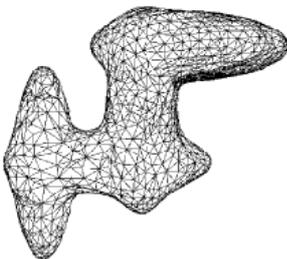


GDV-Seminar 1997

Fusion of 3D-Shapes

1997 von T. Strasser

Basierend auf



Fusion of 3D-Shapes

P. Decaudin, A. Gagalowicz, 1994 [1]



Volumetric Shape Description of Range Data using "Blobby Model"

S. Muraki, 1991 [2]

1 Inhalt

1 EINLEITUNG	2
2 VERSCHMELZUNG DURCHDRINGENDER OBJEKTE.....	3
2.1 IDEE	3
2.2 VERBINDUNGSREGELN.....	3
2.3 VERBINDEN VON OBJEKTEN.....	4
2.4 2D.....	4
2.5 3D.....	5
2.6 PROBLEME	6
2.6.1 Scharfe Kanten.....	7
2.6.2 Nicht sternförmige Objekte	7
2.7 ANWENDUNGEN	7
2.7.1 Erstellungstool.....	7
2.7.2 Animiertes Deformationstool	7
2.7.3 Morphingtool	8
3 VOLUMETRISCHE OBJEKTDESCHEIBUNG	8
3.1 IDEE	9
3.2 GRUNDLAGEN: "BLOBBY MODEL"	9
3.2.1 2D Skalarfelder.....	9
3.2.2 3D Skalarfelder.....	10
3.3 REKONSTRUKTION VON GESCANNTEN OBERFLÄCHEN.....	10
3.4 PROBLEM.....	11
3.5 ALGORITHMUS	12
4 GEGENÜBERSTELLUNG / AUSBLICKE.....	13
5 LITERATUR.....	13

1 Einleitung

Ein wichtiges Gebiet in der Computer Grafik ist die Darstellung und Animation von organischen Objekten wie menschliche Körper, Gesichter etc.. Eine einfache Weise solche Objekte zu erstellen ist die geeignete Verschmelzung einzelner einfachen Objekten zu einer glatten Oberfläche.

Auch im Bereich der Datenvisualisierung ist es nützlich, zusammengehörige Datenobjekte als Clusters durch eine Verschmelzung einzelner Objekte zu realisieren. Es gibt natürlich viele Arten solche Verschmelzungen zu erstellen.

Im wesentlichen gibt es zwei grundlegende Möglichkeiten eine Oberfläche zu beschreiben bzw. zu modellieren: Als Isofläche eines Skalarfeldes zum Beispiel, oder als Netzwerk von Ecken, Kanten und Flächen z.B. mit B-Spline Patches oder Nurbs. Im folgenden werden vereinfachte Verschmelzungsmethoden zu den oben genannten Oberflächenrepräsentationen betrachtet.

2 Verschmelzung durchdringender Objekte

2.1 Idee

Die Idee dieser Methode ist es, zwei (fast) beliebige Objekte zu einem neuen zu verschmelzen, so dass die neue Oberfläche möglichst glatt und visuell "richtig" ist.

Die Objekte, bestehend aus einer Knoten/Kanten/Flächen Vermaschung, werden Polygonal bzw. Polyhedral angenommen. Für die hier vorgestellte Methode müssen sich die Objekte durchdringen. Das neu erzeugte Objekt wird das selbe Volumen wie die Summe der Originale haben. Die Volumenerhaltung wird als Randbedingung für die Konstruktion benötigt.

2.2 Verbindungsregeln

Seien A und B zwei 2D oder 3D Objekte polygonaler bzw. polyhedraler Form. Wir wollen nun ein neues Objekt C von gleichem Charakter erstellen, so dass

- A und B vollständig in C liegen (d.h. jeder Punkt von A und von B liegt auch in C)
- die Fläche (bzw. Volumen) von C gleich der Summe der Flächen (bzw. Volumina) von A und B ist.
- die Oberfläche von C so glatt wie möglich ist (d.h. Wenn die Oberflächen von A und B C^n -Stetig sind, ist auch die Oberfläche von C C^n -Stetig)

Abbildung 1 zeigt ein Beispiel der Verschmelzung von zwei Kreisen.

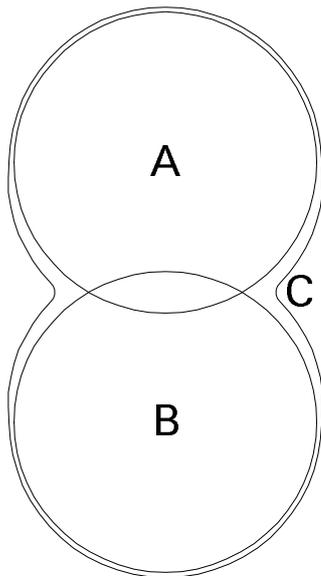


Abbildung 1: Verschmelzung zweier Kreise

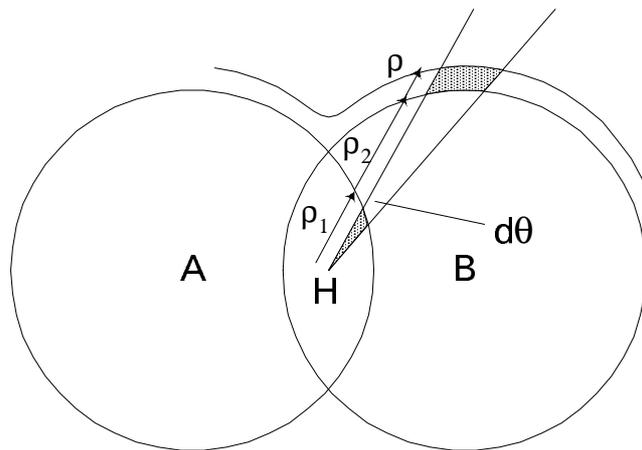


Abbildung 2: Notation (2D-Shapes)

2.3 Verbinden von Objekten

Das Verbinden von Objekten ist relativ einfach. Zur Erläuterung beschränken wir uns zunächst auf Objekte, welche sternförmig¹ zu einem Punkt H sind. Gegeben seien zwei Objekte A und B, so dass H in der Schnittfläche (bzw. -volumen) der Objekte liegt.

Nehmen wir an, die Objekte bestehen aus homogenen Material. Beim Durchdringen der Objekte entsteht in der Schnittfläche (bzw. -volumen) ein Überschuss an Material. Die Idee ist nun, dieses überschüssige Material auf den Rand zu transportieren, entlang der radialen Projektion von H aus.

2.4 2D

Widmen wir uns zunächst dem einfachsten 2-Dimensionalen Fall. Die Formen der Objekte A und B können mit Funktionen $\rho_1(\theta)$ und $\rho_2(\theta)$ in Polarkoordinaten mit Zentrum H dargestellt werden. Seien $\Delta v_1(\theta)$ und $\Delta v_2(\theta)$ die Flächeninhalte der Sektoren $(\rho_1(\theta), d\theta)$ und $(\rho_2(\theta), d\theta)$.

Gesucht ist nun eine Funktion $\rho(\theta)$ für die Form des Objektes C mit der Fläche $\Delta v(\theta)$ des Sektors $(\rho(\theta), d\theta)$.

$$\begin{aligned} \Delta v_1(\theta) &= \frac{1}{2} \rho_1(\theta)^2 d\theta \\ \Delta v_2(\theta) &= \frac{1}{2} \rho_2(\theta)^2 d\theta \\ \Delta v(\theta) &= \frac{1}{2} \rho(\theta)^2 d\theta \end{aligned} \tag{1}$$

¹ Ein Objekt A heisst **sternförmig** im Bezug zu einem Punkt H, wenn für alle Punkte M in A gilt: Strecke \overline{HM} liegt in A.

Um der zweiten Verbindungsregel (Flächenerhaltung) zu genügen muss gelten:

$$\Delta v(\theta) = \Delta v_1(\theta) + \Delta v_2(\theta) \quad (2)$$

bzw.

$$\frac{1}{2}\rho(\theta)^2 d\theta = \frac{1}{2}\rho_1(\theta)d\theta + \frac{1}{2}\rho_2(\theta)d\theta \quad (3)$$

nach Auflösung nach $\rho(\theta)$ erhalten wir

$$\rho(\theta) = \sqrt{\rho_1(\theta)^2 + \rho_2(\theta)^2} \quad (4)$$

Mit (4) kann die Form von C einfach berechnet und gezeichnet werden. Das Objekt C erfüllt die Verbindungsregeln (siehe 2.2):

- C beinhaltet A und B (für alle θ gilt: $\rho(\theta) \geq \rho_1(\theta)$ und $\rho(\theta) \geq \rho_2(\theta)$)
- C ist derart konstruiert, dass die Fläche von C gleich der Summe der Flächen von A und B ist.
- Die Oberfläche von C ist so glatt wie die Oberflächen von A und B (wenn ρ_1 und ρ_2 in C^n sind, dann ist auch ρ in C^n)

2.5 3D

Die Erweiterung nach 3D erfolgt analog. Als Polarfunktion der Oberfläche von C können wir schreiben:

$$\rho(\theta, \varphi) = \sqrt[3]{\rho_1(\theta, \varphi)^3 + \rho_2(\theta, \varphi)^3} \quad (5)$$

Nun benötigen wir einen guten Algorithmus zum erstellen der polyhedralen Form des Objektes C . Zur Einfachheit nehmen wir an, dass A und B keine scharfe Kanten haben. Weiterhin werden die Objekte durch eine Knoten/Kanten/Fläche Vermaschung repräsentiert.

Als Grundlage des neuen Objektes C gehen wir von einem Tetraheder aus, welches in H zentriert wird. Die Oberfläche des Tetraheders wird nun auf die Oberfläche von C projiziert, und wenn nötig eine Verfeinerung der Tetrahederseiten durchgeführt um C möglichst gut zu approximieren (stepwise refinement). Nebst der Projektionsformel wollen die Details des Algorithmus hier nicht beschrieben werden (für genauere Informationen siehe [1], p. 4).

Jeder Vertex M des Tetraheders (oder einer Verfeinerung davon) wird auf den Vertex M' von C projiziert gemäss:

$$M' = H + \left(\sqrt[3]{\|\vec{HI}\|^3 + \|\vec{HJ}\|^3} \right) \frac{\vec{HM}}{\|\vec{HM}\|} \quad (6)$$

$$\text{nach (5) ist } \sqrt[3]{\|\vec{HI}\|^3 + \|\vec{HJ}\|^3} = \rho(\theta, \varphi)$$

wobei I bzw. J der Schnittpunkt der Halbgerade HM mit der Oberfläche von A bzw. B ist.

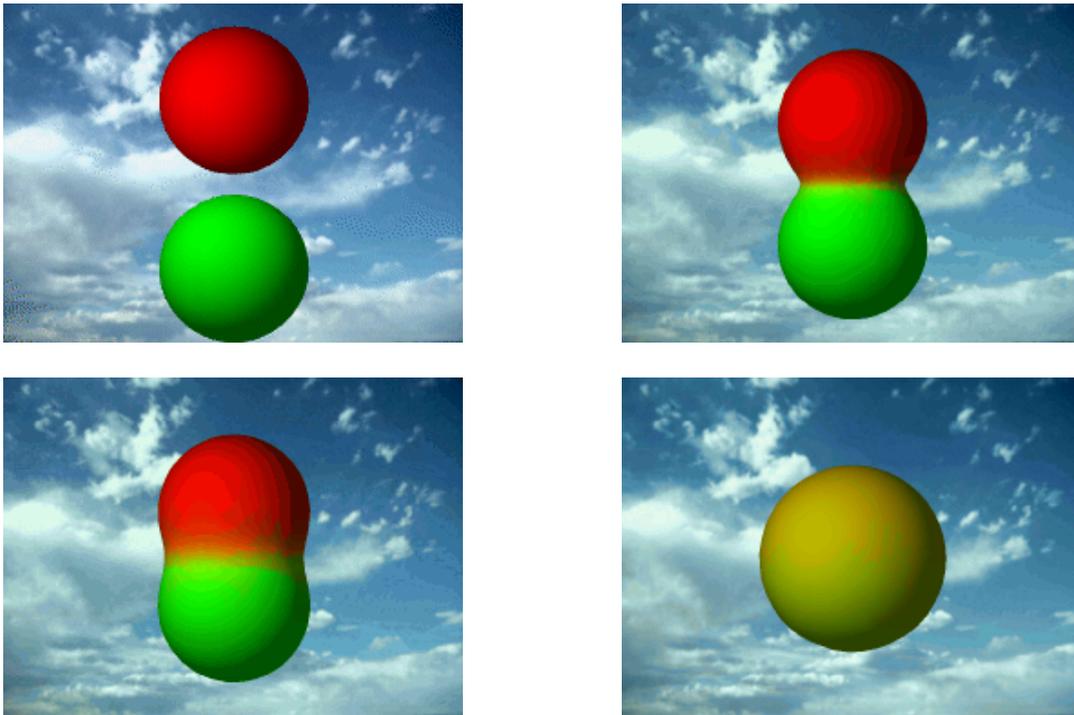


Abbildung 3: Verschmelzung von farbigen Kugeln

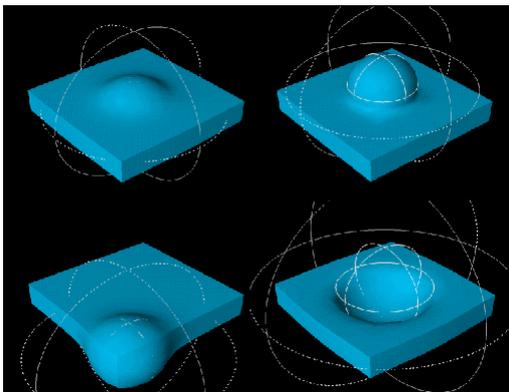


Abbildung 4: Verschmelzung von Kugel und Quader

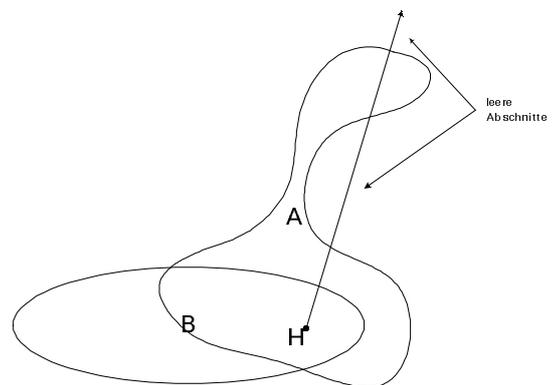


Abbildung 5: Nicht sternförmige Objekte

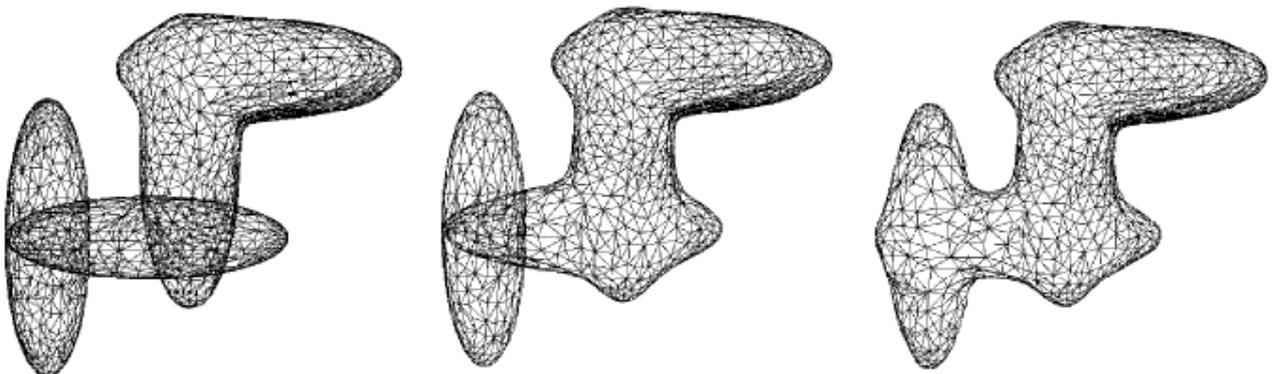


Abbildung 6: Verschmelzung nicht sternförmiger Objekte

2.6 Probleme

2.6.1 Scharfe Kanten

Falls die Objekte A und B scharfe Kanten haben, kann der obige Algorithmus nicht so angewandt werden, da C als glatt angenommen wird. Um trotzdem scharfe Kanten zu zulassen betrachten wir die Oberfläche von C als Menge von glatten Patches, getrennt durch scharfe Kanten. Der Algorithmus von vorher wird nun auf jedes Patch und von A und B angewandt. Zusätzlich werden die scharfe Kanten von A und B auch auf C projiziert und alle einzelnen Netzwerke miteinander verbunden.

Am Anfang des Algorithmus wird anstatt des Tetraheders eine Kugel verwendet. Die Vermaschungen von A und B werden mitsamt den scharfen Kanten zuerst auf die Kugel projiziert und anschliessend miteinander verbunden. Danach wird die "neue" Kugel gemäss (5) auf C projiziert. Als Beispiel siehe (Abbildung 4).

2.6.2 Nicht sternförmige Objekte

Wenn wir nicht sternförmige Objekte verschmelzen wollen treten vor allem zwei Probleme auf. Als erstes wissen wir nicht, wie wir den Überschuss an Material auf dem Rand von C verteilen wollen, als zweites müssen wir einen neuen Weg suchen, das Model von C zu erstellen. Die hier vorgestellte Lösung gilt nur, falls die Objekte A und B nur ein einziges zusammenhängendes Schnittvolumen haben.

Wie vorhin sei H ein Punkt in $A \cap B$. A muss nicht notwendigerweise sternförmig sein, aber B . Ein Projektionsstrahl von H aus schneidet B nur einmal, kann dafür A mehrmals schneiden (siehe Abbildung 5). Die Schnittgerade besteht demzufolge aus Abschnitten in A , B , $A \cap B$, und "leeren" Abschnitten $\overline{A \cap B}$.

Die Idee ist nun, den Überschuss von Material in den ersten, leeren Abschnitt zu transportieren. Falls dieser Abschnitt nicht gross genug ist, wird der Rest des Materials zum nächsten leeren Abschnitt verschoben usw.

Um ein Model von C zu erstellen, wird zunächst eine Kopie von A erstellt und solange deformiert, bis C genügend gut approximiert ist. Für eine detailliertere Beschreibung des Algorithmus siehe [1, p. 9].

2.7 Anwendungen

Es gibt nun verschiedene Möglichkeiten diese neue Technik anzuwenden. Hier werden deren drei kurz vorgestellt.

2.7.1 Erstellungstool

Diese Anwendung ist eine direkte Weiterführung der oben beschriebenen Methode zur Verschmelzung von Objekten.

Der Benutzer kann zwei Objekte (A und B) und einen Punkt (H) auswählen. Das Tool prüft ob die Objekte und der Punkt gültig sind, und verschmilzt sie wenn es möglich ist. Das neu erstellte Objekte unterscheidet sich nicht von den anderen, und kann insbesondere wieder für neue Verschmelzungen verwendet werden.

2.7.2 Animiertes Deformationstool

Auf eine einfach Weise können mit dieser Technik auch Deformationen bewerkstelligt werden, indem der Benutzer Material zu Objekten hinzufügen kann. Durch eine leichte Modifikation der Verschmelzungsregeln kann sogar Material weggenommen werden. Die Bedingung der Volumenerhaltung (2) $\Delta v(\theta) = \Delta v_1(\theta) + \Delta v_2(\theta)$ (siehe 2.4) kann so abgeändert

werden, dass Material abgezogen wird: $\Delta v(\theta) = \Delta v_1(\theta) - \Delta v_2(\theta)$. Damit können bemerkenswerte Effekte erzielt werden (Abbildung 4, letztes Bild).

Durch einfache Transformationen (Skalierung, Schärung) und Translation der Grundobjekte während der Zeit, kann die Deformation animiert werden, indem bei jedem Frame die Verschmelzungen neu berechnet werden. Auch kann das Zentrum der Projektion H variiert werden, um weitere Effekte zu erhalten (Abbildung 7).

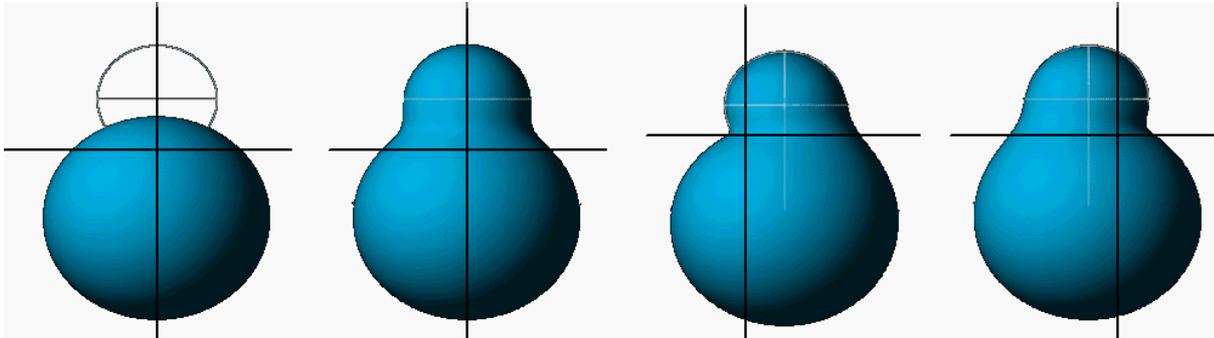


Abbildung 7: Zentrum der Projektion

2.7.3 Morphingtool

Mit dieser Methode können auch auf eine einfache Art Morphs zwischen zwei Objekten erstellt werden. Unser Ziel ist es ein Zeitabhängiges Objekt $C(t)$ zu generieren, welches sich in einem Intervall $t \in [0,1]$ kontinuierlich von einem Objekt A ($C(0) = A$) zu einem Objekt B ($C(1) = B$) verwandelt.

Dies kann einfach erreicht werden, indem wir die Oberfläche von C linear zwischen der Oberfläche von A und B interpolieren:

$$\rho(\theta, \varphi) = \sqrt[3]{(1-t)\rho_1(\theta, \varphi)^3 + t\rho_2(\theta, \varphi)^3} \tag{8}$$

Als Beispiel hierzu siehe Abbildung 8. Hier wurde ein Würfel in ein Kreuz gemorphet (das Kreuz wurde erstellt, indem zwei Ellipsoide verschmolzen wurden).

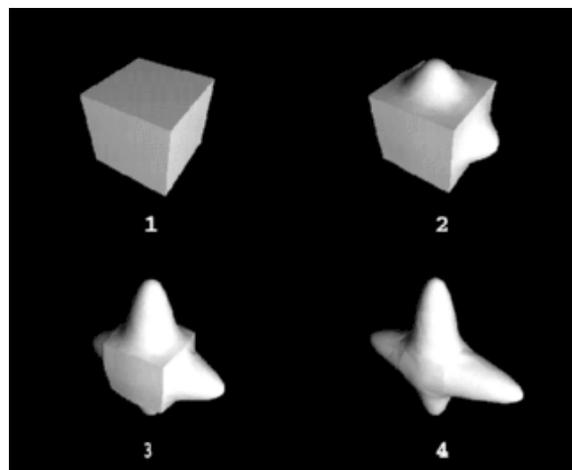


Abbildung 8: Morphing zwischen Würfel und Kreuz

3 Volumetrische Objektbeschreibung

3.1 Idee

Ein ganz anderer Ansatz zur Beschreibung von 3D-Objekten basiert auf Isoflächen von Skalarfeldern oder Volumendaten. Letztere sind eigentlich eine Diskretisierung ersterer welche z.B. von einem CT-Scanner erstellt werden. Dabei wird ein Volumen mit skalaren Werten, wie z.B. Intensitäten oder Attenuationskoeffizienten, gefüllt. Das Objekt wird dann beschrieben durch eine Isofläche gleicher Werte im Volumen, welche dann z.B. mit Marching Cubes oder Marching Tetrahedron trianguliert werden kann.

3.2 Grundlagen: "Blobby Model"

Um Skalarfelder erstellen zu können, verwenden wir einzelne Primitiven, wie Punktquellen oder Superquadrics, als "Strahler". Diese erzeugen je ein Strahlungsfeld welche superponiert werden können um ein resultierendes Feld zu erhalten. Zur Veranschaulichung wollen wir zunächst zweidimensionale Skalarfelder betrachten.

3.2.1 2D Skalarfelder

Gegeben sei nun eine Primitive P_i an einem Punkt (x_i, y_i) welche ein Feld V_i an jedem Punkt (x, y) in \mathfrak{R}^2 erzeugt:

$$V_i(x, y) = b_i e^{-a_i f_i(x, y)} \quad (9)$$

Die Funktion $f_i(x, y)$ beschreibt die Form des Skalarfeldes. Bei einem radial symmetrischen Feld definiert der quadratische Abstand zwischen (x_i, y_i) und (x, y) die Intensität:

$$f_i(x, y) = (x - x_i)^2 + (y - y_i)^2 \quad (10)$$

Der Parameter a_i von (9) beeinflusst das Abklingen des Feldes gegen weiter entfernte Punkte und b_i bestimmt die Stärke des Feldes. Befinden sich mehrere Primitiven in dem Medium, werden diese superponiert. Ausgehend von (9) kann das resultierende Feld V aus N Primitiven wie folgt beschrieben werden:

$$V(x, y) = \sum_{i=1}^N b_i e^{-a_i f_i(x, y)} \quad (11)$$

Abbildung 9 zeigt die Visualisierung eines Feldes generiert durch zwei radial symmetrischen Primitiven. Um nun ein Objekt aus dem Skalarfeld zu erstellen, schneiden wir den Raum $(x, y, V(x, y))$ mit einer Ebene (x, y, T) . Wir erhalten die Höhenlinie (Abbildung 10) durch

$$V(x, y) = T \quad (12)$$

Durch Variation von T kann die "Blobbyness" bestimmt werden. Wird z.B. T eher hoch gewählt, zerfällt das Objekt. Tiefe Werte von T produzieren klumpigere Objekte.

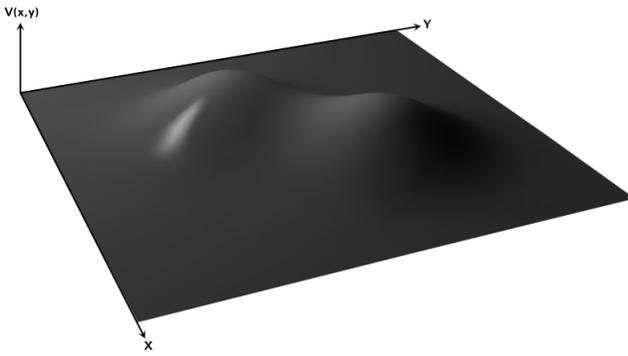


Abbildung 9: Skalarfeld von 2 kreisförmigen Primitiven

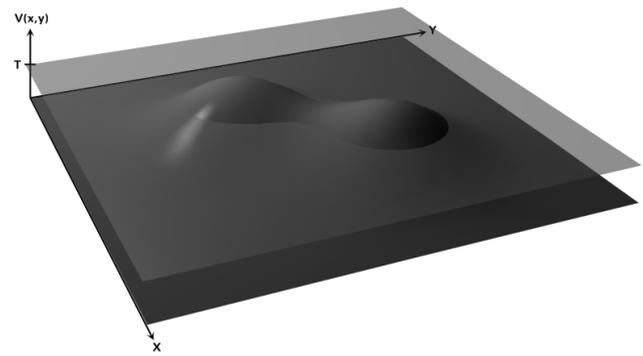


Abbildung 10: Schnitt mit Ebene

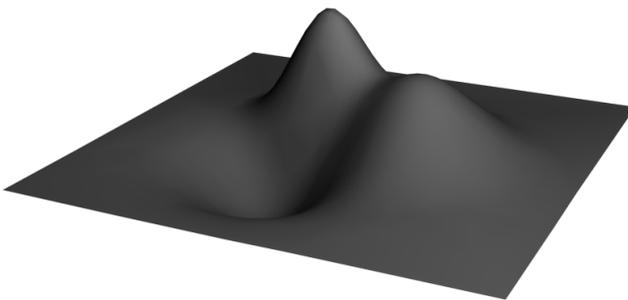
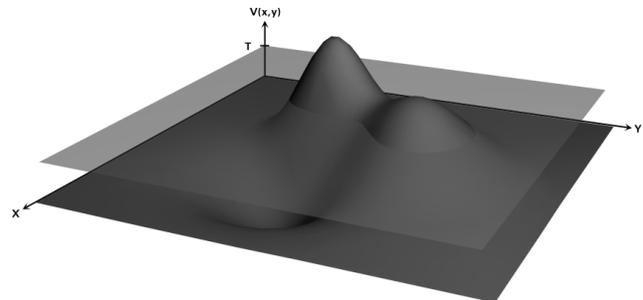


Abbildung 11: Negative Felder



3.2.2 3D Skalarfelder

Die Erweiterung nach 3D geht analog, indem wir ein Feld in \mathfrak{R}^3 definieren. Aus (9) wird

$$V_i(x_i, y_i, z_i) = b_i e^{-a_i f_i(x, y, z)} \tag{13}$$

bzw.

$$V(x, y, z) = \sum_{i=1}^N b_i e^{-a_i f_i(x, y, z)} \tag{14}$$

und

$$V(x, y, z) = T \tag{15}$$

Um das Objekt, gegeben durch eine Isofläche gemäss (15), zu bestimmen, muss jetzt der Raum mit einer Hyperebene geschnitten werden. Leider können wir das hier nicht so richtig visualisieren.

Spezielle Effekte können erzielt werden, wenn der Parameter b_i negativ gewählt wird. Dadurch erzeugt die Primitive P_i ein negatives Feld. In Abbildung 11 hat die vordere Primitive ein negatives b_i und erzeugt so ein Tal im 2D-Skalarfeld.

3.3 Rekonstruktion von gescannten Oberflächen

Das "Blobby Modell" bietet einen guten Ansatz zur Rekonstruktion von Oberflächen. Reelle Gegenstände können z.B. mit Range Scanner abgetastet werden. Dabei wird ein sog. 2½-Model des Objekts erstellt, indem von einem fixen Punkt aus die Abstände zur Oberfläche gemessen werden. Was fehlt, ist die volle 3D Information des Objektes.

3.4 Problem

Zur Rekonstruktion des Objektes muss nun die Oberfläche möglichst gut mit einer Isofläche eines Skalarfeldes approximiert werden. Im wesentlichen muss die Funktion

$$E_{value} = \sum_{j=1}^M \left(V(x_j, y_j, z_j) - T \right)^2 \quad (16)$$

minimiert werden. Wobei M Punkte $P_j(x_j, y_j, z_j)$ der gemessenen Oberfläche gegeben sind. Es geht also darum eine passende Menge von Primitiven zu finden, welche (16) minimiert. Ein weiteres Problem entsteht, da wir nicht wissen, welche Seite der Isofläche wir approximieren sollen. Dazu müssen wir die Flächennormalen der Isofläche und der Range Daten betrachten. Die Normalen \mathbf{n}_j in jedem Punkt P_j der Oberfläche des Objekts können wir mittels den umliegenden Punkte approximieren. Für die Normalen $\mathbf{N}(x, y, z)$ auf die Isofläche verwenden wir die negative Richtung des Gradienten

$$\mathbf{N}(x, y, z) = -\nabla V(x, y, z) \quad (17)$$

wobei der Operator ∇ definiert ist als

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right) \quad (18)$$

Die beste Konfiguration der Primitiven muss also nicht nur (16) minimieren sondern auch noch

$$E_{normal} = \sum_{j=1}^M \left| \mathbf{n}_j - \frac{\mathbf{N}(x_j, y_j, z_j)}{\|\mathbf{N}(x_j, y_j, z_j)\|} \right|^2 \quad (19)$$

Man kann nun zeigen, dass jede Primitive mit $a_i = 0$ und $b_i = T$, welche unendlich weit weg von der Oberfläche ist, beide Bedingungen (16) und (19) erfüllt. Zudem wirken die Bedingungen nur in Bereichen wo Range Datenpunkte existieren. Es kann sein, dass eine Primitive im Bereich nahe an Datenpunkte die Oberfläche gut approximiert, aber weit weg vom Einfluss der Datenpunkte "komische" Artefakte generiert. Darum müssen die Primitiven möglichst kompakt gehalten werden. Aus (10) und (13) kann man zeigen dass

$$\iiint V_i(x_i, y_i, z_i) dx dy dz = \left(\frac{\pi}{a_i} \right)^2 b_i \quad (20)$$

(20) repräsentiert so etwas wie die Masse der Primitive P_i . Nun können wir noch eine weitere Bedingung einführen:

$$E_{shrink} = \left(\sum_{i=1}^N a_i^{-\frac{3}{2}} |b_i| \right)^2 \quad (21)$$

Die Minimierung dieser Bedingung bewirkt eine Minimierung der "Grösse" der Primitive. Die gesuchte Konfiguration von Primitiven kann nun erreicht werden, wenn folgende Funktion minimiert werden kann:

$$E = \frac{1}{M} (E_{value} + \alpha E_{normal}) + \beta E_{shrink} \quad (22)$$

wobei die Gewichte α und β je nach Daten gewählt werden können, so dass eine optimale Approximation gefunden werden kann.

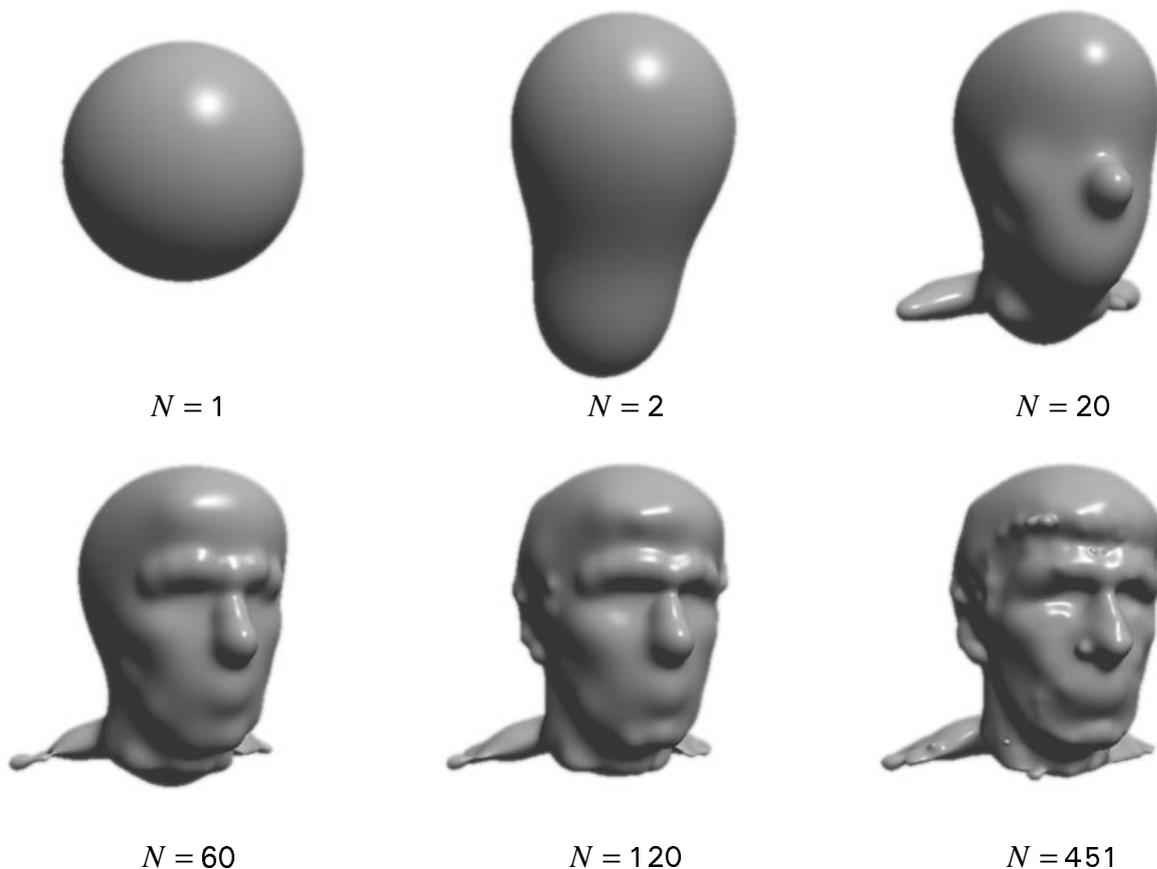


Abbildung 12: Rekonstruktion eines Gesichtes mit Verschiedener Anzahl Primitiven

3.5 Algorithmus

Bei einer Menge von N Primitiven mit jeweils 5 Parametern müssen insgesamt $5N$ Unbekannte gefunden werden. Da (22) nicht linear ist, kann keine analytische Lösung gefunden werden. Für grosse N sind Numerische Methoden sehr aufwendig, da alle $5N$ Unbekannten miteinander gefunden werden müssen. Der hier vorgestellte Algorithmus geht iterativ vor, indem die Oberfläche der Objektes sukzessive durch mehr Primitive approximiert wird. Ausgehend von einer initialen Primitive wird in jedem Schritt eine Primitive ausgewählt und in zwei neue aufgeteilt. Für die neuen Primitiven werden die Parameter gemäss (22) bestimmt. Der Algorithmus wird solange fortgesetzt bis die gewünschte Anzahl Primitiven erreicht worden ist. Für eine genauere Beschreibung siehe [2]. Abbildung 12 zeigt ein Beispiel einer Rekonstruktion eines Gesichtes.

4 Gegenüberstellung / Ausblicke

Beide hier vorgestellten Methoden bieten sehr viele Möglichkeiten zur Modellierung, Deformation, Animation etc. von dreidimensionalen Objekten. Insbesondere sind sie sehr geeignet um organische Gegenstände darzustellen.

Im Gegensatz zum "Blobby Model" wird wohl eher die erste Technik in gängigen 3D-Systemen Anwendung finden. Viele solcher Systeme bieten offene Schnittstellen für Plugins an, und können mit den entsprechenden Tools ergänzt werden. Ein Nachteil dieser Methode ist die Einschränkung, dass die Objekte sich durchdringen müssen. Hier besteht sicher die Möglichkeit die Algorithmen derart weiter zu entwickeln, dass sie mit solchen Objekten umgehen können.

Das "Blobby Model" ist ein mächtiges Werkzeug welches vielseitig eingesetzt werden kann. In der Datenvisualisierung können Objekte gleicher Kategorie als Blobb dargestellt werden. Durch Variation des Isoflächenparameters T könnte z.B. die Stärke der Verschmelzung gewählt werden. Auch zur Approximation von Range- oder Volumendaten ist diese Methode sicher ein guter Ansatz. Falls die Oberfläche des Skalarfeldes als Mesh benötigt wird, stellt die Skalarfeldrepräsentation ein Nachteil dar, da die Isofläche zuerst Triangulation werden muss. Ein Vorteil gegenüber normalen Volumendaten besteht trotzdem, weil die Oberflächennormalen analytisch bestimmt werden können.

5 Literatur

- [1] Fusion of 3D-Shapes
P. Decaudin, A. Gagalowicz, Eurographics Workshop on Animation and simulation. Oslo, Norway, sept. 1994

- [2] Volumetric Shape Description of Range Data using "Blobby Model"
S. Muraki, ACM Computer Graphics, Vol. 25, Nr. 4, July 1991, p. 227