußeren des zweiten Objekts schneiden. Es ergibt sich:

$$disjoint(A, B) \stackrel{\text{def}}{=} A^\circ \cap B^\circ = \emptyset \wedge A^\circ \cap \partial B = \emptyset \wedge \partial A \cap B^\circ = \emptyset \wedge \partial A \cap \partial B = \emptyset$$

Zwei Objekte treffen sich, wenn sich beide Inneren nicht schneiden, aber das Innere oder die Grenze eines Objekts jeweils von der Grenze des anderen Objekts geschnitten werden:

$$meet(A, B) \stackrel{\text{def}}{=} A^\circ \cap B^\circ = \emptyset \wedge (A^\circ \cap \partial B \neq \emptyset \vee \partial A \cap \partial B \neq \emptyset \vee \partial A \cap B^\circ \neq \emptyset)$$

Befindet sich ein Objekt innerhalb eines anderen Objekts, so schneiden sich beide Inneren. Weiterhin hat das Innere Objekt keine echten Teile im Äußeren des anderen Objekts; jedoch muß das beinhaltende Objekt teilweise mit seinem Inneren im Äußeren des anderen Objekts liegen. Außerdem darf es keinen Schnitt beider Grenzen geben.

$$inside(A, B) \stackrel{\text{def}}{=} A^\circ \cap B^\circ \neq \emptyset \wedge A^\circ \cap B^{-1} = \emptyset \wedge A^{-1} \cap B^\circ \neq \emptyset \wedge \partial A \cap \partial B = \emptyset$$

Die zu innerhalb symmetrischen Beziehungen werden durch das enthält-Prädikat beschrieben.

$$contains(A, B) \stackrel{\text{def}}{=} A^\circ \cap B^\circ \neq \emptyset \wedge A^{-1} \cap B^\circ = \emptyset \wedge A^\circ \cap B^{-1} \neq \emptyset \wedge \partial A \cap \partial B = \emptyset$$

Ein Objekt wird von einem anderen Objekt überdeckt, wenn es sich im Inneren dieses Objekts befindet und es zusätzlich einen Grenzschnitt gibt.

$$\text{coveredBy}(A, B) \stackrel{\text{def}}{=} A^\circ \cap B^\circ \neq \emptyset \wedge A^\circ \cap B^{-1} = \emptyset \wedge A^{-1} \cap B^\circ \neq \emptyset \wedge \partial A \cap \partial B \neq \emptyset$$

Das symmetrische Prädikat drückt aus, das Objekt A das Objekt B überdeckt.

$$\text{covers}(A, B) \stackrel{\text{def}}{=} A^\circ \cap B^\circ \neq \emptyset \wedge A^{-1} \cap B^\circ = \emptyset \wedge A^\circ \cap B^{-1} \neq \emptyset \wedge \partial A \cap \partial B \neq \emptyset$$

Zwei Objekte sind gleich, wenn sich höchstens die einander entsprechenden Teile von ihnen schneiden.

$$\text{equals}(A, B) \stackrel{\text{def}}{=} A^\circ \cap \partial B = \emptyset \wedge A^\circ \cap B^{-1} = \emptyset \wedge \partial A \cap B^\circ = \emptyset \wedge \partial A \cap B^{-1} = \emptyset \wedge A^{-1} \cap B^\circ = \emptyset \wedge A^{-1} \cap \partial B = \emptyset$$

Zwei Objekte überlappen sich, wenn Teile des Inneren eines Objekts jeweils im Inneren und Äußeren des anderen Objekts liegen.

$$\text{overlap}(A, B) \stackrel{\text{def}}{=} A^\circ \cap B^\circ \neq \emptyset \wedge A^\circ \cap B^{-1} \neq \emptyset \wedge A^{-1} \cap B^\circ \neq \emptyset$$

Es läßt sich überprüfen, daß sich diese „Schablonen“ gegenseitig ausschließen.

Diese Clusterung kann für eine effiziente Implementierung verwendet werden. Bei einer Datenbankabfrage, ob sich zwei Objekte in der topologischen Beziehung X befinden (X bezeichne einen der hier vorgestellten Cluster), ist es nicht mehr nötig, alle 9 Schnitte dieser Objekte zu berechnen. Es genügt, daß die durch die Cluster vorgegebenen Schnitte geprüft werden. Beispielsweise müssen beim Test auf die *overlap*-Beziehung nur 3 der 9 Schnitte geprüft werden.

Die realisierbaren Matrizen der letzten Abschnitte lassen sich wie folgt in die Cluster einordnen:

Cluster	Punkt/Punkt	Linie/Linie	Region/Region
disjoint	1	1 2 3 4	1
meet	/	5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32	1 3 4
covers	/	49 52 69 72	10 20
coveredBy	/	37 38 41 42	6 8
inside	3	34 35 39 40	7
contains	4	43 46 63 66	17
overlap	5	44 45 47 48 50 51 53 54 55 56 57 58 59 60 61 62 64 65 67 68 70 71 73 74 75 76 77 78 79 80	9 11 12 13 14 15 16 18 19 21 22 23 24 25 26 27 28
equals	2	81 82 33 36	5

Cluster	Punkt/Linie	Punkt/Region	Linie/Region
disjoint	1 2	1	1 2
meet	3 4 5 6	2 3	3 4 5 6 7 8 9 10 11 12 13
covers	/	/	/
coveredBy	/	/	15 17 28 31
inside	7 8 11 12	4 6	14 16 26 27 29 30
contains	/	/	/
overlap	9 10 13 14	5 7	18 19 20 21 22 23 24 25 32 33 34 35 36 37 38 39 40 41 42 43
equals	/	/	/

Kapitel 6

Topologische Beziehungen auf komplexen unscharfen geometrischen Objekten

Das vorherige Kapitel beschäftigte sich mit den topologischen Beziehungen von scharfen Objekten. Da sich diese Arbeit jedoch mit unscharfen Objekten befaßt, müssen auch die Beziehungen zwischen solchen Objekten beschrieben werden können.

Ein unscharfes Objekt wurde als Erweiterung eines scharfen Objekts definiert. Es gibt zwei Erweiterungen, die Zugehörigkeitsfunktion sowie den Skalierungsfaktor. Dabei kann der Skalierungsfaktor auch als Teil der Zugehörigkeitsfunktion angesehen werden (durch Multiplikation mit den Graden).

Eine Beziehung zwischen zwei unscharfen Objekten läßt sich beschreiben, indem man die topologische Beziehung der Grundmengen und die Beziehungen der Zugehörigkeitsfunktion angibt. Über die Topologie scharfer Objekte wurde im letzten Kapitel ausführlich diskutiert. Es fehlt also noch eine Darstellung der Beziehungen zwischen den Zugehörigkeitsfunktionen. Diese können durch eine Menge von Prädikaten beschrieben werden. Es sei die Beziehung der Funktionen der beiden Objekte $A = (B_A, F_A, SF_A)$ und $B = (B_B, F_B, SF_B)$ zu untersuchen.

Um die Beziehungen zwischen Zugehörigkeitsfunktionen auszudrücken, sind folgende Fragestellungen wichtig:

- Ist der Zugehörigkeitsgrad eines Objekts an einer Stelle größer als Null oder ist er gleich Null?
Dies ist deshalb interessant, weil ein Zugehörigkeitsgrad von Null eine Sonderstellung einnimmt, da hiermit die Nichtzugehörigkeit zum Objekt ausgedrückt wird.
- Ist ein Zugehörigkeitsgrad eines Objekts an einer Stelle größer, kleiner oder gleich dem Grad des anderen Objekts an dieser Stelle? Damit sind alle möglichen Beziehungen zwischen zwei Zahlen (den Zugehörigkeitsgraden) abgedeckt.

Die Kombination dieser Fragestellungen führt zu folgenden (sich gegenseitig ausschließenden) Prädikaten (Q1-Q6) für jeden Punkt P des \mathbb{R}^2 .

- beide Grade sind 0
 $Q1(P) \Leftrightarrow F_A(P) = F_B(P) = 0$
- ein Grad ist Null, der andere ist größer als Null
 $Q2(P) \Leftrightarrow F_A(P) > F_B(P) = 0$
 $Q3(P) \Leftrightarrow F_B(B) > F_A(P) = 0$
- beide Grade sind größer als Null, wobei noch weitere Unterscheidungen getroffen werden können

- beide Werte sind gleich
 $Q4(P) \Leftrightarrow F_A(P) = F_B(P) > 0$
- ein Wert ist größer als der andere
 $Q5(P) \Leftrightarrow F_A(P) > F_B(P) > 0$
 $Q6(P) \Leftrightarrow F_B(P) > F_A(P) > 0$

Nun kann nicht für jeden Punkt des \mathbb{R}^2 angegeben werden, welches der obigen Prädikate erfüllt ist. Um eine topologische Beziehung anzugeben, wird daher geprüft, ob überhaupt ein Punkt existiert, an denen das betrachtete Prädikat erfüllt ist. Würde man für diese Existenzprüfung den gesamten \mathbb{R}^2 betrachten, hätte man nur ein sehr grobes Ergebnis. Beispielsweise wäre $Q1$ immer erfüllt, da beide Objekte in ihrem Äußeren den Grad Null aufweisen. Daher wird solch eine Existenzprüfung in einzelnen Partitionen des \mathbb{R}^2 durchgeführt. Die Partitionen sind hierbei durch das 9-Intersection-Modell gegeben. Als Eintrag wird jeweils ein 6-Tupel verwendet, welches an jeder seiner Stellen aussagt, ob es einen Punkt in der jeweils betrachteten Menge gibt, an der das entsprechende Prädikat erfüllt ist. Dies entspricht dem Testen auf Leere der entsprechenden Menge. Dies führt zur folgenden Matrix:

$$fT[A, B] = \left(\begin{array}{c|ccc} & B_A^\circ & \partial B_A & B_A^{-1} \\ \hline B_B^\circ & E(B_A^\circ, B_B^\circ) & E(\partial B_A, B_B^\circ) & E(B_A^{-1}, B_B^\circ) \\ \partial B_B & E(B_A^\circ, \partial B_B) & E(\partial B_A, \partial B_B) & E(B_A^{-1}, \partial B_B) \\ B_B^{-1} & E(B_A^\circ, B_B^{-1}) & E(\partial B_A, B_B^{-1}) & E(B_A^{-1}, B_B^{-1}) \end{array} \right)$$

mit:

$$E(X, Y) \Leftrightarrow (\exists P \in (X \cap Y) : P1(P), \exists P \in (X \cap Y) : P2(P), \exists P \in (X \cap Y) : P3(P), \\ \exists P \in (X \cap Y) : P4(P), \exists P \in (X \cap Y) : P5(P), \exists P \in (X \cap Y) : P6(P))$$

Dies entspricht einer Verfeinerung der Matrixeinträge gegenüber der 9-Intersection-Matrix für scharfe Objekte. In der Hauptsache werden die Schnittmengen dieser Matrix noch einmal in verschiedene Teilmengen zerlegt, die dann auf Leere (eben die Existenz eines Punktes) geprüft werden.

Man kann feststellen, daß sich die topologischen Beziehungen der Grundmengen der Objekte A und B vollständig aus der unscharfen 9-Intersection-Matrix ableiten lassen. Dies läßt sich relativ einfach begründen. Die 6 Prädikate decken alle Möglichkeiten ab, wie sich zwei Punkte zueinander verhalten können. Das heißt, jeder Punkt einer beliebigen Menge erfüllt genau eins dieser Prädikate. Gibt es nun einen Punkt in der Menge (also eine 1 in der 9-Intersection-Matrix der Grundmengen), so enthält der Eintrag an dieser Stelle der erweiterten Matrix mindestens eine 1. Auch die Umkehrung gilt, wenn eine 1 an einer beliebigen Stelle eines Eintrags steht, so gibt es in dieser Menge auch mindestens einen Punkt. Ist diese Menge leer, so ist keins dieser Prädikate erfüllt und der Eintrag der Matrix für die unscharfe Beziehung enthält nur Nullen. Auch hier gilt die Umkehrung.

Es gibt theoretisch $(2^6)^9$ mögliche Matrizen, von denen sich durch oben genannte Beziehungen zu den Matrizen der scharfen Grundmengen und den Betrachtungen des letzten Kapitels sehr viele ausschließen lassen.

Es lassen sich jedoch noch weitere Einträge in Abhängigkeit von den Mengen ausschließen.

- Beim Schnitt der beiden Äußeren läßt sich erkennen, daß nur das erste Prädikat erfüllt werden kann, da bei den anderen Prädikaten mindestens ein Wert größer sein muß als Null. Die Zugehörigkeitsfunktionen liefern im Äußeren der Grundmenge nach ihrer Definition jedoch stets 0.
- Innerhalb der Schnittmenge des Äußeren von A und der Grenze von B, können höchstens die Prädikate P1 und P3 erfüllt werden, da in allen anderen Fällen ein Punkt im Äußeren von A existieren müßte, der einen Grad größer als Null hat.

- Analog läßt sich die Schnittmenge der Äußeren von A mit dem Inneren von B betrachten.
- Die Schnitte des Äußeren von B mit dem Inneren oder der Grenze von A sind symmetrisch zu den beiden letzten Fällen. Es können hier nur die Prädikate P1 und P2 erfüllt werden.
- Bei allen anderen Schnittmengen können (abgesehen von Einschränkungen durch die Objekttypen) alle Prädikate erfüllt werden.

Es wird nun untersucht, wieviele unterscheidbare Beziehungen zwischen einzelnen Objekttypen bestehen können.

Um die Anzahl topologischer Beziehungen zwischen zwei Objekten berechnen zu können, wird in Abhängigkeit von den Objekttypen untersucht, welche Einträge innerhalb der einzelnen Schnittmengen möglich sind. Aus diesen Berechnungen und den 9-Intersection-Matrizen der Grundmengen läßt sich dann die Anzahl der möglichen Matrizen für die unscharfen Objekte ermitteln.

Allgemein läßt sich zunächst festhalten, daß jeder Null-Eintrag in der scharfen Matrix auch nur einen einzigen Eintrag in der Matrix der unscharfen topologischen Beziehung zuläßt, nämlich wo jedes der 6 möglichen Prädikate nicht erfüllt ist. Aus einem 1-Eintrag lassen sich theoretisch $2^6 - 1 = 63$ Einträge ableiten. D.h. alle Kombinationen, bis auf den Fall, daß alle Einträge Null sind, wären möglich.

6.1 Beziehungen zwischen zwei unscharfen Punkten

Zwischen zwei Punkten müssen nur die Schnittmengen der Inneren und Äußeren mit- und untereinander untersucht werden, da bei den Grenzschnitten nur Null-Einträge in den Matrizen für scharfe Punkte vorhanden sind.

Der Schnitt beider Äußeren führt nur zu einer einzigen Kombination der Prädikate, nämlich $(1, 0, 0, 0, 0, 0)$. Die Begründung hierfür wurde bereits gegeben.

Der Schnitt der Inneren von A und des Äußeren von B führt bei zwei Punkten ebenfalls nur zu einer möglichen Kombination der Prädikate: $(0, 1, 0, 0, 0, 0)$. Die letzten vier Prädikate können nicht erfüllt werden, da dort B einen Grad größer als Null haben muß, was jedoch im Äußeren nicht möglich ist. Das erste Prädikat kann nicht erfüllt werden, da jede Koordinate eines Punktes nach Definition einen Grad größer als Null haben muß. Somit sind alle anderen Kombinationen ausgeschlossen.

Beim symmetrischen Fall ist bei analoger Begründung nur der Eintrag $(0, 0, 1, 0, 0, 0)$ möglich. Innerhalb der Schnittmenge der beiden Inneren sind zwischen zwei Punkten nur die Prädikate erfüllbar, bei denen beide Grade größer als Null sind. Dies sind die Prädikate P4, P5 und P6. Es sind also bei einer 1 in der Matrix für die scharfen Punkte $2^3 - 1 = 7$ Einträge innerhalb der Matrix für die unscharfen Punkte möglich.

Mit diesem Wissen ausgerüstet, kann man die Anzahl der unterscheidbaren topologischen Beziehungen zwischen zwei unscharfen Punkten bestimmen.

Aus der ersten Matrix der Abb 14 $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$ läßt sich auch nur eine einzige erweiterte Matrix

ableiten, da jeder Eintrag in dieser Matrix einen eindeutigen Eintrag in der erweiterten Matrix fordert. Diese lautet

$$\begin{pmatrix} (0, 0, 0, 0, 0, 0) & (0, 0, 0, 0, 0, 0) & (0, 1, 0, 0, 0, 0) \\ (0, 0, 0, 0, 0, 0) & (0, 0, 0, 0, 0, 0) & (0, 0, 0, 0, 0, 0) \\ (0, 0, 1, 0, 0, 0) & (0, 0, 0, 0, 0, 0) & (1, 0, 0, 0, 0, 0) \end{pmatrix}$$

Die zweite Matrix von Abb 14 $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ kann zu genau 7 erweiterten Matrizen führen. Dies

resultiert aus der Tatsache, daß die 1 beim Schnitt der beiden Inneren zu 7 möglichen Einträgen in der erweiterten Matrix führt, wogegen alle anderen Einträge nur eine Möglichkeit offenlassen. Auch die Matrizen 3,4 und 5 führen bei gleicher Begründung zu 7 erweiterten Matrizen. Demnach lassen sich mit der hier entwickelten Erweiterung des 9-Intersection-Modells $1 + 4 \cdot 7 = 29$ topologische Beziehungen zwischen unscharfen Punkten unterscheiden.

6.2 Topologische Beziehungen zwischen unscharfen Linien

Um die Anzahl unterscheidbarer topologischer Beziehungen zwischen zwei Linien bestimmen zu können, werden wieder die erfüllbaren Prädikate innerhalb der einzelnen Schnittmengen untersucht.

Der Schnitt beider Äußeren läßt wieder nur die Möglichkeit $(1, 0, 0, 0, 0, 0)$ offen.

Der Schnitt des Inneren von A mit dem Äußeren von B kann zu den Einträgen $(0, 1, 0, 0, 0, 0)$ und $(1, 1, 0, 0, 0, 0)$ führen. Das erste Prädikat kann dann erfüllt werden, wenn A in seinem Inneren einzelne Punkte enthält, die den Grad Null besitzen. Das zweite Prädikat ist zwingend erfüllt. Ist die hier betrachtete Schnittmenge nicht leer, so liegen unendlich viele Punkte von A im Äußeren von B. (In jeder Umgebung des Inneren von A liegen unendlich viele Punkte von A und es gibt für jeden Punkt des Äußeren von B eine Umgebung, die sich vollständig im Äußeren befindet.) Wäre das zweite Prädikat nun nicht erfüllt, bliebe nur der Eintrag $(1, 0, 0, 0, 0, 0)$ übrig. Dies würde jedoch heißen, daß alle Inneren Punkte von A, die im Äußeren von B liegen den Grad Null besitzen. Dies ist jedoch nicht möglich, da das Innere von A nur endlich viele solcher Punkte enthalten darf. Die hinteren vier Prädikate können nicht erfüllt werden, da jeder Punkt des Äußeren von B den Grad 0 besitzt.

Mit einer analogen Begründung sind beim symmetrischen Schnitt nur die Einträge $(0, 0, 1, 0, 0, 0)$ und $(1, 0, 1, 0, 0, 0)$ möglich.

Beim Schnitt Grenze von A mit Äußeren von B sind $(1, 0, 0, 0, 0, 0)$, $(0, 1, 0, 0, 0, 0)$ und $(1, 1, 0, 0, 0, 0)$ die möglichen Einträge in der erweiterten Matrix. Da die Grade von B im Äußeren stets Null sind, fallen die Prädikate P3-P6 aus der Menge erfüllbarer Prädikate heraus. Die Erfüllbarkeit der anderen beiden Prädikate hängt nur von den Graden der Grenze von A im Äußeren von B ab, wobei keine weiteren Einschränkungen gemacht werden können.

Die möglichen Einträge des symmetrischen Schnitts lauten demnach $(1, 0, 0, 0, 0, 0)$, $(0, 0, 1, 0, 0, 0)$ und $(1, 0, 1, 0, 0, 0)$.

Beim Schnitt der beiden Grenzen können keine weiteren Einschränkungen gemacht werden. Daher kann ein Grenzschnitt auf bis zu $2^6 - 1 = 63$ Einträge in der erw. Matrix für die topologischen Beziehungen führen.

Genauso verhält es sich mit den Schnitten der Inneren mit den Grenzen und den Inneren untereinander.

Die folgende Matrix gibt die Anzahlen der möglichen Einträge in der erweiterten Matrix wieder, die aus einem 1-Eintrag der 9-Intersection-Matrix der beiden Grundmengen resultieren:

$$N_{ij} = \begin{pmatrix} 63 & 63 & 2 \\ 63 & 63 & 3 \\ 2 & 3 & 1 \end{pmatrix}$$

Da die Zugehörigkeitsfunktionen recht allgemein gehalten sind, gibt es keine Abhängigkeiten zwischen den einzelnen Einträgen außer denen, die bereits durch die Matrizen für die Grundmengen gegeben sind. So läßt sich z.B. aus dem Verlauf beider Funktionen im Inneren nicht auf deren Verlauf im Äußeren schließen, da die Funktionen endlich viele Unstetigkeitsstellen aufweisen dürfen.

Daher läßt sich die Anzahl der aus einer 9-Intersection-Matrix T_{ij} der Grundmengen ableitbaren

erweiterten Matrizen wie folgt berechnen:

$$a(T_{ij}) = \prod_{T_{ij}=1} N_{ij}$$

So gibt es zur Matrix Nr. 1 der Abb.15

$$M_1 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

insgesamt $2 \cdot 2 \cdot 1 = 4$ ableitbare erweiterte Matrizen.

Die Gesamtanzahl der möglichen Matrizen ist durch die Summe aller von den realisierbaren 9-Intersection-Matrizen ableitbaren erweiterten Matrizen gegeben. Es gibt daher insgesamt 1.077.874.624 Matrizen.

6.3 Topologische Beziehungen zwischen unscharfen Regionen

Auch die Anzahl der möglichen Beziehungen zwischen zwei unscharfen Regionen läßt sich berechnen. Hierzu werden wieder die möglichen Kombinationen von Prädikaten der einzelnen Schnittmengen betrachtet.

Beim Schnitt beider Äußeren ist wieder nur die Kombination $(1, 0, 0, 0, 0, 0)$ möglich.

Beim Schnitt der Grenze von A mit dem Äußeren von B sind nur die Prädikate erfüllbar, bei denen die Grade von B Null sind. Da die Grade der Grenze von A keinen Einschränkungen unterliegen (also komplett Null sein können, stets größer als Null sein können und gemischt auftreten können), gibt es die drei Möglichkeiten $(1, 0, 0, 0, 0, 0)$, $(0, 1, 0, 0, 0, 0)$ und $(1, 1, 0, 0, 0, 0)$.

Beim symmetrischen Fall gibt es daher ebenfalls 3 Möglichkeiten, nämlich $(1, 0, 0, 0, 0, 0)$, $(0, 0, 1, 0, 0, 0)$ und $(1, 0, 1, 0, 0, 0)$.

Schneidet das Innere von A das Äußere von B können wieder nur die Prädikate P1 und P2 erfüllt werden. Jedoch lassen sich von den hierdurch 3 möglichen Kombinationen nur zwei realisieren. Der Eintrag $(1, 0, 0, 0, 0, 0)$ ist mit folgender Begründung nicht realisierbar. Angenommen, dieser Eintrag folgt aus dem Schnitt des Inneren und Äußeren der Grundmengen. Dann hat jeder Punkt des Inneren von A, der an diesem Schnitt beteiligt ist, den Grad 0. (Sonst wäre noch ein weiteres Prädikat erfüllt). Man betrachte nun einen Punkt im Schnitt dieser Mengen. Für diesen läßt sich eine Umgebung finden, die vollständig im Inneren von A liegt. Weiterhin läßt sich eine Umgebung dieses Punktes finden, die vollständig im Äußeren von B liegt. Der Schnitt dieser beiden Umgebungen ist wieder eine Umgebung, d.h. ein zweidimensionales Gebilde. Daher ist auch die Schnittmenge des Inneren von A mit dem Äußeren von B ein zweidimensional. Dann kann jedoch nicht jedem Punkt von A innerhalb dieser Menge der Grad Null zugewiesen worden sein, da dies nur für eindimensionale Teile erlaubt ist.

Beim symmetrischen Fall gibt es daher ebenfalls 2 mögliche Einträge in der erweiterten Matrix, die aus einem 1-Eintrag in der Matrix der Grundmengen resultieren.

Beim Schnitt beider Grenzen sind wieder alle 63 Möglichkeiten realisierbar.

Auch beim Schnitt von Grenze und Innerem ist dies der Fall.

Beim Schnitt beider Inneren muß eins der Prädikate P4, P5 oder P6 erfüllt sein. Mit einer ähnlichen Begründung wie oben kann man zeigen, daß dieser Schnitt ein zweidimensionales Objekt darstellt. Da dies nicht durch eindimensionale Gebilde abgedeckt werden kann, muß eins der drei ersten Prädikate für eine Fläche gelten. Dies würde jedoch bedeuten, daß für diese Fläche die Grade im Inneren von A oder im Inneren von B vollständig Null wären. Das ist aber nur für eindimensionale Teile zulässig. Da es genau 7 mögliche Kombinationen gibt, bei denen mindestens ein Prädikat erfüllt ist, jedoch keins der hier geforderten, bleiben von den 63 Möglichkeiten noch 56 übrig.

Die Matrix für die Anzahlen ist daher:

$$N_{i,j} = \begin{pmatrix} 56 & 63 & 2 \\ 63 & 63 & 3 \\ 2 & 3 & 1 \end{pmatrix}$$

Somit lassen sich 942.068.016 topologische Beziehungen zwischen unscharfen Regionen unterscheiden.

6.4 Topologische Beziehungen zwischen unscharfen Punkten und Linien

Bei den topologischen Beziehungen zwischen Punkt und Linie gibt es beim Schnitt der Äußeren ebenfalls nur eine mögliche Kombination der Prädikate.

Beim Schnitt des Äußeren des Punktes mit dem Inneren der Linie gibt es die drei Möglichkeiten, bei denen die Werte des Punktes Null sind. Von diesen 3 Kombinationen ist diejenige, bei der das dritte Prädikat nicht erfüllt ist, nicht realisierbar. Gibt es diesen Schnitt, so liegt ein ganzer Kurventeil im Äußeren des Punktes. Dieser kann jedoch nach Definition nicht überall den Grad Null besitzen.

Der Fall des Äußeren der Linie mit dem Inneren des Punktes hat nur eine Ableitung, da der Punkt im Inneren stets einen Wert größer als Null annehmen muß und die Linie in ihrem Äußeren stets den Grad 0 besitzt.

Beim Schnitt des Inneren des Punktes mit der Grenze der Linie sind alle Prädikate erfüllbar, an denen der Punkt einen positiven Wert hat. D.h. P1 und P3 sind innerhalb dieser Menge nicht erfüllbar. Es bleiben daher $2^4 - 1 = 15$ ableitbare Kombinationen.

Das Gleiche gilt für den Schnitt der beiden Inneren zwischen diesen Objekttypen.

Beim Schnitt der Grenze der Linie mit dem Äußeren des Punktes sind die drei Kombinationsmöglichkeiten realisierbar, bei denen der Punkt einen Grad von Null aufweist.

Man erhält somit:

$$N_{i,j} = \begin{pmatrix} 15 & 15 & 1 \\ x & x & x \\ 2 & 3 & 1 \end{pmatrix}$$

Das x bedeutet hierbei, daß hier keine Untersuchungen notwendig sind, da dieser Eintrag in den (scharfen) 9-Intersection-Matrizen der Grundmengen stets 0 ist.

Es gibt daher insgesamt 4088 unterscheidbare topologische Beziehungen zwischen diesen Objekttypen.

6.5 Topologische Beziehungen zwischen unscharfen Punkten und Regionen

Der Schnitt der Äußeren zwischen Punkt und Region ergibt eine mögliche Prädikatenkombination.

Der Schnitt des Inneren des Punktes mit dem Äußeren der Region läßt ebenfalls nur eine Möglichkeit offen. Die Begründung ist die gleiche wie die zwischen zwei Punkten.

Der Schnitt der Grenze der Region mit dem Äußeren des Punktes führt zu drei unterschiedlichen Kombinationsmöglichkeiten der Prädikate. Begründung analog zu Region/Region.

Der Schnitt der Grenze der Region mit dem Inneren des Punktes ergibt alle 15 Kombinationsmöglichkeiten, bei denen der Wert des Inneren des Punktes größer als Null ist.

Das Gleiche gilt für den Schnitt der beiden Inneren.

Beim Schnitt des Inneren der Region mit dem Äußeren des Punktes gibt es 2 Möglichkeiten. (Begründung analog zu Region/Region)
Damit ergibt sich folgende Matrix:

$$N_{i,j} = \begin{pmatrix} 15 & 15 & 1 \\ x & x & x \\ 2 & 3 & 1 \end{pmatrix}$$

Damit lassen sich insgesamt 3066 Matrizen realisieren.

6.6 Topologischen Beziehungen zwischen unscharfen Linien und Regionen

Für den Schnitt der beiden Äußeren läßt sich wieder nur ein realisierbarer Eintrag finden. Beim Schnitt des Äußeren einer Region mit der Grenze einer Linie sind alle Prädikate erfüllbar, bei denen der Wert der Region Null ist. Somit gibt es 3 Möglichkeiten.

Auch beim symmetrischen Fall ergibt sich diese Anzahl.

Gibt es einen Schnitt der beiden Grenzen von Linie und Region, sind alle 63 Kombinationen möglich.

Beim Schnitt des Äußeren der Region mit dem Inneren der Linie, gibt es 2 Kombinationen. Die Begründung erfolgt analog zu der zwischen zwei Linien.

Ebenfalls 2 Möglichkeiten gibt es beim symmetrischen Schnitt, wobei die Begründung zwischen zwei Regionen verwendet werden kann.

Beim Schnitt der Grenze der Linie mit dem Inneren der Region lassen sich alle 63 Kombinationsmöglichkeiten realisieren.

Ebenso verhält es sich mit dem symmetrischen Schnitt.

Schneidet das Innere einer Linie das Innere einer Region, so schneidet ein ganzes Kurventeil diese Region. Da dieses nicht vollständig den Grad 0 besitzen darf, ist im entsprechenden Eintrag mindestens 1 der Prädikate P2, P4, P5 oder P6 erfüllt. Da es 3 Kombinationen gibt, bei denen keines dieser Prädikate erfüllt ist, ergeben sich $63 - 3 = 60$ realisierbare Einträge.

Damit ergibt sich folgende Matrix:

$$N_{ij} = \begin{pmatrix} 60 & 63 & 2 \\ 63 & 63 & 3 \\ 2 & 3 & 1 \end{pmatrix}$$

Insgesamt können damit 853.923.054 topologische Beziehungen unterschieden werden.

6.7 Clusterung der topologischen Beziehungen unscharfer Objekte

Die enorme Anzahl der unterscheidbaren Matrizen erfordert dringend eine Clusterung für einen sinnvollen Umgang. Hierzu wird zunächst untersucht, wie die 64 möglichen Einträge zusammengefaßt werden können.

Es kann festgehalten werden, daß der Eintrag $(0, 0, 0, 0, 0, 0)$ eine Sonderstellung einnimmt und daher nicht mit anderen Einträgen zusammengefaßt werden sollte. Somit bildet dieser Eintrag den ersten Cluster, der mit „function_isempty“ bezeichnet wird.

Der Grad 0 besagt, daß ein Punkt nicht zum betrachteten Objekt gehört. Werden nur die ersten drei Prädikate erfüllt, haben daher die beiden Objekte keine „echten“ gemeinsamen Punkte. Daraus läßt sich der nächste Cluster ableiten, der „function_disjoint“ benannt wird. Dieser enthält nach obigen Ausführungen die 7 Einträge, bei denen die letzten 3 Prädikate nicht erfüllt sind, jedoch mindestens eins der ersten drei.

Der „function_equal“-Cluster wird durch die beiden Einträge gebildet, bei denen es mind. einen echten gemeinsamen Punkt gibt und die Funktionswerte gleich sind.

Ist ein Funktionswert eines stets größer als der des anderen Objekts und gibt es mindestens einen echten gemeinsamen Punkt, wird dieser Eintrag dem „function_over“-Cluster zugeordnet. Aus den symmetrischen Bedingungen ergibt sich der „function_under“-Cluster.

Gibt es größere, und gleiche Funktionswerte und jeweils auch echte solcher Punkte, handelt es sich um den „function_over_equal“-Cluster.

Symmetrisch dazu ist der „function_under_equal“-Cluster.

Gibt es echte Punkte, bei denen der Funktionwert eines Objekts größer ist als der des anderen und gibt es (echte oder unechte) Punkte, an denen der umgekehrte Fall eintritt, so werden diese Einträge dem „function_overlap“-Cluster zugeordnet.

Eine vollständige Zuordnung der Einträge zu den Clustern kann der folgenden Tabelle entnommen werden, wobei „function“ bei den Clusterbezeichnungen aus Gründen der Übersichtlichkeit weggelassen wurde.

Eintrag	Cluster	Eintrag	Cluster	Eintrag	Cluster
000000	Empty	000001	Under	000010	Over
000011	Overlap	000100	Equal	000101	UnderEqual
000110	OverEqual	000111	Overlap	001000	Disjoint
001001	Under	001010	Overlap	001011	Overlap
001100	UnderEqual	001101	UnderEqual	001110	Overlap
001111	Overlap	010000	Disjoint	010001	Overlap
010010	Over	010011	Overlap	010100	OverEqual
010101	Overlap	010110	OverEqual	010111	Overlap
011000	Disjoint	011001	Overlap	011010	Overlap
011011	Overlap	011100	Overlap	011101	Overlap
011110	Overlap	011111	Overlap	100000	Disjoint
100001	UnderEqual	100010	OverEqual	100011	Overlap
100100	Equal	100101	UnderEqual	100110	OverEqual
100111	Overlap	101000	Disjoint	101001	UnderEqual
101010	Overlap	101011	Overlap	101100	UnderEqual
101101	UnderEqual	101110	Overlap	101111	Overlap
110000	Disjoint	110001	Overlap	110010	OverEqual
110011	Overlap	110100	OverEqual	110101	Overlap
110110	OverEqual	110111	Overlap	111000	Disjoint
111001	Overlap	111010	Overlap	111011	Overlap
111100	Overlap	111101	Overlap	111110	Overlap
111111	Overlap				

Bei unscharfen Punkten befinden sich die 7 Einträge des Schnittes der beiden Inneren (wenn es einen Schnitt in den entsprechenden Grundmengen gibt) in 6 verschiedenen Clustern. Somit lassen sich mit dieser Clusterung $1 + 4 \cdot 6 = 25$ topologische Beziehungen zwischen unscharfen Punkten mit dieser Clusterung unterscheiden.

Bei zwei unscharfen Linien kann wieder eine Matrix angegeben werden, die die Anzahl ableitbarer Cluster aus einem 1-Eintrag in der Matrix für die Grundmengen angibt:

$$\begin{matrix} 7 & 7 & 1 \\ 7 & 7 & 1 \\ 1 & 1 & 1 \end{matrix}$$

Dazu wird lediglich getestet, in wievielen Clustern die möglichen Einträge liegen. Es lassen sich hiermit 18.832 Matrizen realisieren. Auch für alle anderen Objektkombinationen ergeben sich

solch unpraktikablen Werte.

Man erkennt, daß die Anzahl der unterscheidbaren topologischen Beziehungen immer noch viel zu hoch ist. Dies bedeutet, daß größere Cluster gebildet werden müssen. Eine Möglichkeit hierfür ist es, die Objekte nicht mehr nach Innerem, Äußerem und Grenze zu unterscheiden, sondern nur noch nach Gesamtobjekt und Äußerem. Es ist klar, daß dies zu recht groben Clustern führt. So kann z.B. aus den entstehenden Matrizen nicht mehr abgelesen werden, ob sich z.B. zwei Regionen nur berühren oder ob sie eine gemeinsame Fläche haben. Allerdings kann dies aus der topologischen Beziehung der Grundmengen abgelesen werden. Somit beschränkt sich die Aussagekraft der entstehenden Cluster allein auf die Beschreibung des Verlaufs der Zugehörigkeitsfunktion.

Die Zusammenfassung der Objektteile führt zu einer elementweisen ODER-Verknüpfung der betreffenden Einträge:

	B°	∂B	B^{-1}
A°		\vee	\vee
∂A		\vee	
A^{-1}		\vee	

In den vorangegangenen Betrachtungen hat sich herausgestellt, daß (mit den Clustern) nicht mehr Beziehungen der Äußeren mit Teilen des zweiten Objekts unterschieden werden können, als bei den topologischen Beziehungen der Grundmengen. Im Einzelnen gibt es nur die beiden Einträge „disjoint“ oder „empty“. Genauso verhält es sich mit dem Schnitt der beiden Äußeren. Daher wird die Information gegenüber der topologischen Beziehung der Grundmengen nicht höher. Die gesamte Information über den eigentlichen Funktionsverlauf ist somit im ersten Eintrag der Matrix zu finden. Dieser Cluster-Eintrag genügt somit als Beschreibung der topologischen Beziehung der unscharfen Objekte, wenn zusätzlich die topologische Beziehung der Grundmengen betrachtet wird. Das bedeutet, daß es nun 8 Cluster gibt, die die Beziehung der Zugehörigkeitsfunktionen zweier unscharfer Objekte beschreiben.

Kapitel 7

Realisierung des abstrakten durch ein diskretes Modell

7.1 Das Repräsentationsproblem

Die gewünschten Objekte können nun bereits als unscharfe Mengen dargestellt werden. Versucht man nun jedoch diese Theorie in einen Computer zu übertragen, steht man vor einem enormen Problem. Das oben beschriebene abstrakte Modell geht nämlich von einer unendlichen Präzision aus, die es in einem Computer nicht gibt. Selbst die Darstellung der Basis als Punktmenge ist nicht 100 prozentig möglich. Da diese Menge überabzählbar viele Punkte enthält und eine nahezu beliebige Form annehmen kann, bräuchte man unendlich viel Speicherplatz, um die Basis darzustellen.

Regionen werden daher in GIS meist durch Polygone approximiert. So kann die Region durch endlich viele Eckpunkte beschrieben werden. Aber auch diese Darstellung hat ihre Probleme mit der endlichen Darstellbarkeit.

Beispiel:

Aufgrund des begrenzten Speicherplatzes, der in einem Computer zur Verfügung steht, kann nur eine endliche Menge der reellen Zahlen dargestellt werden. Man betrachte nun die folgende Graphik:

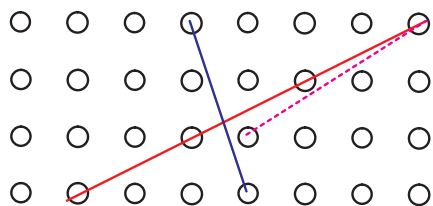


Abb. 20 Schnittpunkt zweier Linien in einem Raster

Die eingezeichneten Kreise seien darstellbare reelle Zahlen. (Arbeitet der Computer mit Festkommaarithmetik, so entspricht dies genau diesem Bild, arbeitet er mit Fließkommaarithmetik, so sind lediglich die Abstände zwischen den einzelnen Punkten nicht einheitlich, was jedoch für die weiteren Betrachtungen keine Rolle spielt.)

Die zu lösende Aufgabe hinter diesem Bild sei die Folgende : „Bestimme das Teilstück einer Linie, welches mit dem Schnittpunkt mit einer anderen Linie beginnt“.

Da der Schnittpunkt der Linie nicht genau dargestellt werden kann, wird der am nächsten liegende Punkt als neuer Endpunkt verwendet. Das Ergebnis dieser Aufgabe ist die gestrichelte Linie. Bei der im abstrakten Modell angenommenen unendlichen Präzision liegt diese Linie genau auf der einen Ursprungslinie. Wie im Bild zu erkennen ist, ist dies in der Computerdarstellung nicht der Fall. Ein anderes Beispiel ist. z.B. in [45] zu finden. Hier wird gezeigt, wie sich eine

Linie durch sukzessive Schnitte mit vertikalen Linien verändert. (Solche Schnitte werden z.B. bei Sweep-Algorithmen berechnet).

7.2 Die geometrische Basis

7.2.1 Partitionierung

Die Grundfläche, also die zweidimensionale Fläche, in der sich die geometrischen Objekte befinden, ist eine Teilmenge der euklidischen Ebene. Sie wird zusammen mit den in ihr befindlichen geometrischen Objekten und einigen Operatoren als Basis des Modells bezeichnet.

Um die endliche Repräsentation zu sichern und gleichzeitig Rundungsfehler zu vermeiden, wird aus dem \mathbb{R}^2 eine endliche Anzahl von Gitterpunkten ausgesucht. Diese müssen so gewählt werden, daß sie unter Verwendung der Realisierung auf einem Computer vollständig darstellbar sind. Das bedeutet, daß Komponenten der Gitterpunkte keine irrationalen Zahlen wie π oder $\sqrt{2}$ sein dürfen.

Um z.B. die Ebene $[0, 400] \times [0, 400]$ darzustellen, verwendet man z.B. als Gitterpunkte die Integerzahlen des Computers. Diese haben die besagte Eigenschaft, daß sie ohne Verluste in einem Computer dargestellt werden können und somit keine Rundungsfehler bei den Rechenoperationen auftreten, auf denen ganze Zahlen abgeschlossen sind.

Wie Abb.20 zeigt, ist es jedoch nicht ausreichend, daß alle Punkte darstellbar sind, da Schnitte von Linien (z.B. Grenzen von Regionen) außerhalb dieser Gitterpunkte liegen können. Aus diesem Grund werden Linien als Mengen von Segmenten dargestellt. Ein solches Segment verbindet zwei benachbarte Gitterpunkte.

Die Zugehörigkeitsfunktion soll linear approximiert werden. D.h. die Funktion wird innerhalb eines Gitterbausteins einer Region durch eine Ebene dargestellt. Da eine Ebene eindeutig durch 3 Punkte repräsentiert werden kann, wird als Gitterbaustein ein Dreieck gewählt. Die Kanten eines solchen Dreiecks sind Segmente und die Eckpunkte sind Elemente der Menge aller Gitterpunkte. Es ist also eine Triangulierung der Basis gesucht, bei der die folgenden Eigenschaften gelten:

- Die entstehenden Dreiecke sind zusammenhängend, d.h. Für jedes Dreieck läßt sich mindestens ein weiteres Dreieck finden, welches zwei verschiedene Gitterpunkte mit dem gegebenen Dreieck gemeinsam hat:

$$\forall T \in \mathbf{BT} : \exists T_2 \neq T : \dim(\text{Pointset}(T) \cap \text{Pointset}(T_2)) = 2$$

- Alle Dreiecke zusammen bilden die gesamte gewünschte Grundmenge :

$$\bigcup_{T \in \mathbf{Basictriangles}} T = \mathbf{Grundmenge}$$

- Die Dreiecke überlappen sich nicht, d.h. sie haben höchstens einen gemeinsamen Eckpunkt oder ein gemeinsames Segment :

$$\forall T_1, T_2 \in \mathbf{BasisTriangles} : \dim(T_1 \cap T_2) \leq 1$$

Der Schnitt beschreibt hier den Schnitt der Punktmenge der Dreiecke, und \dim gibt die Dimension einer Punktmenge an.

Aus der Bedingung, daß die Dreiecke sich nicht überlappen, kann man folgern, daß es keine Dreieckskanten gibt, welche einen Schnittpunkt außerhalb der Gitterpunkte haben.

7.2.2 Wahl der Gitterpunkte

Es ist grundsätzlich möglich, die Gitterpunkte und die dazugehörige Triangulierung beliebig zu wählen, solange die oben angegebenen Eigenschaften erfüllt werden. Um jedoch Kompatibilität zwischen Anwendungen zu erreichen, muß eine möglichst gute einheitliche Einteilung der Grundfläche gefunden werden.

Da eine allgemeine Aufteilung der Basis keine bestimmten Eigenschaften von Anwendungen nutzen kann, bietet sich eine gleichmäßige Partitionierung an.

Soll das Modell nur für eine bestimmte Anwendung verwendet werden, so kann man auch folgendermaßen vorgehen:

Man kann nämlich dann die Grundfläche so einteilen, daß bestimmten Bereichen ‚mehr Aufmerksamkeit geschenkt‘ wird. Dies kann dadurch erreicht werden, indem die Gitterpunkte in solchen Bereichen enger liegen als in weniger wichtigen Bereichen. Falls alle möglichen verwendeten Regionen in einer bestimmten Anwendung im Voraus bekannt sind, kann man die Triangulierung so wählen, daß alle Regionen möglichst gut durch die Dreiecke angenähert werden.

Es ist ebenfalls möglich, dieses Modell so zu erweitern, daß eine hierarchische Auflösung der Triangulierung verwendet wird. Eine solche Vorgehensweise wird z.B. in [26],[39], [3] oder [2] verfolgt.

Eine unterschiedliche Triangulierung führt jedoch zur Inkompatibilität zwischen einzelnen Anwendungen. Man betrachte z.B. diese unterschiedlichen Triangulierungen:

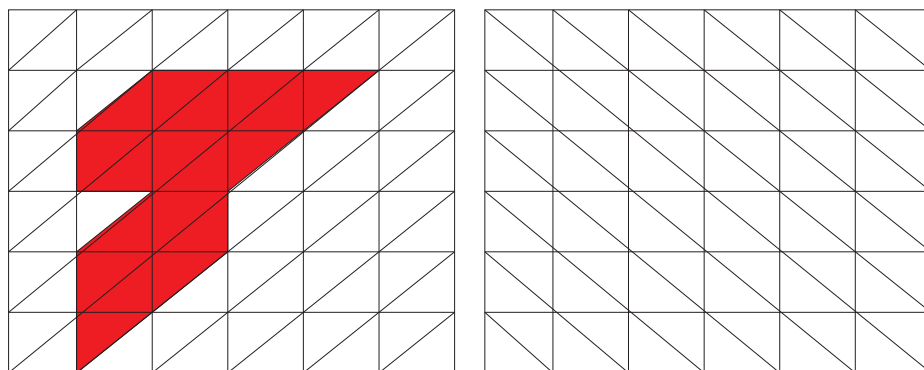


Abb. 21 Unterschiedliche Triangulierungen

Im linken Bild ist eine Region eingezeichnet. Nun wird versucht, diese Region in der rechten Triangulierung darzustellen. Die Region wird hierzu in die Grundfläche des rechten Bildes verschoben:

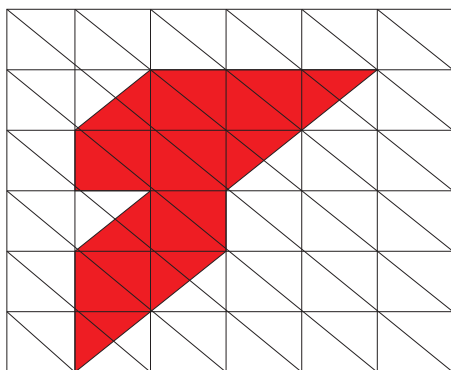


Abb. 22 Verschiebung einer Region

Hier sieht man, daß selbst bei gleichen Gitterpunkten (alle Eckpunkte der Region liegen auf Gitterpunkten) eine exakte Darstellung der Region in einer anderen Triangulierung nicht möglich

ist. (Sie benötigt ‚Teildreiecke‘ und hat Schnittpunkte mit Segmenten, die keine Gitterpunkte sind.)

Man könnte zwar versuchen, diese Region durch die vorhandenen Dreiecke zu approximieren, jedoch ist es möglich, daß die „Rückumwandlung“ der Region nicht wieder die ursprüngliche Region ergibt. Wenn dies nicht der Fall sein darf, benötigt man eine bijektive Abbildung zwischen den Dreiecken unterschiedlicher Triangulierungen. Diese ist jedoch schwer zu finden und kann nur in Einzelfällen eine gute Approximation der Region in der anderen Triangulierung darstellen. Verwendet man zusätzlich unterschiedliche Gitterpunkte, wird es unmöglich eine solche bijektive Abbildung zwischen den Dreiecken zu finden, weil dann die Anzahl der Dreiecke in beiden Partitionen gleich sein müßte, was bei unterschiedlichen Gitterpunkten i.A. nicht gegeben ist. Die folgende Trianguilierung wird vorgeschlagen, um eine gute Darstellung verschiedener Objekte zu ermöglichen. Sie wird im Folgenden als X-Triangulierung bezeichnet.

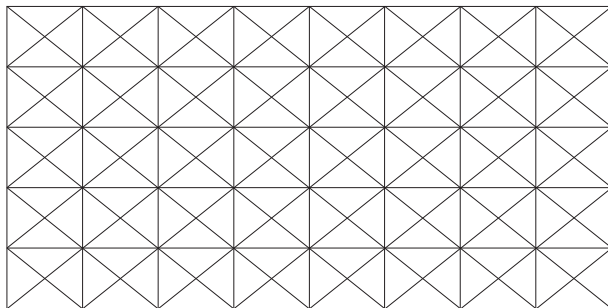


Abb. 23 Vorgeschlagene Triangulierung

Man verwendet als Gitterpunkte einen Teil der positiven Integerzahlen.

Sei **Basicpoint** die Menge der verfügbaren Gitterpunkte und **Basicsegment** die Menge aller Segmente zwischen diesen Gitterpunkten. **Basictriangle** bezeichne die Menge der verfügbaren Dreiecke der Basis. Abhängig von zwei fest zu wählenden gradzahligen Parametern a,b können nun die Basisobjekte definiert werden.

7.2.3 Elementardatentypen der Basis

Ein Basispunkt wird durch ein Koordinatenpaar dargestellt:

$$BP = (x, y)$$

welches die Position des Punktes in der Basisebene widerspiegelt. Die Punktmenge eines Basispunktes im Punktmenge Modell entspricht genau dieser Koordinate.

1. $\forall P = (n \cdot a, m \cdot b); n, m \in \mathbb{N} : P \in \mathbf{Basicpoint}$
2. $\forall P = (n \cdot a + \frac{a}{2}, m \cdot b + \frac{b}{2}); n, m \in \mathbb{N} : P \in \mathbf{Basicpoint}$
3. Punkte, die obigen Anforderungen nicht genügen sind nicht Element von **Basicpoint**

Basissegmente werden durch ihre beiden Endpunkte dargestellt:

$$BS = (EP_1, EP_2)$$

1. $\forall L = ((n \cdot a, m \cdot b), ((n + 1) \cdot a, m \cdot b)); n, m \in \mathbb{N} : L \in \mathbf{BasicSegment}$
2. $\forall L = ((n \cdot a, m \cdot b), (n \cdot a, (m + 1) \cdot b)); n, m \in \mathbb{N} : L \in \mathbf{BasicSegment}$
3. $\forall L = ((n \cdot a, m \cdot b), ((n + \frac{1}{2}) \cdot a, (m + \frac{1}{2}) \cdot b)); n, m \in \mathbb{N} : L \in \mathbf{BasicSegment}$

4. $\forall L = ((n \cdot a, (m + 1) \cdot b), ((n + \frac{1}{2}) \cdot a, (m + \frac{1}{2}) \cdot b)); n, m \in \mathbb{N} : L \in \mathbf{BasicSegment}$
5. $\forall L = ((n + \frac{1}{2}) \cdot a, (m + \frac{1}{2}) \cdot b), ((n + 1) \cdot a, m \cdot b)); n, m \in \mathbb{N} : L \in \mathbf{BasicSegment}$
6. $\forall L = ((n + \frac{1}{2}) \cdot a, (m + \frac{1}{2}) \cdot b), ((n + 1) \cdot a, (m + 1) \cdot b)); n, m \in \mathbb{N} : L \in \mathbf{BasicSegment}$
7. keine anderen Segmente sind Elemente von **BasicSegment**

Diesen Basissegmenten können verschiedene Ausrichtungen zugeordnet werden. Die Basislinien, die durch 1 definiert werden, verlaufen waagrecht, die durch 2 definiert werden, verlaufen senkrecht, und die durch 3,4,5 und 6 definierten Basislinien verlaufen diagonal.

Ein Basissegment besteht aus genau der Punktmenge, die zwischen den beiden Endpunkten liegt, sowie aus den Endpunkten selbst:

$$((x_1, y_2), (x_2, y_2)) \equiv \{(x, y) : x = x_1 + \delta(x_2 - x_1) \wedge y = y_1 + \delta(y_2 - y_1); \delta \in [0, 1]\}$$

Ein Basisdreieck ist durch die Angabe der drei Eckpunkte eindeutig bestimmt.

$$BT = (CP_1, CP_2, CP_3)$$

Alternativ kann ein Basisdreieck auch durch ein Tupel dargestellt werden, welches den linken oberen Eckpunkt eines Rechtecks sowie eine eindeutige Identifizierung des Dreiecks als Komponenten enthält.

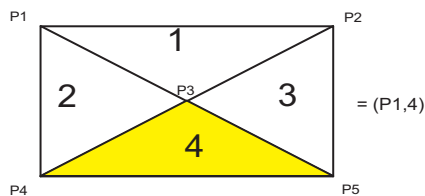


Abb. 24 Alternative Dreiecksdarstellung

Diese Darstellung wird hier nicht weiter verwendet, könnte jedoch bei einer Implementierung zur Einsparung von Speicherplatz dienen.

1. $\forall T = ((n \cdot a, m \cdot b), ((n + 1) \cdot a, m \cdot b), ((n + \frac{1}{2}) \cdot a, (m + \frac{1}{2}) \cdot b)); m, n \in \mathbb{N} : T \in \mathbf{Basictriangle}$
2. $\forall T = ((n \cdot a, m \cdot b), ((n + \frac{1}{2}) \cdot a, (m + \frac{1}{2}) \cdot b), (n \cdot a, (m + 1) \cdot b)); n, m \in \mathbb{N} : T \in \mathbf{Basictriangle}$
3. $\forall T = (((n + \frac{1}{2}) \cdot a, (m + \frac{1}{2}) \cdot b), ((n + 1) \cdot a, m \cdot b), ((n + 1) \cdot a, (m + 1) \cdot b)); n, m \in \mathbb{N} : T \in \mathbf{Basictriangle}$
4. $\forall T = ((n \cdot a, (m + 1) \cdot b), ((n + \frac{1}{2}) \cdot a, (m + \frac{1}{2}) \cdot b), ((n + 1) \cdot a, (m + 1) \cdot b)); n, m \in \mathbb{N} : T \in \mathbf{Basictriangle}$
5. keine anderen Dreiecke sind Elemente von **Basictriangle**

Man beachte, daß die Reihenfolge der Eckpunkte des Dreiecks durch diese Definition fest vorgeschrieben ist. Der erste Eckpunkt ist der lexikographisch kleinste der drei Punkte. Die beiden weiteren Komponenten sind so angeordnet, daß die Seitenlinien des Dreiecks im Uhrzeigersinn orientiert sind.

Weiterhin ist es möglich, ein Basisdreieck durch die beiden ersten Komponenten eindeutig zu identifizieren: bei fester Wahl von n und m gibt es keine zwei Alternativen, bei der die ersten beiden Komponenten gleich sind.

Ein Basisdreieck besteht aus drei Eckpunkten und den Basislinien, die diese Punkte verbinden sowie der durch diese Basislinien eingeschlossene Fläche.

Das Basisdreieck sei durch $((x_1, y_1), (x_2, y_2), (x_3, y_3))$ gegeben. Um festzustellen, ob ein Punkt innerhalb des Dreiecks oder auf einer seiner Seiten liegt, muß dieser sich rechts von jeder der drei orientierten Seiten befinden. Im Einzelnen wird geprüft, ob der zu untersuchende Punkt (x, y) sich rechts der Geraden, die durch $(x_1, y_1) \rightarrow (x_2, y_2)$, $(x_2, y_2) \rightarrow (x_3, y_3)$ und $(x_3, y_3) \rightarrow (x_1, y_1)$ definiert sind, befindet oder auf ihnen liegt. Alle Punkte, die diesen Anforderungen entsprechen, gehören zur durch das Basisdreieck definierten Punktmenge.

Im Folgenden wird gezeigt, wie geprüft werden kann, ob der Punkt (x, y) sich rechts der Geraden, die durch (x_1, y_1) und (x_2, y_2) gegeben ist, befindet. Die Tests mit den anderen Linien sind analog vorzunehmen.

Als Hilfsmittel wird dazu der gerichtete Flächeninhalt des durch $((x_1, y_1), (x_2, y_2), (x, y))$ definierten Dreiecks berechnet. Weist dieser einen positiven Wert auf, so ist die Orientierung dieses Dreiecks entgegengesetzt zum Uhrzeigersinn, und somit liegt (x, y) links der Geraden. Ist er gleich 0, so liegt der Punkt genau auf der Geraden, bei einem Wert größer als 0 ist das Dreieck mit dem Uhrzeigersinn gerichtet und der Punkt liegt rechts der Geraden. Der gerichtete Flächeninhalt kann durch die folgende Formel bestimmt werden [40]:

$$F = (x - x_1) \frac{y + y_1}{2} + (x_2 - x) \frac{y_2 - y}{2} + (x_1 - x_2) \frac{y_1 + y_2}{2}$$

Da geprüft werden soll, ob sich ein negativer Flächeninhalt ergibt, wird die Ungleichung $F \leq 0$ getestet. Zunächst kann die Ungleichung auf jeder Seite mit 2 multipliziert werden und es ergibt sich:

$$(x - x_1)(y + y_1) + (x_2 - x)(y_2 - y) + (x_1 - x_2)(y_1 + y_2) \leq 0$$

als zu prüfende Ungleichung.

Somit kann die Punktmenge von $BT = ((x_1, y_1), (x_2, y_2), (x_3, y_3))$ wie folgt beschrieben werden:

$$\begin{aligned} ((x_1, y_1), (x_2, y_2), (x_3, y_3)) \equiv \{ (x, y) : & (x - x_i)(y - y_i) + (x_j - x)(y_j - y) + \\ & (x_i - x_j)(y_i + y_j) \leq 0; \\ & i = 1 \dots 3; j = (i \bmod 3) + 1 \} \end{aligned}$$

7.2.4 Operatoren und Prädikate auf den Elementardatentypen

Die Fläche eines Basisdreiecks, kann berechnet werden, indem die Fläche eines halben Dreiecks verdoppelt wird. Diese Fläche läßt sich recht leicht bestimmen, da sich bei entsprechender Halbierung ein rechtwinkliges Dreieck ergibt.

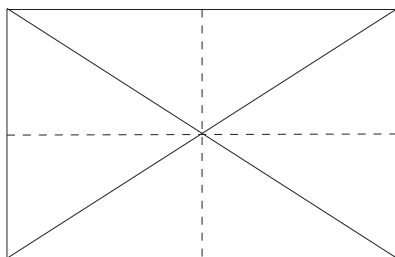


Abb. 25 Fläche eines Basisdreiecks

$$area(BT) \stackrel{\text{def}}{=} 2 \cdot \frac{a}{2} \cdot \frac{b}{2} = \frac{ab}{4}$$

Da a und b ganzzahlig gewählt worden sind, ergibt sich auch eine ganze Zahl als Flächeninhalt für solch ein Dreieck. Das ist sehr vorteilhaft, da so die Fläche einer Menge solcher Dreiecke ohne Rundungsfehler berechnet werden kann.

Die oben eingeführten Basiselemente können untereinander in verschiedenen Beziehungen stehen, die in diesem Abschnitt betrachtet werden sollen.

Die Gleichheit zweier Basiselemente wird mittels des folgenden Prädikats dargestellt:

$$\mathit{equal} : \mathbf{BasicObject}_i \times \mathbf{BasicObject}_i \rightarrow \mathbf{boolean}$$

Die Indizierung bedeutet, daß es sich um zwei gleichartige Basiselemente handeln muß. Basiselemente, die dieser Forderung nicht entsprechen, sind nicht vergleichbar.

Gleichheit von Punkten:

$$\mathit{equal}((x_1, y_1), (x_2, y_2)) \stackrel{\text{def}}{=} (x_1 = x_2) \wedge (y_1 = y_2)$$

Auf den Basispunkten kann eine Ordnung definiert werden, die der gewöhnlichen lexikographischen Ordnung entspricht:

$$< : \mathbf{BasicPoint} \times \mathbf{BasicPoint} \rightarrow \mathbf{boolean}$$

$$(x_1, y_1) < (x_2, y_2) \stackrel{\text{def}}{=} x_1 < x_2 \vee ((x_1 = x_2) \wedge (y_1 < y_2))$$

Ein Basispunkt kann Endpunkt eines Basissegments sein:

$$\mathit{isendpoint} : \mathbf{BasicSegment} \times \mathbf{BasicPoint} \rightarrow \mathbf{boolean}$$

$$\mathit{isendpoint}((EP_1, EP_2), P) \stackrel{\text{def}}{=} \mathit{equal}(P, EP_1) \vee \mathit{equal}(P, EP_2)$$

Ob ein Basispunkt ein Eckpunkt eines Basisdreiecks ist, kann durch folgendes Prädikat ermittelt werden:

$$\mathit{iscornerpoint} : \mathbf{BasicTriangle} \times \mathbf{BasicPoint} \rightarrow \mathbf{boolean}$$

$$\mathit{iscornerpoint}((CP_1, CP_2, CP_3), P) \stackrel{\text{def}}{=} \mathit{equal}(P, CP_1) \vee \mathit{equal}(P, CP_2) \vee \mathit{equal}(P, CP_3)$$

Zwei Basissegmente sind gleich, wenn sie die gleichen Endpunkte besitzen.

$$\mathit{equal} : \mathbf{BasicSegment} \times \mathbf{BasicSegment} \rightarrow \mathbf{boolean}$$

$$\mathit{equal}((EP_1^1, EP_2^1), (EP_1^2, EP_2^2)) \stackrel{\text{def}}{=} (\mathit{equal}(EP_1^1, EP_1^2) \wedge \mathit{equal}(EP_2^1, EP_2^2))$$

Weiterhin kann ein Segment die Seite eines Basisdreiecks darstellen:

$$\mathit{isSide} : \mathbf{BasicTriangle} \times \mathbf{BasicSegment} \rightarrow \mathbf{boolean}$$

$$\mathit{isSide}(T, L) \stackrel{\text{def}}{=} \exists P_1 \neq P_2 \in \mathbf{Basicpoints} : \mathit{isendpoint}(P_1, L) \wedge \mathit{isendpoint}(P_2, L) \wedge \mathit{iscornerPoint}(P_1, T) \wedge \mathit{iscornerPoint}(P_2, T)$$

Zwei Basissegmente können miteinander verbunden sein. Dies ist genau dann der Fall, wenn sie genau einen gemeinsamen Endpunkt besitzen:

$$\mathit{connected} : \mathbf{BasicSegment} \times \mathbf{BasicSegment} \rightarrow \mathbf{boolean}$$

$$\mathit{connected}(BL_1, BL_2) \stackrel{\text{def}}{=} \neg \mathit{equal}(BL_1, BL_2) \wedge \exists P \in \mathbf{Basicpoint} : \mathit{isendpoint}(P, BL_1) \wedge \mathit{isendpoint}(P, BL_2)$$

Die Länge eines Basissegments entspricht der gewöhnlichen euklidischen Länge einer Linie.

$len : \mathbf{BasicSegment} \rightarrow \mathbb{R}$

$$len((x_1, y_1), (x_2, y_2)) \stackrel{\text{def}}{=} \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Je nach „Typ“ des Basissegments (senkrecht, waagrecht, diagonal) kann die Länge des Basissegments auch wie folgt dargestellt werden:

$$len(BL) = \begin{cases} a & \text{wenn BL vertikal} \\ b & \text{wenn BL horizontal} \\ \frac{\sqrt{a^2+b^2}}{2} & \text{wenn BL diagonal} \end{cases}$$

Zwei Basisdreiecke sind gleich, wenn die dazugehörigen Punkte gleich sind. Wie oben bereits bemerkt, werden die Basisdreiecke durch ihre beiden ersten Komponenten eindeutig bestimmt, wodurch ein Vergleich dieser Komponenten ausreichend ist:

$equal : \mathbf{BasicTriangle} \times \mathbf{BasicTriangle} \rightarrow \mathbf{boolean}$

$$equal((P_1, P_2, P_3), (Q_1, Q_2, Q_3)) \stackrel{\text{def}}{=} equal(P_1, Q_1) \wedge equal(P_2, Q_2)$$

Basisdreiecke sind verbunden, wenn sie genau ein gemeinsames Segment besitzen:

$connected : \mathbf{BasicTriangle} \times \mathbf{BasicTriangle} \rightarrow \mathbf{boolean}$

$$connected(BT_1, BT_2) \stackrel{\text{def}}{=} \exists L \in \mathbf{BasicSegment} : isSide(L, BT_1) \wedge isSide(L, BT_2) \wedge \neg equal(BT_1, BT_2)$$

7.3 Repräsentation der Datentypen

7.3.1 Unscharfe Regionen

Unscharfe Elementardreiecke

Ein Elementardreieck (ET) entspricht einem Basisdreieck, bei dem jedem einzelnen Eckpunkt ein Zugehörigkeitsgrad zugewiesen wurde. Es kann somit als Triple $((p_1, Z_1), (p_2, Z_2), (p_3, Z_3))$ dargestellt werden, wobei alle $Z_i, i = 1 \dots 3$ aus dem Intervall $[0, 1]$ stammen. Weiterhin gilt: $(p_1, p_2, p_3) \in \mathbf{Basictriangle}$.

Durch diese Zuordnung kann man das so entstehende Objekt auch als dreidimensionales Dreieck auffassen.

Das zugrundeliegende Basisdreieck erhält man durch den Operator *basic*.

$$basic(p_1, Z_1, p_2, Z_2, p_3, Z_3) \stackrel{\text{def}}{=} (p_1, p_2, p_3)$$

Definition unscharfer Regionen

Eine unscharfe Region wird als eine Menge von ET's und einem Skalierungsfaktor definiert:

$$FREGION = (scal \in \mathbb{R}, \mathbf{ST} = \text{set of ET})$$

Dies entspricht etwa der Definition des abstrakten Modells, jedoch ist hier keine Zugehörigkeitsfunktion angegeben, sondern die entsprechenden Grade sind explizit den Eckpunkten von Basisdreiecken zugeordnet. Die Elemente aus ST müssen der Bedingung genügen, daß mindestens einer der Zugehörigkeitsgrade einen Wert größer 0 hat. Dadurch wird eine eindeutige Repräsentation der Regionen möglich (vgl. dazu die Diskussion im abstrakten Modell). Die Menge der skal. ET's hat die Eigenschaft, daß die zugehörigen Basisdreiecke eine Menge bilden, d.h. es gibt keine zwei unterschiedlichen Elemente in der Menge, denen das gleiche Basisdreieck zugrunde liegt:

$$\forall T^1 = ((p_1^1, Z_1^1), (p_2^1, Z_2^1), (p_3^1, Z_3^1)) \in \mathbf{ST}, T^2 = ((p_1^2, Z_1^2), (p_2^2, Z_2^2), (p_3^2, Z_3^2)) \in \mathbf{ST} : \\ equal(basic(T^1), basic(T^2)) \Rightarrow Z_i^1 = Z_i^2; i = 1 \dots 3$$

Zugehörigkeitsgrad 0

Um zu sichern, daß keine flächigen Gebilde den Grad 0 erhalten, genügt es hier zu fordern, daß mindestens einer der drei Zugehörigkeitsgrade einen Wert größer als Null hat. Durch die lineare Approximation entsteht so maximal ein einzelnes Segment, welches vollständig Null ist.

Darstellung von Unstetigkeiten

Es fällt auf, daß die Zugehörigkeitsfunktion des abstrakten Modells hier keine Funktion mehr darstellt. Dies ist deshalb so, weil mehrere Dreiecke, die einen gemeinsamen Basispunkt besitzen, diesem unterschiedliche Zugehörigkeitsgrade zuweisen können. Haben z.B. zwei Dreiecke ein gemeinsames Segment und wurden mind. einem der Endpunkte durch diese Dreiecke zwei Grade zugewiesen, so sind auf dem gesamten Segment unterschiedliche Zugehörigkeitsgrade zu finden. Einzige Ausnahme sind dabei evtl. Schnittpunkte der 3-dimensionalen Linien. Somit handelt es sich im diskreten Modell um eine Relation, statt um eine Funktion. Diese „mathematische Unkorrektheit“ ist beabsichtigt, um Unstetigkeiten innerhalb der Funktion darzustellen. Da es sich hier um ein diskretes Modell handelt, stellt dies eine vernünftige Variante dar. Es stellt sich dabei die Frage, welcher Zugehörigkeitsgrad dem Punkt angerechnet werden soll. Diesem Problem wird durch die Einführung mehrerer Operatoren begegnet. Je nach Anwendungsfall kann der geeignetste Operator ausgewählt werden. Anders verhält es sich bei der Betrachtung von topologischen Beziehungen. Hier wird stets der maximale Zugehörigkeitsgrad zur Berechnung verwendet. Dies hat folgenden Grund: Scharfe Regionen stellen eine Teilmenge der durch dieses Modell darstellbaren Regionen dar. Diese werden als reguläre, abgeschlossene Punktmenge vorausgesetzt. D.h. die Grenze gehört vollständig zur Region. Im Modell haben Grenzteile scharfer Regionen die beiden Grade 0 und 1 zugewiesen bekommen. Würde man hier einen anderen als den maximalen Grad verwenden, so würden diese Grenzteile nicht (ganz) zur Region gehören, womit diese nicht mehr regulär wäre.

7.3.2 Unscharfe Linien

Unscharfe Segmente

Die Grundbausteine unscharfer Linien bilden unscharfe Segmente (fSegment). Dazu werden den Eckpunkten eines Basissegments Zugehörigkeitsgrade zugewiesen. Es ergibt sich ein Quadrupel (P_1, Z_1, P_2, Z_2) , wobei gilt: $(P_1, P_2) \in \mathbf{BasicSegment}$ und $Z_1, Z_2 \in [0, 1]$. Diese Struktur kann als dreidimensionale Strecke aufgefaßt und als solche dargestellt werden.

Wie auch bei Regionen, wird das zugrundeliegende Basissegment durch den *basic*-Operator ermittelt:

$$\mathit{basic}((p_1, Z_1, p_2, Z_2)) \stackrel{\text{def}}{=} (p_1, p_2)$$

Die Grade der beiden Endpunkte lassen sich durch die folgenden Operatoren ermitteln:

$$Z1((P_1, Z_1, P_2, Z_2)) \rightarrow Z_1$$

$$Z2((P_1, Z_1, P_2, Z_2)) \rightarrow Z_2$$

Definition unscharfer Linien

Eine unscharfe Linie wird durch eine Menge von unscharfen Segmenten sowie einem Skalierungsfaktor beschrieben.

$$FLine = (SF \in \mathbb{R}, \mathbf{fS} = \text{Set of fSegment})$$

Es muß hier gefordert werden, daß die zugrundeliegenden Basissegmente eine Menge bilden, so daß nicht derselben Basislinie unterschiedliche Zugehörigkeitsgrade zugeordnet werden können:

$$\forall L^1 = ((p_1^1, Z_1^1), (p_2^1, Z_2^1)), L^2 = ((p_1^2, Z_1^2), (p_2^2, Z_2^2)) \in \mathbf{fS} : \\ \mathit{equal}((p_1^1, p_2^1), (p_1^2, p_2^2)) \Rightarrow Z_1^1 = Z_1^2 \wedge Z_2^1 = Z_2^2$$

7.3.3 Unscharfe Punkte

Unscharfe Elementarpunkte

Ein unscharfer Elementarpunkt ist der Grundbaustein der unscharfen Punkte. Er wird als Tupel bestehend aus einem Basispunkt und einem Zugehörigkeitsgrad $fEP = (BP, Z)$ dargestellt. Dabei muß gelten:

$$BP \in \mathbf{BasicPoints} \wedge Z \in [0, 1]$$

Der *basic*-Operator stellt eine Projektion auf den entsprechenden Basispunkt dar:

$$basic(BP, Z) \stackrel{\text{def}}{=} BP$$

Definition unscharfer Punkte

Ein unscharfer Punkt wird als ein Tupel (SF, \mathbf{EPs}) mit :

\mathbf{EPs} : eine Menge von Elementarpunkten

SF : ist ein Skalierungsfaktor.

dargestellt.

Die den Elementarpunkten zugrundeliegenden Basispunkte müssen eine Menge bilden, d.h. es ist nicht möglich, daß es in dieser Menge zwei unterschiedliche Elementarpunkte gibt, denen der gleiche Basispunkt zugrundeliegt :

$$\forall EP_1, EP_2 \in \mathbf{EPs} : equal(basic(EP_1), basic(EP_2)) \Rightarrow equal(EP_1, EP_2)$$

Weiterhin wird nicht zugelassen, daß es Zugehörigkeitsgrade gibt, die 0 sind. Dies führt zu einer einheitlichen Darstellung eines unscharfen Punktes.

7.4 Realisierung der Operationen

$PM(fEO)$ bezeichne die Punktmenge von $basic(fEO)$. Diese ist für jedes einzelne Basisobjekt bereits definiert worden.

Die Punktmenge eines unscharfen Objekts ist nun die Vereinigung der Punktmenge der enthaltenen unscharfen Elementarobjekte:

$$PM : \mathbf{fObjekt} \rightarrow \text{set of } (\mathbb{R}, \mathbb{R})$$

$$PM((SF, SfEO = \text{set of } fEO)) \stackrel{\text{def}}{=} \bigcup_{O \in SfEO} PM(O)$$

Er liefert also die zugrundeliegende Punktmenge eines unscharfen Objekts. Bei unscharfen Punkten handelt es sich um eine endliche Menge, wogegen es sich bei Linien und Regionen um eine unendliche (überabzählbare) Menge handelt.

Um die Basis eines unscharfen Objekts zu bestimmen, wird der *basics*-Operator verwendet. Dieser liefert die Menge der jeweiligen vom unscharfen Objekt abgedeckten Basisobjekte.

$$basics : \mathbf{FObjekt}_i \rightarrow \text{set of } \mathbf{BasicObject}_i$$

$$basics(SF, \mathbf{fEO}) \stackrel{\text{def}}{=} \bigcup_{fEO \in \mathbf{fEO}} basic(fEO)$$

Nun wird angegeben, wie die Zugehörigkeit einer Koordinate zu einem geometrischen Objekt berechnet wird. Da dies selbstverständlich für alle Objekttypen unterschiedlich ist, erhält man insgesamt drei Operatoren (jedoch mit dem gleichen Namen). Es wird davon ausgegangen, daß die Zugehörigkeit an einem beliebigen Punkt berechnet werden soll. Man sollte sich jedoch dabei vor Augen halten, daß die Werte nur an den Gitterpunkten korrekt berechnet werden können, da an anderen Punkten Rundungsfehler auftreten können.

Der Zugehörigkeitsgrad ist bei unscharfen Punkten eindeutig und stellt somit eine Funktion dar. Für eine einheitliche Darstellung bei Punkten, Regionen und Linien wird trotzdem eine Relation verwendet. $ZRel$ stellt somit eine Abbildung eines unscharfen Punktes und einer (zweidimensionalen) Koordinate auf eine Menge von reellen Zahlen aus $[0,1]$ dar.

$$ZRel : \mathbf{FPoint} \times (x, y) \rightarrow \mathbf{Set\ of\ Z} \in [0, 1]$$

$$ZRel((SF, \mathbf{SP}), (x, y)) \stackrel{\text{def}}{=} \begin{cases} \{Z\} & \text{wenn } ((x, y), Z) \in \mathbf{SP} \\ \{0\} & \text{sonst} \end{cases}$$

Um den Zugehörigkeitsgrad einer Linie zu berechnen, muß man in der Lage sein, den Grad eines unscharfen Segments berechnen zu können. Dies kann folgendermaßen geschehen:

$$Zfkt : \mathbf{fS} \times (\mathbb{R} \times \mathbb{R}) \rightarrow [0, 1]$$

$$Zfkt(S = ((x_1, y_1), Z_1, (x_2, y_2), Z_2)), (x, y) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } (x, y) \notin PM(basic(S)) = \text{Bed.1} \\ Z_1 + \frac{x-x_1}{x_2-x_1}(Z_2 - Z_1) & \text{if } x_2 \neq x_1 \wedge \neg \text{Bed.1} \\ Z_1 + \frac{y-y_1}{y_2-y_1}(Z_2 - Z_1) & \text{sonst} \end{cases}$$

Gehört die zu untersuchende Koordinate zur Punktmenge der Linie, so werden die Grade aller Segmente aus der Linie, zu deren Punktmenge der Punkt gehört, zum Ergebnis vereinigt, ansonsten ist das Ergebnis die Menge $\{0\}$. Es bleibt festzuhalten, daß die maximale Kardinalität der Ergebnismenge nur von Triangulierung der Basis abhängt. Sie kann maximal so groß werden, wie Basissegmente in einer Koordinate zusammenlaufen. Bei der hier verwendeten Triangulierung also maximal 8 Werte.

$$ZRel((SF, SS), (x, y)) \stackrel{\text{def}}{=} \begin{cases} GS & \text{wenn } (x, y) \text{ in } PM(SF, SS) \\ \{0\} & \text{sonst} \end{cases}$$

mit:

$$GS := \{Zfkt(S) : (x, y) \in PS(S) \wedge S \in \mathbf{fS}\}$$

Wie bei Linien muß man auch bei Regionen den Zugehörigkeitsgrad von Grundbausteinen berechnen können.

$$Zfkt : \mathbf{fET} \times (\mathbb{R}, \mathbb{R}) \rightarrow [0, 1]$$

Die Zugehörigkeitsfunktion einer Koordinate im Inneren des Dreiecks kann in der Weise berechnet werden, daß zunächst die Zugehörigkeitsgrade der Randpunkte des Dreiecks mit dem gegebenen y-Wert des Punktes berechnet werden und anschließend zwischen diesen Punkten der Wert des eigentlichen Punktes durch eine lineare Gleichung gefunden wird.

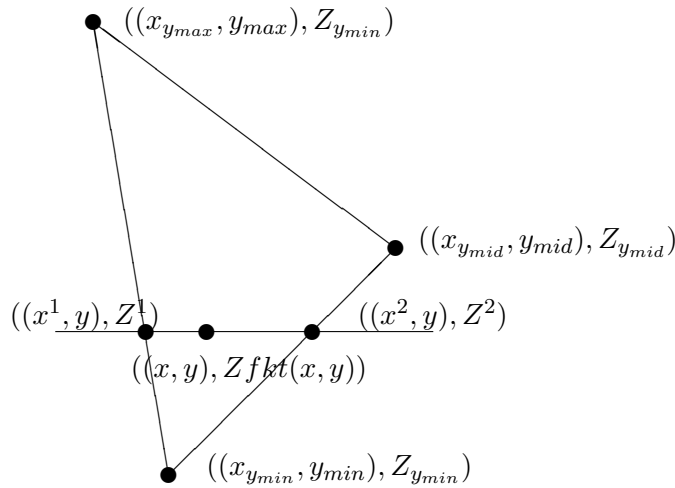


Abb. 26 Berechnung der Zugehörigkeitsfunktion eines fET

Somit wird insgesamt 3 mal ein Zugehörigkeitsgrad auf einer Linie ermittelt, um den Grad eines inneren Dreieckspunktes zu ermitteln:

Sei (x, y) der Punkt, dessen Zugehörigkeitsgrad gesucht ist. Der Punkt liege in der Punktmenge des zu $((x_1, y_1), Z_1), ((x_2, y_2), Z_2), ((x_3, y_3), Z_3)$ gehörenden Basisdreiecks. Es werden zuerst die Punkte ermittelt, die durch die zur x-Achse parallele Gerade mit Höhe y geschnitten werden. Es wird ebenfalls der Zugehörigkeitsgrad dieser Punkte bestimmt. Dazu muß zunächst festgestellt werden, welche beiden der drei Seiten durch diese Gerade geschnitten werden. Es bezeichne $(x_{y_{max}}, y_{max})$ den Punkt der drei Eckpunkte des zugrundeliegenden Basisdreiecks, dessen y -Wert maximal ist:

$$(x_{y_{max}}, y_{max}) = (x_i, y_i)$$

mit

$$y_i = \text{MAX}\{y_1, y_2, y_3\}$$

Analog bezeichne $(x_{y_{min}}, y_{min})$ den Punkt mit minimalem y -Wert. $(x_{y_{mid}}, y_{mid})$ bezeichne den letzten der drei Punkte:

$$(x_{y_{mid}}, y_{mid}) = \{(x_1, y_1), (x_2, y_2), (x_3, y_3)\} - \{(x_{y_{min}}, y_{min}), (x_{y_{max}}, y_{max})\}$$

Die oben beschriebene Gerade schneidet in jedem Fall die Linie $(x_{y_{min}}, y_{min}) \rightarrow (x_{y_{max}}, y_{max})$. Welche der anderen beiden Seiten geschnitten wird, hängt vom y -Wert der Geraden ab. Ist dieser kleiner als y_{mid} , so wird die Linie $(x_{y_{min}}, y_{min}) \rightarrow (x_{y_{mid}}, y_{mid})$ geschnitten, im anderen Fall die Linie $(x_{y_{mid}}, y_{mid}) \rightarrow (x_{y_{max}}, y_{max})$. Die Werte $Z_{y_{min}}$ usw. bezeichnen die Zugehörigkeitsgrade an den entsprechenden Punkten.

Somit können die Schnittpunkte $(x^1, y), (x^2, y)$ und ihre entsprechenden Grade Z^1, Z^2 wie folgt ermittelt werden:

$$x^1 = x_{y_{min}} + \delta^1 \cdot (x_{y_{max}} - x_{y_{min}})$$

$$Z^1 = Z_{y_{min}} + \delta^1 \cdot (Z_{y_{max}} - Z_{y_{min}})$$

mit:

$$\delta^1 = \frac{y - y_{min}}{y_{max} - y_{min}}$$

Hierbei kann es zu keiner Division durch Null kommen, da in einem Dreieck y_{min} und y_{max} verschieden sein müssen.

$$x^2 = \begin{cases} x_{y_{min}} + \delta^2 \cdot (x_{y_{mid}} - x_{y_{min}}) & \text{wenn } y < y_{mid} \\ x_{y_{mid}} + \delta^2 \cdot (x_{y_{max}} - x_{y_{mid}}) & \text{sonst} \end{cases}$$

$$Z^2 = \begin{cases} Z_{y_{min}} + \delta^2 \cdot (Z_{y_{mid}} - Z_{y_{min}}) & \text{wenn } y < y_{mid} \\ Z_{y_{mid}} + \delta^2 \cdot (Z_{y_{max}} - Z_{y_{mid}}) & \text{sonst} \end{cases}$$

mit:

$$\delta^2 = \begin{cases} \frac{y - y_{min}}{y_{mid} - y_{min}} & \text{wenn } y < y_{mid} \\ \frac{y - y_{mid}}{y_{max} - y_{mid}} & \text{sonst} \end{cases}$$

Im ersten Teil dieser Formel, kann es nicht zu einer Division durch Null kommen, da (x,y) nach Voraussetzung zum Inneren des betrachteten Dreiecks gehört. Daher und nach der Bedingung dieses Formelteils gilt:

$$y_{min} < y < y_{mid}$$

Daher sind y_{min} und y_{mid} verschieden. Beim zweiten Teil der Formel läßt sich bei analoger Begründung auch die Division durch Null ausschließen.

Um nun den Zugehörigkeitsgrad am inneren Punkt (x,y) zu berechnen, wird die folgende Formel verwendet:

$$Zfkt(fT, (x, y)) = Z^1 + \frac{x - x^1}{x^2 - x^1} \cdot (Z^2 - Z^1)$$

Bei Punkten, die nicht zum Inneren jedoch zur Punktmenge des Dreiecks gehören, kann die Formel für Segmente angewendet werden, da diese Punkte sich auf einer Dreiecksseite befinden müssen.

Der Zugehörigkeitsgrad einer Koordinate, die sich im Inneren eines Basisdreiecks befindet, ist durch oben angegebene Berechnungen eindeutig bestimmt. Koordinaten, die im Inneren zweier Basissegmente liegen, können einen oder zwei Zugehörigkeitsgrade besitzen. Koordinaten, die auf einem Basispunkt liegen, können maximal so viele Grade zugeordnet bekommen haben, wie Dreiecke diesen als Eckpunkt haben. Bei der gewählten Triangulierung also maximal 8 Grade. Ist die Kardinalität dieser Menge 1, so ist die Region an dieser Stelle stetig, wenn sie größer als 1 ist, so hat die Region dort eine Sprungstelle.

$$ZRel((SF, \mathbf{ET}), (x, y)) \stackrel{\text{def}}{=} \begin{cases} GS & \text{wenn } (x, y) \in PM((SF, \mathbf{ET})) \\ \{0\} & \text{sonst} \end{cases}$$

mit:

$$GS := \{Zfkt(ET, (x, y)) : (x, y) \in PS(ET) \wedge ET \in \mathbf{ET}\}$$

Die speziellen Zugehörigkeitsfunktionen, die maximalen, mittleren bzw. minimalen Zugehörigkeitsgrad berechnen, können über $ZRel$ definiert werden. Daraus ergeben sich:

$$\min Zfkt(fObject, (x, y)) \stackrel{\text{def}}{=} \text{MIN}(ZRel(fObject, (x, y)))$$

$$\text{mid} Zfkt(fObject, (x, y)) \stackrel{\text{def}}{=} \frac{\sum_{Z \in ZRel(fObject, (x, y))} Z}{|ZRel(fObject, (x, y))|}$$

$$\text{maxZfkt}(fObject, (x, y)) \stackrel{\text{def}}{=} \text{MAX}(Z\text{Rel}(fObject, (x, y)))$$

Bei einigen Berechnungen ist es wünschenswert, wenn man den Zugehörigkeitsgrad einer Koordinate für ein unscharfes Objekt eindeutig bestimmen kann. Dies wird dann möglich, wenn das entsprechende Basisobjekt mit in die Berechnung einbezogen wird. Bei Punkten ist dies eine reine Formsache, da es sich ohnehin um eine Funktion handelt. Bei Linien und Regionen jedoch lassen sich viele Operatoren vereinfacht darstellen.

$$\text{Zfkt} : \mathbf{fPoint} \times (\mathbb{R}, \mathbb{R}) \times \mathbf{BasicPoint} \rightarrow [0, 1]$$

$$\text{Zfkt}((SF, \mathbf{fEP}), (x, y), (bx, by)) \stackrel{\text{def}}{=} \begin{cases} Z & \text{wenn } bx = x \wedge by = y \wedge ((x, y), Z) \in \mathbf{fEP} \\ 0 & \text{sonst} \end{cases}$$

$$\text{ZFkt} : \mathbf{FLine} \times (\mathbb{R}, \mathbb{R}) \times \mathbf{BasicSegment} \rightarrow [0, 1]$$

$$\text{Zfkt} : (FL = (SF, \mathbf{SS}), (x, y), BS) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{wenn } BS \notin \text{basics}(BL) \vee (x, y) \notin PS(BS) \\ \text{ZFkt}(Seg, (x, y)) & \text{sonst} \end{cases}$$

mit:

$$\text{Seg} = S \in \mathbf{SS} : \text{equal}(S, BS)\}$$

$$\text{ZFkt} : \mathbf{FRegion} \times (\mathbb{R}, \mathbb{R}) \times \mathbf{BasicTriangle} \rightarrow [0, 1]$$

$$\text{ZFkt}(R = (SF, \mathbf{fET}), (x, y), BT) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{wenn } (x, y) \notin PS(BT) \vee \\ & \exists T \in \mathbf{fET} : \text{equal}(\text{basic}(T), BT) \\ \text{Zfkt}(T, (x, y)) & \text{mit } T \in \mathbf{fET} \wedge (x, y) \in PS(T) \text{ sonst} \end{cases}$$

Um den maximalen Grad des unscharfen Objekts festzustellen, kann zunächst beobachtet werden, daß dieser stets an einem Basispunkt liegt. Da Punkte „nur“ an Basispunkten Zugehörigkeitsgrade zugewiesen bekommen, ist dies für diese Objekte klar. Bei Segmenten kann unterschieden werden, ob die Zugehörigkeitsgrade der Endpunkte gleich sind oder nicht. Im ersten Fall ist klar, daß alle Punkte (speziell auch die Endpunkte) diesen maximalen Grad besitzen. Im anderen Fall sorgt die Linearität der Funktion eines Segments für das gewünschte Ergebnis. Bei unscharfen Dreiecken ist ebenfalls die lineare Approximation zwischen den Graden der Punkte für diese Erscheinung verantwortlich. *MaxZ* kann aufgrund dieser Beobachtung wie folgt berechnet werden:

$$\text{maxZ}(SF, \mathbf{fEP}) \stackrel{\text{def}}{=} \text{MAX}\{Z : ((x, y), Z) \in \mathbf{fEP}\}$$

$$\text{maxZ}(SF, \mathbf{fS}) \stackrel{\text{def}}{=} \text{MAX}\{Z : (((x_1, y_1), Z_1), ((x_2, y_2), Z)) \in \mathbf{fS} \vee \\ (((x_1, y_1), Z), ((x_2, y_2), Z_2)) \in \mathbf{fS}\}$$

$$\text{maxZ}((SF, \mathbf{fET})) \stackrel{\text{def}}{=} \text{MAX}\{Z : \exists((p_1, Z_1), (p_2, Z_2), (p_3, Z_3)) \in \mathbf{fET} : Z = Z_i; i \in \{1, 2, 3\}\}$$

Entsprechendes gilt für den *minZ*-Operator:

$$\text{minZ}(SF, \mathbf{fEP}) \stackrel{\text{def}}{=} \text{MIN}\{Z : ((x, y), Z) \in \mathbf{fEP} \vee Z = 0\}$$

$$\text{minZ}(SF, \mathbf{fS}) \stackrel{\text{def}}{=} \text{MIN}\{Z : (((x_1, y_1), Z_1), ((x_2, y_2), Z)) \in \mathbf{fS} \vee \\ (((x_1, y_1), Z), ((x_2, y_2), Z_2)) \in \mathbf{fS} \vee \\ Z = 0\}$$

$$\min Z((SF, \mathbf{fET})) \stackrel{\text{def}}{=} \text{MIN}\{Z : \exists((p_1, Z_1), (p_2, Z_2), (p_3, Z_3)) \in \mathbf{fET} : \\ Z = Z_i; i \in \{1, 2, 3\} \vee Z = 0\}$$

Die „Größe“ eines Punktes wird durch seine Kardinalität ausgedrückt. Man kann zwischen der Basiskardinalität und der unscharfen Kardinalität unterscheiden:

$$\text{basic_card}((SF, \mathbf{fEP})) \stackrel{\text{def}}{=} |\mathbf{fEP}|$$

$$\text{card}((SF, \mathbf{fEP})) \stackrel{\text{def}}{=} \sum_{((x,y),Z) \in \mathbf{fEP}} Z$$

Für die Berechnung der Länge einer Linie können drei unterschiedliche Interpretationen des Begriffes „Länge“ verwendet werden (vgl. dazu die Ausführungen im abstrakten Modell).

$$\text{basic_len}((SF, \mathbf{fS})) \stackrel{\text{def}}{=} \sum_{fS \in \mathbf{fS}} \text{len}(\text{basic}(fS))$$

Hier wird lediglich die Länge der zugrundeliegenden Basislinie ermittelt. Man beachte hierbei, daß bei der Berechnung der Längenoperator für Basissegmente verwendet wird.

$$\text{len}((SF, \mathbf{fS})) \stackrel{\text{def}}{=} \sum_{(((x_1, y_1), Z_1), ((x_2, y_2), Z_2)) \in \mathbf{fS}} \frac{Z_1 + Z_2}{2} \cdot \text{len}(((x_1, y_1), (x_2, y_2)))$$

Hier wird zu jedem enthaltenen Segment der mittlere Zugehörigkeitsgrad bestimmt und dieser mit der Länge des entsprechenden Basissegments multipliziert. Damit wird der Tatsache Rechnung getragen, daß ein Segment nicht vollständig zur Linie gehört. Die Summe dieser gewichteten Segmente führt schließlich zum endgültigen Ergebnis. Man kann leicht sehen, daß das Resultat dieser Operation stets kleiner oder gleich der Basislänge ist. Gleichheit tritt genau dann auf, wenn es sich um eine scharfe Linie handelt, also einer Linie, die an jedem ihrer Punkte den Grad 1 besitzt.

$$\mathbb{3D_len}((SF, \mathbf{fS})) \stackrel{\text{def}}{=} \sum_{fS \in \mathbf{fS}} \mathbb{3D_len}(fS)$$

Hier findet die gewöhnliche euklidische Länge im \mathbb{R}^3 als Länge des unscharfen Segments Verwendung.

$$\text{len}(((x_1, y_1), Z_1), ((x_2, y_2), Z_2)) \stackrel{\text{def}}{=} \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (Z_2 - Z_1)^2}$$

Hier ist das Ergebnis mindestens so groß wie das der Basislänge. Das ist leicht zu sehen, denn im Fall $Z_2 = Z_1$ fällt der hintere Term weg und man erhält die Formel für die Basislänge eines Segments. Da dieser Term nur größer als Null sein kann (durch das Quadrat), muß das Ergebnis mindestens so groß sein wie die Basislänge.

Die Basis-Fläche einer unscharfen Region ergibt sich analog zu den Linien als Summe der Größe der Einzelkomponenten:

$$\text{basic_area}(SF, \mathbf{fET}) \stackrel{\text{def}}{=} \sum_{fET \in \mathbf{fET}} \text{area}(fET)$$

Da in der gewählten Triangulierung die Flächen aller Basisdreiecke identisch sind, vereinfacht sich diese Formel zu:

$$\text{basic_area}(SF, \mathbf{fET}) \stackrel{\text{def}}{=} |\mathbf{fET}| \cdot \frac{ab}{4}$$

wobei a und b die in der Basis gewählten Parameter darstellen.

Der Flächeninhalt einer unscharfen Region ist im abstrakten Modell als Integral der Zugehörigkeitsfunktion über alle Punkte der Basis dieser Region definiert. Aus der Analysis ist bekannt, daß dieses Integral dem Volumen des durch die Zugehörigkeitsfunktion definierten Körpers entspricht. Ist man nun in der Lage, das Volumen über einem Basisdreieck zu berechnen, so kann der Flächeninhalt der Region als Summe der Einzelvolumen definiert werden.

Volumen über einem Basisdreieck

Verbindet man die Punkte des dreidimensionalen Dreiecks mit den passenden Basispunkten (dritte Dimension = 0), so erhält man einen dreidimensionalen Körper. Von diesem Körper kann das Volumen berechnet werden.

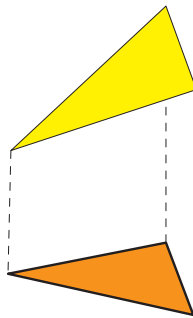


Abb. 27 Durch ein Elementardreieck bestimmter Körper

Dieser Körper kann in ein Prisma und eine darüberliegende Pyramide zerlegt werden. Das Volumen ergibt sich aus der Addition der Einzelvolumen.

Das Volumen des Prismas ist einfach dadurch zu berechnen, indem die Fläche des Basisdreiecks mit dem minimalen Zugehörigkeitsgrad des fET multipliziert wird. Bei der Berechnung des Volumens der 'aufgesetzten' Pyramide können 3 Fälle unterschieden werden. Sind die drei Zugehörigkeitsgrade aller Eckpunkte gleich, so existiert die Pyramide nicht, und es ergibt sich ein Volumen von 0. Sind die beiden minimalen Grade gleich, so erhält man eine Pyramide mit einer dreieckigen Grundfläche. Ihre Grundfläche entspricht einem "nach oben geschobenen" Basisdreieck, insbesondere haben Basisdreieck und die Grundfläche den gleichen Flächeninhalt. Die Höhe dieser Pyramide ist genau die Differenz des maximalen und minimalen Zugehörigkeitsgrades.

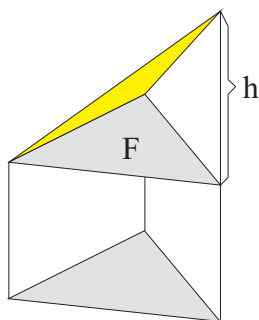


Abb. 28 Dreiseitige Pyramide

Das Volumen dieser Pyramide ergibt sich nun durch [24]:

$$V = \frac{F \cdot h}{3} = \frac{\frac{ab}{4} \cdot (Z_{max} - Z_{min})}{3} = \frac{ab(Z_{max} - Z_{min})}{12}$$

mit:

- a,b = Parameter, die in der Basis gewählt wurden

- $Z_{min} = MIN\{Z_1, Z_2, Z_3\}$
- $Z_{max} = MAX\{Z_1, Z_2, Z_3\}$

Sind die beiden minimalen Zugehörigkeitsgrade nicht gleich, so ist das Volumen einer (auf die Seite gekippten) Pyramide mit einem Viereck als Grundfläche zu berechnen. Hierzu werden zunächst die folgenden Bezeichnungen eingeführt:

- $(p_{Z_{min}}, Z_{min}) = (p_i, Z_i) : Z_i = MAX\{Z_1, Z_2, Z_3\}$
- $(p_{Z_{max}}, Z_{max}) = (p_i, Z_i) : Z_i = MIN\{Z_1, Z_2, Z_3\}$
- $(p_{Z_{mid}}, Z_{mid}) = \{(p_1, Z_1), (p_2, Z_2), (p_3, Z_3)\} - \{(p_{Z_{min}}, Z_{min}), (p_{Z_{max}}, Z_{max})\}$

Die Grundfläche der Pyramide wird nun durch das folgende Viereck gebildet:

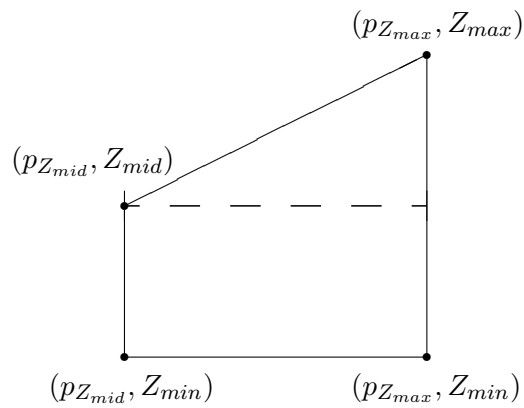


Abb. 29 Grundfläche der Pyramide

Der Flächeninhalt dieses Vierecks kann am einfachsten dadurch berechnet werden, daß die beiden Flächeninhalte des Rechtecks und des rechtwinkligen Dreiecks addiert werden. Dadurch ergibt sich:

$$\begin{aligned}
 F &= \text{lenght}((p_{Z_{mid}}, p_{Z_{max}})) \cdot (Z_{mid} - Z_{min}) + \frac{\text{lenght}((p_{Z_{mid}}, p_{Z_{max}})) \cdot (Z_{max} - Z_{mid})}{2} \\
 &= \text{lenght}((p_{Z_{mid}}, p_{Z_{max}})) \cdot \left((Z_{mid} - Z_{min}) + \frac{Z_{max} - Z_{min}}{2} \right)
 \end{aligned}$$

Die Höhe dieser Pyramide kann durch das folgende Dreieck bestimmt werden.

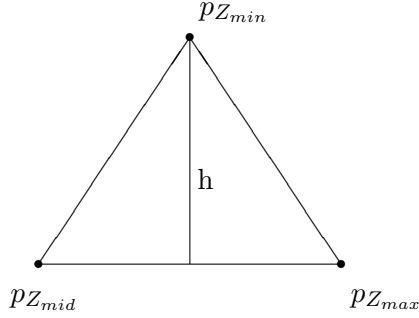


Abb. 30 Höhe der Pyramide

Es gilt für beliebige Dreiecke: $A = \frac{1}{2}ah_a$ [22]. In obigen Dreieck gilt demnach:

$$h = \frac{2A}{\text{length}((pZ_{mid}, pZ_{max}))}$$

Da es sich hier um ein Basisdreieck handelt, kann die Höhe nun durch:

$$h = \frac{a \cdot b}{2 \cdot \text{length}((pZ_{mid}, pZ_{max}))}$$

berechnet werden, wobei a und b die in der Basis gewählten Parameter der Triangulierung darstellen.

Das Volumen der Pyramide kann nun wie folgt berechnet werden:

$$\begin{aligned} V &= \frac{F \cdot h}{3} \\ &= \frac{F \cdot a \cdot b}{6 \text{length}((pZ_{mid}, pZ_{max}))} \\ &= \frac{\text{lenght}((pZ_{mid}, pZ_{max})) \cdot \left((Z_{mid} - Z_{min}) + \frac{Z_{max} - Z_{min}}{2} \right) \cdot a \cdot b}{6 \text{length}((pZ_{mid}, pZ_{max}))} \\ &= \frac{\left((Z_{mid} - Z_{min}) + \frac{Z_{max} - Z_{min}}{2} \right) \cdot a \cdot b}{6} \end{aligned}$$

Man kann leicht erkennen, daß im Fall $Z_{mid} = Z_{min}$ das Ergebnis dasselbe ist, wie bei der Berechnung der Pyramide im oben betrachteten Fall. Daher gilt allgemein:

$$\text{volume}((p_1, Z_1), (p_2, Z_2), (p_3, Z_3)) \stackrel{\text{def}}{=} \frac{a \cdot b}{4} \cdot Z_{min} + \frac{\left((Z_{mid} - Z_{min}) + \frac{Z_{max} - Z_{min}}{2} \right) \cdot a \cdot b}{6}$$

Mit den oben definierten Z_{min} , Z_{mid} , Z_{max} sowie den in der Basis gewählten Parametern a und b.

Der Flächeninhalt (das Volumen) einer unscharfen Region ergibt sich durch Aufaddieren der Einzelvolumen der unscharfen Elementardreiecke.

$$\text{area}((SF, \mathbf{fET})) \stackrel{\text{def}}{=} \sum_{fET \in \mathbf{fET}} \text{volume}(fET)$$

Es ist klar, daß dieser Operator stets einen Wert liefert, der kleiner oder gleich dem Ergebnis des *basic_area*-Operators ist.

Als dritter Operator zur Angabe der Größe einer unscharfen Region kann die Oberfläche der durch sie definierten 3D-Struktur ermittelt werden. Dies kann wieder durch die Addition der Einzeloberflächen der in der Region enthaltenen fET's berechnet werden. Flächen, die durch Sprungstellen entstehen, werden nicht mit in die Oberfläche einbezogen.

$$\text{surface}((x_1, y_1), Z_1), ((x_2, y_2), Z_2), ((x_3, y_3), Z_3)) \stackrel{\text{def}}{=} S$$

S bezeichnet hierbei die Oberfläche eines dreidimensionalen Dreiecks und ist in [24] wie folgt definiert:

$$\begin{aligned} S &= \sqrt{s(s-a)(s-b)(s-c)} && \text{mit } s = \frac{(a+b+c)}{2} \\ &= \sqrt{\frac{(a+b+c)(-a+b+c)(a-b+c)(a+b-c)}{2}} \\ &= \sqrt{\frac{(-a^2+b^2+2bc+c^2)(a^2-b^2+2bc-c^2)}{2}} \\ &= \sqrt{\frac{-a^4-b^4-c^4+2a^2b^2+2a^2c^2+2b^2c^2}{2}} \end{aligned}$$

Mit

$$a = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (SZ_2 - SZ_1)^2}$$

$$b = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2 + (SZ_3 - SZ_2)^2}$$

$$c = \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2 + (SZ_3 - SZ_1)^2}$$

$$\text{und } SZ_i = Z_i \cdot SF$$

kann der Flächeninhalt des Dreiecks ermittelt werden.

Betrachtet man die unteren Terme, so erkennt man den Sinn der Umstellung der ersten Formel. Man kann hier erkennen, daß nur die Quadrate der unteren drei Bezeichner verwendet werden. Dies ermöglicht es, die Fläche zu berechnen, wobei nur einmal eine Wurzel berechnet werden muß.

Die Oberfläche der unscharfen Region:

$$\text{surface}(SF, \mathbf{fET}) \stackrel{\text{def}}{=} \sum_{fET \in \mathbf{fET}} \text{surface}(fET)$$

Hier ist das Ergebnis mindestens so groß wie das des *basic_area*-Operators.

Filter

Man kann zwei Filter-Operatoren auf geometrischen Objekten definieren, welche die Teile eines Objekts herausfiltern, deren Zugehörigkeitsgrad einen bestimmten Wert unterschreitet.

Diese sind für Punkte leicht definierbar:

$$\alpha\text{-cut}((SF, \mathbf{fEP}), \alpha) \stackrel{\text{def}}{=} (SF, \mathbf{fEP}^{\text{new}})$$

mit:

$$\mathbf{fEP}^{\text{new}} = \{(BP, Z) : (BP, Z) \in \mathbf{fEP} \wedge Z \geq \alpha\}$$

Der strenge α -Schnitt arbeitet im Prinzip wie der α -Schnitt, nur mit dem Unterschied, daß die Gleichheit mit α nicht zugelassen wird:

$$\text{strong-}\alpha\text{-cut}((SF, \mathbf{fEP}), \alpha) \stackrel{\text{def}}{=} (SF, \mathbf{fEP}^{\text{new}})$$

mit:

$$\mathbf{fEP}^{\text{new}} = \{(BP, Z) : (BP, Z) \in \mathbf{fEP} \wedge Z > \alpha\}$$

Bei Linien gibt es mehrere Möglichkeiten den α -Schnitt zu definieren. Diese sind:

- ein Endpunkt muß einen Zugehörigkeitsgrad $\geq \alpha$ haben
- beide Endpunkte müssen einen Zugehörigkeitsgrad $\geq \alpha$ aufweisen
- der mittlere Zugehörigkeitsgrad muß $\geq \alpha$ sein

Die gleichen Ausführungen gelten natürlich auch für den strengen α -Schnitt.

In der vorliegenden Arbeit wird die letzte Alternative verwendet. Der Grund für diese Entscheidung läßt sich am besten durch ein Beispiel erklären: Man betrachte ein Segment, welches an seinen Endpunkten den Grad 0 und den Grad 1 besitzt. (Man kann jedoch auch andere unterschiedliche Grade betrachten). Man beachte, daß durch die lineare Approximation der Zugehörigkeitsfunktion nur die Endpunkte die angegebenen Grade besitzen. Bei der zweiten Alternative würde der strenge-0-Schnitt das Segment aus der Linie entfernen. Jedoch ist nur ein einzelner Punkt (nämlich der entsprechende Endpunkt) aus einer unendlichen Menge von Punkten für diese Entscheidung verantwortlich. Eine ähnliche Situation ergibt sich, wenn man sich den 1-Schnitt bei der ersten Alternative ansieht. Das Segment würde in der Linie verbleiben, obwohl nur ein einzelner Punkt aus einer unendlichen Menge von Punkten die Bedingung des α -Schnittes erfüllt.

α -cut : **FLine** \rightarrow **FLine**

$$\alpha\text{-cut}((SF, \mathbf{fS}), \alpha) \stackrel{\text{def}}{=} (SF, \mathbf{fS}^{\text{new}})$$

mit:

$$\mathbf{fS}^{\text{new}} = \{((P_1, Z_1), (P_2, Z_2)) \in \mathbf{fS} : \frac{Z_1 + Z_2}{2} \geq \alpha\}$$

strong- α -cut : **FLine** \rightarrow **FLine**

$$\text{strong-}\alpha\text{-cut}((SF, \mathbf{fS}), \alpha) \stackrel{\text{def}}{=} (SF, \mathbf{fS}^{\text{new}})$$

mit:

$$\mathbf{fS}^{\text{new}} = \{((P_1, Z_1), (P_2, Z_2)) \in \mathbf{fS} : \frac{Z_1 + Z_2}{2} > \alpha\}$$

Wie bei Linien gibt es auch bei Regionen unterschiedliche Möglichkeiten der Definition des α -Schnitts. Prinzipiell bieten sich die folgenden Alternativen an:

- alle Eckpunkte müssen einen Grad $\geq \alpha$ besitzen
- mind. ein Eckpunkt hat einen Grad $\geq \alpha$
- mind. zwei Eckpunkte haben einen Grad $\geq \alpha$
- der mittlere Grad ist $\geq \alpha$

Aus den gleichen Gründen, wie bei unscharfen Linien, werden die ersten beiden Alternativen ausgeschlossen. Auch hier genügt ein Punkt, für die Entscheidung, ob das Segment in der Region verbleibt oder nicht. Die dritte Alternative ist aus ähnlichem Grund ebenfalls nicht günstig. Man betrachte z.B. den strengen- α -Schnitt für ein Dreieck $(P_1, 0.5, P_2, 0.5, P_3, 0.9)$ für $\alpha = 0.5$. Obwohl nur eine Linie (der zweidimensionalen Dreiecksfläche) der α -Bedingung widerspricht, würde das Dreieck entfernt werden. Es läßt sich leicht ein weiteres Beispiel finden, bei dem eine einzelne Linie für eine positive Entscheidung verantwortlich ist.

In diesem Modell wird deshalb verlangt, daß der mittlere Zugehörigkeitsgrad der drei Eckpunkte eines Dreiecks größer (oder gleich) α ist:

$\alpha\text{-cut} : \mathbf{FRegion} \rightarrow \mathbf{FRegion}$

$$\alpha\text{-cut}((SF, \mathbf{fET}), \alpha) \stackrel{\text{def}}{=} (SF, \mathbf{fET}^{\text{new}})$$

mit:

$$\mathbf{fET}^{\text{new}} = \{fET = ((P_1, Z_1), (P_2, Z_2), (P_3, Z_3)) \in \mathbf{fET} : \frac{Z_1 + Z_2 + Z_3}{3} \geq \alpha\}$$

$\text{strong} - \alpha - \text{cut} : \mathbf{FRegion} \rightarrow \mathbf{FRegion}$

$$\text{strong} - \alpha - \text{cut}((SF, \mathbf{fET}), \alpha) \stackrel{\text{def}}{=} (SF, \mathbf{fET}^{\text{new}})$$

mit:

$$\mathbf{fET}^{\text{new}} = \{fET = ((P_1, Z_1), (P_2, Z_2), (P_3, Z_3)) \in \mathbf{fET} : \frac{Z_1 + Z_2 + Z_3}{3} > \alpha\}$$

Allgemeines zu den Filtern

Man kann feststellen, daß der strenge- α -Schnitt dazu verwendet werden kann, um „ungültige“ unscharfe Objekte in äquivalente gültige Objekte zu konvertieren. Mit „ungültig“ sind hier solche Objekte gemeint, die Elemente enthalten, deren Zugehörigkeitsgrade alle 0 sind. Da solche Objekte natürlich nicht das Ergebnis von Operatoren sein dürfen aber als „Zwischenprodukt“ bei Berechnungen auftreten können, wird der strenge- α -Schnitt für diesen Zweck verwendet. Es wird hierbei $\text{strong-0-cut}(fObject)$ als Abkürzung für $\text{strong-}\alpha\text{-cut}(fObject, 0)$ verwendet.

Bei einigen Operatoren (z.B. die additive Verknüpfung von Objekten) kann es passieren, daß der Bereich der zulässigen Zugehörigkeitsgrade $[0,1]$ verlassen wird. Es wird eine Hilfsfunktion benötigt, die solche Unregelmäßigkeiten bereinigt, ohne dabei das eigentliche Objekt zu verändern. Die Information wird dazu von den Zugehörigkeitsgraden in den Skalierungsfaktor verschoben. Durch diesen Operator kann sichergestellt werden, daß die Ergebnisse von Operatoren stets korrekt sind.

Bei der Normierung wird die Information von Zugehörigkeitsgraden jenseits der zulässigen Werte in den Skalierungsfaktor verschoben. Im Prinzip arbeitet dieser Operator in mehreren Schritten. Zunächst wird die Differenz zwischen dem maximalen und minimalen Zugehörigkeitsgrad ermittelt. Diese Differenz bildet zusammen mit dem ursprünglichen Skalierungsfaktor den Wert für den Skalierungsfaktor nach dieser Operation. Ist der minimale Zugehörigkeitsgrad kleiner als Null, so wird das Objekt insgesamt soweit angehoben, daß der minimale Grad nun Null beträgt. Dazu wird von jedem enthaltenen Punkt der minimale Grad subtrahiert. Da dieser negativ ist, ergeben sich stets Werte, die größer oder gleich Null sind. Nun ist zwar jeder Zugehörigkeitsgrad des Objekts größer oder gleich Null, kann aber noch über 1 liegen. Um dies auszugleichen, wird jeder Punkt des Objekts durch die Differenz zwischen minimalem und maximalem Grad geteilt. Daraus ergibt sich, daß der neue maximale Grad genau 1 beträgt. Die Information wurde bereits in den Skalierungsfaktor verschoben, so daß sich am eigentlichen Objekt nichts ändert, sondern nur die Repräsentation in eine gültige konvertiert wird.

Normierung von Punkten

$norm : \mathbf{FPoint}^* \rightarrow \mathbf{FPoint}$

Der * soll kennzeichnen, daß die Basispunkte auch Zugehörigkeitsgrade jenseits von (0,1] zugewiesen bekommen haben können. Es bezeichne Z_{min}^* den kleinsten Zugehörigkeitsgrad, der einem Basispunkt durch einen unscharfen Punkt $P=(SF, \mathbf{fEP})$ zugewiesen wurde.

$$Z_{min}^* = MIN\{Z : (BP, Z) \in \mathbf{fEP}\}$$

Weiterhin bezeichne $Z_{min} = min(0, Z_{min}^*)$. Außerdem sei mit Z_{max} entsprechend der größte zugeordnete Grad eines Punktes gegeben. Dann läßt sich $norm$ wie folgt definieren:

$$norm((SF, \mathbf{fEP})) \stackrel{\text{def}}{=} strong - 0 - cut((SF \cdot (Z_{max} - Z_{min}), \mathbf{fEP}_{new}))$$

mit:

$$\mathbf{fEP}_{new} = \{(BP, \frac{Z - Z_{min}}{Z_{max} - Z_{min}} : (BP, Z) \in \mathbf{fEP}\}$$

Normierung von Linien

$norm : \mathbf{FLine}^* \rightarrow \mathbf{FLine}$

* : auch „ungültige“ Linien (siehe oben)

Es bezeichnen:

$$Z_{min}^* = MIN\{Z : ((p_1, Z), (p_2, Z_2)) \in \mathbf{fS} \vee ((p_1, Z_1), (p_2, Z)) \in \mathbf{fS}\}$$

$$Z_{min} = max(0, Z_{min}^*)$$

$$Z_{max} = MAX\{Z : ((p_1, Z), (p_2, Z_2)) \in \mathbf{fS} \vee ((p_1, Z_1), (p_2, Z)) \in \mathbf{fS}\}$$

Formel:

$$norm(L = (SF, \mathbf{fS})) \stackrel{\text{def}}{=} strong - 0 - cut((SF \cdot (Z_{max} - Z_{min}), \mathbf{fS}_{new}))$$

mit:

$$\mathbf{fS}_{new} = \{((p_1, \frac{Z_1 - Z_{min}}{Z_{max} - Z_{min}}), (p_2, \frac{Z_2 - Z_{min}}{Z_{max} - Z_{min}})) : ((p_1, Z_1), (p_2, Z_2)) \in \mathbf{fS}\}$$

Normierung von Regionen

Bezeichnungen:

$$Z_{min}^* = MIN\{Z : \exists((p_1, Z_1), (p_2, Z_2), (p_3, Z_3)) \in \mathbf{scET} \wedge (Z = Z_1 \vee Z = Z_2 \vee Z = Z_3)\}$$

$$Z_{min} = MIN(Z_{min}^*, 0)$$

$$Z_{max} = MAX\{Z : \exists((p_1, Z_1), (p_2, Z_2), (p_3, Z_3)) \in \mathbf{scET} \wedge (Z = Z_1 \vee Z = Z_2 \vee Z = Z_3)\}$$

Formel:

$$norm(R = (SF, \mathbf{fET})) \stackrel{\text{def}}{=} (SF^*, \mathbf{fET}_{new})$$

mit:

$$SF^* = SF \cdot (Z_{max} - Z_{min})$$

und

$$\mathbf{fET}_{new} = \left\{ \left(\left(p_1, \frac{Z_1 - Z_{min}}{Z_{max} - Z_{min}} \right), \left(p_2, \frac{Z_2 - Z_{min}}{Z_{max} - Z_{min}} \right), \left(p_3, \frac{Z_3 - Z_{min}}{Z_{max} - Z_{min}} \right) \right) : \right. \\ \left. ((p_1, Z_1), (p_2, Z_2), (p_3, Z_3)) \in \mathbf{fET} \right\}$$

Wie im abstrakten Modell beschrieben, wird manchmal ein Operator benötigt, der die unscharfen Objekte in scharfe umwandelt. Wieder gibt es für die unterschiedlichen Objekttypen verschiedene Operatoren.

sharp : FPoint → FPoint

$$sharp((SF, \mathbf{fEP})) \stackrel{\text{def}}{=} (SF, \{(P, 1) \mid (P, Z) \in \mathbf{fEP}\})$$

Hier wird einfach jedem enthaltenen Punkt der Grad 1 zugewiesen.

sharp : FLine → FLine

$$sharp((SF, \mathbf{fS})) \stackrel{\text{def}}{=} (SF, \{(P_1, 1, P_2, 1) \mid (P_1, Z_1, P_2, Z_2) \in \mathbf{fS}\})$$

Allen Endpunkten von enthaltenen Segmenten wird der Grad 1 zugewiesen. Durch die lineare Approximation zwischen den Endpunkten haben so auch alle inneren Punkte des Segments diesen Grad.

sharp : FRegion → FRegion

$$sharp((SF, \mathbf{fT})) \stackrel{\text{def}}{=} (SF, \{(P_1, 1, P_2, 1, P_3, 1) \mid (P_1, Z_1, P_2, Z_2, P_3, Z_3) \in \mathbf{fT}\})$$

Hier wird jedem Eckpunkt eines Dreiecks der Grad 1 zugewiesen. Durch die lineare Approximation haben so auch alle anderen Punkte des Dreiecks den Grad 1.

Vereinigung von Punkten

Der nichtinterpretierte Vereinigungsoperator stellt die klassische Vereinigung zweier unscharfer Mengen dar. Es wird hierbei das Maximum der beiden zu betrachtenden Zugehörigkeitsgrade gebildet:

$$union(P_1 = (SF_1, \mathbf{fEP}_1), P_2 = (SF_2, \mathbf{fEP}_2)) \stackrel{\text{def}}{=} (1, \mathbf{EP}_{S_{new}})$$

mit:

$$\mathbf{EP}_{S_{new}} = \{(BP, Z) : BP \in (basics(\mathbf{fEP}_1) \cup basics(\mathbf{fEP}_2)) \wedge \\ Z = \max(ZFkt(P_1, BP, BP), ZFkt(P_2, BP, BP))\}$$

Dies entspricht einer sehr direkten Umsetzung der Definition des abstrakten Modells. Die Variante, welche den Skalierungsfaktor mit einbezieht, arbeitet folgendermaßen:

scaled_union : FPoint × FPoint → FPoint

$$scaled_union(P_1 = (SF_1, \mathbf{fEP}_1), P_2 = (SF_2, \mathbf{fEP}_2)) \stackrel{\text{def}}{=} norm((1, \mathbf{fEP}))$$

mit:

$$\mathbf{fEP} = \{(BP, Z) : BP \in (basics(\mathbf{fEP}_1) \cup basics(\mathbf{fEP}_2)) \wedge \\ Z = \max((SF_1 \cdot ZFkt(P_1, BP, BP)), (SF_2 \cdot ZFkt(P_2, BP, BP)))\}$$

Wie im abstrakten Modell beschrieben, wird zunächst die Information des Skalierungsfaktors in die Zugehörigkeitsgrade gezogen. Die Normierung wandelt das erhaltene Objekt wieder in ein gültiges Objekt um.

Vereinigung von Linien

$union : \mathbf{FLine} \times \mathbf{FLine} \rightarrow \mathbf{FLine}$

$$union(L^1 = (SF^1, \mathbf{fS}^1), L^2 = (SF^2, \mathbf{fS}^2)) \stackrel{\text{def}}{=} (1, \mathbf{fS}^{\text{neu}})$$

mit:

$$\mathbf{fS}^{\text{neu}} = \{((p_1, Z_1^{\text{neu}}), (p_2, Z_2^{\text{neu}})) : BL = (p_1, p_2) \in basics(L^1) \cup basics(L^2) \wedge Z_i^{\text{neu}} = \max(ZFkt(L^1, p_i, BL), ZFkt(L^2, p_i, BL))\}$$

Bei dieser Vereinigung kann Information durch die lineare Approximation verlorengehen. Im Einzelnen entsprechen die Zugehörigkeitsgrade im Inneren der Vereinigung nicht unbedingt denen des abstrakten Modells.

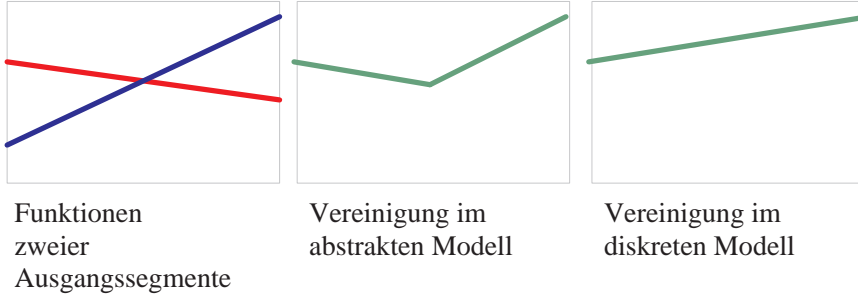


Abb. 31 Vergleich von abstrakter und diskreter Vereinigung

Dies verdeutlicht, daß die Zugehörigkeitsgrade tatsächlich nur in den Basispunkten eindeutig bestimmt werden können.

$scaled_union : \mathbf{FLine} \times \mathbf{FLine} \rightarrow \mathbf{FLine}$

$$scaled_union(L^1 = (SF^1, \mathbf{fS}^1), L^2 = (SF^2, \mathbf{fS}^2)) \stackrel{\text{def}}{=} norm((1, \mathbf{fS}^{\text{neu}}))$$

mit:

$$\mathbf{fS}^{\text{neu}} = \{((p_1, Z_1^{\text{neu}}), (p_2, Z_2^{\text{neu}})) : BL = (p_1, p_2) \in basics(L^1) \cup basics(L^2) \wedge Z_i^{\text{neu}} = \max(SF^1 * ZFkt(L^1, p_i, BL), SF^2 * ZFkt(L^2, p_i, BL))\}$$

Vereinigung von Regionen

Bei der Vereinigung von Regionen besteht die Ergebnismenge aus der Menge der vereinigten Basisdreiecke beider Operanden. Für jeden Eckpunkt eines solchen Dreiecks wird das Maximum der Grade der beiden Operanden gewählt. Wie bei Linien kann Information über die Grade im Inneren der Dreiecke verlorengehen.

$union : \mathbf{FRegion} \times \mathbf{FRegion} \rightarrow \mathbf{FRegion}$

$$union(R_1 = (SF_1, \mathbf{fET}_1), R_2 = (SF_2, \mathbf{fET}_2)) \stackrel{\text{def}}{=} (1, \mathbf{fET}_3)$$

mit:

$$\mathbf{fET}_3 = \{((BP_1, Z_1), (BP_2, Z_2), (BP_3, Z_3)) : (BP_1, BP_2, BP_3) \in \text{basics}(R_1) \cup \text{basics}(R_2) \wedge \\ Z_i = \max(Zfkt(R_1, BP_i, (BP_1, BP_2, BP_3)), \\ Zfkt(R_2, BP_i, (BP_1, BP_2, BP_3))); \\ i = i \dots 3\}$$

scaled_union : **FRegion** × **FRegion** → **FRegion**

$$\text{scaled_union}(R_1 = (SF_1, \mathbf{fET}_1), R_2 = (SF_2, \mathbf{fET}_2)) \stackrel{\text{def}}{=} \text{norm}((1, \mathbf{fET}_3))$$

mit:

$$\mathbf{fET}_3 = \{((BP_1, Z_1), (BP_2, Z_2), (BP_3, Z_3)) : (BP_1, BP_2, BP_3) \in \text{basics}(R_1) \cup \text{basics}(R_2) \wedge \\ Z_i = \max(SF_1 \cdot Zfkt(R_1, BP_i, (BP_1, BP_2, BP_3)), \\ SF_2 \cdot Zfkt(R_2, BP_i, (BP_1, BP_2, BP_3))); \\ i = i \dots 3\}$$

Vereinigung und Stetigkeit

Werden zwei Regionen oder Linien vereinigt, so kann man feststellen, daß Unstetigkeiten im Ergebnis nur dort auftreten können, wo in den Ausgangsobjekten Unstetigkeiten waren. (Für Punkte stellt sich das Problem von Unstetigkeiten nicht.) Hat ein Objekt keine Unstetigkeit an einer Koordinate (x,y), so besitzt diese genau einen Zugehörigkeitsgrad, den sie durch alle anliegenden unscharfen Elementardreiecke bzw. Segmente bekommt. Haben nun beide Objekte keine Unstetigkeit an (x,y), so gibt es für jedes angrenzende Dreieck(Segment) das gleiche Maximum. Somit besitzen im Ergebnisobjekt alle angrenzenden Dreiecke (Segmente) an dieser Koordinate genau einen Zugehörigkeitsgrad und somit ist die Vereinigung dort stetig. Hat ein Objekt (oder beide) in (x,y) eine Unstetigkeit, so gibt es angrenzende unscharfe Elementardreiecke in diesem Objekt, die an dieser Stelle unterschiedliche Zugehörigkeitsgrade besitzen. Somit ergibt auch das Maximum der Grade für die verschiedenen Dreiecke(Segmente) evtl. unterschiedliche Werte, wodurch im Ergebnis eine Unstetigkeit entstehen kann. Es kann jedoch vorkommen, daß Unstetigkeitsstellen eliminiert werden.

Der Schnitt von Punkten

Um den Schnitt zweier unscharfer Punkte zu berechnen, wird zunächst für alle Koordinaten, die in einem der beiden Punkte enthalten sind der minimale Grad bestimmt, der dieser Koordinate von beiden Punkten zugeordnet wird. Anschließend werden alle Koordinaten entfernt, die dadurch den Grad Null erhalten haben.

intersection : **FPoint** × **FPoint** → **FPoint**

$$\text{intersection}(P^1, P^2) \stackrel{\text{def}}{=} \text{strong-0-cut}((1, \mathbf{fEP}^{\text{new}}))$$

mit:

$$\mathbf{fEP}^{\text{new}} = \{(BP, Z) : BP \in (\text{basics}(P^1) \cup \text{basics}(P^2)) \wedge \\ Z = \min(ZFkt(P^1, BP, BP), ZFkt(P^2, BP, BP))\}$$

Der entsprechende interpretierte Operator kann wie folgt beschrieben werden:

scaled_intersection : **FPoint** × **FPoint** → **FPoint**

$$\text{scaled_intersection}(P^1 = (SF^1, \mathbf{fEP}^1), P^2 = (SF^2, \mathbf{fEP}^2)) \stackrel{\text{def}}{=} \text{norm}((1, \mathbf{fEP}^{\text{new}}))$$

mit:

$$\mathbf{fEP}^{\text{new}} = \{(BP, Z) : BP \in (\text{basics}(P^1) \cup \text{basics}(P^2)) \wedge \\ Z = \min(SF^1 \cdot ZFkt(P^1, BP, BP), SF^2 \cdot ZFkt(P^2, BP, BP))\}$$

Der Schnitt von Linien

Beim Schnitt von Linien werden allen Endpunkten der beteiligten Segmente das Minimum der Grade zugewiesen, die von beiden Operanden stammen. Anschließend werden die Linien entfernt, die an beiden Endpunkten den Grad Null erhalten haben.

$intersection : \mathbf{FLine} \times \mathbf{FLine} \rightarrow \mathbf{FLine}$

$intersection(L^1, L^2) \stackrel{\text{def}}{=} strong-0-cut((1, \mathbf{fS}^{\text{new}}))$

mit:

$$\mathbf{fS}^{\text{new}} = \{((p_1, Z_1), (p_2, Z_2)) : BL = (p_1, p_2) \in basics(L^1) \cup basics(L^2) \wedge Z_i = \min(ZFkt(L^1, p_i, BL), ZFkt(L^2, p_i, BL)); i = 1, 2)\}$$

Entsprechendes gilt auch für die interpretierte Version:

$scaled_intersection : \mathbf{FLine} \times \mathbf{FLine} \rightarrow \mathbf{FLine}$

$scaled_intersection(L^1 = (SF^1, \mathbf{fS}^1), L^2 = (SF^2, \mathbf{fS}^2) \stackrel{\text{def}}{=} norm((1, \mathbf{fS}^{\text{new}}))$

mit:

$$\mathbf{fS}^{\text{new}} = \{((p_1, Z_1), (p_2, Z_2)) : BL = (p_1, p_2) \in basics(L^1) \cup basics(L^2) \wedge Z_i = \min(SF^1 \cdot ZFkt(L^1, p_i, BL), SF^2 \cdot ZFkt(L^2, p_i, BL)); i = 1, 2\}$$

Man beachte, daß sich beim Schnitt zweier unscharfer Mengen die Basis gegenüber dem abstrakten Modell ändern kann:



Abb. 32 Vergleich des Schnitts im diskreten und abstrakten Modell

Da im diskreten Fall das gesamte Segment den Grad Null erhalten hat, wird dieses aus der Menge der Segmente des Resultats entfernt.

Schnitt von Regionen

Der Schnitt von Regionen funktioniert wie der Schnitt zwischen zwei unscharfen Linien, nur daß hier für jedes Dreieck 3 Basispunkte betrachtet werden müssen.

$intersection : \mathbf{FRegion} \times \mathbf{FRegion} \rightarrow \mathbf{FRegion}$

$intersection(R^1, R^2) \stackrel{\text{def}}{=} strong-0-cut((1, \mathbf{fET}^{\text{new}}))$

mit:

$$\mathbf{fET}^{\text{new}} = \{((p_1, Z_1), (p_2, Z_2), (p_3, Z_3)) : BT = (p_1, p_2, p_3) \in (basics(R^1) \cup basics(R^2)) \wedge Z_i = \min(ZFkt(R^1, p_i, BT), ZFkt(R^2, p_i, BT)) i = 1, 2, 3\}$$

Auch die interpretierte Version ähnelt der der Linie stark:

$scaled_intersection : \mathbf{FRegion} \times \mathbf{FRegion} \rightarrow \mathbf{FRegion}$

$scaled_intersection(R^1 = (SF^1, \mathbf{fET}^1), R^2 = (SF^2, \mathbf{fET}^2)) \stackrel{\text{def}}{=} norm((1, \mathbf{fET}^{\text{new}}))$

mit:

$\mathbf{fET}^{\text{new}} = \{(p_1, Z_1), (p_2, Z_2), (p_3, Z_3)\} :$
 $BT = (p_1, p_2, p_3) \in basics(R^1) \cup basics(R^2) \wedge$
 $Z_i = \min(SF^1 \cdot ZFkt(R^1, p_i, BT), SF^2 \cdot ZFkt(R^2, p_i, BT)); i = 1, 2, 3\}$

Auch hier kann sich die Basis gegenüber dem abstrakten Modell ändern. Man betrachte hierzu z.B. die beiden unscharfen Elementardreiecke $(P_1, 0, P_2, 0, P_3, 1)$ und $(P_1, 1, P_2, 1, P_3, 0)$. Im diskreten Modell ergibt sich $(P_1, 0, P_2, 0, P_3, 0)$, womit dieses Dreieck aus dem Ergebnis herausfällt. Im abstrakten Modell hätten alle inneren Punkte einen Grad größer als Null. Daher bleibt dieses Dreieck im abstrakten Modell erhalten.

Schnitt und Stetigkeit

Auch beim Schnitt können Unstetigkeiten im Ergebnis nur dort auftauchen, wo Unstetigkeiten in den Ausgangsobjekten (Regionen/Linien) auftraten. Die Ausführungen sind hier analog zu denen bei der Vereinigung.

Differenz von Punkten

Die nichtinterpretierte Differenz entspricht der klassischen Differenz zweier unscharfer Mengen. Hier werden die beteiligten Zugehörigkeitsgrade subtrahiert. Ergibt sich dadurch ein Wert kleiner als Null, so wird er auf Null gesetzt. Dies geschieht innerhalb der Formel durch den *max*-Operator.

$difference : \mathbf{FPoint} \times \mathbf{FPoint} \rightarrow \mathbf{FPoint}$

$difference(P^1, P^2) \stackrel{\text{def}}{=} strong-0-cut((1, \mathbf{fEP}^{\text{new}}))$

mit:

$\mathbf{fEP}^{\text{new}} = \{(BP, Z) : BP \in basics(P^1) \wedge$
 $Z = \max(0, ZFkt(P^1, BP, BP) - ZFkt(P^2, BP, BP))\}$

Bei der interpretierten Version dieses Operators werden Werte unter Null nicht einfach abgeschnitten. Daher genügt es hier auch nicht, nur die Koordinaten des ersten Punktes zu betrachten. Die Gültigkeit des Ergebnisses wird hier durch die Normierung gewährleistet.

$scaled_difference : \mathbf{FPoint} \times \mathbf{FPoint} \rightarrow \mathbf{FPoint}$

$scaled_difference(P^1 = (SF^1, \mathbf{fEP}^1), P^2 = (SF^2, \mathbf{fEP}^2)) \stackrel{\text{def}}{=} norm((1, \mathbf{fEP}^{\text{new}}))$

mit:

$\mathbf{fEP}^{\text{new}} = \{(BP, Z) : BP \in (basics(P^1) \cup basics(P^2)) \wedge$
 $Z = SF^1 \cdot ZFkt(P^1, BP, BP) - SF^2 \cdot ZFkt(P^2, BP, BP)\}$

Differenz von Linien

Bei der Differenz zweier unscharfer Linien wird mit den Endpunkten der betreffenden Segmente so verfahren, wie mit den Koordinaten eines unscharfen Punktes.

$$\text{difference} : \mathbf{FLine} \times \mathbf{FLine} \rightarrow \mathbf{FLine}$$

$$\text{difference}(L^1, L^2) \stackrel{\text{def}}{=} \text{strong-0-cut}((1, \mathbf{fS}^{\text{new}}))$$

mit:

$$\mathbf{fS}^{\text{new}} = \{((p_1, Z_1), (p_2, Z_2)) : BL = (p_1, p_2) \in \text{basics}(L^1) \wedge Z_i = \max(0, (\text{Zfkt}(L^1, p_i, BL) - \text{Zfkt}(L^2, p_i, BL))) ; i = 1, 2\}$$

Es läßt sich erkennen, daß auch hier Informationsverlust gegenüber dem abstrakten Modell auftreten kann. Dies ist dann der Fall, wenn eine Differenz in den Basispunkten größer ist als Null und sich am anderen Ende des Segments eine negative Differenz ergibt:

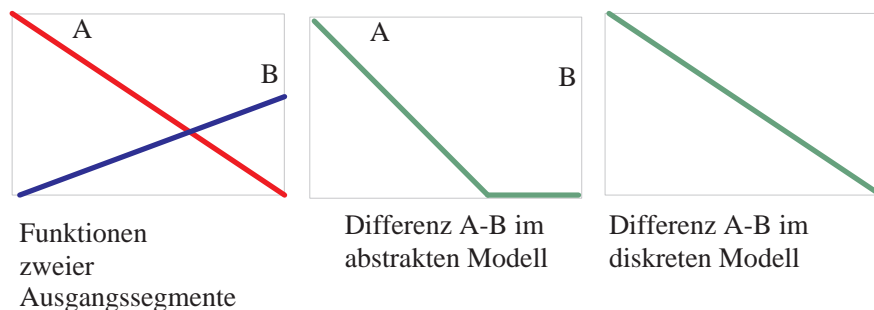


Abb. 33 Vergleich von abstrakter und diskreter Differenz

$$\text{scaled_difference} : \mathbf{FLine} \times \mathbf{FLine} \rightarrow \mathbf{FLine}$$

$$\text{scaled_difference}(L^1 = (SF^1, \mathbf{fS}^1), L^2 = (SF^2, \mathbf{fS}^2)) \stackrel{\text{def}}{=} \text{norm}((1, \mathbf{fS}^{\text{new}}))$$

mit:

$$\mathbf{fS}^{\text{new}} = \{((p_1, Z_1), (p_2, Z_2)) : BL = (p_1, p_2) \in \text{basics}(L^1) \cup \text{basics}(L^2) \wedge Z_i = (SF^1 \cdot \text{Zfkt}(L^1, p_i, BL) - SF^2 \cdot \text{Zfkt}(L^2, p_i, BL)) ; i = 1, 2\}$$

Bei der skalierten Differenz tritt kein Informationsverlust im Inneren eines Segments gegenüber dem abstrakten Modell auf.

Differenz von Regionen

Bei der Differenz von Regionen werden die Eckpunkte jedes Dreiecks so behandelt wie die Koordinaten unscharfer Punkte.

$$\text{difference} : \mathbf{FRegion} \times \mathbf{FRegion} \rightarrow \mathbf{FRegion}$$

$$\text{difference}(R^1, R^2) \stackrel{\text{def}}{=} \text{strong-0-cut}((1, \mathbf{fET}^{\text{new}}))$$

mit:

$$\mathbf{fET}^{\text{new}} = \{((P_1, Z_1), (P_2, Z_2), (P_3, Z_3)) : BT = (P_1, P_2, P_3) \in \text{basics}(R^1) \wedge Z_i = \max(0, \text{Zfkt}(R^1, P_i, BT) - \text{Zfkt}(R^2, P_i, BT)) ; i = 1 \dots 3\}$$

Wie bei Linien kann es hier zum Informationsverlust im Inneren von Dreiecken kommen, wenn es Differenzwerte über und unter Null gibt.

$scaled_difference : \mathbf{FRegion} \times \mathbf{FRegion} \rightarrow \mathbf{FRegion}$

$$scaled_difference(R^1 = (SF^1, \mathbf{fET}^1), R^2 = (SF^2, \mathbf{fET}^2)) \stackrel{\text{def}}{=} norm((1, \mathbf{fET}^{\text{new}}))$$

mit:

$$\mathbf{fET}^{\text{new}} = \{((P_1, Z_1), (P_2, Z_2), (P_3, Z_3)) : BT = (P_1, P_2, P_3) \in basics(R^1) \cup basics(R^2) \wedge \\ Z_i = Zfkt(R^1, P_i, BT) - Zfkt(R^2, P_i, BT) \\ i = 1..3\}$$

Auch bei Regionen tritt kein Informationsverlust im Inneren eines Dreiecks bei diesem Operator auf.

Addition von Punkten

Beim *add*-Operator werden die Zugehörigkeitsgrade der beteiligten Koordinaten aufaddiert. Werte größer als 1 werden hier abgeschnitten. Dafür sorgt der *min*-Operator.

$add : \mathbf{FPoint} \times \mathbf{FPoint} \rightarrow \mathbf{FPoint}$

$$add(P^1 = (SF^1, \mathbf{fEP}^1), P^2 = (SF^2, \mathbf{fEP}^2)) \stackrel{\text{def}}{=} (1, \mathbf{fEP}^{\text{new}})$$

mit:

$$\mathbf{fEP}^{\text{new}} = \{(BP, Z) : BP \in basics(P^1) \cup basics(P^2) \wedge \\ Z = \min(1, Zfkt(P^1, BP, BP) + Zfkt(P^2, BP, BP))\}$$

Bei der skalierten Version werden Werte größer als 1 nicht abgeschnitten, sondern durch den *norm*-Operator angepaßt:

$scaled_add : \mathbf{FPoint} \times \mathbf{FPoint} \rightarrow \mathbf{FPoint}$

$$scaled_add(P^1 = (SF^1, \mathbf{fEP}^1), P^2 = (SF^2, \mathbf{fEP}^2)) \stackrel{\text{def}}{=} norm((1, \mathbf{fEP}^{\text{new}}))$$

mit:

$$\mathbf{fEP}^{\text{new}} = \{(BP, Z) : BP \in basics(P^1) \cup basics(P^2) \wedge \\ Z = SF^1 \cdot Zfkt(P^1, BP, BP) + SF^2 \cdot Zfkt(P^2, BP, BP)\}$$

Addition von Linien

Hier werden die Zugehörigkeitsgrade beider Endpunkte aller beteiligten Segmente aufaddiert. Werte über 1 werden dabei abgeschnitten.

$add : \mathbf{FLine} \times \mathbf{FLine} \rightarrow \mathbf{FLine}$

$$add(L^1 = (SF^1, \mathbf{fS}^1), L^2 = (SF^2, \mathbf{fS}^2)) \stackrel{\text{def}}{=} (1, \mathbf{fS}^{\text{new}})$$

mit:

$$\mathbf{fS}^{\text{new}} = \{((P_1, Z_1), (P_2, Z_2)) : BL = (P_1, P_2) \in basics(L^1) \cup basics(L^2) \wedge \\ Z_i = \min(1, Zfkt(L^1, P_i, BL) + Zfkt(L^2, P_i, BL)); i = 1, 2\}$$

Hier können, wie bei der Differenz Informationsverluste im Inneren von Segmenten auftreten,

wenn eine Summe größer und die Summe am anderen Endpunkt kleiner als 1 ist. Diesen Informationsverlust gibt es in der interpretierten Version dieses Operator nicht mehr:

$scaled_add : \mathbf{FLine} \times \mathbf{FLine} \rightarrow \mathbf{FLine}$

$$scaled_add(L^1 = (SF^1, \mathbf{fS}^1), L^2 = (SF^2, \mathbf{fS}^2)) \stackrel{\text{def}}{=} norm((1, \mathbf{fS}^{\text{new}}))$$

mit:

$$\mathbf{fS}^{\text{new}} = \{((P_1, Z_1), (P_2, Z_2)) : BL = (P_1, P_2) \in basics(L^1) \cup basics(L^2) \wedge \\ Z_i = SF^1 \cdot Zfkt(L^1, P_i, BL) + SF^2 \cdot Zfkt(L^2, P_i, BL); i = 1, 2\}$$

Addition von Regionen

Werden zwei Regionen addiert, so werden die 3 Eckpunkte der beteiligten Dreiecke so behandelt, wie die Koordinaten unscharfer Punkte bei diesem Operator.

$add : \mathbf{FRegion} \times \mathbf{FRegion} \rightarrow \mathbf{FRegion}$

$$add(R^1 = (SF^1, \mathbf{fET}^1), R^2 = (SF^2, \mathbf{fET}^2)) \stackrel{\text{def}}{=} (1, \mathbf{fET}^{\text{new}})$$

mit:

$$\mathbf{fET}^{\text{new}} = \{((P_1, Z_1), (P_2, Z_2), (P_3, Z_3)) : BT = (P_1, P_2, P_3) \in basics(R^1) \cup basics(R^2) \wedge \\ Z_i = \min(1, Zfkt(R^1, P_i, BT) + Zfkt(R^2, P_i, BT)); \\ i = 1..3\}$$

Auch hier kann es zu Informationsverlusten im Inneren von Dreiecken kommen. Die interpretierte Version hat keinen solchen Effekt.

$scaled_add : \mathbf{FRegion} \times \mathbf{FRegion} \rightarrow \mathbf{FRegion}$

$$scaled_add(R^1 = (SF^1, \mathbf{fET}^1), R^2 = (SF^2, \mathbf{fET}^2)) \stackrel{\text{def}}{=} norm((1, \mathbf{fET}^{\text{new}}))$$

mit:

$$\mathbf{fET}^{\text{new}} = \{((P_1, Z_1), (P_2, Z_2), (P_3, Z_3)) : BT = (P_1, P_2, P_3) \in basics(R^1) \cup basics(R^2) \wedge \\ Z_i := Zfkt(R^1, P_i, BT) + Zfkt(R^2, P_i, BT); \\ i = 1..3\}$$

Mittelwert einer Menge von Punkten

Hier wird für jede beteiligte Koordinate die Summe der Zugehörigkeitsgrade der Operanden ermittelt und anschließend durch die Anzahl der Operanden geteilt:

$mid : \text{Set of FPoint} \rightarrow \mathbf{FPoint}$

$$mid(\text{SoFP}) \stackrel{\text{def}}{=} (1, \mathbf{fEP}^{\text{new}})$$

mit:

$$\mathbf{fEP}^{\text{new}} = \{(P, Z) \mid \exists FP \in \text{SoFP} : P \in basics(FP) \wedge Z = \frac{\sum_{FP \in \text{SoFP}} Zfkt(FP, P, P)}{|\text{SoFP}|}\}$$

Bei der interpretierten Version wird genauso mit den skalierten Graden verfahren. Die Normierung sichert die Gültigkeit des Ergebnisses.

scaled_mid : Set of FPoint → FPoint

$$\text{scaled_mid}(SoFP) \stackrel{\text{def}}{=} \text{norm}((1, \mathbf{fEP}^{\text{new}}))$$

mit:

$$\mathbf{fEP}^{\text{new}} = \{(P, Z) \mid \exists FP \in SoFP : P \in \text{basics}(FP) \wedge \\ Z = \frac{\sum_{FP=(SF, \mathbf{fEP}) \in SoFP} Zfkt(FP, P, P) \cdot SF}{|SoFP|}\}$$

Mittelwert einer Menge von unscharfen Linien

Hier wird der mittlere Zugehörigkeitsgrad aller Endpunkte der beteiligten Segmente ermittelt:

mid : Set of FLine → FLine

$$\text{mid}(SoFL) \stackrel{\text{def}}{=} (1, \mathbf{fS}^{\text{new}})$$

mit:

$$\mathbf{fS}^{\text{new}} = \{(P_1, Z_1, P_2, Z_2) \mid \exists FL \in SoFL : BS = (P_1, P_2) \in \text{basics}(FL) \wedge \\ Z_i = \frac{\sum_{FL \in SoFL} Zfkt(FL, P_i, BS)}{|SoFL|} ; i = 1, 2\}$$

Bei der interpretierten Version werden zusätzlich die Zugehörigkeitsgrade berücksichtigt. Durch die Normierung wird gesichert, daß hier keine Grade über 1 im Ergebnis auftauchen können.

scaled_mid : Set of FLine → FLine

$$\text{scaled_mid}(SoFL) \stackrel{\text{def}}{=} \text{norm}((1, \mathbf{fS}^{\text{new}}))$$

mit:

$$\mathbf{fS}^{\text{new}} = \{(P_1, Z_1, P_2, Z_2) \mid \exists FL \in SoFL : BS = (P_1, P_2) \in \text{basics}(FL) \wedge \\ Z_i = \frac{\sum_{FL=(SF, \mathbf{fS}) \in SoFL} Zfkt(FL, P_i, BS) \cdot SF}{|SoFL|} ; i = 1, 2\}$$

Mittelwert einer Menge von Regionen

Im Prinzip arbeitet dieser Operator auf Regionen genauso wie auf Linien:

mid : Set of FRegion → FRegion

$$\text{mid}(SoFR) \stackrel{\text{def}}{=} (1, \mathbf{fT}^{\text{new}})$$

mit:

$$\mathbf{fT}^{\text{new}} = \{(P_1, Z_1, P_2, Z_2, P_3, Z_3) \mid \exists FR \in SoFR : BT \in \text{basics}(FT) \wedge \\ Z_i = \frac{\sum_{FR \in SoFR} Zfkt(FR, P_i, BT)}{|SoFR|} ; i = 1 \dots 3 \wedge \\ BT = (P_1, P_2, P_3)\}$$

Auch bei der interpretierten Version gibt es keine Unterschiede in der Arbeitsweise gegenüber unscharfen Linien:

$scaled_mid : \mathbf{Set\ of\ FRegion} \rightarrow \mathbf{FRegion}$

$$scaled_mid(SoFR) \stackrel{\text{def}}{=} norm((1, \mathbf{fT}^{\text{new}}))$$

mit:

$$\mathbf{fT}^{\text{new}} = \{(P_1, Z_1, P_2, Z_2, P_3, Z_3) \mid \exists FR \in SoFR : BT \in basics(FR) \wedge \\ Z_i = \frac{\sum_{FR=(SF, \mathbf{fT}) \in SoFR} Z_{fkt}(FR, P_i, BT) \cdot SF}{|SoFL|}; i = 1 \dots 3 \wedge \\ BT = (P_1, P_2, P_3)\}$$

Prüfen auf Leere

Um zu prüfen, ob ein unscharfes Objekt leer ist, genügt es zu testen, ob die Menge der unscharfen Elementarobjekte dieses Objekts leer ist. Da es dabei keine Unterschiede zwischen den verschiedenen Objekttypen gibt, kann dies innerhalb eines einzigen Operators getestet werden:

$isempty : \mathbf{FObject} \times \mathbf{FObject} \rightarrow \mathbf{boolean}$

$$isempty(SF, \mathbf{fEO}) \stackrel{\text{def}}{=} (\mathbf{fEO} = \emptyset)$$

Ähnlichkeiten von Objekten in der Basis

In [33] beschreiben M.Molenaar und Tao Cheng, wie durch das Verhältnis der Flächen von Schnitt und Vereinigung eine Ähnlichkeit zwischen Objekten definiert werden kann. Dies läßt sich leicht erweitern, um Ähnlichkeiten zwischen allen hier verwendeten Objekttypen zu beschreiben. Es ergeben sich folgende Operatoren:

$basic_similar : \mathbf{FPoint} \times \mathbf{FPoint} \rightarrow [0, 1]$

$$basic_similar(P^1, P^2) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{wenn } isempty(P^1) \wedge isempty(P^2) \\ \frac{basic_card(intersection(P^1, P^2))}{basic_card(union(P^1, P^2))} & \text{sonst} \end{cases}$$

Man kann hier leicht erkennen, daß dieser Operator genau dann 1 liefert, wenn beiden Punkten die gleiche Basis zugrundeliegt. Haben beide Punkte keine gemeinsame Komponente, liefert der Schnitt einen leeren Punkt (mit Kardinalität 0) und somit liefert der gesamte Operator Null. Der Operator funktioniert für die restlichen Typen analog.

$basic_similar : \mathbf{FLine} \times \mathbf{FLine} \rightarrow [0, 1]$

$$basic_similar(L^1, L^2) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{wenn } isempty(L^1) \wedge isempty(L^2) \\ \frac{basic_len(intersection(L^1, L^2))}{basic_len(union(L^1, L^2))} & \text{sonst} \end{cases}$$

$basic_similar : \mathbf{Region} \times \mathbf{FRegion} \rightarrow [0, 1]$

$$basic_similar(R^1, R^2) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{wenn } isempty(R^1) \wedge isempty(R^2) \\ \frac{basic_area(intersection(R^1, R^2))}{basic_area(union(R^1, R^2))} & \text{sonst} \end{cases}$$

Ähnlichkeit von unscharfen Objekten

Die Ähnlichkeit in der Basis läßt nur teilweise Rückschlüsse auf die Ähnlichkeit der eigentlichen Objekte zu. Daher muß ein weiterer Operator definiert werden, der diese Aufgabe erfüllt. Dieser arbeitet im Prinzip genauso, wie der vorherige Operator, nur daß jeweils die unscharfen Größen verwendet werden müssen.

$$\textit{similar} : \mathbf{FPoint} \times \mathbf{FPoint} \rightarrow [0, 1]$$

$$\textit{similar}(P^1, P^2) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{wenn } \textit{isempty}(P^1) \wedge \textit{isempty}(P^2) \\ \frac{\textit{card}(\textit{intersection}(P^1, P^2))}{\textit{card}(\textit{union}(P^1, P^2))} & \text{sonst} \end{cases}$$

$$\textit{similar} : \mathbf{FLine} \times \mathbf{FLine} \rightarrow [0, 1]$$

$$\textit{similar}(L^1, L^2) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{wenn } \textit{isempty}(L^1) \wedge \textit{isempty}(L^2) \\ \frac{\textit{len}(\textit{intersection}(L^1, L^2))}{\textit{len}(\textit{union}(L^1, L^2))} & \text{sonst} \end{cases}$$

$$\textit{similar} : \mathbf{Region} \times \mathbf{FRegion} \rightarrow [0, 1]$$

$$\textit{similar}(R^1, R^2) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{wenn } \textit{isempty}(R^1) \wedge \textit{isempty}(R^2) \\ \frac{\textit{area}(\textit{intersection}(R^1, R^2))}{\textit{area}(\textit{union}(R^1, R^2))} & \text{sonst} \end{cases}$$

Dieser Operator liefert genau dann 1, wenn beide Objekte an jedem Punkt den gleichen Zugehörigkeitsgrad aufweisen. Sind sie quasi-disjunkt (d.h. sie haben keine gemeinsamen Teile ihrer eigenen Dimension), liefert der Operator 0.

Es kann nun ein weiterer Operator definiert werden, der die Ähnlichkeit der skalierten Objekte berechnet. Dieser trägt den Namen *scaled_similar* und wird für die einzelnen Objekttypen genauso definiert wie *similar*, nur daß *intersection* und *union* durch *scaled_intersection* bzw. *scaled_union* zu ersetzen sind.

Endpunkte von Linien

Der Endpunkt einer Linie läßt sich durch folgenden Operator berechnen:

$$\textit{endpoint} : \mathbf{FLine} \rightarrow \mathbf{FPoint}$$

$$\textit{endpoint}(SF, \mathbf{fS}) \stackrel{\text{def}}{=} (1, \mathbf{fEP})$$

mit:

$$\mathbf{fEP} = \{(BP, 1) : |fS \in \mathbf{fS} : \textit{isendpoint}(\textit{basic}(fS), BP)| = 1\}$$

Die Endpunkte einer Linie sind genau die Koordinaten, die Endpunkte genau eines Basissegments dieser Linie sind.

Man könnte fragen, warum nicht die Zugehörigkeitsgrade der entsprechenden Punkte der Linie in die Punkte übernommen wurden. Dies hat eine recht einfache Erklärung. Jeder Endpunkt einer Linie ist 100%iger Endpunkt der Linie. Würde man den Grad der Linie an diesem Punkt übernehmen, könnte dieser Null sein und demnach keinen Endpunkt mehr darstellen.

Grenze einer unscharfen Region

Durch eine analoge Begründung wie bei Linien, liefert auch der *boundary*-Operator für einer FRegion eine „scharfe“ Grenze.

$boundary : \mathbf{FRegion} \rightarrow \mathbf{FLine}$

$boundary((SF, \mathbf{fET})) \stackrel{\text{def}}{=} (1, \mathbf{fS})$

mit:

$\mathbf{fS} = \{((P_1, 1), (P_2, 1)) : |\{fET \in \mathbf{fET} : isborderline(basic(fET), (P_1, P_2))\}| = 1\}$

Ähnlich wie bei Linien gehört ein Segment genau dann zur Grenze einer Region, wenn es in der Region genau eine Dreiecksbasis gibt, die dieses Segment als Seite hat.

Gemeinsame Punkte zweier Linien

Gemeinsame Koordinaten zweier Linien stellen Schnitt oder Berührungspunkte dieser Linien dar. Jede dieser Koordinaten darf dabei nicht zu einem gemeinsamen Segment beider Linien gehören. Jedoch spielt die Anzahl der Segmente, die mit diesem Punkt verbunden sind keine Rolle:

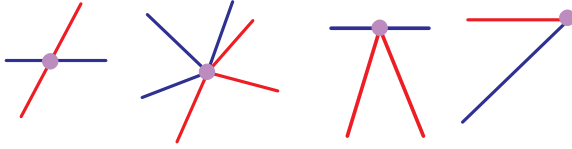


Abb. 34 Gemeinsame Punkte zweier Linien

Der entsprechende Operator läßt sich wie folgt beschreiben:

$commonPoints : \mathbf{FLine} \times \mathbf{FLine} \rightarrow \mathbf{FPoint}$

$commonPoints(L_1, L_2) \stackrel{\text{def}}{=} (1, \mathbf{feP})$

mit:

$\mathbf{feP} = \{(P, 1) \mid \begin{array}{l} \exists BS_1 \in basics(L_1) : isEndpoint(BS_1, P) \wedge \\ \exists BS_2 \in basics(L_2) : isEndpoint(BS_2, P) \wedge \\ \nexists BS_3 \in basics(intersection(sharp(L_1), sharp(L_2))) : isEndPoint(BS_3, P) \end{array}\}$

Gemeinsame Linien von Regionen

Um gemeinsame Linien zweier Regionen zu untersuchen, d.h. gemeinsame Linien, die nicht Teil eines gemeinsamen Dreiecks sind, werden in der Menge aller Basissegmente mehrere Fälle unterschieden.

1.Fall Es handelt sich um ein inneres Segment der einen Region.

Dann gehören beide anliegenden Dreiecke zur Region. Deckt die zweite Region ebenfalls dieses Segment ab, gehört mind. eins der beiden anliegenden Dreiecke zur zweiten Region.

Dann gehört dieses Segment aber zu einem gemeinsamen Dreieck. *widerspruch*

Deckt die andere Region dieses Segment nicht ab, so handelt es sich um kein gemeinsames Segment. *widerspruch*

2.Fall Es handelt sich um kein Segment der ersten Region.

Dann kann es sich nicht um ein gemeinsames Segment handeln. *widerspruch*

3.Fall Das Segment wird von der Grenze der einen Region abgedeckt.

Dann kann dieses Segment innerer, äußerer oder Grenzteil der anderen Region sein. Die ersten beiden Fälle sind symmetrisch zum Fall 1 bzw. Fall 2 und führen daher nicht zum Erfolg. Es verbleibt nur die dritte Möglichkeit, daß es sich ebenfalls um ein Grenzsegment handelt.

Die Bedingung, daß es sich um zwei Grenzsegmente einer Region handelt, genügt jedoch noch nicht. Die jeweils anliegenden Dreiecke müssen auch noch auf verschiedenen Seiten des betrachteten Segments liegen. Um dies zu sichern, kann man feststellen, daß wenn beide anliegenden Dreiecke auf der gleichen Seite liegen, sie sich im Schnitt beider Regionen befinden. Daher gehört das betrachtete Segment in diesem Fall zur Grenze des Schnitts beider Regionen. Im anderen Fall (Dreiecke liegen auf verschiedenen Seiten) gehört das Segment nicht zum Schnitt beider Regionen. Hier kann jedoch nicht der normale *intersection*-Operator verwendet werden. Dazu betrachte man zwei unscharfe Dreiecke $(P_1, 0, P_2, 0, P_3, 1)$ und $(P_1, 1, P_2, 0, P_3, 0)$ der beiden Regionen. Durch die Minimumsbildung ergibt sich $(P_1, 0, P_2, 0, P_3, 0)$, d.h. dieses Segment taucht nicht im Schnitt beider Regionen auf. Um diesem Problem zu begegnen, wird ein Hilfsoperator eingeführt:

***sharp_intersection* : FRegion \times FRegion \rightarrow FRegion**

$sharp_intersection(R_1, R_2) \stackrel{\text{def}}{=} intersection(sharp(R_1), sharp(R_2))$

Damit kann die gemeinsame Linie wie folgt definiert werden:

***commonLine* : FRegion \times FRegion \rightarrow FLine**

$commonLine(R_1, R_2) \stackrel{\text{def}}{=} difference($
 $sharp_intersection(boundary(R_1), boundary(R_2)),$
 $boundary(\sharp_intersection(R_1, R_2)))$

Gemeinsame Punkte von Regionen

Ähnlich wie bei den gemeinsamen Segmenten zweier Regionen, kann man bei den gemeinsamen Punkten von zwei Regionen feststellen, daß sich diese auf den Grenzen beider Regionen befinden. Sie dürfen jedoch nicht Teil gemeinsamer Grenzsegmente sein. Zur vereinfachten Definition dieses Operators wird eine Hilfsfunktion eingeführt, die die Menge der durch eine Region überdeckten Basispunkte zurückliefert:

***basicPoints* : FRegion \rightarrow set of BasicPoint**

$basicPoints(R) \stackrel{\text{def}}{=} \{BP : \exists BT \in basics(R) : iscornerpoint(BT, BP)\}$

Nun können die gemeinsamen Punkte innerhalb des diskreten Modells definiert werden:

***commonPoints* : FRegion \times FRegion \rightarrow FPoint**

$commonPoints(R_1, R_2) \stackrel{\text{def}}{=} (1, \mathbf{fEP})$

mit:

$\mathbf{fEP} = \{(BP, 1) : BP \in (basicPoints(R_1) \cap basicPoints(R_2)) \wedge$
 $\exists BT \in (basics(R_1) \cap basics(R_2)) : iscornerpoint(BT, BP) \wedge$
 $\exists BS \in basics(intersection(boundary(R_1), boundary(R_2))) : isendpoint(BS, BP)\}$

Zusammenhang von Linien und Regionen

In diesem Modell heißen zwei Regionen zusammenhängend, wenn ihre Basis mindestens ein gemeinsames Segment enthält.

$connected : \mathbf{FRegion} \times \mathbf{FRegion} \rightarrow \mathbf{boolean}$

$$connected(R^1, R^2) \stackrel{\text{def}}{=} \exists BT^1 \in basics(R^1), BT^2 \in basics(R^2) : \\ connected(BT^1, BT^2) \vee equal(BT^1, BT^2)$$

Entsprechend kann der Zusammenhang von Linien definiert werden:

$connected : \mathbf{FLine} \times \mathbf{FLine} \rightarrow \mathbf{boolean}$

$$connected(L^1, L^2) \stackrel{\text{def}}{=} \exists BL^1 \in basics(L^1), BL^2 \in basics(L^2) : \\ connected(BL^1, BL^2) \vee equal(BL^1, BL^2)$$

Faces von Regionen und Linien

$faces : \mathbf{FRegion} \rightarrow \mathbf{Set\ of\ FRegion}$

$$faces(R = (SF, \mathbf{fET})) \stackrel{\text{def}}{=} \mathbf{F} = \{(SF, \mathbf{fET}^{\text{new}}) \mid \bigcup \mathbf{fET}^{\text{new}} = \mathbf{fET}\}$$

mit:

$$\forall R^1 \neq R^2 \in (F) : \neg connected(R^1, R^2)$$

Für FLines läßt sich dieser Operator analog definieren:

$faces : \mathbf{FLine} \rightarrow \mathbf{Set\ of\ FLine}$

$$faces(L = (SF, \mathbf{fS})) \stackrel{\text{def}}{=} \mathbf{F} = \{(SF, \mathbf{fS}^{\text{new}}) \mid \bigcup \mathbf{fS}^{\text{new}} = \mathbf{fS}\}$$

mit:

$$\forall L^1 \neq L^2 \in \mathbf{F} : \neg connected(L^1, L^2)$$

Löcher in Regionen:

Um Löcher in Regionen zu definieren, werden zunächst einige Hilfsoperatoren und Hilfsobjekte eingeführt. Zunächst werden einfache Kreise definiert.

$$SimpleCircle = (BS_1, \dots, BS_n)$$

mit:

$$\begin{aligned} &BS_i \in \mathbf{BasicSegment}; i = 1 \dots n \wedge \\ &connected(BS_i, BS_{i+1}); i = 1 \dots n - 1 \wedge \\ &\neg connected(BS_m, BS_k) k, m = 1 \dots n - 1 \wedge |k - m| > 1 \wedge \\ &\neg connected(BS_i, BS_n) i = 2 \dots n - 2 \wedge \\ &connected(BS_1, BS_n) \wedge \\ &BS_1 < BS_i; i = 2 \dots n \end{aligned}$$

Es handelt sich also um eine Folge verbundener Basissegmente, die einen Kreis bilden, jedoch sonst keine Selbstschnitte besitzen. Um die Eindeutigkeit der Darstellung zu sichern, wird zusätzlich gefordert, daß das lexikographisch kleinste Basissegment an der ersten Stelle der Folge steht. Der Operator *containing_Region* ermittelt die in einem solchen Kreis enthaltene Region.

containing_Region : **SimpleCircle** → **FRegion**

$$\text{containing_Region}((BS_1, \dots, BS_n)) \stackrel{\text{def}}{=} R = (1, \mathbf{fET})$$

mit:

$$\mathbf{fET} = \{(P_1, 1, P_2, 1, P_3, 1) : \forall BS \in \text{basics}(\text{boundary}(R)) : \exists i \in \{1, \dots, n\} : \text{equal}(BS, BS_i)\}$$

Ein weiterer Operator liefert die Menge aller Grenzkreise einer Region. Aufgrund der Darstellung der Regionen besteht jede Regionsgrenze aus einer solchen Menge.

boundary_circles : **FRegion** → **Set of SimpleCircle**

$$\text{boundary_circles}((SF, \mathbf{fET})) \stackrel{\text{def}}{=} \{(B_1, \dots, B_n) : \exists F \in \text{faces}(R) : B_i \in \text{basics}(\text{boundary}(F)) \\ i = 1 \dots n\}$$

Mit diesen Hilfsoperatoren lassen sich nun die Löcher von Regionen definieren.

holes : **FRegion** → **FRegion**

$$\text{holes}(R = (SF, \mathbf{fET})) \stackrel{\text{def}}{=} (1, \mathbf{fET}^{\text{new}})$$

mit:

$$\mathbf{fET}^{\text{new}} = \{(P_1, 1, P_2, 1, P_3, 1) : \\ (P_1, P_2, P_3) \in \bigcup_{C \in \text{boundary_circles}(R)} \text{basics}(\text{containing_Region}(C)) \wedge \\ (P_1, P_2, P_3) \notin \text{basics}(R)\}$$

Hier werden also genau die Dreiecke berechnet, die innerhalb der Grenzkreise einer Region liegen, jedoch selbst nicht zur Region gehören.

Da Löcher nun definiert sind, läßt sich auch unproblematisch die Grenze ohne Betrachtung der Löcher berechnen:

contour : **FRegion** → **FLine**

$$\text{contour}(R) \stackrel{\text{def}}{=} \text{boundary}(\text{union}(R, \text{holes}(R)))$$

7.5 Berechnung von topologischen Beziehungen

7.5.1 Beziehungen in der Basis

Dieser Abschnitt befaßt sich mit der Frage, wie die Matrizen berechnet werden können, die topologische Beziehungen in der Basis zweier unscharfer Objekte darstellen. Ist die Matrix berechnet, so können die Ergebnisse des Abschnitts 5.5 verwendet werden, um die entsprechenden topologischen Prädikate zuzuordnen. Im Folgenden werden nur die Beziehungen zwischen zwei nichtleeren Objekten betrachtet.

7.5.1.1 Matrizen für zwei Punkte

Die Matrix BT für die topologische Beziehung der Punkte $A = (SF_A, \mathbf{fEP}_A)$ und $B = (SF_B, \mathbf{fEP}_B)$ wird nun berechnet. Aus dem Abschnitt 5.1 lassen sich die folgenden Matrixeinträge ableiten:

$$BT[\partial A \cap B^\circ] \stackrel{\text{def}}{=} 0$$

$$BT[\partial A \cap \partial B] \stackrel{\text{def}}{=} 0$$

$$BT[\partial A \cap B^{-1}] \stackrel{\text{def}}{=} 0$$

$$BT[A^\circ \cap \partial B] \stackrel{\text{def}}{=} 0$$

$$BT[A^{-1} \cap \partial B] \stackrel{\text{def}}{=} 0$$

$$BT[A^{-1} \cap B^{-1}] \stackrel{\text{def}}{=} 1$$

Während die bisher betrachteten Matrixeinträge unabhängig von den beiden betrachteten Punkten waren, muß bei den restlichen Schnitten die Struktur der Objekte mit berücksichtigt werden. Die Inneren schneiden sich, wenn es eine Koordinate gibt, die zur Basis beider Punkte gehört (aufgrund der leeren Grenze entspricht die Basis genau dem Inneren der Punkte):

$$BT[A^\circ \cap B^\circ] \stackrel{\text{def}}{=} \exists P \in \text{basics}(A) : P \in \text{basics}(B)$$

Zuletzt müssen noch die Schnitte der Inneren mit den Äußeren berechnet werden können. Hier muß es eine Koordinate in der Basis eines Punktes geben, der nicht in der Basis des anderen Punktes auftritt:

$$BT[A^\circ \cap B^{-1}] \stackrel{\text{def}}{=} \exists P : P \in \text{basics}(A) \wedge P \notin \text{basics}(B)$$

Der symmetrische Schnitt $A^{-1} \cap B^\circ$ kann analog berechnet werden.

7.5.1.2 Beziehungen zwischen Punkt und Linie

In diesem Abschnitt werden die topologischen Beziehungen zwischen einem Punkt $P = (SF_P, \mathbf{fEP})$ und einer Linie $L = (SF_L, \mathbf{fS})$ in der Basis betrachtet.

Die Ausführungen von Abschnitt 5.4.1 ergeben zunächst die folgenden Einträge, welche unabhängig von den Objekten sind:

$$BT[P^{-1} \cap L^{-1}] \stackrel{\text{def}}{=} 1$$

$$BT[\partial P \cap L^\circ] \stackrel{\text{def}}{=} 0$$

$$BT[\partial P \cap \partial L] \stackrel{\text{def}}{=} 0$$

$$BT[\partial P \cap L^{-1}] \stackrel{\text{def}}{=} 0$$

$$BT[P^{-1} \cap L^\circ] \stackrel{\text{def}}{=} 1$$

Die restlichen Matrixeinträge berechnen sich wie folgt:

$$BT[P^\circ \cap \partial L] \stackrel{\text{def}}{=} \neg \text{isempty}(\text{intersection}(P, \text{boundary}(L)))$$

Hier handelt es sich um eine sehr direkte Umsetzung der Definition des Matrixeintrages.

$$BT[P^{-1} \cap \partial L] \stackrel{\text{def}}{=} \neg \text{isempty}(\text{difference}(\text{boundary}(L), \text{sharp}(P)))$$

Das Äußere eines Punktes schneidet die Grenze einer Linie genau dann, wenn diese auch Koordinaten abdeckt, welche nicht zum betrachteten Punkt gehören. Dies ist genau dann der Fall, wenn die oben angegebene Differenz nicht leer ist.

Es bezeichne $P^* = (SF^*, \mathbf{feP}^*) = \text{difference}(P, \text{boundary}(L))$, also den Punkt der Menge aller Koordinaten von P, die nicht durch die Grenze der Linie abgedeckt werden. Dann läßt sich der Schnitt der beiden Inneren wie folgt beschreiben:

$$BT[P^\circ \cap L^\circ] \stackrel{\text{def}}{=} \exists(P_0, Z_0) \in \mathbf{feP}^*, (P_1, Z_1, P_2, Z_2) \in \mathbf{fS} : \text{isEndpoint}((P_1, P_2), P_0)$$

Das Innere des Punktes schneidet das Äußere der Linie genau dann, wenn er Koordinaten enthält, welche nicht Endpunkt eines Segments der Linie sind:

$$BT[P^\circ \cap L^{-1}] \stackrel{\text{def}}{=} \exists(P_0, Z_0) \in \mathbf{fEP} : \forall(P_1, Z_1, P_2, Z_2) \in \mathbf{fS} : \neg \text{isEndpoint}((P_1, P_2), P_0)$$

7.5.1.3 Beziehung zwischen Punkt und Region

Es gilt nun, die Beziehung zwischen einem Punkt $P = (SF_P, \mathbf{fEP})$ und einer Region $R = (SF_R, \mathbf{fT})$ zu berechnen. Aus dem Abschnitt 5.4.2 läßt sich folgendes (unabhängig von den betrachteten Objekten) aussagen:

$$BT[\partial P \cap R^\circ] \stackrel{\text{def}}{=} 0$$

$$BT[\partial P \cap \partial R] \stackrel{\text{def}}{=} 0$$

$$BT[\partial P \cap R^{-1}] \stackrel{\text{def}}{=} 0$$

$$BT[P^{-1} \cap R^\circ] \stackrel{\text{def}}{=} 1$$

$$BT[P^{-1} \cap \partial R] \stackrel{\text{def}}{=} 1$$

$$BT[P^{-1} \cap R^{-1}] \stackrel{\text{def}}{=} 1$$

Die beiden Inneren schneiden sich, wenn der Punkt eine Koordinate enthält, die Eckpunkt eines in der Region enthaltenen Dreiecks ist und dieser Eckpunkt nicht in der Grenze der Region enthalten ist.

$$BT[P^\circ \cap R^\circ] \stackrel{\text{def}}{=} \exists(P_0, Z_0) \in \mathbf{fEP}, (P_1, Z_1, P_2, Z_2, P_3, Z_3) \in \mathbf{fT} : \\ \text{isCornerpoint}((P_1, P_2, P_3), P_0) \wedge \\ \exists(P_a, P_b) \in \text{basics}(\text{boundary}(R)) : \text{isEndpoint}((P_a, P_b), P_0)$$

Man kann feststellen, daß die Grenze einer Region eine Linie ohne Endpunkte ist, da sie aus einem oder mehreren geschlossenen Kurven besteht. Daraus läßt sich ableiten:

$$BT[P^\circ \cap \partial R] \stackrel{\text{def}}{=} BT[P^\circ \cap (\text{boundary}(R)^\circ)]$$

Ein Punkt schneidet das Äußere einer Region genau dann, wenn er eine Koordinate enthält, die nicht Eckpunkt eines Dreiecks der Region ist.

$$BT[P^\circ \cap R^{-1}] \stackrel{\text{def}}{=} \exists(P_0, Z_0) \in \mathbf{fEP} : \\ \forall(P_1, P_1, P_2, Z_2, P_3, Z_3) \in \mathbf{fT} : \neg \text{isCornerpoint}((P_1, P_2, P_3), P_0)$$

7.5.1.4 Beziehungen zwischen zwei Linien

In diesem Abschnitt werden die topologischen Beziehungen zwischen den beiden Linien $A = (SF_A, \mathbf{fS}_A)$ und $B = (SF_B, \mathbf{fS}_B)$ untersucht.

Nach Abschnitt 5.2 gilt :

$$BT[A^{-1} \cap B^{-1}] \stackrel{\text{def}}{=} 1$$

Der Schnitt beider Grenzen kann wie folgt berechnet werden:

$$BT[\partial A \cap \partial B] \stackrel{\text{def}}{=} \neg \text{isempty}(\text{intersection}(\text{boundary}(A), \text{boundary}(B)))$$

Um den Schnitt beider Inneren auf Leere zu prüfen, genügt ein Test auf gemeinsame Segmente nicht. Zwei sich kreuzende Linien haben ebenfalls einen gemeinsamen inneren Punkt. Somit muß auch getestet werden, ob die Linien einen gemeinsamen Punkt haben, der nicht zur Grenze einer der Linien gehört. Es bezeichne $\mathbf{BPs} = \{P | P \in \text{basics}(\text{boundary}(A) \cup \text{boundary}(B))\}$ die Menge aller Grenzpunkte der Linien A und B. Dann ist der Schnitt beider Inneren gegeben durch:

$$\begin{aligned} BT[A^\circ \cap B^\circ] \stackrel{\text{def}}{=} & (\exists P, (P_{A1}, Z_{A1}, P_{A2}, Z_{A2}) \in \mathbf{fS}_A, (P_{B1}, Z_{B1}, P_{B2}, Z_{B2}) \in \mathbf{fS}_B : \\ & \text{isEndpoint}((P_{A1}, P_{A2}), P) \wedge \text{isEndpoint}((P_{B1}, P_{B2}), P) \wedge P \notin \mathbf{BPs}) \\ \vee \\ & (\exists (P_{A1}, Z_{A1}, P_{A2}, Z_{A2}) \in \mathbf{fS}_A, (P_{B1}, Z_{B1}, P_{B2}, Z_{B2}) \in \mathbf{fS}_B : \\ & \text{isequal}((P_{A1}, P_{A2}), (P_{B1}, P_{B2}))) \end{aligned}$$

Für den Schnitttest des Inneren mit dem Äußeren genügt es zu überprüfen, daß eine Linie in ihrer Basis ein Segment besitzt, welches jedoch nicht zur Basis der anderen Linie gehört. Daher ergeben sich die beiden symmetrischen Fälle:

$$BT[A^\circ \cap B^{-1}] \stackrel{\text{def}}{=} \exists BS \in \text{basics}(A) : BS \notin \text{basics}(B)$$

$$BT[A^{-1} \cap B^\circ] \stackrel{\text{def}}{=} \exists BS \in \text{basics}(B) : BS \notin \text{basics}(A)$$

Der Schnitt zwischen Grenze und Innerem läßt sich wie folgt beschreiben:

Es bezeichne $(SF^*, \mathbf{fEP}^*) = \text{difference}(\text{boundary}(A), \text{boundary}(B))$ die Menge aller Grenzpunkte von A, die nicht zur Grenze von B gehören. Weiterhin bezeichne $B_B = \text{basics}(\text{boundary}(B))$ die Menge aller Grenzpunkte von B. Dann gilt:

$$BT[\partial A \cap B^\circ] \stackrel{\text{def}}{=} \exists (P, Z) \in \mathbf{fEP}^*, (P_1, Z_1, P_2, Z_2) \in \mathbf{fS}_B : \text{isEndPoint}((P_1, P_2), P) \wedge P \notin B_B$$

Die symmetrische Beziehung $A^\circ \cap \partial B$ läßt sich analog berechnen.

Nun fehlt noch die Berechnung des Schnitts von Grenze und Äußeren:

$$BT[\partial A \cap B^{-1}] \stackrel{\text{def}}{=} \exists (P, Z) \in \mathbf{fEP}^* : \forall (P_1, Z_1, P_2, Z_2) \in \mathbf{fS}_B : \neg \text{isEndPoint}((P_1, P_2), P)$$

Der umgekehrte Schnitt berechnet sich analog.

7.5.1.5 Beziehungen zwischen Linie und Region

Hier geht es um die Berechnung der topologischen Beziehung zwischen einer Linie $L = (SF_L, \mathbf{fS})$ und einer Region $R = (SF_R, \mathbf{fT})$.

Durch die Betrachtungen in Abschnitt 5.4.3 können zunächst folgende Matrixeinträge festgelegt werden:

$$BT[L^{-1} \cap R^{-1}] \stackrel{\text{def}}{=} 1$$

$$BT[L^{-1} \cap R^\circ] \stackrel{\text{def}}{=} 1$$

Da die Grenze einer Region aus einer (oder mehr) geschlossenen Kurven besteht, läßt sich der Schnitt beider Grenzen wie folgt definieren:

$$BT[\partial L \cap \partial R] \stackrel{\text{def}}{=} BT[(\text{boundary}(L))^\circ \cap (\text{boundary}(R))^\circ]$$

Es wird also der Matrixeintrag zwischen einem Punkt und einer Linie übernommen.

Der Schnitt beider Inneren ist in der Basis genau dann nicht leer, wenn die Linie ein Basissegment besitzt, welches eine Seite eines Basisdreiecks der Region darstellt, die nicht zur Grenze der Region gehört:

$$BT[L^\circ \cap R^\circ] \stackrel{\text{def}}{=} \exists BS \in \text{basics}(L), BT \in \text{basics}(R) : \text{isSide}(BT, BS) \wedge \\ BS \notin \text{basics}(\text{boundary}(R))$$

Beim Schnitt des Inneren der Linie mit dem Äußeren der Region läßt sich der entsprechende Matrixeintrag für zwei Linien verwenden, da die Grenze der Region nur aus inneren Punkten besteht:

$$BT[L^\circ \cap \partial R] \stackrel{\text{def}}{=} BT[L^\circ, (\text{boundary}(R))^\circ]$$

Das Innere einer Linie schneidet das Äußere der Region, wenn ein Basissegment der Linie nicht zu einer Seite eines Basisdreiecks der Region gehört:

$$BT[L^\circ \cap R^{-1}] \stackrel{\text{def}}{=} \exists BS \in \text{basics}(L) : \forall BT \in \text{basics}(R) : \neg \text{isSide}(BT, BS)$$

Die Grenze einer Linie schneidet das Innere der Region genau dann, wenn ein Grenzpunkt der Linie einen Eckpunkt eines zur Region gehörenden Basisdreiecks darstellt, der jedoch nicht zur Grenze der Region gehören darf:

$$BT[\partial L \cap R^\circ] \stackrel{\text{def}}{=} \exists P \in \text{basics}(\text{boundary}(L)), BT \in \text{basics}(R) : \text{isCornerPoint}(BT, P) \wedge \\ \forall BS \in \text{basics}(\text{boundary}(R)) : \neg \text{isEndPoint}(S, P)$$

Der Test, ob ein Grenzpunkt der Linie im Äußeren der Region liegt, läßt sich auf die Berechnung des Matrixeintrages zwischen einem Punkt und einer Region zurückführen:

$$BT[\partial L \cap R^{-1}] \stackrel{\text{def}}{=} BT[(\text{boundary}(L))^\circ \cap R^{-1}]$$

Das Äußere der Linie schneidet die Grenze der Region genau dann, wenn die entsprechende Differenz nicht leer ist:

$$BT[L^{-1} \cap \partial R] \stackrel{\text{def}}{=} \neg \text{isempty}(\text{difference}(\text{boundary}(R), \text{sharp}(L)))$$

7.5.1.6 Topologische Beziehungen zwischen zwei Regionen

Es wird die 9-Intersection-Matrix zwischen den beiden Regionen $A = (SF_A, \mathbf{fT}_A)$ und $B = (SF_B, \mathbf{fT}_B)$ berechnet.

Es gelten die Ausführungen des Abschnitts 5.3:

$$BT[A^{-1} \cap B^{-1}] \stackrel{\text{def}}{=} 1$$

Die Inneren zweier Regionen schneiden sich genau dann, wenn sie ein gemeinsames Basisdreieck besitzen.

$$BT[A^\circ B^\circ] \stackrel{\text{def}}{=} \exists BT \in \text{basics}(A) : BT \in \text{basics}(B)$$

Es gilt weiterhin:

$$BT[A^\circ \cap \partial B] \stackrel{\text{def}}{=} BT[A^\circ \cap (\text{boundary}(R))^\circ]$$

Wieder kann der symmetrische Eintrag analog berechnet werden.

$$BT[A^\circ \cap B^{-1}] \stackrel{\text{def}}{=} \exists BT \in \text{basic}(A) : BT \notin \text{basic}(B)$$

Auch hier wird der umgekehrte Fall analog betrachtet.

$$BT[\partial A \cap \partial B] \stackrel{\text{def}}{=} BT[(\text{boundary}(A))^\circ, (\text{boundary}(B))^\circ]$$

Es verbleibt, den Schnitt von Grenze und Äußerem zu betrachten: Es bezeichne

$$S^A = \{BS \mid \exists BT \in \text{basics}A : \text{isSide}(BT, BS)\}$$

die Menge aller der durch die Region A abgedeckten Basissegmente.

$$BT[\partial A \cap B^{-1}] \stackrel{\text{def}}{=} \exists BS \in \text{basics}(\text{boundary}(A)) : BS \notin S^B$$

7.5.2 Beziehungen zwischen den unscharfen Objekten

Während sich der letzte Abschnitt mit der Topologie in der Basis beschäftigte, wird nun die unscharfe Topologie innerhalb des diskreten Modells betrachtet. Genauer gesagt, es wird angegeben wie die einzelnen 6-Tupel der Clusterung (vgl. Abschnitt 6.7) innerhalb des diskreten Modells berechnet werden können. Hier werden also die Objekte nicht mehr in Grenze und Inneres aufgespalten, sondern als Gesamtheit betrachtet. Dazu wird untersucht, welche der Prädikate des Abschnitts 6 erfüllt werden. Im diskreten Modell gibt es noch das Problem, daß eine Koordinate des \mathbb{R}^2 auch mehr als einen Grad besitzen kann, was im abstrakten Modell nicht möglich ist. In einem solchen Fall wird stets der maximal zugeordnete Grad zur Bestimmung der topologischen Beziehung herangezogen.

7.5.2.1 Unscharfe Topologie zweier Punkte

Es werden die beiden Punkte $A = (SF_A, \mathbf{fEP}_A)$ und $B = (SF_B, \mathbf{fEP}_B)$ betrachtet. Da die Grenze eines Punktes stets leer ist, kann hierdurch kein Prädikat erfüllt werden, da stets vom Vorhandensein einer Koordinate ausgegangen wird. Es müssen demnach nur noch die gemeinsamen inneren Koordinaten untersucht werden. Man kann feststellen, daß die ersten drei Prädikate bei Punkten nicht erfüllt werden können, da deren Koordinaten keinen Zugehörigkeitsgrad von 0 haben dürfen. Daher gilt:

$$Q1(A, B) \stackrel{\text{def}}{=} 0$$

$$Q2(A, B) \stackrel{\text{def}}{=} 0$$

$$Q3(A, B) \stackrel{\text{def}}{=} 0$$

Auch die weiten Einträge sind leicht zu ermitteln:

$$Q4(A, B) \stackrel{\text{def}}{=} \exists (P, Z) \in \mathbf{fEP}_A : (P, Z) \in \mathbf{fEP}_B$$

$$Q5(A, B) \stackrel{\text{def}}{=} \exists (P, Z_1) \in \mathbf{fEP}_A, (P, Z_2) \in \mathbf{fEP}_B : Z_1 > Z_2$$

$$Q6(A, B) \stackrel{\text{def}}{=} \exists (P, Z_1) \in \mathbf{fEP}_A, (P, Z_2) \in \mathbf{fEP}_B : Z_1 < Z_2$$

7.5.2.2 Unscharfe Topologie zwischen Punkt und Linie

Es seien $P = (SF_P, \mathbf{fEP})$ und $L = (SF_L, \mathbf{fS})$ die beiden betrachteten unscharfen Objekte. Man erkennt, daß die Prädikate Q1 und Q3 niemals erfüllt sein können, da ein unscharfer Punkt keine Koordinaten mit dem Grad 0 zuläßt. Daher ergibt sich:

$$Q1(P, L) \stackrel{\text{def}}{=} 0$$

$$Q3(P, L) \stackrel{\text{def}}{=} 0$$

Die weiteren Prädikate erhalten folgende Werte:

$$Q2(P, L) \stackrel{\text{def}}{=} \exists(P_0, Z_0) \in \mathbf{fEP}, (P_1, Z_1, P_2, Z_2) \in \mathbf{fS} : (\text{equal}(P_0, P_1) \vee \text{equal}(P_0, P_2)) \wedge \text{maxZfkt}(L, P_0) = 0$$

$$Q4(P, L) \stackrel{\text{def}}{=} \exists(P_0, Z_0) \in \mathbf{fEP} : \text{maxZfkt}(L, P_0) = Z_0$$

$$Q5(P, L) \stackrel{\text{def}}{=} \exists(P_0, Z_0) \in \mathbf{fEP} : \text{maxZfkt}(L, P_0) > 0 \wedge \text{maxZfkt}(L, P_0) < Z_0$$

$$Q6(P, L) \stackrel{\text{def}}{=} \exists(P_0, Z_0) \in \mathbf{fEP} : \text{maxZfkt}(L, P_0) > Z_0$$

7.5.2.3 Unscharfe Topologie zwischen Punkt und Region

Die betrachteten Objekte seien $P = (SF_P, \mathbf{fEP})$ und $R = (SF_R, \mathbf{fT})$. Mit einer analogen Begründung wie oben ergibt sich:

$$Q1(P, R) \stackrel{\text{def}}{=} 0$$

$$Q3(P, R) \stackrel{\text{def}}{=} 0$$

Desweiteren erhält man:

$$Q2(P, R) \stackrel{\text{def}}{=} \exists(P_0, Z_0) \in \mathbf{fEP}, (P_1, Z_1, P_2, Z_2, P_3, Z_3) \in \mathbf{fET} : \text{equals}(P_0, P_i); i \in \{1..3\} \wedge \text{maxZfkt}(L, P_0) = 0$$

$$Q4(P, R) \stackrel{\text{def}}{=} \exists(P_0, Z_0) \in \mathbf{fEP} : \text{maxZfkt}(R, P_0) = Z_0$$

$$Q5(P, R) \stackrel{\text{def}}{=} \exists(P_0, Z_0) \in \mathbf{fEP} : \text{maxZfkt}(R, P_0) > 0 \wedge \text{maxZfkt}(R, P_0) < Z_0$$

$$Q6(P, R) \stackrel{\text{def}}{=} \exists(P_0, Z_0) \in \mathbf{fEP} : \text{maxZfkt}(R, P_0) > Z_0$$

7.5.2.4 Unscharfe Topologie zwischen zwei Linien

Hier werden die Formeln zur Berechnung der 6 Prädikate für zwei unscharfe Linien $A = (SF_A, \mathbf{fS}_A)$ und $B = (SF_B, \mathbf{fS}_B)$ angegeben.

Bei unscharfen Linien kann man beobachten, daß zwar Grade von Null auftreten können, dies jedoch nur auf Basispunkten (d.h. Endpunkten von Segmenten) der Fall sein kann. Es bezeichne \mathbf{Pts}_A die Menge aller durch Linie A überdeckten Basispunkte:

$$\mathbf{Pts}_A = \{P : \exists BS \in \text{basics}(A) : \text{isEndpoint}(BS, P)\}$$

Entsprechendes gelte für \mathbf{Pts}_B .

$$Q1(A, B) \stackrel{\text{def}}{=} \exists P \in (\mathbf{Pts}_A \cap \mathbf{Pts}_B) : \text{maxZfkt}(A, P) = 0 \wedge \text{maxZfkt}(B, P) = 0$$

$$Q2(A, B) \stackrel{\text{def}}{=} \exists P \in (\mathbf{Pts}_A \cap \mathbf{Pts}_B) : \max Zfkt(A, P) > 0 \wedge \max Zfkt(B, P) = 0$$

$$Q3(A, B) \stackrel{\text{def}}{=} \exists P \in (\mathbf{Pts}_A \cap \mathbf{Pts}_B) : \max Zfkt(A, P) = 0 \wedge \max Zfkt(B, P) > 0$$

Man kann weiterhin feststellen, daß es gemeinsame Punkte mit Graden größer als Null auch jenseits von Basispunkten geben kann. Zwei Grade können dann gleich sein, wenn es einen gemeinsamen Basispunkt mit gleichem maximalen Grad gibt oder sich die Zugehörigkeitsfunktionen über einem gemeinsamen Segment schneiden. Es wird zwischen diesen beiden Fällen unterschieden.

$$Q4_1(A, B) \stackrel{\text{def}}{=} \exists P \in (\mathbf{Pts}_A \cap \mathbf{Pts}_B) : (\max Zfkt(A, P) = \max Zfkt(B, P)) \wedge (\max Zfkt(A, P) > 0)$$

Dieses Prädikat beschreibt die Existenz eines gemeinsamen Basispunktes, der für beide Objekte den gleichen Grad (> 0) aufweist.

$$Q4_2(A, B) \stackrel{\text{def}}{=} \exists (P_1, Z_{A1}, P_2, Z_{A2}) \in \mathbf{fS}_A, (P_1, Z_{B1}, P_2, Z_{B2}) \in \mathbf{fS}_B : \\ ((Z_{A1} - Z_{B1}) * (Z_{A2} - Z_{B2}) < 0) \vee ((Z_{A1} = Z_{B1}) \wedge (Z_{A2} = Z_{B2}))$$

Hier wird die Existenz eines gemeinsamen Basissegments beschrieben, über dem sich die Zugehörigkeitsfunktionen kreuzen oder gleich sind. Insgesamt ergibt sich:

$$Q4(A, B) = Q4_1(A, B) \vee Q4_2(A, B)$$

Ähnlich muß auch bei den letzten beiden Prädikaten zwischen Basispunkten und inneren Punkten der Segmente unterschieden werden, da den Basispunkten mehrere Grade zugeordnet sein können und hier stets die maximalen Grade verwendet werden. Wie schon bei $Q4$ werden diese Fälle einzeln betrachtet.

$$Q5_1(A, B) \stackrel{\text{def}}{=} \exists P \in (\mathbf{Pts}_A \cap \mathbf{Pts}_B) : (\max Zfkt(A, P) > \max Zfkt(B, P)) \wedge (\max Zfkt(B, P) > 0)$$

$$Q5_2(A, B) \stackrel{\text{def}}{=} \exists (P_1, Z_A^1, P_2, Z_A^2) \in \mathbf{fS}_A, (P_1, Z_B^1, P_2, Z_B^2) \in \mathbf{fS}_B : (Z_A^1 > Z_B^1) \vee (Z_A^2 > Z_B^2)$$

$$Q5(A, B) \stackrel{\text{def}}{=} Q5_1(A, B) \vee Q5_2(A, B)$$

$$Q6_1(A, B) \stackrel{\text{def}}{=} \exists P \in (Pts_A \cap Pts_B) : (\max Zfkt(A, P) < \max Zfkt(B, P)) \wedge (\max Zfkt(A, P) > 0)$$

$$Q6_2(A, B) \stackrel{\text{def}}{=} \exists (P_1, Z_A^1, P_2, Z_A^2) \in \mathbf{fS}_A, (P_1, Z_B^1, P_2, Z_B^2) \in \mathbf{fS}_B : (Z_A^1 < Z_B^1) \vee (Z_A^2 < Z_B^2)$$

$$Q6(A, B) \stackrel{\text{def}}{=} Q6_1(A, B) \vee Q6_2(A, B)$$

7.5.2.5 Beziehungen zwischen Linie und Region

In diesem Abschnitt werden die topologischen Beziehungen einer unscharfen Linie $L = (SF_L, \mathbf{fS})$ und der unscharfen Region $R = (SF_R, \mathbf{fT})$ untersucht. Zunächst kann man feststellen, daß bei Regionen alle inneren Punkte von Dreiecken Grade größer als Null besitzen. Ebenso verhält es sich bei Linien, wodurch diese nur auf Basispunkten einen Grad von 0 haben können. Es bezeichne \mathbf{Pts}_L die Menge der durch die Linie L abgedeckten Basispunkte und \mathbf{Pts}_R die Basispunkte innerhalb von R:

$$\mathbf{Pts}_L \stackrel{\text{def}}{=} \{P : \exists BS \in \text{basics}(L) : \text{isEndpoint}(BS, P)\}$$

$$\mathbf{Pts}_R \stackrel{\text{def}}{=} \{P : \exists BT \in \text{basics}(R) : \text{isCornerpoint}(BT, P)\}$$

Aus den obigen Überlegungen ergibt sich, daß Q1 nur auf Basispunkten erfüllt sein kann:

$$Q1(L, R) \stackrel{\text{def}}{=} \exists P \in (\mathbf{Pts}_L \cap \mathbf{Pts}_R) : \max Zfkt(L, P) = 0 \wedge \max Zfkt(R, P) = 0$$

Q2 kann ebenfalls auf einzelnen Basispunkten gelten, jedoch auch auf inneren Punkten von Segmenten. Zur Übersichtlichkeit wird Q1 in diese beiden Teile aufgespalten:

$$Q2_1(L, R) \stackrel{\text{def}}{=} \exists P \in (\mathbf{Pts}_L \cap \mathbf{Pts}_R) : \max Zfkt(L, P) > 0 \wedge \max Zfkt(R, P) = 0$$

Es bezeichne \mathbf{S}_R die Menge aller durch R überdeckten Segmente.

$$\mathbf{S}_R \stackrel{\text{def}}{=} \{BS : \exists BT \in \text{basics}(R) : \text{isSide}(BT, BS)\}$$

Dann lassen sich die inneren Punkte von Segmenten wie folgt betrachten:

$$Q2_2(L, R) \stackrel{\text{def}}{=} \exists BS = (P_1, P_2) \in (\text{basics}(L) \cap \mathbf{S}_R) : \forall BT \in \text{basics}(R) : \neg \text{isSide}(BT, BS) \vee Zfkt(R, P_2, BT) = 0 \wedge Zfkt(R, P_1, BT) = 0$$

Insgesamt ergibt sich dann:

$$Q2(L, R) \stackrel{\text{def}}{=} Q2_1(L, R) \vee Q2_2(L, R)$$

Q3 kann wieder nur über Basispunkten erfüllt sein:

$$Q3(L, R) \stackrel{\text{def}}{=} \exists P \in (\mathbf{Pts}_L \cap \mathbf{Pts}_R) : \max Zfkt(L, P) = 0 \wedge \max Zfkt(R, P) > 0$$

Die Grade von L und R über einem Punkt des \mathbb{R}^2 können bei Basispunkten oder bei inneren Segmentpunkten gleich sein. Wie schon früher angewendet, werden diese beiden Fälle zur besseren Lesbarkeit unterschieden:

$$Q4_1(L, R) \stackrel{\text{def}}{=} \exists P \in (\mathbf{Pts}_L \cap \mathbf{Pts}_R) : \max Zfkt(L, P) = \max Zfkt(R, P) \wedge \max Zfkt(L, P) > 0$$

Bei den inneren Punkten stellt sich das Problem, daß nicht nur zwei Funktionsverläufe betrachtet werden müssen, sondern bis zu drei unterschiedliche Funktionen beteiligt sein können. Dabei handelt es sich zum einen um die Funktion, die dem Segment durch die Linie zugeordnet wird, zum anderen steuern die beiden anliegenden Dreiecke evtl. unterschiedliche Zugehörigkeitsfunktionen bei. Es lassen sich zwei „Unterfälle“ unterscheiden. Im ersten Fall, wird die Lage der Zugehörigkeitsgrade im Inneren des Segments betrachtet, wenn die Funktionen der beiden anliegenden Dreiecke gleich sind oder eine Funktion stets über der anderen liegt. Im zweiten Fall kreuzen sich die beiden Funktionen. Während im ersten Fall die Betrachtungen von zwei Segmenten übernommen werden können, ist es im zweiten Fall notwendig, den Kreuzungspunkt der beiden Funktionen zu berücksichtigen. Folgendes Bild soll dies veranschaulichen:

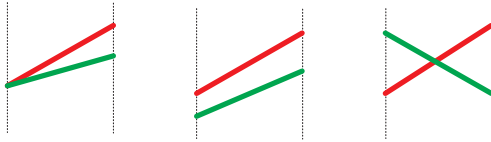


Abb. 35 Verlauf von Zugehörigkeitsfunktionen über einem Segment

Hier sind mehrere Funktionsverläufe eingezeichnet, die zwei Dreiecke einer Region einem gemeinsamen Segment zuordnen können. Man erkennt, daß in den linken beiden Fällen der maximale Zugehörigkeitsgrad eines inneren Punktes nur durch eins der beiden Dreiecke bestimmt wird. Bei den sich kreuzenden Funktionen sind dagegen beide Dreiecke beteiligt.

Somit ist es sinnvoll, hier die Fälle zu unterscheiden, ob sich die Funktionen der Dreiecke kreuzen oder nicht. Dazu wird ein Hilfsprädikat eingeführt, welches **true** ergibt, wenn sich die Funktionsverläufe der beiden an einem Basissegment anliegenden Dreiecke kreuzen.

$crossing : \mathbf{FRegion} \times \mathbf{BasicSegment} \rightarrow \mathbf{boolean}$

$$crossing(R = (SF, \mathbf{fT}), BS = (P_1, P_2)) \stackrel{\text{def}}{=} \exists FT_1, FT_2 \in \mathbf{fT} : isSide(FT_1, BS) \wedge isSide(FT_2, BS) \wedge (Zfkt(R, P_1, FT_1) - Zfkt(R, P_1, FT_2)) * (Zfkt(R, P_2, FT_1) - Zfkt(R, P_2, FT_2)) < 0$$

Damit lassen sich die beiden oben beschriebenen Fälle für ein beliebiges Segment unterscheiden.

$$Q_{4_2S}(L, R, S) \stackrel{\text{def}}{=} \begin{cases} Q_{4_2S_{cross}}(L, R, S) & \text{wenn } crossing(R, S) \wedge S \in (basics(L) \cap S_R) \\ Q_{4_2S_{simple}}(L, R, S) & \text{wenn } \neg crossing(R, S) \wedge S \in (basics(L) \cap S_R) \\ false & \text{sonst} \end{cases}$$

Um den einfachen Fall (also sich nicht kreuzende Funktionen) zu berechnen, wird eine weitere Hilfsfunktion eingeführt. Diese gibt zu einem Basissegment das unscharfe Segment einer unscharfen Region zurück, welche durch die maximalen Zugehörigkeitsgrade der anliegenden Dreiecke bestimmt ist:

$maxSegment : \mathbf{FRegion} \times \mathbf{BasicSegment} \rightarrow \mathbf{FSegment}$

$$maxSegment(R, BS) \stackrel{\text{def}}{=} (P_1, Z_1, P_2, Z_2) : equals((P_1, P_2), BS) \wedge Z_i = Max\{Z : Z = Zfkt(R, P_i, BT) \wedge isSide(BT, BS); i = 1, 2\}$$

$$Q_{4_2S_{simple}}((SF_L, \mathbf{fS}), R, S = (P_1, P_2)) \stackrel{\text{def}}{=} \exists (P_1, Z_1, P_2, Z_2) \in \mathbf{fS}, (P_1, Z_3, P_2, Z_4) = maxSegment(R, S) : ((Z_1 - Z_3) * (Z_2 - Z_4) < 0) \vee ((Z_1 = Z_3) \wedge (Z_2 = Z_4))$$

Bei den sich kreuzenden Linien muß berechnet werden, welchen Zugehörigkeitsgrad die Region am Schnittpunkt der beiden Funktionen hat. Ebenfalls muß der Grad der unscharfen Linie über diesen Punkt bestimmt werden. Dabei ist es unerheblich, wo sich dieser (im Inneren des Segments) Schnittpunkt befindet. Da nur Zugehörigkeitsgrade über einem gemeinsamen Segment betrachtet werden, ist es möglich, den Funktionsverlauf als eine Funktion von \mathbb{R} nach \mathbb{R} zu betrachten. Man kann die drei Funktionen dann wie folgt beschreiben:

$$S_1 = \frac{Z_{2S_1} - Z_{1S_1}}{len} \cdot x + Z_{1S_1}$$

$$S_2 = \frac{Z_{2S_2} - Z_{1S_2}}{len} \cdot x + Z_{1S_2}$$

$$S_3 = \frac{Z_{2S_3} - Z_{1S_3}}{len} \cdot x + Z_{1S_3}$$

Die Bedeutung der Variablen für die einzelnen Funktionen ist der nachfolgenden Skizze zu entnehmen:

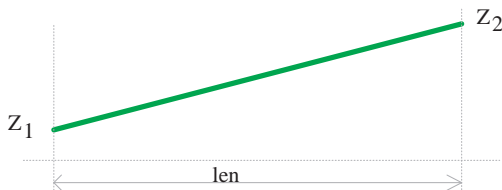


Abb. 36 Bedeutung der Variablen

S_1 und S_2 bezeichnen die sich kreuzenden Funktionen der beiden Dreiecke und S_3 die Funktion des unsharpen Segments der Linie. Da der Schnittpunkt der Funktionen berechnet werden soll, sind diese gleichzusetzen:

$$S_1 = S_2$$

$$\frac{(Z_{2S_1} - Z_{1S_1})}{len} \cdot x + Z_{1S_1} = \frac{(Z_{2S_2} - Z_{1S_2})}{len} \cdot x + Z_{1S_2}$$

$$\frac{(Z_{2S_1} - Z_{1S_1}) - (Z_{2S_2} - Z_{1S_2})}{len} \cdot x = Z_{1S_2} - Z_{1S_1}$$

$$\frac{x}{len} = \frac{Z_{1S_2} - Z_{1S_1}}{(Z_{2S_1} - Z_{1S_1}) - (Z_{2S_2} - Z_{1S_2})}$$

Dies kann nun in die obigen Formeln eingesetzt werden, um die beiden gewünschten Funktionswerte zu erhalten:

$$S_{1C} = S_{2C} = \frac{(Z_{2S_1} - Z_{1S_1}) \cdot (Z_{1S_2} - Z_{1S_1})}{(Z_{2S_1} - Z_{1S_1}) - (Z_{2S_2} - Z_{1S_2})} + Z_{1S_1}$$

$$S_{3C} = \frac{(Z_{2S_3} - Z_{1S_3}) \cdot (Z_{1S_2} - Z_{1S_1})}{(Z_{2S_1} - Z_{1S_1}) - (Z_{2S_2} - Z_{1S_2})} + Z_{1S_3}$$

Diese zwei Werte werden über einem gegebenen Segment durch die nachfolgende Funktion berechnet:

$$getInnerValues : \mathbf{FLine} \times \mathbf{FRegion} \times \mathbf{BasicSegment} \rightarrow [0, 1]^2$$

$$getInnerValues(L = (SF_L, \mathbf{fS}), R = (SF_R, \mathbf{fT}), BS = (P_1, P_2)) \stackrel{\text{def}}{=} (S_{1C}, S_{3C})$$

mit:

$$\begin{aligned} Z_{1S_1} &= ZFkt(R, P_1, BT_1); & Z_{2S_1} &= Zfkt(R, P_2, BT_1); \\ Z_{1S_2} &= ZFkt(R, P_1, BT_2); & Z_{2S_2} &= Zfkt(R, P_2, BT_2); \\ Z_{1S_3} &= Zfkt(L, P_1, BS); & Z_{2S_3} &= Zfkt(L, P_2, BS); \\ & & & isSide(BT_1, BS) \wedge isSide(BT_2, BS) \wedge \neg isequal(BT_1, BT_2) \end{aligned}$$

Nun kann auch $Q_{4_{2S_{cross}}}$ definiert werden:

$$Q_{4_{2S_{cross}}}(L = (SF_L, \mathbf{fS}), R, S = (P_1, P_2)) \stackrel{\text{def}}{=} ((Z_{1R} - Z_{1L}) * (Z_{MR} - Z_{ML}) < 0) \vee ((Z_{MR} - Z_{ML}) * (Z_{2R} - Z_{2L}) < 0) \vee (Z_{MR} = Z_{ML})$$

mit:

$$\begin{aligned} (P_1, Z_{1R}, P_2, Z_{2R}) &= maxSegment(R, S) \\ (P_1, Z_{1L}, P_2, Z_{2L}) &\in \mathbf{fS} \\ (Z_{MR}, Z_{ML}) &= getInnerValues(L, R, S) \end{aligned}$$

Jetzt läßt sich auch der zweite Teil (die Betrachtung der inneren Punkte) von Q4 ableiten:

$$Q_{4_2}(L, R) \stackrel{\text{def}}{=} \exists BS \in basics(L) : Q_{4_{2S}}(L, R, BS)$$

Letztendlich kann dann noch Q4 selbst definiert werden:

$$Q4(L, R) = Q_{4_2} \vee Q_{4_2}$$

Auch $Q5$ und $Q6$ lassen sich auf diese Weise darstellen.

$$Q5(L, R) \stackrel{\text{def}}{=} Q5_1(L, R) \vee Q5_2(L, R)$$

$$Q5_1(L, R) \stackrel{\text{def}}{=} \exists P \in (Pts_L \cap Pts_R) : (\maxZfkt(L, P) > \maxZfkt(R, P)) \wedge (\maxZfkt(R, P) > 0)$$

$$Q5_2(L, R) \stackrel{\text{def}}{=} \exists BS \in basics(L) : Q5_{2S}(L, R, BS)$$

$$Q5_{2S}(L, R, S) \stackrel{\text{def}}{=} \begin{cases} Q5_{2S_{cross}} & \text{wenn } crossing(R, S) \wedge S \in (basics(L) \cap S_R) \\ Q5_{2S_{simple}} & \text{wenn } \neg crossing(R, S) \wedge S \in (basics(L) \cap S_R) \\ false & \text{sonst} \end{cases}$$

$$Q5_{2S_{simple}}((SF_L, \mathbf{fS}), R, S = (P_1, P_2)) \stackrel{\text{def}}{=} \exists (P_1, Z_1, P_2, Z_2) \in \mathbf{fS}, (P_1, Z_3, P_2, Z_4) = \maxSegment(R, S) : (Z_1 > Z_3) \vee (Z_2 > Z_4)$$

$$Q5_{2S_{cross}}(L = (SF_L, \mathbf{fS}), R, S = (P_1, P_2)) \stackrel{\text{def}}{=} (Z_{1L} > Z_{1R}) \vee (Z_{ML} > Z_{MR}) \vee (Z_{2L} > Z_{2R})$$

mit:

$$\begin{aligned} (P_1, Z_{1R}, P_2, Z_{2R}) &= \maxSegment(R, S) \\ (P_1, Z_{1L}, P_2, Z_{2L}) &\in \mathbf{fS} \\ (Z_{MR}, Z_{ML}) &= getInnerValues(L, R, S) \end{aligned}$$

$$Q6(L, R) \stackrel{\text{def}}{=} Q6_1(L, R) \vee Q6_2(L, R)$$

$$Q6_1(L, R) \stackrel{\text{def}}{=} \exists P \in (Pts_L \cap Pts_R) : (\maxZfkt(L, P) < \maxZfkt(R, P)) \wedge (\maxZfkt(L, P) > 0)$$

$$Q6_2(L, R) \stackrel{\text{def}}{=} \exists BS \in basics(L) : Q6_{2S}(L, R, BS)$$

$$Q6_{2S}(L, R, S) \stackrel{\text{def}}{=} \begin{cases} Q6_{2S_{cross}} & \text{wenn } crossing(R, S) \wedge S \in (basics(L) \cap S_R) \\ Q6_{2S_{simple}} & \text{wenn } \neg crossing(R, S) \wedge S \in (basics(L) \cap S_R) \\ false & \text{sonst} \end{cases}$$

$$Q6_{2S_{simple}}((SF_L, \mathbf{fS}), R, S = (P_1, P_2)) \stackrel{\text{def}}{=} \exists (P_1, Z_1, P_2, Z_2) \in \mathbf{fS}, (P_1, Z_3, P_2, Z_4) = \maxSegment(R, S) : (Z_1 < Z_3) \vee (Z_2 < Z_4)$$

$$Q6_{2S_{cross}}(L = (SF_L, \mathbf{fS}), R, S = (P_1, P_2)) \stackrel{\text{def}}{=} (Z_{1L} < Z_{1R}) \vee (Z_{ML} < Z_{MR}) \vee (Z_{2L} < Z_{2R})$$

mit:

$$\begin{aligned} (P_1, Z_{1R}, P_2, Z_{2R}) &= \maxSegment(R, S) \\ (P_1, Z_{1L}, P_2, Z_{2L}) &\in \mathbf{fS} \\ (Z_{MR}, Z_{ML}) &= getInnerValues(L, R, S) \end{aligned}$$

7.5.2.6 Beziehungen zwischen zwei Regionen

Hier werden die unscharfen topologischen Beziehungen der Regionen $A = (SF_A, \mathbf{f}\Gamma_A)$ und $B = (SF_B, \mathbf{f}\Gamma_B)$ untersucht.

$Q1, Q2$ und $Q3$ können bei zwei Regionen nur auf Basispunkten oder Segmenten, jedoch nicht bei inneren Punkten von Dreiecken gelten:

$$Q1_1(A, B) \stackrel{\text{def}}{=} \exists P \in (Pts_A \cap Pts_B) : (maxZfkt(A, P) = 0) \wedge (maxZfkt(B, P) = 0)$$

$$Q1_2(A, B) \stackrel{\text{def}}{=} \exists S = (P_1, P_2) \in (S_A \cap S_B) : equals((P_1, 0, P_2, 0), maxSegment(A, S)) \wedge equals((P_1, 0, P_2, 0), maxSegment(B, S))$$

$$Q1(A, B) = Q1_1 \vee Q1_2$$

$$Q2_1(A, B) \stackrel{\text{def}}{=} \exists P \in (Pts_A \cap Pts_B) : (maxZfkt(A, P) > 0) \wedge (maxZfkt(B, P) = 0)$$

$$Q2_2(A, B) \stackrel{\text{def}}{=} \exists BS = (P_1, P_2) \in (S_A \cap S_B) : equals((P_1, 0, P_2, 0), maxSegment(B, BS)) \wedge ((Z1(maxSegment(A, BS)) > 0) \vee (Z2(maxSegment(A, BS)) > 0))$$

$$Q2(A, B) \stackrel{\text{def}}{=} Q2_1(A, B) \vee Q2_2(A, B)$$

$$Q3_1(A, B) \stackrel{\text{def}}{=} \exists P \in (Pts_A \cap Pts_B) : (maxZfkt(A, P) = 0) \wedge (maxZfkt(B, P) > 0)$$

$$Q3_2(A, B) \stackrel{\text{def}}{=} \exists BS = (P_1, P_2) \in (S_A \cap S_B) : equals((P_1, 0, P_2, 0), maxSegment(A, BS)) \wedge ((Z1(maxSegment(B, BS)) > 0) \vee (Z2(maxSegment(B, BS)) > 0))$$

$$Q3(A, B) \stackrel{\text{def}}{=} Q3_1(A, B) \vee Q3_2(A, B)$$

Bei den restlichen Prädikaten stellt sich wieder das Problem, daß die Zugehörigkeitsfunktionen sich über Dreiecksseiten kreuzen können. Hier kommt noch erschwerend hinzu, daß dies bei beiden Operanden der Fall sein kann. Somit müssen folgende Fälle für jedes gemeinsame Segment unterschieden werden:

- die Zugehörigkeitsfunktionen der anliegenden Dreiecke der beiden Regionen sind kreuzungsfrei.
- die Zugehörigkeitsfunktionen der anliegenden Dreiecke einer Region kreuzen sich; die der anderen Region sind kreuzungsfrei
- die Zugehörigkeitsfunktionen der anliegenden Dreiecke beider Regionen kreuzen sich

Außerdem müssen natürlich auch die inneren Punkte von Dreiecken betrachtet werden. Zuerst wird wieder eine Funktion bereitgestellt, welche die Funktionswerte über dem Kreuzungspunkt der Zugehörigkeitsfunktionen zurückgibt.

$$getInnerValues : \mathbf{FRegion} \times \mathbf{FRegion} \rightarrow [0.1]^2$$

Hier kann die Berechnungsvorschrift zwischen einer Linie und einer Region verwendet werden. Auf die Ausführung wird daher verzichtet. Zu erwähnen ist noch, daß der Kreuzungspunkt des zweiten Parameters verwendet wird.

Die Betrachtung der Segmente wird aufgespalten. Somit ergibt sich:

$$Qi_2(A, B) \stackrel{\text{def}}{=} \exists S \in (S_A \cap S_B) : Qi_2(A, B, S)$$

mit:

$$Qi_2(A, B, S) = \begin{cases} Qi_{2_{cross1}}(A, B, S) & \text{wenn } crossing(A, S) \wedge \neg crossing(B, S) \wedge S \in (S_A \cap S_B) \\ Qi_{2_{cross2}}(A, B, S) & \text{wenn } \neg crossing(A, S) \wedge crossing(B, S) \wedge S \in (S_A \cap S_B) \\ Qi_{2_{cross12}}(A, B, S) & \text{wenn } crossing(A, S) \wedge crossing(B, S) \wedge S \in (S_A \cap S_B) \\ Qi_{2_{simple}}(A, B, S) & \text{wenn } \neg crossing(A, S) \wedge \neg crossing(B, S) \wedge S \in (S_A \cap S_B) \\ \text{false} & \text{sonst} \end{cases}$$

und

$$i \in \{4, 5, 6\}$$

$$Q4_{2_{simple}}(A, B, S) \stackrel{\text{def}}{=} equals(maxSegment(A, S), maxSegment(B, S)) \wedge ((Z1(maxSegment(A, S)) > 0) \vee (Z2(maxSegment(A, S)) > 0))$$

$$Q5_{2_{simple}}(A, B, S) \stackrel{\text{def}}{=} (Z1(maxSegment(A, S)) > Z1(maxSegment(B, S))) \vee (Z2(maxSegment(A, S)) > Z2(maxSegment(B, S)))$$

$$Q6_{2_{simple}}(A, B, S) \stackrel{\text{def}}{=} (Z1(maxSegment(A, S)) < Z1(maxSegment(B, S))) \vee (Z2(maxSegment(A, S)) < Z2(maxSegment(B, S)))$$

$$Q4_{2_{cross1}}(A, B, S = (P_1, P_2)) \stackrel{\text{def}}{=} ((Z_{1A} - Z_{1B}) * (Z_{MA} - Z_{MB}) < 0) \vee ((Z_{MA} - Z_{MB}) * (Z_{2A} - Z_{2A}) < 0) \vee (Z_{MA} = Z_{MB})$$

mit:

$$(P_1, Z_{1A}, P_2, Z_{2A}) = maxSegment(A, S)$$

$$(P_1, Z_{2A}, P_2, Z_{2B}) = maxSegment(B, S)$$

$$(Z_{MA}, Z_{MB}) = getInnerValues(A, B, S)$$

$Q4_{2_{cross2}}$ wird analog definiert. Nur Z_{MA} und Z_{MB} müssen anders definiert werden:

$$(Z_{MB}, Z_{MA}) = getInnerValues(B, A, S)$$

$$Q5_{2_{cross1}}(A, B, S) \stackrel{\text{def}}{=} (Z_{1A} > Z_{1B}) \vee (Z_{2A} > Z_{2B}) \text{opor}(Z_{MA} > Z_{MB})$$

$$Q6_{2_{cross1}}(A, B, S) \stackrel{\text{def}}{=} (Z_{1A} < Z_{1B}) \vee (Z_{2A} < Z_{2B}) \text{opor}(Z_{MA} < Z_{MB})$$

mit den gleichen Bezeichnungen wie bei $Q4_{2_{cross1}}$

$Q5_{2_{cross2}}$ und $Q6_{2_{cross2}}$ werden vollständig analog definiert, wobei jedoch die Bezeichnungen von $Q4_{2_{cross2}}$ verwendet werden.

Zuletzt muß noch der Fall betrachtet werden, bei dem sich die Funktionsverläufe der Dreiecke beider Regionen kreuzen. Hierzu wird der *getInnerValues*-Operator derart erweitert, daß er die Werte beider Kreuzungspunkte ergibt:

$$getAllInnerValues : \mathbf{FRegion} \times \mathbf{FRegion} \times \mathbf{BasicSegment} \rightarrow [0, 1]^4$$

$$getAllInnerValues(A, B, (P_1, P_2)) \rightarrow (Z_{1A}, Z_{1B}, Z_{2A}, Z_{2B})$$

Die Berechnungsvorschrift entspricht derjenigen, die zwischen einer unscharfen Linie und einer Region gilt. Hier wird daher nur noch die Bedeutung der einzelnen Variablen erklärt:

Z_{1A} : maximaler Funktionswert von A in dem Kreuzungspunkt, der P_1 am nächsten liegt

Z_{1B} : maximaler Funktionswert von B in dem Kreuzungspunkt, der P_1 am nächsten liegt

Z_{2A} : maximaler Funktionswert von A in dem Kreuzungspunkt, der P_2 am nächsten liegt

Z_{2B} : analog für B

Damit lassen sich die fehlenden Prädikatenteile für die Segmentbetrachtung definieren:

$$Q4_{2_{cross12}}(A, B, S = (P_1, P_2)) \stackrel{\text{def}}{=} ((Z_{0A} - Z_{0B}) \cdot (Z_{1A} - Z_{1B}) < 0) \vee \\ ((Z_{1A} - Z_{1B}) \cdot (Z_{2A} - Z_{2B}) < 0) \vee \\ ((Z_{2A} - Z_{2B}) \cdot (Z_{3A} - Z_{3B}) < 0) \vee \\ (Z_{1A} = Z_{1B}) \vee (Z_{2A} = Z_{2B})$$

$$Q5_{2_{cross12}}(A, B, S = (P_1, P_2)) \stackrel{\text{def}}{=} (Z_{0A} > Z_{0B}) \vee (Z_{1A} > Z_{1B}) \vee (Z_{2A} > Z_{2B}) \vee (Z_{3A} > Z_{3B})$$

$$Q6_{2_{cross12}}(A, B, S = (P_1, P_2)) \stackrel{\text{def}}{=} (Z_{0A} < Z_{0B}) \vee (Z_{1A} < Z_{1B}) \vee (Z_{2A} < Z_{2B}) \vee (Z_{3A} < Z_{3B})$$

jeweils mit:

$$(P_1, Z_{0A}, P_2, Z_{3A}) = \text{maxSegment}(A, S)$$

$$(P_1, Z_{0B}, P_2, Z_{3B}) = \text{maxSegment}(B, S)$$

$$(Z_{1A}, Z_{1B}, Z_{2A}, Z_{2B}) = \text{getAllInnerValues}(A, B, S)$$

Letztlich fehlt noch die Betrachtung innerer Dreieckspunkte:

$$Q4_3(A, B) \stackrel{\text{def}}{=} \exists (P_{1A}, Z_{1A}, P_{2A}, Z_{2A}, P_{3A}, Z_{3A}) \in \mathbf{fT}_A, (P_{1B}, Z_{1B}, P_{2B}, Z_{2B}, P_{3B}, Z_{3B}) \in \mathbf{fT}_B : \\ ((Z_{1A} = Z_{1B}) \wedge (Z_{2A} = Z_{2B}) \wedge (Z_{3A} = Z_{3B})) \vee \\ ((Z_{1A} - Z_{1B}) \cdot (Z_{2A} - Z_{2B}) < 0) \vee \\ ((Z_{2A} - Z_{2B}) \cdot (Z_{3A} - Z_{3B}) < 0) \vee \\ ((Z_{1A} - Z_{1B}) \cdot (Z_{3A} - Z_{3B}) < 0)$$

$$Q5_3(A, B) \stackrel{\text{def}}{=} \exists (P_{1A}, Z_{1A}, P_{2A}, Z_{2A}, P_{3A}, Z_{3A}) \in \mathbf{fT}_A, (P_{1B}, Z_{1B}, P_{2B}, Z_{2B}, P_{3B}, Z_{3B}) \in \mathbf{fT}_B : \\ (Z_{1A} > Z_{1B}) \vee (Z_{2A} > Z_{2B}) \vee (Z_{3A} > Z_{3B})$$

$$Q6_3(A, B) \stackrel{\text{def}}{=} \exists (P_{1A}, Z_{1A}, P_{2A}, Z_{2A}, P_{3A}, Z_{3A}) \in \mathbf{fT}_A, (P_{1B}, Z_{1B}, P_{2B}, Z_{2B}, P_{3B}, Z_{3B}) \in \mathbf{fT}_B : \\ (Z_{1A} < Z_{1B}) \vee (Z_{2A} < Z_{2B}) \vee (Z_{3A} < Z_{3B})$$

Zum Schluß müssen die Prädikate zusammengesetzt werden. Es ergibt sich:

$$Q4(A, B) \stackrel{\text{def}}{=} Q4_1(A, B) \vee Q4_2(A, B) \vee Q4_3(A, B)$$

$$Q5(A, B) \stackrel{\text{def}}{=} Q5_1(A, B) \vee Q5_2(A, B) \vee Q5_3(A, B)$$

$$Q6(A, B) \stackrel{\text{def}}{=} Q6_1(A, B) \vee Q6_2(A, B) \vee Q6_3(A, B)$$

Kapitel 8

Implementierung des diskreten Modells

8.1 Wahl der Programmiersprache

Um das Modell zu implementieren, muß man sich auch Gedanken darüber machen, welche Programmiersprache verwendet werden soll. Dazu ist es notwendig, den Zweck dieser Implementierung zu untersuchen. Hier soll gezeigt werden, daß das Modell tatsächlich effizient implementierbar ist. Das entstehende Programm soll der Veranschaulichung dienen, wie die definierten Operatoren in der Realität arbeiten.

Bei der Wahl der Programmiersprache wurde hauptsächlich darauf geachtet, daß das Programm möglichst unabhängig von Hardware und verwendetem Betriebssystem ist. Mit anderen Worten sollte die entstehende Anwendung plattformunabhängig arbeiten. Es ist selbstverständlich, daß hierdurch die Laufzeit des Programms nicht gerade begünstigt wird, da spezielle Eigenschaften nicht verwendet werden können. Da hier jedoch nicht gleich riesige Datenmengen verarbeitet werden sollen, ist dieser Nachteil nicht so gravierend. Um die geometrischen Objekte auch darstellen zu können (reine Textausgabe ist zur Demonstration der Operatoren nicht geeignet), muß die Programmiersprache auch über die Möglichkeit verfügen, entsprechende graphische Ausgaben machen zu können.

Aus oben genannten Gründen ist Java als Sprache zur Implementierung des Modells gewählt worden. Diese Sprache zeichnet sich durch ihre Plattformunabhängigkeit aus. Auch kann das entstehende Programm mit Einschränkungen zu einer Online-Version umgearbeitet werden.

Damit das Programm nicht auf spezielle Hardware angewiesen ist, wurde auf die Verwendung zusätzlicher Möglichkeiten, z.B. Packages, die 3D-Graphikkarten unterstützen, verzichtet.

8.2 Datenmodellierung

Ein wichtiger Aspekt, der auch die Laufzeit der einzelnen Operatoren entscheidend beeinflusst, ist die Wahl der entsprechenden Datenstrukturen für die Objekte.

Für die Modellierung der hier vorgestellten Objekte ist nur wichtig, wie die Menge von unscharfen Elementarobjekten dargestellt werden soll. Wie unscharfe Elementarpunkte, Segmente und Dreiecke dargestellt werden, spielt für die Laufzeit keine große Rolle, da die Anzahl ihrer Basispunkte konstant ist. So kann eine Klasse für die unscharfen Dreiecke derart dargestellt werden, daß Koordinaten und Zugehörigkeitsgrade als Attribute der Klasse implementiert werden. Genausogut kann aber auch eine Instanz der Klasse für Basisdreiecke verwendet werden, um diese Klasse zu realisieren.

Als Datenstruktur für die Menge wurde ein sortiertes Array gewählt. Für die Sortierung dient die Ordnung der zugrundeliegenden Basiselemente. Dies ist möglich, da ein Basiselement höchstens

einmal in der Menge auftreten kann. In Java ist es möglich, die Arraygröße während der Laufzeit festzulegen. Zudem bietet Java mit der Klasse Vector die Möglichkeit, ein Array dynamischer Größe zu verwalten. Hieraus ergibt sich der Nachteil, daß viele Objekte (bzw. ihre Referenzen) umkopiert werden müssen, wenn die Anzahl der Elemente der Menge größer als das Fassungsvermögen des Arrays wird. Das ist allerdings nicht so schlimm, da Objekte relativ konstant sind, d.h. größere Änderungen nur während ihrer Definition auftreten. Auch beim Einfügen von Elementen können viele Verschiebungen auftreten. Es wird sich jedoch herausstellen, daß das Einfügen in die Mitte des Arrays bei den meisten Operationen nicht benötigt wird. Weiterhin kann die maximale Größe des Ergebnisses eines Operator berechnet werden, so daß auch hier keine größeren Elementverschiebungen auftreten müssen.

Durch den Direktzugriff, den ein Array bietet, ist klar, daß ein Element in $O(\log(n))$ gefunden werden kann, wenn n die Größe der Menge ist.

Zusammenfassend hat die so gegebene Datenstruktur folgende Eigenschaften:

- ein geordneter Durchlauf ist in Linearzeit möglich
- ein Objekt läßt sich in $O(\log(n))$ in der Menge finden
- das Einfügen eines Objekts an einer beliebigen Position benötigt lineare Zeit
- das Einfügen eines Elements am Ende dauert
 - konstante Zeit, wenn noch Platz im Array ist
 - lineare Zeit, wenn das Array voll ist

8.3 Algorithmen

8.3.1 Mengenoperatoren

Bei den Algorithmen werden zunächst die binären Mengenoperatoren untersucht. Im Einzelnen handelt es sich hierbei um:

- *add* bzw. *scaled_add*
- *union* bzw. *scaled_union*
- *intersection* bzw. *scaled_intersection*
- *difference* bzw. *scaled_difference*

Man kann feststellen, daß bei all diesen Operationen unterschieden werden muß, ob ein Baselement in beiden oder nur in einer der Mengen vorkommt. Weiterhin kann die maximal mögliche Größe des Ergebnisses nicht nur abgeschätzt, sondern sogar berechnet werden, so daß ein Einfügen am Ende der zu berechnenden Menge stets in konstanter Zeit möglich ist.

Bezeichnen m und n die Größen der Mengen der beiden Operanden, dann haben die Ergebnismengen die folgenden maximalen Größen:

- *add* : $m + n$
- *union* : $m + n$
- *intersection* : $\min(m,n)$
- *difference* : m

Die gleichen Werte gelten natürlich auch für die jeweiligen interpretierten Operatoren. Der folgende Algorithmus berechnet das Ergebnis eines der oben genannten Operatoren in Linearzeit. Es wird vorausgesetzt, dass die Arraygröße des Ergebnisses gemäß obiger Angaben gesetzt wurde. Eine später näher erläuterte Prozedur (`processElement`) verarbeitet die übergebenen Argumente abhängig vom Operator. Für den linearen (geordneten) Durchlauf durch die Mengen werden folgende Funktionen bzw. Prozeduren verwendet:

hasMoreElements : liefert `true`, wenn die entsprechende Menge weitere Elemente besitzt.

goStart : setzt die aktuelle Position der Menge auf den Anfang

next : setzt die aktuelle Position auf das nächstgrößere Element

get : liefert das Element an der aktuellen Position in der Menge

set(O) : überschreibt das aktuelle Element mit `O`

insert(O) : fügt `O` an der richtigen Stelle ein (gemäß der Ordnung)

search(BO) : sucht das übergebene Basisobjekt in `M` und liefert das dazugehörige unscharfe Objekt oder `null`, wenn kein Element mit dieser Basis enthalten ist

getPos(BO) : liefert die Position von `BO` in `M` zurück; ist kein Elementarobjekt mit `BO` als Basis in `M` vorhanden, so liefert diese Prozedur `-1`

Desweiteren gibt es eine `compare`-Funktion, die folgende Ergebnisse zurückliefert:

$$\text{compare}(E1, E2) \stackrel{\text{def}}{=} \begin{cases} -1 & \text{wenn } E1 < E2 \\ 0 & \text{wenn } E1 = E2 \\ 1 & \text{wenn } E1 > E2 \end{cases}$$

Algorithmus

Eingabe :

SF1 : Skalierungsfaktor des 1.Operanden

M1 : Menge der Elementarobjekte des 1.Operanden

SF2 : Skalierungsfaktor des 2.Operanden

M2 : Menge der Elementarobjekte des 2.Operanden

OP : der gewünschte Operator

Ausgabe :

Result : Ergebnis des entsprechenden Operators

lokale Variablen :

E1 : Elementarobjekt

E2 : Elementarobjekt

c : integer

`M1.goStart();`

`M2.goStart();`

`while(M1.hasMoreElements() AND M2.hasMoreElements()){`

`E1 = M1.get();`

`E2 = M2.get();`

`c = compare(E1,E2);`

```

    if (c==1) {
        processElements(E1,SF1,null ,SF2,Result,OP);
        M1.next();
    }
    if(c==0){
        processElement(E1,SF1,E2,SF2,Result,OP);
        M2.next();
        M1.next();
    }
    if(c==1){
        processElement(null,SF1,E2,SF2,Result,OP);
        M2.next();
    }
}

while(M1.hasMoreElements()){
    E1 = M1.get();
    processElement(E1,SF1,null,SF2,Result,OP);
    M1.next();
}
while(M2.hasMoreElements()){
    E2 = M2.get();
    processElement(null,SF1,E2,SF2,Result,OP);
    M2.next();
}
return Result;
}

```

Es ist leicht zu sehen, daß dieser (Merge-Sort-ähnliche) Algorithmus unter oben genannten Voraussetzungen eine Laufzeit von $O(m+n)$ besitzt, wenn `processElement(...)` eine konstante Laufzeit besitzt. Ein Teil dieser Prozedur kann das folgende Aussehen haben:

```

processElement( E1,SF1, E2,SF2, Result, OP) {
// beide Elemente haben die gleiche Basis oder mind. 1
// Element ist null
switch(OP){
    case(union) :  if( (E1!=null) OR (E2!=null) ){
                    if(E1==null)
                        Result.insert(E2);
                    else if(E2==null)
                        Result.insert(E1);
                    else{
                        Erzeuge neues Element E3, mit der Basis von E1 und
                        den jeweils maximalen Zugehoerigkeitsgraden;
                        Result.insert(E3);
                    }
                };
                break;
            ...
        }
    }
}

```

Hier ist zu erkennen, daß diese Prozedur tatsächlich in konstanter Zeit abläuft. Der einzige Kandidat für eine längere Laufzeit ist das Einfügen des Elements in das Ergebnis. Aber auch dies geschieht in konstanter Zeit. Nach Voraussetzung ist das Ergebnis groß genug, um alle notwendigen Elemente aufnehmen zu können, daher sind keine Arrayvergrößerungen und speziell der damit verbundene Kopieraufwand notwendig. Aufgrund der Reihenfolge des Aufrufs dieser Prozedur wird stets am Ende eingefügt, was in konstanter Zeit möglich ist.

8.3.2 Normierung

Zu bemerken ist noch, daß die Normierung, die bei den interpretierten Versionen der Operatoren notwendig ist, in einem separaten Schritt durchgeführt wird. Aber auch dies kann in Linearzeit erfolgen. Der minimale und maximale Zugehörigkeitsgrad lassen sich durch einen linearen Durchlauf ermitteln. Bei einem weiteren Durchlauf werden die Zugehörigkeitsgrade entsprechend angepaßt. Es bezeichne M die Menge der Elementarobjekte des betrachteten Objekts. Die Änderung der Zugehörigkeitsgrade ist ab Seite 94 beschrieben.

```
norm(){
  // 1. Feststellen der minimalen und maximalen Z-Werte
  z_min = 1;
  z_max = 0;
  M.goStart();
  while(this.hasMoreElements()){
    Current = M.get();
    if(current.getMinZ()<Z_min)
      minZ = Current.getMinZ();
    if(current.getMaxZ()>Z_max)
      maxZ = Current.getZMax();
    M.next();
  }
  // 2. Aendern der Z-Werte
  M.goStart();
  while(this.hasMoreElements()){
    aendere Z-Werte entsprechend Text;
    M.next();
  }
}
```

8.3.3 Berechnung von Zugehörigkeitsgraden

Die Zugehörigkeitsgrade eines unscharfen Objekts an einem bestimmten Punkt lassen sich in $O(\log(n))$ bestimmen, wobei n die Größe der Menge darstellt. Dies zu zeigen ist recht einfach. Aufgrund der gleichmäßigen Triangulierung lassen sich alle Basisobjekte, die an dieser Stelle liegen in konstanter Zeit ermitteln. Die zugehörigen unscharfen Elementarobjekte können aufgrund der Datenstruktur in $O(\log(n))$ in der entsprechenden Menge gefunden werden. Von jedem dieser Elementarobjekte läßt sich der Zugehörigkeitsgrad wieder in konstanter Zeit bestimmen (vgl. Ausführungen im abstrakten Modell). Das Verfahren, mit dem sich die Basisobjekte an einem gegebenen Punkt bestimmen lassen, ist in Kapitel 10 beschrieben. Hier wird von einer gegebenen Prozedur `getObjects(P)` ausgegangen, die für eine gegebene Koordinate die Menge der Basisobjekte zurückliefert, die diese überdecken. Bei einer realen Implementierung gäbe es drei solcher Prozeduren, entsprechend den betrachteten Objekttypen. Die Berechnung der Zugehörigkeitsgrade für Elementarobjekte wurde bereits beim diskreten Modell beschrieben (siehe

Seite 84ff). Es ist klar, daß die speziellen Zugehörigkeitsgrade (mittlerer, kleinster, größter) sowie der Grad bei Angabe des zugrundeliegenden Basisobjekts ebenfalls auf diese Weise berechnet werden können.

Algorithmus

Eingabe :

SF : Skalierungsfaktor des Operanden
M : Menge des Elementarobjekte des Operanden
P : der Punkt für den die Menge der Grade zu bestimmen ist

Ausgabe :

Result : Menge der Grade am gegebenen Punkt

lokale Variablen :

BOs : Menge von Basisobjekten
EO : ein Elementarobjekt

```
BOs = getObjects(P);
Result = {};
for (all BO in BOs) {
    EO = M.search(BO);
    if(EO==null)
        fuege 0 zu Result hinzu
    else
        fuege EO.Zfkt(P) zu Result hinzu
}
return Result;
```

8.3.4 Gleichheit unscharfer Objekte

Um zwei unscharfe Objekte auf Gleichheit zu testen, genügt ein einfacher gemischter Durchlauf. Das jeweils kleinste Element muß bei beiden Objekten gleich sein. Dies kann in konstanter Zeit erledigt werden. Somit beträgt die Laufzeit $O(n + m)$.

8.3.5 Alpha-Cuts

Auch die beiden Versionen des *alpha-cuts* lassen sich in linearer Zeit berechnen. Es wird dazu ein linearer Durchlauf durch die Menge benötigt, bei dem die Elemente in das Ergebnis eingefügt werden, die der geforderten Bedingung (vgl.S.92) entsprechen. Da der Durchlauf in aufsteigender Reihenfolge durchgeführt wird, wird stets das nächstgrößere Element ins Ergebnis eingefügt, was dadurch in konstanter Zeit möglich ist. Da die Größe der Ergebnismenge maximal der Ausgangsmenge entspricht, kann die Arraygröße bereits zu Beginn der Berechnung festgelegt werden, wodurch keine Umkopierarbeiten notwendig werden.

8.3.6 Der sharp-Operator

Genauso ist es möglich, daß der *sharp*-Operator in linearer Zeit berechnet wird. Die Größe des Ergebnisses entspricht genau der des Ausgangsobjekts. Daher werden bei geeigneter Größeninitialisierung der Ergebnismenge keine Arrayvergrößerungen notwendig. Es erfolgt ein einfacher geordneter Durchlauf, bei dem die angepaßten Elementarobjekte (alle Grade 1) ins Ergebnis eingefügt werden.

8.3.7 Grenze von Linie oder Region

Die Grenze einer Linie oder einer Region kann in $O(n(\log(n) + n)) = O(n^2)$ berechnet werden. Das Prinzip der Berechnung wird hier für Regionen beschrieben und kann auf Linien übertragen werden. Es findet ein linearer Durchlauf durch die Menge der unscharfen Dreiecke statt. Eine Seitenlinie des aktuellen Dreiecks gehört genau dann zur Grenze der Region, wenn kein weiteres Dreieck in der Region diese als Seite besitzt. Aufgrund der gleichmäßigen Triangulierung der Grundmenge, lassen sich die Dreiecke, die ein gegebenes Basissegment als Seite besitzen in konstanter Zeit bestimmen. Das zweite Dreieck (also das, welches nicht das aktuelle Dreieck ist), läßt sich in $O(\log(n))$ in der Region finden, wenn es existiert. Somit kann innerhalb dieser Zeit getestet werden, ob eine Seite des aktuellen Dreiecks zur Grenze gehört. Dieses wird nun in die Ergebnislinie eingetragen, was in linearer Zeit zu erreichen ist. Da die Menge der Segmente der Linie maximal 3 mal so groß werden kann, wie die Menge der Dreiecke, ergibt sich die oben angegebene Zeitschranke.

Algorithmus

Eingabe :

SF : Skalierungsfaktor der Region

M : Menge der unscharfen Elementardreiecke

Ausgabe :

Result : Grenze der übergebenen Region

lokale Variablen :

BSs : Menge von Basissegmenten

S : Basissegment

BT1, BT2 : Basisdreieck

BTs : Menge von Basisdreiecken

T : unscharfes Elementardreieck

```
M.goStart();
while(M.hasMoreElements()){
  T = M.get();
  BT1 = T.basic();
  BSs = BT1.Sides(); // die drei Seiten von BT1
  for (all S in BSs){
    BTs = S.getTriangles(); // bestimmt die beiden anliegenden Dreiecke
    for (all BT2 in BTs ){
      if(!BT1.equals(BT2)) {
        if(M.search(BT2)==null){
          fuege S zu Result hinzu;
        }
      }
    }
  }
  M.next();
}
```

8.3.8 Gemeinsame Linien von Regionen

Um die gemeinsamen Linien zweier Regionen zu bestimmen, werden alle enthaltenen Dreiecke der einen Region durchlaufen. Für jede Dreiecksseite des aktuellen Dreiecks wird geprüft, ob dies zur Grenze der Region gehört ($O(\log(n))$). Ist dies der Fall, werden in logarithmischer Zeit die anliegenden Dreiecke in der zweiten Region gesucht. Wird ausschließlich das dem aktuell betrachteten Basisdreieck gegenüberliegende Dreieck gefunden, wird das Segment ins Ergebnis übernommen. Dies geschieht in linearer Zeit. Somit beträgt hier die Laufzeit $O(n \cdot (\log(m) + n))$.

8.3.9 Gemeinsame Punkte von Linien und Regionen

Hier werden die Elementarobjekte eines Operanden durchlaufen. Zu jedem Eckpunkt werden die anliegenden Basissegmente bestimmt. Diese (oder beide anliegenden Dreiecke) werden nun in beiden Objekten gesucht ($O(\log(n) + \log(m))$). Sind beide Ergebnismengen nicht leer und überschneidungsfrei, so gehört dieser Punkt zum Ergebnis. Er kann in linearer Zeit eingetragen werden, woraus sich eine Gesamtlaufzeit von $O(n \cdot (\log(m) + n))$ ergibt.

8.3.10 Größenangaben

Die Berechnung der Basisgrößen ist für Regionen und Punkte in konstanter Zeit möglich, wenn die Anzahl der Elemente mitgezählt wird. Man beachte hierbei, daß die Basisdreiecke in der gewählten Triangulierung stets die gleiche Größe besitzen. Die Berechnung der Basislänge einer Linie hingegen benötigt lineare Zeit, da Segmente unterschiedliche Längen besitzen. Es ist jedoch möglich, die Länge der Linie als zusätzliches Attribut abzuspeichern, womit auch diese Abfrage in konstanter Zeit erfolgen kann. Dieses Längenattribut ist beim Hinzufügen oder Entfernen von Segmenten zu aktualisieren, was in konstanter Zeit möglich ist.

Die anderen (unscharfen) Ausmaße lassen sich in Linearzeit berechnen oder können über ein zusätzliches Attribut (was beim Einfügen, Überschreiben oder Löschen von Elementen aktualisiert wird) in konstanter Zeit abgefragt werden. Man beachte hierbei, daß die Aktualisierung dieses zusätzlichen Attributs nur konstanten Aufwand bedeutet und somit die Laufzeit für die Änderung der Menge nicht beeinflusst.

8.3.11 Der Mittelwert

Der Mittelwert und seine interpretierte Version lassen sich in der Zeit $O(m \cdot (m + \sum_{i=1}^m n_i))$ berechnen. Dabei bezeichne m die Anzahl der beteiligten Operanden und n_i die jeweilige Größe des Operanden.

Hierzu wird von allen Operanden das aktuelle Minimum gesucht. Verweise der betroffenen Operanden werden in einem Vektor gespeichert. Dies benötigt m Schritte. Die Zugehörigkeitsgrade der Minima werden jeweils addiert und das Ergebnis durch m , geteilt. Bei diesem Durchlauf können die entsprechenden aktuellen Werte „weitergesetzt“ werden. Dies benötigt maximal m Schritte. Das Erzeugen eines neuen Elements sowie das Einfügen in das Ergebnis geschieht in konstanter Zeit. Dieser Vorgang wird n_{max} mal wiederholt, wobei n_{max} die Größe der mächtigsten Menge aller Operanden darstellt. Somit ergibt sich die oben angegebene Zeitschranke.

Algorithmus

Eingabe :

Ops : Menge von unscharfen Objekten eines Typs

Ausgabe :

Result : unscharfes Objekt

lokale Variablen :

aktMin : unscharfes Elementarobjekt
Minima : Vektor (Verweise auf Elemente in Ops)
aktOp : unscharfes Objekt
tmp : unscharfes Elementarobjekt
c,i : integer
Z1,Z2,Z3 : double
fO : unscharfes Elementarobjekt
neu : unscharfes Elementarobjekt

```
do{
  // bestimme die Ops mit minimalem ersten Element
  Ops.goStart();
  aktMin=null;
  while(Ops.hasMoreElements()){
    aktOp = Ops.get();
    Ops.next();
    if(aktOp.fEO.hasMoreElements()){
      tmp = aktOp.fEO.get();
      if(aktMin==null){ // erstes Elementarobjekt
        aktMin = tmp;
        Minima.add(aktOp);
      }
      else{
        c = compare(aktMin,tmp);
        if(c>0) { // neues Minimum gefunden
          Minima.empty();
          aktMin = tmp;
          Minima.add(aktOp);
        }
        if(c==0){ // zusaetzliches Minimum gefunden
          Minima.add(aktOp);
        }
      }
    }
  }
}
if(aktMin!=null){
  // aktualisiere Ergebnis
  Z1 = Z1 = Z3 = 0.0;
  for(i=0;i<Minima.size();i++){
    fO = Minima.get(i).fEO.get(); // ermitteln des aktuellen Elements
    Z1 = Z1 + fO.getZ1();
    Z2 = Z2 + fO.getZ2();
    Z3 = Z3 + fO.getZ3();
    Minima.get(i).fEO.next(); // auf das naechste Element setzen
  }
  Z1 = Z1 / Ops.size();
}
```

```

    Z2 = Z2 / Ops.size();
    Z3 = Z3 / Ops.size();
    neu = neues Element mit Basis aktMin und Z1,Z2,Z3 als Grade,
    Result.add(neu);
}
while(aktMin!=null); // keine Elemente mehr vorhanden
return Result;

```

8.3.12 Basis von unscharfen Objekten

Diese läßt sich relativ einfach in Linearzeit berechnen. Bei einem einfachen geordneten Durchlauf durch die Menge der unscharfen Elementarobjekte wird die Basis des aktuellen Elementarobjekts, die sich in konstanter Zeit bestimmen läßt, ins Ergebnis eingefügt. Somit ist das Ergebnis geordnet.

8.3.13 Faces von Linien und Regionen

Um zusammenhängende Teile von Linien und Regionen zu berechnen, wird ein Algorithmus verwendet, der bei gegebenem Basisobjekt BO und gegebenem unscharfen Objekt UO die zusammenhängende Komponente von UO ermittelt, in der BO liegt. Dabei wird vorausgesetzt, daß BO in UO enthalten ist. Dieser verwaltet eine Liste L von noch nicht verarbeiteten Dreiecken und arbeitet folgendermaßen:

Algorithmus

Eingabe :

UO : eine unscharfe Region oder eine unscharfe Linie

BO : ein Basisdreieck oder ein Basissegment aus UO

Ausgabe :

Result : unscharfes Objekt

lokale Variablen :

L : eine Liste von Basiselementen

N : eine Menge von Basiselementen maximaler Größe

E : ein Basiselement

```

aktuell = Eingabedreieck;
do{
    fuege aktuell in Result ein;
    ermittle alle Nachbarn N von aktuell;
    for( alle Elemente E in N ) {
        if (E in der Basis von UO) and
            (E nicht in der Basis von Result) and
            (E nicht in L )
            fuege E in L ein;
    } // for
    if (L enthaelt Elemente){
        aktuell = erstes Element von L;
        entferne erstes Element von L;
    }
}

```



```

else
    aktuell= null;
} while aktuell!=null;
return result;

```

Ist eine solche Komponente gefunden, wird in linearer Zeit die Differenz von Ausgangsobjekt (bzw. eine Kopie desselben) und dieser Komponente gebildet. Der Algorithmus wird solange aufgerufen, wie Elemente in diesem Objekt vorhanden sind. Es ergibt sich (durch den oben angegebenen Algorithmus) insgesamt eine quadratische Laufzeit. Der Algorithmus verwendet jedes Element des Ausgangsobjekts einmal. Die Listenoperationen sowie das Einfügen ins Ergebnis benötigen Linearzeit, woraus diese Zeitschranke resultiert.

8.3.14 Löcher in Regionen

Da Löcher innerhalb von Faces liegen, werden zunächst zusammenhängende Komponenten der Ausgangsregion gebildet, was quadratische Zeit in Anspruch nimmt. Von jedem Face wird die Grenze ermittelt und von dieser die Faces berechnet. Das kann insgesamt wieder in quadratischer Zeit geschehen. Aus diesen Segmentmengen müssen die einzelnen Grenzkreise extrahiert werden. Dazu dient der folgende Algorithmus:

Algorithmus

Eingabe :

Segs : eine Menge von Segmenten

Ausgabe :

result : eine Menge von einfachen Kreisen

lokale Variablen :

Bps : eine Menge von Basispunkten

EP,BP,CurrentBP,L1 : ein Basispunkt

S : ein Segment

Ps,Ps2 : eine Menge von Pfaden

P,P2,CurrentPath,Rest : ein Pfad

Ns : eine Menge von Segmenten

N : ein Segment

circle : boolean

```

// Berechnung von Selbstschnitten
for( all S in Segs ) {
    berechne fuer jeden Endpunkt EP von S die anliegenden Segmente;
    pruefe fuer jedes anliegende Segment das Enthaltensein in Segs;
    gibt es mehr als zwei solcher Segmente fuege EP zu BPs hinzu;
}
// Sonderfall einfacher Kreis (keine Selbstschnitte)
if( BPs == {} )
    fuege einen Endpunkt eines in Segs enthaltenen Segments zu Bps zu;
markiere alle S in Segs als 'nicht benutzt';
for( all BP in BPs ) {
    ermittle alle an BP anliegenden Segmente Ns;
    for(all N in Ns) {

```

```

if(N in Segs enthalten und als 'nicht benutzt' markiert) {
  // beginne neuen Pfad
  P = new Path;
  fuege BP dem Pfad P hinzu;
  CurrentBP = zweiter Endpunkt von N; // !=BP
  markiere N in Segs als 'benutzt';
  fuege CurrentBP dem Pfad P hinzu;
  while( CurrentBP nicht in BPs) {
    ermittle naechstes Segment S mit folgenden Eigenschaften:
      -S liegt an CurrentBP an &
      -S ist in Segs enthalten &
      -S ist als 'nicht benutzt' markiert
    markiere S als 'benutzt'
    CurrentBP = zweiter Endpunkt von S;
    fuege CurrentBP zu P hinzu
  }
  fuege P zu Ps hinzu
} // if
}
}
for(all P in Ps){
  if(P bildet einen Kreis)
    fuege P zu result hinzu;
  else
    fuege P zu Ps2 hinzu
}
// Ps2 enthaelt nun einfache Pfade, die keine Kreise sind
markiere alle Pfade in Ps2 als 'nicht benutzt';
for( all P in Ps2) {
  if(P als 'nicht benutzt' markiert){
    BPs = leere Punktmenge;
    fuege Anfangs und Endpunkt von P in BPs ein;
    markiere P als benutzt;
    circle = false;
    CurrentPath=P;
    while(!circle) {
      suche einen Pfad P2 mit folgenden Eigenschaften:
        * Anfangs- oder Endpunkt von P2 == Endpunkt von CurrentPath
        * P2 ist als 'nicht benutzt' markiert

      markiere P2 als benutzt;
      if(EndPunkt von P2 == Endpunkt von CurrentPath)
        drehe P2 um;
      L1 = letzter Punkt von P2;
      if( L1 = Anfangspunkt von CurrentPath){
        verlaengere CurrentPath mit P2;
        fuege CurrentPath zu result hinzu;
        circle = true;
      }
    }
  }
  else if( L1 in Bps enthalten) {
    Teile CurrentPath an der Stelle L1, setze dabei

```

```

        CurrentPath = Anfangspfad // Position 0 bis L1
        Rest = Endpfad // L1 bis Ende
        verlaengere Rest um P2;
        fuege Rest zu result hinzu;
        loesche in Bps alle Punkte, die nach L1 eingefuegt wurden;
    }
    else{ // einfache Verlaengerung
        erweitere CurrentPath mit P2;
        fuege L1 zu Bps hinzu
    }
} // while !circle
} // P noch nicht benutzt
} // for all P in Ps2
return result;

```

Der Algorithmus nutzt aus, daß die Menge der übergebenen Segmente aus einfachen Kreisen besteht, die sich in einzelnen Punkten berühren können. Zunächst werden diese Berührungspunkte ermittelt ($O(n \cdot \log(n))$). Dann werden alle Pfade zum nächsten (evtl. den gleichen) solchen Punkt ermittelt. Die Berechnung aller Pfade nimmt dabei $O(n \cdot \log(n))$ Zeit in Anspruch, da für jedes Segment der nächste Nachbar in logarithmischer Zeit bestimmt wird. Solch ein Pfad bildet entweder selbst einen Kreis oder läßt sich durch weitere Pfade zu einem Kreis erweitern. Dies benötigt (in Abhängigkeit von der Anzahl der ermittelten Pfade) quadratische Zeit. Die Anzahl der Pfade kann zwar linear von der Anzahl der Segmente abhängen, liegt jedoch meist wesentlich darunter.

Die Pfade werden in Mengen geordneter Segmente umgewandelt. Es wird ein eingeschlossenes Dreieck für jeden Grenzkreis ermittelt. Dazu wird das kleinste Segment ermittelt (konstante Zeit) und beide angrenzenden Dreiecke bestimmt. Das größere Dreiecke befindet sich innerhalb des Kreises. Dann wird getestet, ob dieses Dreieck zur Region gehört (logarithmische Zeit). Ist dies der Fall, handelt es sich beim ermittelten Pfad um die Kontur des Faces und muß daher nicht weiter beachtet werden. Von allen anderen Kreisen werden die eingeschlossenen Dreiecke ermittelt, die die Löcher des Face bilden ($O(n^2)$). Es wird (wie oben beschrieben), ein eingeschlossenes Dreieck ermittelt und in die Menge der enthaltenen Segmente aufgenommen. Seiten, die nicht zum aktuellen Grenzkreis gehören, werden hinzugefügt, die Anderen werden gelöscht. Dadurch wird die Grenze ohne das gerade berechnete Dreieck ermittelt. Dies wird solange wiederholt, bis die Menge der Segmente leer ist.

Die noch notwendige Differenzbildung zur Berücksichtigung von Inseln geschieht in Linearzeit. Damit ergibt sich insgesamt eine quadratische Gesamtlaufzeit.

8.3.15 Kontur einer Region

Um die Kontur einer Region zu ermitteln, wird diese mit ihren Löcher vereinigt. Anschließend wird die Grenze von diesem Objekt gebildet. Da die Ermittlung der Löcher quadratische Zeit benötigt und die anderen Operationen diese Zeitschranke nicht überschreiten, ergibt sich auch insgesamt eine quadratische Laufzeit.

8.3.16 Ähnlichkeit zweier Objekte

Um die Ähnlichkeit zweier Objekte zu ermitteln, müssen Vereinigung und Schnitt ermittelt werden. Das ist jeweils in Linearzeit möglich. Anschließend müssen noch die entsprechenden Größenangaben ermittelt und ins Verhältnis gesetzt werden. Dies ist in konstanter Zeit möglich (vgl. Absatz 8.3.10). Somit wird auch gesamt nur lineare Zeit $O(n + m)$ benötigt.

8.3.17 Berechnung topologischer Beziehungen

8.3.17.1 Berechnung scharfer topologischer Beziehungen

Scharfe Beziehungen zwischen zwei Punkten: Um die entsprechende Matrix zu berechnen, ist es hier nur notwendig, gemischt durch beide Punktmengen durchzugehen und dabei zu prüfen, ob das jeweils kleinste Element zu beiden oder nur zu einem der Punkte gehört. Somit kann die Berechnung in Linearzeit $O(n + m)$ erfolgen.

Scharfe Beziehungen zwischen Punkt und Linie: Es ist klar, daß bei unterschiedlichen Objekttypen kein gemischter Durchlauf stattfinden kann, da die Elemente der Mengen nicht verglichen werden können. Für einen beliebigen Basispunkt kann man in $O(\log(n))$ feststellen, wieviele Basissegmente innerhalb der Linie an diesem Basispunkt anliegen. Dazu werden in konstanter Zeit alle anliegenden Basiselemente berechnet und in der angegebenen Zeit innerhalb der Linie gesucht. Ist diese Anzahl 0, so liegt dieser Punkt im Äußeren der Linie, ist sie 1, liegt der Punkt auf der Grenze der Linie und sonst liegt der Punkt im Inneren der Linie. Damit lassen sich topologische Beziehungen zwischen Punkt und Linie insgesamt in $O(m \log(n))$ berechnen, indem für jedes Element des Punktes die oben angegebenen Eigenschaften getestet werden. Dabei bezeichne m die Kardinalität des Punktes und n die Anzahl der in der Linie enthaltenen Basissegmente.

Scharfe Beziehungen zwischen Punkt und Region: Für einen beliebigen Basispunkt läßt sich in $O(\log(n))$ ermitteln, ob dieser zur betrachteten Region gehört. Dazu werden alle anliegenden Basisdreiecke in konstanter Zeit berechnet und dann in der Region gesucht. Dabei läßt sich ebenfalls prüfen, ob dieser Basispunkt auf der Grenze der Region liegt. Sind alle anliegenden Dreiecke in der Region vorhanden, so liegt der Punkt im Inneren der Region. Gehört keins dieser Dreiecke zur Region, so liegt der Punkt im Äußeren der Region. In den anderen Fällen, befindet sich der Punkt auf der Grenze der Region. Um die Beziehung von Punkt und Region zu berechnen genügt es, die oben genannten Eigenschaften für alle im Punkt enthaltenen Basispunkte zu testen. Damit wird $O(n \log(m))$ Zeit benötigt.

Scharfe Beziehungen zwischen zwei Linien: Hier ist ein gemischter Durchlauf möglich. Zusätzlich muß für das jeweils betrachtete Element getestet werden, ob die Endpunkte evtl. zu den Grenzen der Linien gehören. Oben wurde erläutert, daß dies in $O(\log(n))$ durchgeführt werden kann, womit die Gesamtzeit zur Berechnung der 9-Intersection-Matrix $O((m + n)(\log(m) + \log(n)))$ beträgt.

Scharfe Beziehungen zwischen Linie und Region: Hierzu werden für jedes Segment in der Linie mehrere Tests durchgeführt. In $O(\log(m))$ läßt sich für ein Basissegment testen, zu welchem Teil einer Region es gehört. Dazu werden beide anliegenden Basisdreiecke in konstanter Zeit bestimmt und in der Region gesucht. Sind beide Dreiecke in der Region vorhanden, so liegt das Segment im Inneren der Region. Wird nur ein Dreieck gefunden, befindet sich das Segment auf der Grenze der Region und ansonsten im Äußeren. In $O(\log(n))$ läßt sich für einen Endpunkt eines Segments testen, ob er zur Grenze der Region gehört und in $O(\log(m))$ kann der Teil der Region ermittelt werden, zu dem der Punkt gehört. Somit kann die gesamte Matrix in $O(n * (\log(m) + \log(n)))$ bestimmt werden.

Scharfe Beziehungen zwischen zwei Regionen: Um die topologischen Beziehungen zwischen zwei Regionen berechnen zu können, werden die beiden Mengen wieder gemischt durchgegangen. Es ist klar, daß sich durch entsprechende Vergleiche feststellen läßt, ob es Schnitte der Inneren mit Nicht-Grenzteilen der jeweils anderen Region gibt. Alle diese Schnitte können damit

in linearer Zeit berechnet werden. Somit fehlen in der Matrix noch die Grenzschnitte. Um diese zu berechnen, eignet sich wieder ein gemischter Durchlauf. Vom aktuell ermittelten minimalen Dreieck werden in konstanter Zeit die drei Seiten ermittelt. Für jede dieser Seiten läßt sich in $O(\log(n) + \log(m))$ der Teil von beiden Regionen ermitteln, zu dem es gehört. Ist das betrachtete Segment Teil der Grenze einer der beiden Regionen, erfolgt der entsprechende Eintrag in der Matrix. Die Gesamtmatrix läßt sich daher in $O((n + m) \cdot (\log(n) + \log(m)))$ ermitteln.

8.3.17.2 Topologische Beziehungen mit Betrachtung der Unschärfe

Unschärfe Topologie zwischen zwei Punkten: Hier müssen die Zugehörigkeitsgrade von gemeinsamen Koordinaten verglichen werden. Dies kann in konstanter Zeit geschehen. Um die gemeinsamen Koordinaten zu ermitteln, findet in gemischter Durchlauf statt. Damit ist das gesuchte 6-Tupel in Linearzeit zu ermitteln.

Unschärfe Topologie zwischen Punkt und Linie: Der maximale Zugehörigkeitsgrad an einem Basispunkt ist für eine Linie in $O(\log(n))$ bestimmbar. Da dieser Grad mit den Graden aller im Punkt enthaltenen Basispunkte verglichen werden muß, ergibt sich die Zeitschranke $O(m(\log(n)))$.

Unschärfe Topologie zwischen Punkt und Region: Hier gelten die gleichen Ausführungen wie zwischen Punkt und Linie.

Unschärfe Topologie zwischen zwei Linien Die Berechnung der Einträge für ein gemeinsames Segment läßt sich in konstanter Zeit ermitteln. Diese Segmente lassen sich durch einen gemischten Durchlauf in linearer Zeit bestimmen. Es fehlen daher nur noch die Einträge für gemeinsame Basispunkte. Um diese zu erhalten, wird die Menge der unscharfen Segmente einer Linie durchlaufen. Für das aktuelle Element werden die maximalen Zugehörigkeitsgrade beider Endpunkte in beiden Linien ermittelt. Dies benötigt für jeden Operanden logarithmische Zeit. Aus den Ergebnissen können dann die Einträge berechnet werden, was nur konstanten Aufwand bedeutet. Insgesamt ergibt sich somit eine Zeitschranke von $O(n \cdot (\log(m) + \log(n)))$.

Unschärfe Topologie zwischen Linie und Region Hier kann man die Elemente der Linie durchlaufen und alle anliegenden Dreiecke innerhalb der Region ermitteln. Für die beiden Endpunkte werden noch die maximal zugewiesenen Grade bestimmt. Die benötigten Einträge sind dann innerhalb konstanter Zeit berechenbar, woraus sich eine Gesamtzeit von $O(n \cdot (\log(n) + \log(m)))$ ergibt.

Unschärfe Topologie zwischen zwei Regionen Eine der beiden Regionen wird hier durchlaufen. Für das aktuelle Element des Durchlaufs, werden in dieser Region alle anliegenden Dreiecke gesucht. Damit lassen sich sowohl die maximalen Grade der Eckpunkte als auch die Funktionsverläufe über den Seiten ermitteln. In der zweiten Region werden das aktuelle und alle anliegenden Dreiecke gesucht, um die entsprechenden Werte zu ermitteln. Hiermit lassen sich alle notwendigen Einträge berechnen. Es ergibt sich also eine Gesamtzeit von $O(n \cdot (\log(m) + \log(n)))$.

8.4 Pakete

Um Programme zu strukturieren, bietet Java die Möglichkeit an, zusammengehörige Klassen in Paketen (packages) zu organisieren.

Das Programm, welches die Möglichkeiten der in dieser Arbeit vorgestellten Objekte zeigen soll, ist daher in verschiedenen Paketen untergebracht.

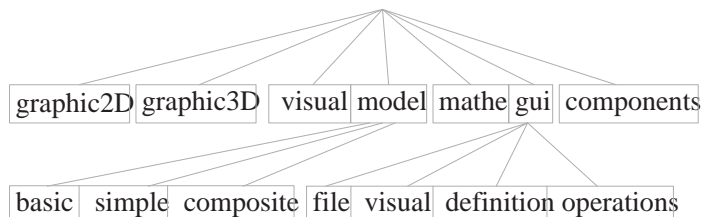


Abb. 37 Verwendete Pakete

Obwohl Java das Konzept von Unterpaketen nicht kennt (basic steht in keinem anderen Verhältnis zu model als graphic2D), ist es sinnvoll, die vorhandenen Pakete zu strukturieren.

Im Paket „graphic2d“ sind solche Klassen enthalten, die es ermöglichen Mengen 2-dimensionaler Objekte auf einem Monitor darzustellen.

„graphic3d“ liefert die notwendigen Klassen, um 3-dimensionale Objekte in eine 2-dimensionale Darstellung umzuwandeln.

Die Klassen des Pakets „visual“ stellen die Verbindung zwischen Modell und seiner Darstellung her.

Das Paket „model“ entspricht der Umsetzung des diskreten Modells in ein Programm. Die Unterpakete „basic“, „simple“ und „composite“ spiegeln den Aufbau von unscharfen Objekten wider.

Das Paket „mathe“ enthält hier nur die Klasse „Matrix“, welche von Klassen des Paketes „graphic3d“ benötigt wird.

Das Paket „gui“ mit seinen Unterpaketen implementiert die Benutzeroberfläche des Programms. Im Paket „components“ sind graphische Komponenten enthalten, die vom Paket „gui“ verwendet werden, jedoch auch innerhalb anderer Anwendungen benutzt werden können.

8.5 Klassen

In diesem Abschnitt werden wichtige Klassen und ihre Funktion beschrieben. Da das Wesentliche dieser Arbeit ein Modell für unscharfe Objekte ist, wird hier mit den Klassen des Pakets model begonnen.

Die Klasse Params hat die Aufgabe die Parameter des Modells festzuhalten. Insbesondere werden hier die Werte von a und b der Triangulierung festgelegt.

Im Paket model.basic werden die Klassen BasicPoint, BasicSegment sowie BasicTriangle definiert, welche den Objekten des diskreten Modells entsprechen. Da in diesem Paket auch die gesamte Information der Triangulierung abgelegt ist, existieren hier auch noch Methoden, die

- alle Basisobjekte berechnen, die eine gegebene Koordinate überdecken
- alle Basisdreiecke berechnen, die an einem Basissegment bzw. einem Basispunkt liegen
- alle Basissegmente berechnen, die einen gegebenen Basispunkt als Endpunkt besitzen
- alle Nachbarn eines Basispunktes berechnen

Durch die gleichmäßige Triangulierung ist es möglich, alle diese Berechnungen in konstanter Zeit auszuführen.

Das Paket model.simple enthält die zu den Elementarobjekten passenden Klassen. Hier finden auch Berechnungen für Zugehörigkeitsgrad, Größe usw. von solchen Objekten statt.

Die eigentlichen unscharfen Objekte befinden sich innerhalb des Paketes model.composite. Eine Klasse SortedObjects übernimmt die Verwaltung der in den Objekten enthaltenen Mengen.

Speziell wird hier ein Vector verwaltet, dessen Elemente sortiert sind. Somit werden Suchanfragen durch eine geeignete Methode in logarithmischer Zeit beantwortet. Das Einfügen am Ende kann bei entsprechender Größe des Vektors in konstanter Zeit erfolgen. Dadurch können die bereits angegebenen Zeitvorgaben erfüllt werden. Die weiteren Klassen enthalten genau eine solche Menge, weitere Attribute wie den Skalierungsfaktor und Methoden, die eine Realisierung der Operatoren darstellen. Auch sind bei den einzelnen unscharfen Objekten (bzw. deren Klassen) Methoden zur Berechnung topologischer Beziehungen untergebracht. Um das Ergebnis einer solchen Berechnung darstellen zu können, gibt es noch die Klasse M9Int, welche Matrizen des 9-Intersection-Modells bereitstellt. Eine weitere Klasse ist für die Darstellung unscharfer topologischer Beziehungen vorhanden.

Einige im Paket visual enthaltenen Klassen erweitern die Klassen für unscharfe Objekte um Methoden, die die unscharfen Elemente dieser Objekte in 2- oder 3-dimensionale Objekte umwandeln, die dann von den entsprechenden Paketen verarbeitet werden können. Andere Klassen erweitern die Klassen der Pakete graphic2d und graphic3d um Möglichkeiten, die zugrundeliegende Basis eines Bildes einzublenden sowie Objekte innerhalb eines Bildes anwählen zu können. Die Klassen von graphic2d bestehen aus den Objektklassen für 2D-Objekte, sowie weiteren Klassen, die es ermöglichen, diese Objekte auf dem Bildschirm darzustellen. Hier sind z.B. auch die Methoden enthalten, die den Farbverlauf von Linien und Dreiecken berechnen.

Die im Paket graphic3d enthaltenen Klassen projizieren 3D-Objekte auf eine Ebene. Dazu kann die Sicht festgelegt werden. Auch werden die Objekte hier an der Sichtpyramide geclippt.

Die Benutzeroberfläche bietet die folgenden Möglichkeiten:

- Erzeugen/Modifizieren neuer unscharfer Objekte
- Visualisierung der Objekte
- Ausführen von Operationen
- Abspeichern/Laden von Objekten

Kapitel 9

Visualisierung

9.1 Darstellungsmöglichkeiten

Um eine unscharfe Region darzustellen, ist es nicht sinnvoll, diese als einfarbige Fläche darzustellen. Dies ist für scharf begrenzte Regionen zwar eine gute Lösung, jedoch geht durch eine solche Darstellung die Unschärfe verloren. Es muß nun nach Möglichkeiten gesucht werden, die Region so darzustellen, daß zu erkennen ist, welche Gebiete zu welchem Grad der Region angehören. Ähnliche Ausführungen gelten auch für Linien und Punkte, auf die in diesem Kapitel jedoch nicht näher eingegangen wird.

9.1.1 Erzeugung von Farben

Um bei einer zweidimensionalen Darstellung die Unschärfe darzustellen, muß man mit verschiedenen Farben arbeiten. Einem Grad wird eine bestimmte Farbe zugewiesen. Für eine einheitliche Darstellung ist ein Farbübergang von der Farbe $C1$ zur Farbe $C2$ angemessen. Durch die Wahl dieser beiden Farben können mehrere Regionen innerhalb einer Darstellung unterschieden werden. Da der Zugehörigkeitsgrad einen Wert $\in [0, 1]$ annehmen kann, ergibt sich als Zuweisung einer Farbe zu einem Grad an Unschärfe :

$$C = C1 + Grad(C2 - C1)$$

In den folgenden Ausführungen wird angenommen, daß Farben im RGB-Modell dargestellt werden. Der Farbanteil jeder Grundfarbe wird durch einen Wert im Intervall $[0, 255]$ dargestellt. Die Zuweisung eines Zugehörigkeitsgrades zu einer Farbe kann in diesem Farbmodell wie folgt berechnet werden.

```
Color getColor(double Grad) {
    double cr, cg, cb;
    cr = c1r + Grad*(c2r-c1r);
    cg = c1g + Grad*(c2g-c1g);
    cb = c1b + Grad*(c2b-c1b);
    return new Color((int)cr, (int)cg, (int)cb);
}
```

Hier stellen $c?r$, $c?g$, $c?b$ die RGB-Werte der beiden Grenzfarben dar.

Aus dieser Fähigkeit eine Farbe aus einem bestimmten Grad an Zugehörigkeit zu berechnen, ergeben sich mehrere Möglichkeiten, die in den folgenden Abschnitten beschrieben werden sollen.

9.1.2 Die einfachste Darstellung

Eine sehr einfache Möglichkeit ist es, jedem Dreieck der Grundfläche eine feste Farbe zuzuweisen, z.B. die Farbe für den durchschnittlichen Grad der 3 Eckpunkte. Andere Alternativen sind die Wahl der Farbe des maximalen oder des minimalen Zugehörigkeitsgrades.

Vorteile der Darstellung :

Durch die Einfachheit der Darstellung ist eine schnelle und effiziente Implementierung möglich. Für Anwendungen, bei denen es nicht auf eine genaue Darstellung ankommt, wo man hingegen schnell Ergebnisse sehen will, ist diese Möglichkeit der Visualisierung angebracht.

Nachteile der Darstellung :

Die Einfachheit ist der größte Nachteil dieser Darstellung. So kann zum Beispiel nicht erkannt werden, ob sich zwischen zwei Dreiecken ein stetiger Übergang befindet. Es wird nicht ausgenutzt, daß es sich um eine stückweise lineare Approximation handelt, sondern einfach eine stückweise konstante Approximation dargestellt.

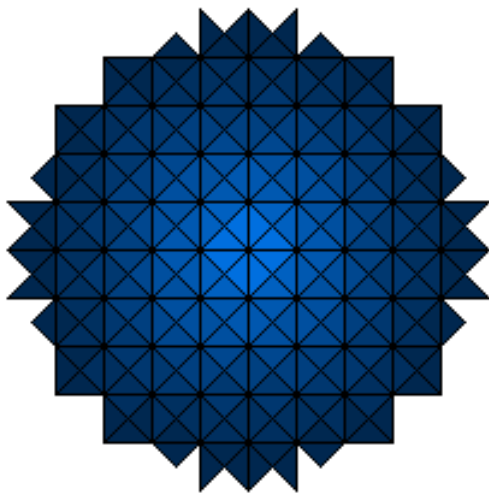


Abb. 38 Einfachste Darstellung einer Region

9.1.3 Eine verbesserte Darstellung

Auch eine zweidimensionale Darstellung kann die stückweise lineare Annäherung an die tatsächliche Zugehörigkeitsfunktion aufzeigen. Füllt man das Dreieck nicht mit einer einzelnen Farbe, sondern mit einem Farbverlauf zwischen den drei Farben, bleibt die Stetigkeitsinformation zwischen den einzelnen Dreiecken erhalten.

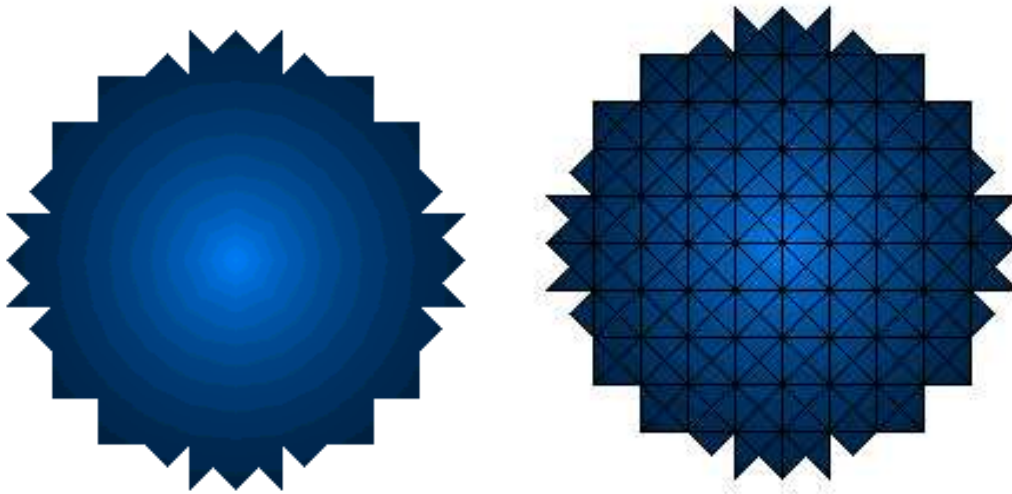


Abb. 39 Darstellung mit Erhalt der Stetigkeitsinformation

9.1.4 3D - Gitterdarstellung

Eine weitere Möglichkeit ist, den Zugehörigkeitsgrad als 3. Dimension eines jeden Punktes zu betrachten. Die Dreiecke der Regionen, die Segmente von Linien sowie die Koordinaten von Punktobjekten sind so als 3-dimensional aufzufassen. Durch Algorithmen, wie sie aus der Computergraphik bekannt sind, lassen sich solche Objekte auf eine Ebene projizieren, d.h. in eine zweidimensionale Darstellung überführen. So können die Objekte als Drahtgittermodelle dargestellt werden. Der Vorteil dieser Darstellung ist, daß sie noch recht einfach zu realisieren ist. Man braucht sich nicht um verdeckte Bereiche zu kümmern. Weiterhin benötigt diese Darstellung keine Farben, so daß selbst auf schwarz-weiß - Ausdrucken der Verlauf der Zugehörigkeitsfunktion gut zu erkennen ist. Nachteilig wirkt sich vor allem bei komplexen Regionen die dann unübersichtlich werdende Darstellung aus.

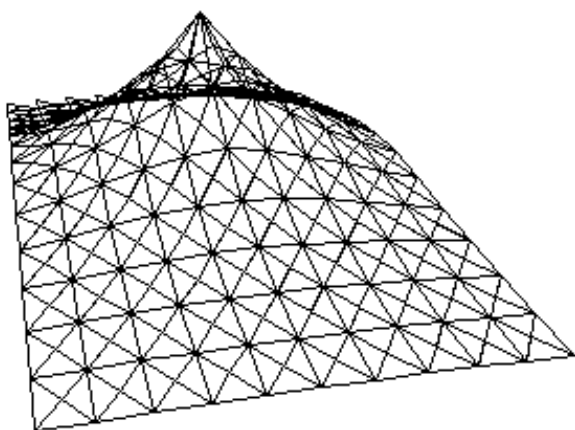


Abb. 40 Gitterdarstellung einer unscharfen Region

9.1.5 3D - solide Darstellung

Diese Unübersichtlichkeit in der Darstellung kann vermieden werden, wenn nicht-sichtbare Bereiche verdeckt werden. Eine einfache Möglichkeit dies zu erreichen, ist der sogenannte Maler-

Algorithmus. Dieser sortiert die (Dreiecks-) Flächen nach ihrer Entfernung vom Betrachter. Alle Flächen werden in dieser Reihenfolge gemalt, wobei mit den am weitesten entfernten Flächen angefangen wird. Durch das „Übermalen“ werden so die nichtsichtbaren Flächen verdeckt. Die Farbgestaltung erfolgt analog zu der einfachen Farbgebung in der zweidimensionalen Fläche.

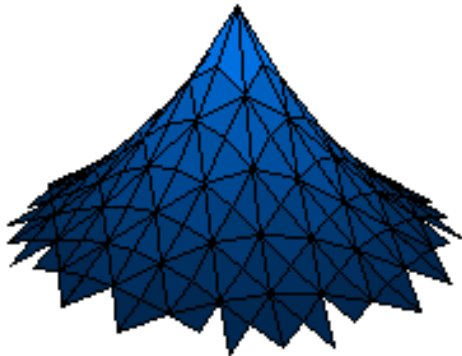


Abb. 41 3D-Darstellung einer unscharfen Region

9.1.6 3D Darstellung mit Verlauf

Man kann ausnutzen, daß durch die Transformationen (um eine dreidimensionale Szene in ein zweidimensionales Bild umzuwandeln) die Farben der Eckpunkte eines Dreiecks nicht verändert werden. Auch an der Tatsache, daß es zwischen diesen Punkten einen linearen Farbverlauf gibt, wird nichts geändert. Daher kann der Füllalgorithmus, der den Farbverlauf in der zweidimensionalen Darstellung berechnet, auch für Dreiecke verwendet werden, die einer dreidimensionalen Szene entstammen. Die entsprechende Graphik hat das folgende Aussehen:



Abb. 42 Verbesserte 3D-Darstellung einer unscharfen Region

9.2 Berechnung der 3D-Darstellung

Möchte man eine dreidimensionale Ansicht eines unscharfen Objekts haben, so muß aus den dreidimensionalen Koordinaten ein zweidimensionales Bild berechnet werden. Als Ausgangsbasis besitzt man eine dreidimensionale Szene, welche Punkte, Linien und Dreiecke enthalten kann. Um diese Szene in ein zweidimensionales Bild zu konvertieren, muß zuerst festgelegt werden, was

zu sehen sein soll. Die Ausführungen werden in verschiedenen Büchern der Computergraphik vorgestellt ([10]). Hierzu werden die folgenden Punkte bzw. Vektoren festgelegt:

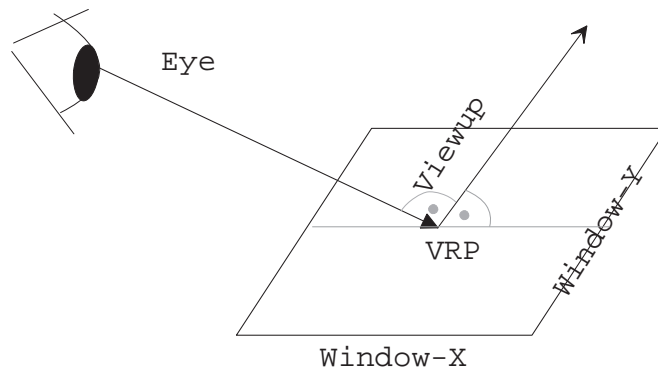


Abb. 43 Festlegen der Sicht

Der Punkt EYE bestimmt, von wo aus die Szene betrachtet werden soll. Der VRP (View Reference Point) gibt an, in welche Richtung der Betrachter der Szene schaut. Um den VRP wird das Window (die Projektionsfläche) festgelegt. Da sich der VRP innerhalb dieser Fläche befindet, bestimmt der Abstand zwischen EYE und VRP auch den Grad der Verzerrung, welcher durch die Projektion auftritt. Je größer der Abstand zwischen diesen beiden Punkten ist, desto geringer ist die Verzerrung. Der durch EYE und VRP bestimmte Vektor steht senkrecht auf der Projektionsfläche. Der ViewUp-Vektor legt die Richtung der y-Achse der Projektionsfläche fest. Die Größe des Window bestimmt den sichtbaren Ausschnitt, i.E. wird durch die Größe des Window das Ausmaß der sogenannten Sichtpyramide bestimmt, an der die Objekte der Szene geclippt werden.

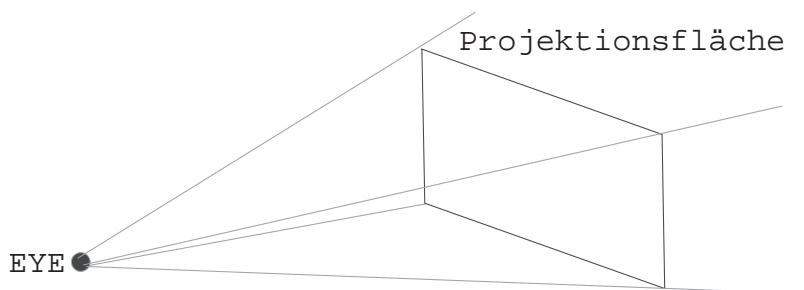


Abb. 44 Die Sichtpyramide

Alle Objekte, die sich innerhalb der nach hinten offenen Sichtpyramide befinden, sind sichtbar. Objekte, welche nur teilweise im Inneren liegen, werden entsprechend abgeschnitten. Alle Objekte werden nun wie folgt verarbeitet:

1. Normalisierung
2. Anpassen an die Einheitspyramide
3. Clippen an der Einheitspyramide
4. Rückanpassung

5. Projektion auf das Window

6. Abbildung des Window auf den Viewport

Durch Translation und Rotation werden EYE, VRP und ViewUp-Vektor so verändert, daß die Projektionsfläche parallel zu der durch ($z=0$) definierten Fläche liegt. EYE liegt dabei genau auf der Koordinate(0,0,0). Alle Objekte werden genauso behandelt, so daß sich die gesamte Szene entsprechend mit verändert, d.h. noch das Gleiche zu sehen ist.

Um an der Pyramide clippen zu können, wird diese zunächst in die Einheitspyramide überführt (Skalierung), d.h. der Abstand zwischen EYE und VRP beträgt 1 und das Ausmaß des Window ist 2×2 . Alle Objekte werden entsprechend angepaßt.

Im nächsten Schritt werden die Objekte der Szene an der Pyramide geclippt. Zuerst wird an einer Hilfsfläche geclippt, die parallel zur Projektionsebene liegt und sich kurz vor dem Auge befindet. Dadurch werden Objekte hinter dem Auge nicht mit projiziert. Diese Ebene muß sich kurz vor dem Auge befinden, da Punkte, die genau auf dem EYE-Punkt liegen, später bei der Projektion zu einer Division durch Null führen würden.

Dann wird die Skalierung mit den nun geclippten Objekten rückgängig gemacht.

Nun findet die Projektion, also die Umwandlung von dreidimensionalen in zweidimensionale Koordinaten statt.

Da die Größe der Projektionsfläche nicht zwingend mit der auf dem Ausgabegerät zur Verfügung stehenden Fläche (Viewport) übereinstimmt, findet nun noch eine Skalierung statt.

Anschließend können die nun zweidimensionalen Objekte auf dem Ausgabegerät (bzw. einem Zwischenpuffer) dargestellt werden.

Kapitel 10

Vorschlag für eine Benutzeroberfläche

Die letzten beiden Kapitel beschäftigten sich damit, wie das Modell implementiert werden kann und wie unscharfe Objekte dargestellt werden können. Hier wird eine Benutzeroberfläche vorgestellt, die das implementierte Modell sowie die vorgestellten Darstellungsmöglichkeiten nutzt. Es werden Themen behandelt, wie sich unscharfe Objekte möglichst komfortabel erstellen und manipulieren lassen.

10.1 Darstellung von unscharfen Objekten

Die Repräsentation von unscharfen Objekten wurde im Kapitel 9 vorgestellt. In diesem Abschnitt soll die Frage geklärt werden, welche Einstellungsmöglichkeiten ein Programmbenutzer erhalten sollte, um die gewünschte Ansicht eines unscharfen Objekts zu erhalten.

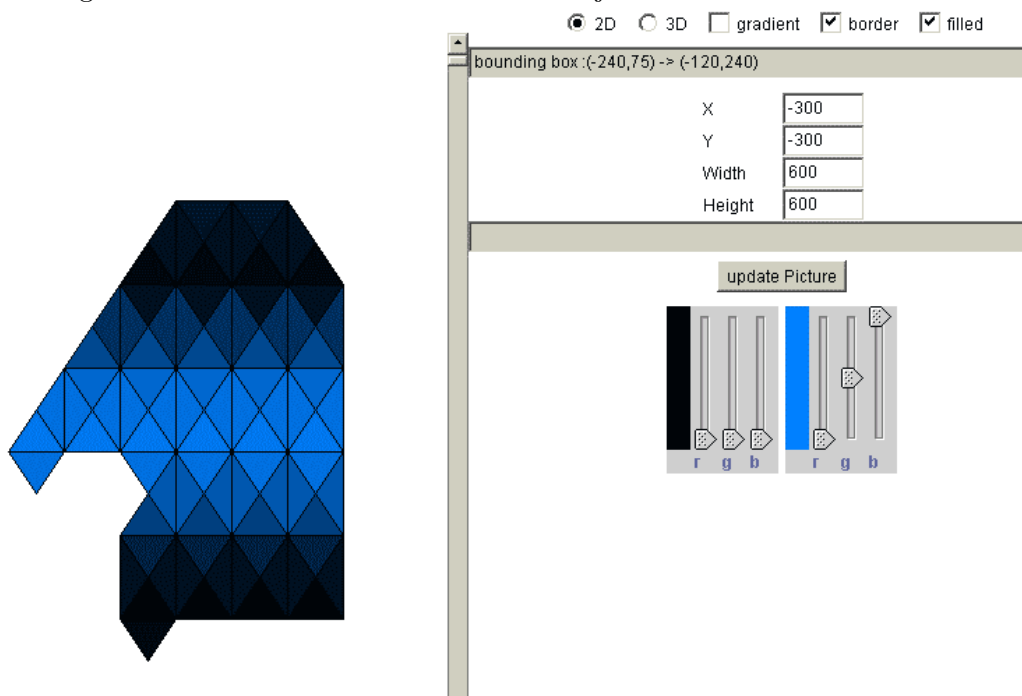


Abb. 45 Parameter in der 2D-Darstellung

10.1.1 Allgemeine Parameter

Um ein unscharfes Objekt darzustellen, sollten die beiden Farben gewählt werden können, die den Grad 0 bzw. 1 repräsentieren. Für Punkte, die einen Zugehörigkeitsgrad in $(0, 1)$ besitzen, kann die entsprechende Farbe dann berechnet werden (vgl. Kapitel 9). Jede der beiden Farben kann durch 3 Slider eingestellt werden, die jeweils den Rot-, Grün- und Blauanteil dieser Farbe darstellen.

Weiterhin läßt sich die Art der Farbdarstellung einstellen. D.h. man kann umschalten, ob eine einzelne Farbe pro unscharfem Elementarobjekt gewählt wird, oder ob innerhalb dieser Bausteine ein Farbverlauf zu sehen sein soll.

Speziell für Regionen läßt sich noch einstellen, ob Rand und/oder Innere Teile der einzelnen Bausteine dargestellt werden sollen. Wird nur der Rand gewählt, so erhält man eine Gitterdarstellung der betreffenden Region.

10.1.2 Spezielle Parameter der 2D-Darstellung

Um die Parameter der 2D-Darstellung zu erläutern, muß erst geklärt werden, wie der darzustellende Ausschnitt des \mathbb{R}^2 auf den Bildschirm gelangt.

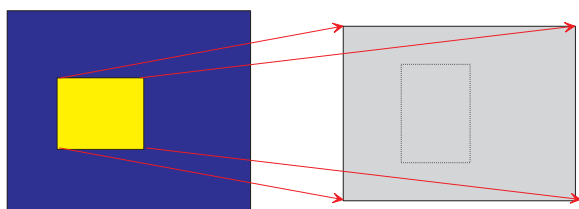


Abb. 46 Vom \mathbb{R}^2 auf den Monitor

Das blaue Rechteck stellt in diesem Bild den \mathbb{R}^2 dar (bzw. den Teil davon, der durch den Zahlenbereich des Computers vorgegeben ist). Innerhalb des blauen Rechtecks kann ein Ausschnitt ausgewählt werden. Dieser Ausschnitt wird auf ein Bild fester Größe abgebildet (graues Rechteck rechts im Bild). Das darin enthaltene Rechteck stellt den sichtbaren Teil dieses Bildes dar. Der sichtbare Ausschnitt kann an eine beliebige Position innerhalb des darunterliegenden Bildes geschoben werden.

Der Ausschnitt des \mathbb{R}^2 läßt sich einfach durch Angabe vom linken unteren Eckpunkt sowie Breite und Höhe per Tastatur eingeben. Um zusätzliche Berechnungen (und somit Zeit) zu sparen, werden bei diesen Eingaben nur Vielfache von a und b (den Parametern der Triangulation) erlaubt. Bei anderen Eingaben wird automatisch ein benachbartes Vielfaches gewählt. Durch die Angabe des umgebenden Rechtecks des zu betrachtenden Objekts ist leicht es möglich, den Ausschnitt so einzustellen, daß das unscharfe Objekt auch tatsächlich angezeigt wird.

Die Größe des sichtbaren Bereichs läßt sich durch Vergrößern bzw. Verkleinern des Programmfensters einstellen. Die Position dieses Ausschnitts läßt sich über zwei Scrollbalken beliebig regeln.

10.1.3 Spezielle Parameter der 3D-Darstellung

Wie im Kapitel 9 gezeigt, müssen für die 3-dimensionale Darstellung von Objekten folgende Parameter einzustellen sein:

- Position des Auges
- Position des Blickpunktes
- Richtung des View-Up-Vectors
- Größe des sichtbaren Ausschnitts

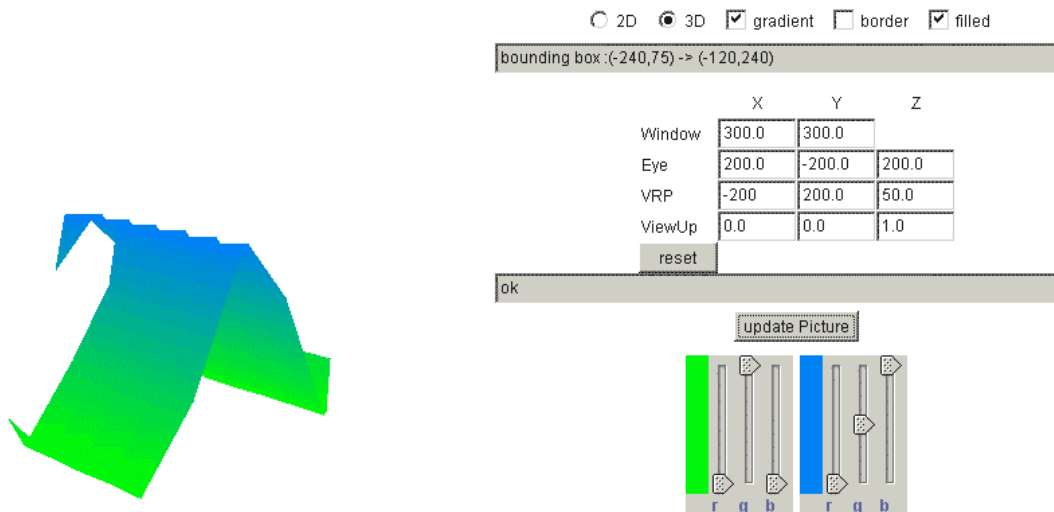


Abb. 47 Parameter der 3D-Darstellung

Da diese Angaben keine festen Unter- und Obergrenzen besitzen, ist es nicht sinnvoll, Scrollbalken o.ä. für deren Eingabe zu verwenden. Daher finden diese Einstellungen per Tastatur statt. Um zu vermeiden, daß jedesmal ein Bild berechnet wird, nachdem ein Parameter geändert wurde, wird das Bild erst dann erneuert, wenn der entsprechende Button gedrückt wurde. Das Ändern mehrerer Parameter würde sonst zu lange dauern.

10.2 Erzeugung unscharfer Objekte

Um mit unscharfen Objekten arbeiten zu können, müssen diese in die gewünschte Computerdarstellung übertragen werden. Dazu ist es notwendig, ausgewählten Teilen des \mathbb{R}^2 Zugehörigkeitsgrade zuzuordnen. Weiterhin muß einem unscharfen Objekt auch ein Skalierungsfaktor zugewiesen werden können. Die Auswahl der gewünschten Objektteile findet stets in der 2-dimensionalen Darstellung statt, da die Positionseingabe in der 3D-Darstellung entweder spezielle Eingabegeräte erfordert oder sehr unkomfortabel ist. Da der Skalierungsfaktor bis auf den begrenzten Zahlenbereich des Computers keine Obergrenze hat, kommt hierfür nur eine einfache Eingabe über die Tastatur in Frage.

10.2.1 Erstellen von unscharfen Punkten

Ein unscharfer Punkt besteht aus einer endlichen Menge von Punkten des \mathbb{R}^2 , denen ein Zugehörigkeitsgrad zugewiesen werden muß. Da der Zugehörigkeitsgrad sich im Bereich zwischen 0 und 1 befindet, erscheint eine Eingabe über einen Slider sinnvoll. Die Auswahl des betreffenden Punktes erfolgt in der aktuellen Darstellung des Ausschnitts des \mathbb{R}^2 per Mouse. Ausgehend von

der Position, an der ein Mouseclick stattfindet, wird der nächstgelegene Basispunkt markiert. Um diesen zu bestimmen, muß zuerst die angeklickte Bildschirmkoordinate in die entsprechende Koordinate des \mathbb{R}^2 transformiert werden. Dies geschieht in mehreren Schritten:

1. Bestimmen der Koordinate im darunterliegenden Bild
Dazu werden die Werte der Scrollbalken auf die Mousekoordinate aufaddiert (X-Werte) bzw. subtrahiert (Y-Werte).
2. Bestimmung der Koordinate innerhalb des Auswahlbereichs (gelbes Rechteck in Abb. 46). Hier handelt es sich um eine einfache Skalierung. Es bezeichne (X, Y) die im Schritt 1 berechnete Koordinate, X_W und Y_W die Breite bzw. die Höhe des Auswahlbereichs und X_P, Y_P die Größe des Bildes (graues Rechteck in Abb. 46). Dann berechnet sich die Koordinate innerhalb des Auswahlbereichs durch:

$$X_A = \frac{X \cdot X_W}{X_P}$$

$$Y_A = \frac{(Y_P - Y) \cdot Y_W}{Y_P}$$
 (Bildschirmdarstellung und mathematisches Koordinatensystem haben unterschiedliche Y-Richtungen)
3. Bestimmung der Koordinate im \mathbb{R}^2
Hier handelt es sich um eine einfache Verschiebung. Die komponentenweise Addition der Position der linken unteren Ecke des Auswahlbereichs führt die gewünschte Transformation durch.

Aus der so ermittelten Koordinate läßt sich der nächstgelegene Basispunkt berechnen. Dies ist in konstanter Zeit möglich. Zunächst wird das Rechteck der X-Triangulierung ermittelt, in dem die Koordinate liegt. Dazu wird die linke untere Ecke dieses Rechtecks ermittelt. Dies geschieht durch die Programmzeilen:

```
Ecke_X = (X / a) * a;
Ecke_Y = (Y / b) * b;
if (X<0)
    Ecke_X = Ecke_X - a;
if (Y<0)
    Ecke_Y = Ecke_Y - b;
```

X und Y bezeichnen hier die gegebene Koordinate und a, b die Parameter der X-Triangulierung. In den ersten beiden Zeilen wird die Ecke des Rechtecks bestimmt, die dem Koordinatenursprung am nächsten liegt. Zuerst wird bestimmt, wieviele Rechtecke zwischen Nullpunkt und dieser Ecke liegen (X/a bzw. Y/b). Da es sich um eine Integer-Division handelt, werden nur volle Rechtecke betrachtet. Anschließend wird durch die Multiplikation mit a bzw. b die gesuchte Ecke berechnet. In den nächsten Zeilen wird die linke untere Ecke des Rechtecks bestimmt. Bei positiven Werten handelt es sich genau um die in Schritt 1 berechneten Koordinaten, im anderen Fall muß eine Verschiebung stattfinden.

Da nun die linke untere Ecke des Rechtecks bekannt ist, und da es sich um eine gleichmäßige Triangulierung handelt, können alle 5 Punkte, die von diesem Rechteck betroffen sind, ermittelt werden. Nun muß nur noch der euklidische Abstand der Eingangskoordinate zu diesen 5 Punkten berechnet werden. Somit ist der nächstgelegene Basispunkt bestimmt.

Dieser wird durch das Programm wieder in Bildschirmdarstellung umgerechnet und markiert. Durch den Slider läßt sich der gewünschte Grad an Zugehörigkeit einstellen. Um die Einstellung der Extremwerte 0 und 1 zu erleichtern, gibt es zu diesem Zweck zusätzliche Buttons. Wird die Einstellung akzeptiert, so wird der Grad innerhalb des unscharfen Punktes übernommen (ggf. wird ein neuer unscharfer Elementarpunkt eingefügt). Bei einem Grad von 0 wird der markierte Basispunkt aus dem unscharfen Punkt entfernt.

10.2.2 Erstellen unscharfer Linien

Wie auch bei unscharfen Punkten, lassen sich hier einzelne Basiselemente (hier Segmente) auswählen. Durch Angabe von zwei Graden (für jeden Endpunkt einen), kann ein unscharfes Segment erzeugt werden, das der unscharfen Linie hinzugefügt werden kann. Sind beide Grade als Null gewählt worden, wird das betreffende Segment (falls vorhanden) aus der Linie entfernt. Es muß zuerst das dem Mouseclick nächstgelegene Segment gefunden werden. Dazu wird wie auch bei Punkten das Rechteck der X-Triangulierung ermittelt, in dem die betreffende Koordinate gewählt wurde. Dieses Rechteck wird nun zusätzlich in Quadranten unterteilt:

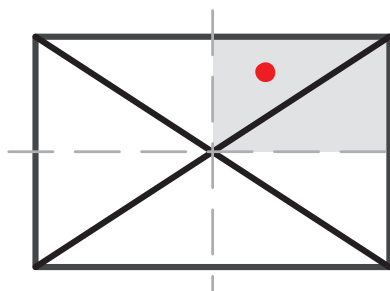


Abb. 48 Einteilung in Quadranten

Man erkennt, daß nur noch 3 von den 8 im Rechteck enthaltenen Segmenten betrachtet werden müssen. Der Abstand zum senkrechten oder zum waagerechten Segment läßt sich einfach durch die Differenz der X- bzw. Y-Werte der Koordinate und des Segments ermitteln.

Bei den Diagonalen läßt sich der Abstand wie folgt berechnen:

Die Gerade sei durch die Gleichung $mx + ny + c = 0$ gegeben. Dann ergibt sich der Abstand vom gegebenen Punkt mit dieser Gerade durch: $d = \frac{xm+yn+c}{\sqrt{m^2+n^2}}$ (vgl. [24]). Durch Verschieben des Rechtecks in den Koordinatenursprung erhält man folgende Gleichungen für die Geraden:

steigende Gerade: $y = \frac{b}{a}x + 0 \Leftrightarrow x \cdot \frac{b}{a} - 1 \cdot y = 0$

fallende Gerade: $y = -\frac{b}{a}x + b \Leftrightarrow x \cdot -\frac{b}{a} - 1 \cdot y + b = 0$

Daraus ergeben sich die Abstände wie folgt:

steigende Gerade: $m = \frac{b}{a}; n = -1; c = 0$

$$\Rightarrow d = \frac{x \frac{b}{a} - y}{\sqrt{\left(\frac{b}{a}\right)^2 + 1}}$$

fallende Gerade: $m = -\frac{b}{a}; n = -1; c = b$

$$\Rightarrow d = \frac{-x \frac{b}{a} - y + b}{\sqrt{\left(\frac{b}{a}\right)^2 + 1}}$$

Da hiermit jedoch der gerichtete Abstand berechnet wird, muß anschließend noch der Absolutbetrag von d berechnet werden. Durch den Vergleich der 3 ermittelten Abstände (zur Senkrechten, Waagerechten und der Diagonalen) läßt sich das am nächsten liegende Segment ermitteln.

Dieses wird wieder in die Bildschirmdarstellung umgerechnet und markiert. Dann kann beiden Eckpunkten der gewünschte Grad zugeordnet werden.

Für eine einfachere Eingabe von langen Linien mit einer kontinuierlichen Steigung, bietet sich eine andere Vorgehensweise an. Zunächst wird die gesamte Linie (eine Folge von Basissegmenten) markiert. Dazu wird als Erstes der Anfangspunkt der Linie gesetzt. Dann werden weitere Basispunkte ausgewählt. Das Programm selbst berechnet einen kürzesten Pfad zwischen dem letzten Punkt des bereits erstellten Pfades und dem neuen Punkt. Die Auswahl der Punkte wird

dabei so gehandhabt, wie die Auswahl einfacher Basispunkte. Dann wird geprüft, ob sich der vorhandene Pfad durch den neu berechneten Pfad verlängern läßt. Dazu dürfen die beiden Pfade keine gemeinsamen Segmente besitzen, wogegen Schnittpunkte erlaubt sind. Bei erfolgreicher Prüfung wird der neue Pfad an den vorhandenen angehängt.

Berechnung des Pfades zwischen zwei Basispunkten:

Durch die gewählte Triangulierung ist die Berechnung eines kürzesten Pfades in linearer Zeit in Abhängigkeit von seiner Länge möglich. Es soll kurz beschrieben werden, mit welchem Algorithmus das nächste Segment gewählt wird. Dazu sei B der aktuelle Punkt und Z der Zielpunkt des Pfades:

```
{ if (B in der Mitte eines X) {
    Berechne die benachbarten Punkte von B;
    Waehle den mit der geringsten Entfernung zu Z;
}
else{
    Berechne die benachbarten Punkte von B;
    Schliesse die Punkte aus, bei denen sich der Abstand zu Z vergroessert
    Pruefe Diagonalen: (B,C)
    if (isequal(Z,C)) {
        Verlaengere Pfad um (B,C)
    }
    else{
        if ( Z.x != C.x) & (Z.y != C.Y) ){
            Verlaengere Pfad um (B,C);
        }
        else{
            Waehle die Senkrechte oder Waagerechte, deren
            Endpunkt Z am nachsten ist
        }
    }
}
```

Die Idee des Algorithmus ist recht simpel. Solange sich X- und Y-Koordinate des aktuellen Punktes auch im nächsten Schritt dem Zielpunkt nähern lassen, wird eine Diagonale gewählt, ansonsten ein achsenparalleles Segment. Dadurch, daß auch der nächste Schritt mit einbezogen wird, ist gesichert, daß sich beide Werte auch dann noch dem Zielpunkt nähern, wenn die vollständige Diagonale eines Rechtecks von der X-Triangulierung durchlaufen wird. Aufgrund der Dreiecksungleichung wird dadurch ein kürzester Pfad ermittelt. Läßt sich das Rechteck nicht vollständig durchlaufen, so würde der Zielpunkt genau über oder neben dem neuen Punkt liegen. Dann ist es jedoch günstiger, sich durch achsenparallele Segmente zu nähern und dann nur noch eine Diagonale zu benutzen.

Ist der Pfad markiert, lassen sich die Zugehörigkeitsgrade der beiden Endpunkte mittels Slider einstellen. Durch einfache lineare Approximation werden die Grade der einzelnen enthaltenen Punkte ermittelt und anschließend die betreffenden Segmente der unscharfen Linie hinzugefügt.

10.2.3 Erstellung unscharfer Regionen

Eine einfache Möglichkeit, unscharfe Regionen zu erstellen, ist die schrittweise Selektion von Basisdreiecken und das Zuweisen von Graden zu den drei Eckpunkten. Um das Dreieck zu ermitteln, welches selektiert wird, wird zuerst wie bei Linien und Punkten das angeklickte Rechteck bestimmt. Dieses wird dann in den Koordinatenursprung verschoben. Welches der Dreiecke

angeklickt wurde, läßt sich herausfinden, indem untersucht wird, ob der Punkt oberhalb oder unterhalb der beiden Diagonalen liegt. Dieses Vorgehen führt zu 4 Fällen, die genau die 4 enthaltenen Dreiecke beschreiben. Ob sich der Punkt über oder unter einer der Diagonalen befindet, ist durch die folgenden Prädikate gegeben: (X und Y stellen die (zusammen mit dem Rechteck) verschobenen Werte des markierten Punktes dar.)

```
steigendueber = Y > X*b/a;
fallendueber  = Y > b-X*b/a;
```

Versucht man jedoch größere Regionen zu erstellen, ist diese Vorgehensweise sehr mühsam. Das Programm bietet daher die Möglichkeit, Regionen durch Angabe der Grenze und Zuweisen einer von mehreren parametrisierten Zugehörigkeitsfunktionen zu definieren. Mit einem solchen Schritt läßt sich eine zusammenhängende Region mit einer kontinuierlichen Zugehörigkeitsfunktion ohne Löcher definieren. Durch wiederholtes Ausführen lassen sich auch komplexe Regionen angeben. Das Markieren der Grenze geschieht analog zum Markieren einer Linie mit dem Unterschied, daß keine Schnittpunkte bei einer Pfadverlängerung zugelassen sind. Wird der Pfad geschlossen, werden alle enthaltenen Basisdreiecke ermittelt und markiert. Die Berechnung dieser inneren Dreiecke entspricht der bei Löchern.

10.3 Manipulation von unscharfen Objekten

Die Manipulation unscharfer Objekte entspricht größtenteils einer Neudefinition. Hier wird jedoch nicht von einem neuen, sondern einem vorhandenen Objekt ausgegangen. Auf eine detaillierte Beschreibung kann daher verzichtet werden.

10.4 Operationen auf unscharfen Objekten

Alle gespeicherten Objekte lassen sich in eine Operanden-Liste eintragen, um Operationen auf ihnen ausführen zu können.

Die Manipulation der Liste ist im folgenden Bild zu erkennen:

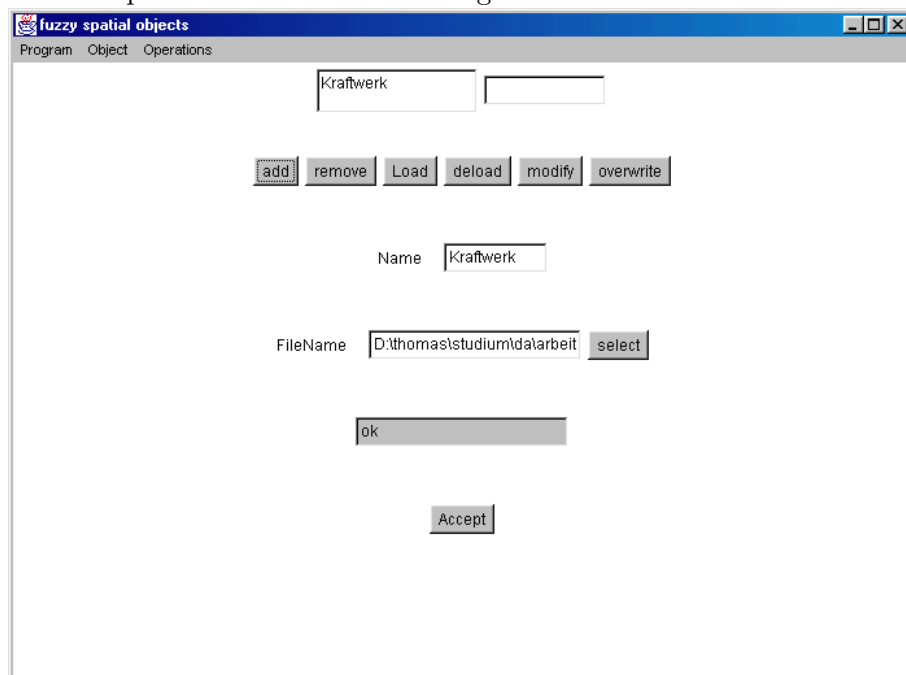


Abb. 49 Manipulation der Operandenliste

Im Feld **Filename** läßt sich der Dateiname eines gespeicherten unscharfen Objekts direkt angeben. Betätigt man den **Select**-Button, kann diese Auswahl auch über einen Dateimanager erfolgen. Im Feld **Name** läßt sich der Name angeben, mit dem das Objekt aus der Datei im Programm angesprochen werden kann. Das hat z.B. den Vorteil, daß auch gleichnamige Dateien aus unterschiedlichen Verzeichnissen verarbeitet werden können. Mittels **add**-Button wird das beschriebene Objekt in die Operandenliste eingefügt. Durch **remove** kann ein in der Liste selektierter Eintrag wieder entfernt werden. Um nicht unnötig Speicherplatz zu verwenden, wird das betreffende Objekt jedoch noch nicht aus der Datei eingeladen; dies geschieht erst bei Bedarf, d.h. wenn das Objekt bei einer Operation benutzt wird. Dieses Vorgehen hat jedoch den Nachteil, daß fehlerhafte Dateien auch erst zu diesem Zeitpunkt erkannt werden. Der **load**-Button bietet daher die Möglichkeit, das Objekt auch sofort in den Speicher zu laden und dabei evtl. Fehler zu erkennen. Der **deload**-Button übergibt den durch das Objekt belegten Speicher an die Java-interne Speicherbereinigung. Mit dem **modify**-Button werden die Name und Dateiname des selektierten Listeneintrags in die unteren Felder übernommen. Der **overwrite**-Button dient noch dazu, einen Listeneintrag durch neue Werte zu überschreiben. Sind die gewünschten Objekte in der Liste vorhanden, gelangt man über den **accept**-Button wieder zu den Operationen.

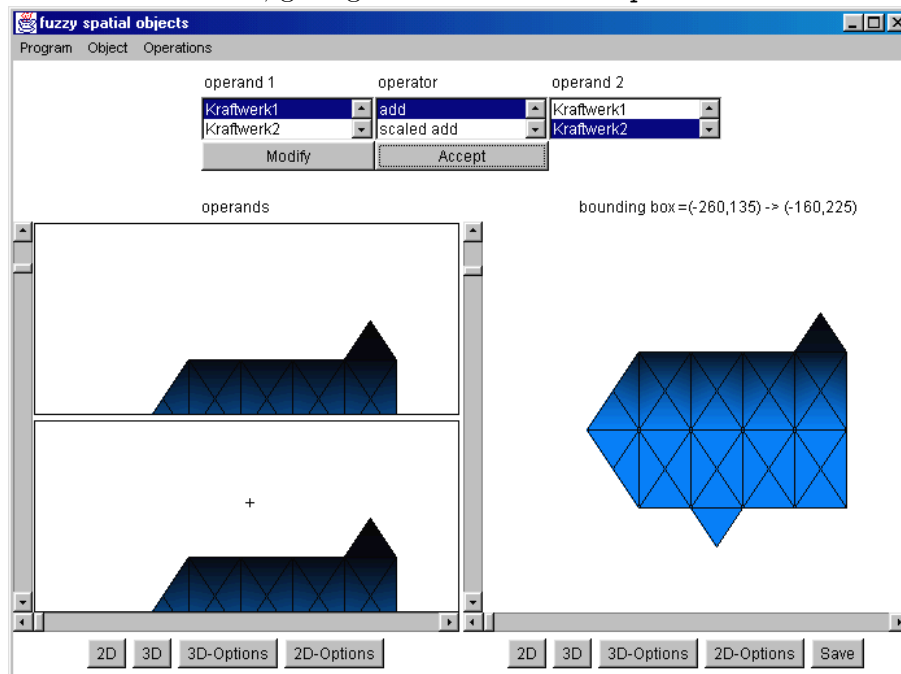


Abb. 50 Anwendung binärer Operationen

Im oberen Teil des Bildes sind drei Listen zu erkennen. Die äußeren beiden Listen stellen die Operandenliste dar. Durch den **modify**-Button gelangt man in den oben beschriebenen Programmteil zu deren Manipulation. In der mittleren Liste lassen sich die verfügbaren Operationen auswählen. Sind die drei notwendigen Markierungen gesetzt (also in jeder Liste ein Eintrag ausgewählt), wird die Berechnung der Operation durch den **accept**-Button gestartet. Beide Operanden sowie das Ergebnis werden im unteren Teil des Bildes dargestellt. Da in der zweidimensionalen Darstellung überlappende Bereiche innerhalb einer Darstellung schlecht dargestellt werden können, wird jedes Objekt für sich dargestellt. Beide Bilder zeigen dabei den gleichen Ausschnitt des \mathbb{R}^2 , wodurch eine bessere Vergleichsmöglichkeit gegeben ist. Befindet sich der Mauszeiger über einem der beiden Darstellungsbereiche, wird die dazugehörige Position zusätzlich im anderen Bereich markiert.

Die Ansichten der Operanden und des Ergebnisses lassen sich über die unteren Buttons wie gewünscht manipulieren. Man kann zwischen 2 und 3-dimensionaler Darstellung wählen und entsprechende Einstellungen vornehmen. Beim Ergebnis besteht zudem die Möglichkeit, es zu

speichern. Damit kann es anschließend auch in die Operanden-Liste eingetragen werden. Über den Menüpunkt **Objekt** kann außerdem zur Visualisierung oder zur Manipulation des Ergebnisses umgeschaltet werden.

Kapitel 11

Einbindung in Datenbanken

Nachdem nun die unscharfen Objekte modelliert und implementiert sind, stellt sich die Frage nach einer effizienten Verwaltung großer Mengen solcher Objekte. Um mit riesigen Mengen an Informationen umzugehen, bieten Datenbanken die entsprechenden Möglichkeiten. In dieser Arbeit sollen die vorgestellten Objekte in keine spezielle Datenbank integriert werden. Vielmehr soll hier gezeigt werden, welche Möglichkeiten sich durch die Einbindung unscharfer Objekte in Datenbanken ergeben.

Hier wird von einem erweiterbaren Datenbanksystem ausgegangen. D.h. die Datenbank kann um Typen erweitert werden. Weiterhin läßt sich die Anfragesprache um die benötigten Operatoren erweitern. Dieses Vorgehen hat den Vorteil, daß das vorgestellte Modell zunächst unabhängig von einem konkreten Datenbankmodell entwickelt werden konnte. Dadurch ist das Ergebnis sehr allgemein und kann an unterschiedliche Datenbanken angepaßt werden.

Ein Datenbanksystem sei nun also um die 3 unscharfen Objekte erweitert worden, d.h. solche Objekte können in der Datenbank gespeichert werden und können das Ergebnis von Anfragen sein. Die um Operationen auf unscharfen Objekten erweiterte Anfragesprache sei mit fuzzy-SQL oder kurz FSQL bezeichnet. Es gebe weiterhin einen sum-Operator, der die wiederholte Anwendung der Vereinigung darstellt. Mit diesen Mitteln lassen sich entsprechende Aufgaben lösen.

In der oben beschriebenen Datenbank seien folgende Tabellen definiert, aus denen sich auch das Schema der Datenbank ableiten läßt:

<i>Verschmutzung</i>			
Gebiet	Art	Verursacher	...
region1	CO	Kraftwerk X	...
region2	N	Kraftwerk X	...
...			

<i>Vorkommen von Lebewesen</i>		
Gebiet	Art	...
region1	Fuchs	...
region2	Hecht	...
...		

<i>Grundstücke</i>			
Gebiet	Besitzer	Nutzung	...
region1	A	Acker	...
region2	B	Bauland	...
...			

Nun sollen einige Szenarien konstruiert werden und jeweils eine Anfrage gestellt werden, die das Problem desselben löst.

Ein Grundstücksbesitzer A möchte sein gesamtes Grundstück einzäunen. Um die Länge dieses

Zaunes zu ermitteln, kann er folgende Anfrage an das DBS stellen:

```
SELECT   len(sum(boundary(Gebiet)))  
FROM     Grundstücke  
WHERE    Besitzer=A
```

Die Anfrage eines Anglers/Fischers, der eine bestimmte Fischart dort fangen will, wo das Wasser möglichst wenig mit Schwermetall belastet ist, könnte so aussehen:

```
SELECT   difference(sum(Vorkommen.region),sum(Verschmutzung.region))  
FROM     Vorkommen, Verschmutzung  
WHERE    Vorkommen.Art=Hecht AND Verschmutzung.Art=Schwermetall
```

Angenommen, ein Gebiet soll als neues Bauland erschlossen werden. Aufgrund gesetzlicher Bestimmungen darf nur dort gebaut werden, wo die Pflanze xy nur vereinzelt auftritt. Es muß dazu das Gebiet ermittelt werden, in dem xy häufig vorkommt. Dies kann mit der folgenden Anfrage ermittelt werden:

```
SELECT   sharp(alpha-cut(sum(Gebiet),0.3))  
FROM     Vorkommen  
WHERE    Art=...
```

Da in einer Stadt ein neues Spaßbad gebaut werden soll, muß hierfür ein geeignetes Grundstück gefunden werden. Da es sich um ein recht großen Komplex handeln soll, müssen die infragekommenden Grundstücke ermittelt werden. Aufgrund der Planung muß das Grundstück eine Mindestgröße von $30.000m^2$ besitzen. Wegen fehlender finanzieller Mittel zum Erwerb eines neuen Grundstücks, muß ein bereits der Stadt gehörendes genommen werden.

Um die zusammenhängenden Grundstücke der Stadt zu ermitteln, kann zunächst die folgende Sicht definiert werden:

```
CREATE VIEW StadtGrundst(Grundst) AS  
    SELECT   faces(Gebiet)  
    FROM     Grundstücke  
    WHERE    Besitzer=Stadt
```

Die eigentliche Anfrage kann dann wie folgt lauten:

```
SELECT   difference(Grundst)  
FROM     StadtGrundst  
WHERE    basic_area(Grundst)>10.000m2
```

Interessiert man sich für die mittlere Verschmutzung, die durch Kraftwerk X verursacht wird, hilft folgende Anfrage:

```
SELECT   mid(Gebiet)  
FROM     Verschmutzung  
WHERE    Verursacher=X AND (ART = CO OR ART = CO2)
```

Die Betriebserlaubnis für Kraftwerk Y kann nur dann aufrechterhalten werden, wenn die Verschmutzung durch Kohlenmonoxyd unter 2% liegt. Die Frage kann wie folgt geklärt werden:

```
SELECT   isempty(sum(alpha_cut(Gebiet,0.05)))  
FROM     Verschmutzung  
WHERE    Verursacher=Y
```

Es lassen sich noch viele weitere Probleme finden, die sich durch entsprechende Anfragen lösen lassen. Somit stellt die Anbindung dieses Modells an ein spezielles DBS eine interessante Aufgabe für die Zukunft dar.

Kapitel 12

Schlußfolgerungen und Ausblick

In der Arbeit wurde gezeigt, daß heutige GIS Probleme bei der Darstellung und Verarbeitung vieler realer Phänomene haben. Als Grund dafür stellte sich die Modellierung von geographischen Objekten durch ein boolesches Modell heraus. In diesem kann nur unterschieden werden, ob ein gegebener Punkt zum Objekt gehört oder nicht. Damit war eine abgestufte Zugehörigkeit nicht darzustellen. Einige der bisherigen Ansätze zur Lösung dieser Probleme waren sehr einfach, wodurch sich nur wenige davon lösen lassen. Andere Ansätze waren rein theoretischer Natur. Sie erlaubten daher nicht die Umsetzung in ein Computerprogramm.

Es wurde hier ein Modell entwickelt, welches komplexe mit Unschärfe behaftete Objekte darstellen und verarbeiten kann. Weiterhin wurde die relative Lage zweier durch dieses Modell darstellbaren Objekte untersucht. Um eine Implementierung zu ermöglichen, wurde ausgehend von diesem Modell, ein weiteres Modell erstellt, welche die endliche Darstellbarkeit von Objekten in Computern berücksichtigte. Schließlich wurden noch Datenstrukturen und Algorithmen vorgestellt, mit denen sich das Modell dann tatsächlich implementieren läßt. Es wurde zusätzlich die Laufzeit der Operatoren bei den gegebenen Datenstrukturen untersucht. Ein weiteres Kapitel dieser Arbeit beschäftigte sich mit den Visualisierungsmöglichkeiten unscharfer Objekte. Zuletzt wurde noch untersucht, wie das Modell in Datenbanken eingebunden werden könnte.

Offen ist noch die reale Einbindung dieses Modells in bestehende Computersysteme. Auch die Ausnutzung spezieller Hardware oder Betriebssysteme kann Gegenstand zukünftiger Arbeiten werden. Speziell sollen hiermit die Möglichkeiten moderner Graphikkarten angesprochen werden, 3-dimensionale Objekte darzustellen. Die Definition der Norm eines Austauschformats für unscharfe räumliche Objekte, mit welchem diese zwischen verschiedenen Anwendungen verschoben werden können, ist ebenfalls ein offenes Thema.

Literaturverzeichnis

- [1] Anthony G Cohn & Nicholas Mark Gotts. Representing Spatial Vagueness: A Mereological Approach. In Luigia Carlucci Aiello, Jon Doyle, and Stuart Shapiro, editors, *KR'96: Principles of Knowledge Representation and Reasoning*, pages 230–241. Morgan Kaufmann, San Francisco, California, 1996.
- [2] William Evans, David Kirkpatrick & Gregg Townsend. Right Triangular Irregular Networks. Technical Report EIPP-90-2, Institute for Environmental Policy and Planning, State University of New York, 1990.
- [3] David Scott Andrews. Simplifying terrain models and measuring terrain model accuracy. Technical report, Dept. of CS, UBC, 1996.
- [4] P. A. Burrough. *Natural Objects with Indeterminate Boundaries*, pages 3–28. 1996.
- [5] E. Clementini and P. Di Felice. *An Algebraic Model for Spatial Objects with Indeterminate Boundaries*, pages 155–169. Springer-Verlag, 1996.
- [6] E. Clementini and P. Di Felice. Approximate Topological Relations. *International Journal of Approximate Reasoning*, 16:173–204, 1997.
- [7] A. G. Cohn and N. M. Gotts. *The ‘Egg-Yolk’ Representation of Regions with Indeterminate Boundaries*, pages 171–191. 1996.
- [8] David Altman. Fuzzy Set Theoretic Approaches for Handling Imprecision in Spatial Analysis. *International Journal of Geographical Information Systems*, 8(3):271–289, 1994.
- [9] Detlef Nauck, Frank Klawonn & Rudolf Kruse. Fuzzy Sets, Fuzzy Controllers, and Neural Networks. *Wissenschaftliche Zeitschrift der Humboldt-Universität zu Berlin, Medizin*, 41:99–120, 1992.
- [10] Dietmar Fox. Computer Grafik, 1994.
- [11] E. Clementini and P. Di Felice and G. Califano. Composite Regions in Topological Queries. *IS*, 20(7):579–594, 1995.
- [12] M. J. Egenhofer and J. Sharma. Topological relations between regions in \mathbb{R}^2 and \mathbb{Z}^2 . In *3rd*, LNCS 692, pages 316–336. Springer-Verlag, 1993.
- [13] Max J. Egenhofer. A formal definition of binary topological relationships. In W.Litwin and H.Schek, editors, *Third International Conference on Foundations of Data Organization and Algorithms (FODO)*, volume 367 of *Lecture Notes in Computer Science*, pages 457–472. Springer-Verlag, June 1989.
- [14] M.J. Egenhofer, E. Clementini, and P. Di Felice. Topological Relations between Regions with Holes. *International Journal of Geographical Information Systems*, 8(2):128–142, 1994.

- [15] Jayant Sharma & Max J. Egenhofer Eliseo Clementini. Modeling topological spatial relations: Strategies for query processing. *Computer and Graphics*, 18(6):815–822, 1994.
- [16] Eliseo Clementini and Paolino di Felice. A Model for Representing Topological Relationships Between Complex Geometric Features in Spatial Databases. *Informations Sciences*, 90(1-4):121–136, 1996.
- [17] Eliseo Clementini, Paoline Di Felice & Peter van Oosterom. A Small Set of Formal Topological Relationships Suitable for End-User Interaction. In *SSD: Advances in Spatial Databases*. LNCS, Springer-Verlag, 1993.
- [18] M. Erwig and M. Schneider. Vague Regions. In *5th Int. Symp. on Advances in Spatial Databases*, LNCS 1262, pages 298–320. Springer-Verlag, 1997.
- [19] Eliseo Clementini & Paolino Di Felice. A comparison of methods for representing topological relationships. *Informations Sciences*, 3:149–178, 1995.
- [20] Stephan Winter & Andrew U. Frank. Topology in raster and vector representation. *Geo-Informatica*, pages 35–65, 2000.
- [21] Max J. Egenhofer & Robert D. Franzosa. On the equivalence of topological relations. In *International Journal of Geographical Information Systems*, volume 9, pages 133–152, 1995.
- [22] Helmut Sieber, Leopold Huber. *Mathematische Formeln*.
- [23] Max J. Egenhofer & John R. Herring. Categorizing binary topological relations between regions, lines, and points in geographic databases. Technical report, Department of Surveying Engineering, University of Maine, Maine, 1990.
- [24] I.N.Bronstein & K.A.Semendjajew. *Taschenbuch der Mathematik*. B.G. Teubner Verlagsgesellschaft, 1991.
- [25] Klaus-Jürgen Förster. *Analysis und Numerik*, 1994.
- [26] Leila De Floriani, Paola Magillo & Enrico Puppo. Data Structured for Simplicial Multi-complexes. volume 1651 of *Lecture Notes in Computer Science*, pages 33–51. Springer-Verlag, 1999.
- [27] Manfred Schrenk. *Informationstechnologie als Instrument und als Gegenstand der Raumplanung*.
- [28] Markus Schneider. *Spatial Data Types for Database Systems*. PhD thesis, FernUniversität Hagen, 1995.
- [29] Markus Schneider. Modeling Spatial Objects with Undetermined Boundaries Using the Realm/ROSE Approach. In *Geographic Objects with Indeterminate Boundaries*, volume 2 of *GISDATA*, pages 141–152. Taylor & Francis, 1996.
- [30] Markus Schneider. Finite Resolution Crisp and Fuzzy Spatial Objects. In *9th Int. Symp. on Spatial Data Handling*, pages 8a.3–17, 2000.
- [31] Martien Molenaar. *A Syntactic Approach for Handling the semantics of Fuzzy Spatial Objects*, 1994.
- [32] Martien Molenaar. The Extensional Uncertainty of Spatial Objects. In *SDH96*, pages 1–14, 1996.

- [33] Martien Molenaar & Tao Cheng. Fuzzy Spatial Objects and their Dynamics. In D. Fritsch, M. Englich & M. Sester, editor, *GIS Between Visions and Applications*, volume 32/4 of *International Archives of Photogrammetry and Remote Sensing*, pages 389–394, Stuttgart, 1998.
- [34] Max J. Egenhofer. A Model for Detailed Binary Topological Relationships. *Geomatica*, 47(3&4):261–273, 1993.
- [35] N.M. Gotts. An Axiomatix Approach to Topology For Spatial Information Systems. Technical report, University of Leeds, 1996.
- [36] N.M.Gotts. Using the 'RCC' Formalism To Describe The Topology Of Spherical Regions. Technical report, University of Leeds, 1996.
- [37] A.G.Cohn & N.M.Gotts. Spatial regions with undetermined boundaries. In *Proceedings of Gaithesburg Workshop on GIS, ACM*, 1994.
- [38] A.G.Cohn & N.M.Gotts. A theory of spatial regions with indeterminate boundaries. In C Eschenbach, C. Habel and B. Smith, editor, *Topological foundations of Cognitive Science*, 1994.
- [39] P.Cignoni, E.Puppo & R.Scopigno. Representation and Visualization of Terrain Surfaces at Variable Resolution. In *Intl. Symp. Scientific Visualization*. World Scientific, Cagliari, Italy.
- [40] Rolf Klein. *Algorithmische Geometrie*, 1997.
- [41] Markus Schneider. Topological predicates for complex crisp and fuzzy regions.
- [42] Markus Schneider. Uncertainty management for spatial data in databases: Fuzzy spatial data types. In *6th Int.Symp. on Advances in Spatial Databases (SSD)*, LNCS 1651, pages 330–351. Springer Verlag, 1999.
- [43] Markus Schneider. Metric operations on fuzzy spatial objects in databases. In *8th ACM Symp. on Geographic Information Systems (ACM GIS)*, pages 21–26, 2000.
- [44] Thomas Behr & Markus Schneider. Topological Relationships of Complex Points and Complex Regions, 2001.
- [45] Werner Hölbing, Werner Kuhn & Andrew U.Frank. Finite-resolution simplicial complexes. In *GeoInformatica*, volume 2(3), pages 281–298, 1998.
- [46] Wilfried Wolf & Gerd Peyke. Genauere Aussagen in der Geographie durch Betrachtung der Unschärfe.