Automatische Rekonstruktion von B-Spline Flächen von beliebigem topologischen Typ

von

Matthias Eck und Hugues Hoppe

Einführung

Auch wenn es heute durch Modellierungs-Software möglich ist sehr detaillierte Flächenmodelle zu erzeugen, gibt es immer noch Schwierigkeiten bei organischen Formen wie beispielsweise menschliche Gesichter und bei Freiformflächen.
Die Benutzung von Laser-Range-Scanning zur Erstellung von Freiformflächen aufgrund realer physischer dreidimensionaler Objekte bietet eine vielversprechende Alternative zur reinen softwaremässigen Konstruktion von solchen Flächen. Viele Modelle der Computergrafik werden zuerst aus Ton oder Holz gefertigt und anschliessend in digitale Form gescannt.
Laser-Range-Scanner produzieren eine grosse Ansammlung von Punkten auf Objektoberflächen. Die Umwandlung dieser Datenpunkte in ein brauchbares geometrisches Modell ist bekannt unter dem Begriff der Flächenrekonstruktion. Es gibt viele Arbeiten, die Flächen von einfacher Topologie rekonstruieren. Andere Arbeiten, die das Problem für beliebige Flächen lösen sind oft überladen, da es sehr viele Patches braucht, um eine solche Fläche zu repräsentieren. Wünschenswert wäre daher eine glatte Flächenbeschreibung. Die gebräuchlichste Formbeschreibung von glatten Oberflächen in gängiger Modellierungssoftware sind B-Spline-Flächen, ihre allgemeinste Form wird durch die NURBS (non-uniform rational B-splines) repräsentiert.
In diesem Paper wird nun zum ersten Mal ein Verfahren dargestellt, das ausgehend von einer Menge von Oberflächenpunkten einer Fläche mit beliebiger Topologie, vollautomatisch eine B-Spline-Flächenrepräsentation dieser Fläche berechnet. Die B-Splinefläche wird anschliessend adaptiv soweit verfeinert, dass sie sich überall einem Fehlermass entsprechend genau an die Ausgangspunkte anschmiegt.

Übersicht

Das Paper baut im wesentlichen auf drei vorangegangene Arbeiten auf. Ausgegangen wird erstens von den früheren Rekonstruktionsarbeiten von Hugues Hoppe, die in den Paper «Surface Reconstruction from Unorganized Points» und «Mesh Optimization» zusammengefasst sind. Nach eher kleineren Abänderungen erhalten wir daraus für die ausgehende Punktmenge ein dichtes Dreiecksnetz als Aproximationsfläche, die uns als Anfangsparametrisierung dient (Schritt 1). Das wichtigste bei diesem Schritt ist, die Topologie des Objektes richtig zu erfassen. Als zweites wird von der Arbeit «Multiresolution Analysis of Arbitrary Meshes» von Matthias Eck Gebrauch gemacht, um eine Parametrisierung zu finden, die unter Beibehaltung der Topologie nur noch wenige Grundflächen hat, und sich so als Ausgangslage zur Erstellung von B-Spline-Flächen besser eignet (Schritt 2). Als dritte Arbeit wurde das Paper «Constructing C1 surfaces of arbitrary topology using biquadratic and bicubic splines» von J. Peters zu Hilfe genommen, um in Schritt 4, über den Umweg von Bézier-Flächen über viereckigen Basisflächen B-Spline-Flächen zu konstruieren. Die Leistung der hier vorgestellten Arbeit ist die Kombination dieser Arbeiten (zum Teil in leicht abgeänderter Form) mit neu entworfenen Algorithmen und die Zusammenfassung zu einem ganzen Verfahren, das eben vollautomatisch aus unorganisierten Punkten eine C1-stetige B-Spline-Flächendarstellung generiert. Was nun ganz neu dazugekommen ist, sind einerseits der Algorithmus zur Umwandlung des dreiecksbasierten Basiskomplex in einen Viereckbasierten und zweitens das Verfahren zur Anpassungsfähigen Verfeinerung.

1. Konstruktion einer Anfangsparametrisierung über eine grobe Aproximationsfläche

In diesem Schritt wird ausgegehend von einer unorganisierten Menge von Punkten X_i , welche die Oberfläche eines Objekts beschreiben, ein dichtes Dreiecksnetz als Aproximationsfläche konstruiert, welches wir dann als Anfangsparametrisierung verwenden. Die Schwierigkeit hierbei ist die Topologie, des durch die Punktmenge repräsentierten Objekts, richtig zu erfassen.

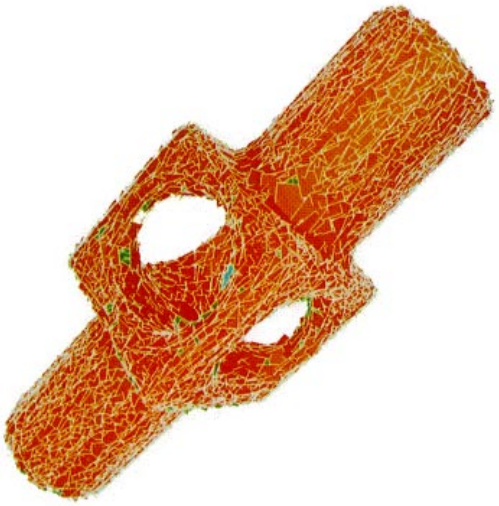
Erzeugung eines Dreiecksgitter

A: Für jeden Punkt X_i wird aufgrund der nächstgelegenen K Nachbarpunkten eine orientierte Tangentenfläche so berechnet, dass die Fläche durch X_i geht und die Summe der quadrierten Abstände der Nachbarpunkte zu dieser Fläche minimal ist. Wir bekommen eine stückweise lineare Funktion f: R^3 -> R, die uns eine Approximation für den Abstand zur Oberfläche des gesuchten Objektes gibt, wenn wir für jeden Punkt in R^3 den Abstand zur nächstgelegenen Tangentenfläche zurückgeben. Es handelt sich bei diesen Abständen um vorzeichenbehaftete Werte, die angeben ob man sich aussserhalb oder innerhalb des Objektes befindet. Die Orientierung der Fläche wird dadurch erreicht, dass für jede Tangentenfläche auch ein Vektor berechnet wird, der nach aussen zeigt. Die Schwierigkeit dabei ist, die Richtung dieser Orientierungsvektoren zwischen den einzelnen Tangentenflächen richtig zu propagieren. (Gelöst wird dieses Problem über die Graphenoptimierung eines Graphen mit einem Knoten pro Tangentenfläche, einer Kante für jedes genügend nahe beieinanderliegende Paar von Punkten und einer Kostenfunktion, die angibt wie konsistent die Orientierungen der Tangentenflächen sind.)

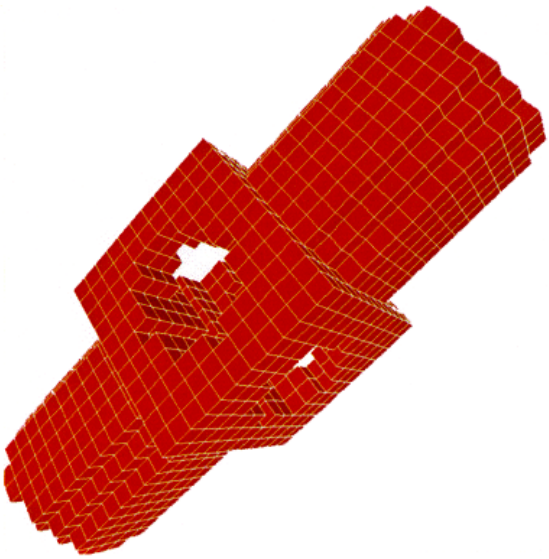
B: Als nächstes wird eine Würfelgrösse gewählt und und der Raum in dem die Punkte X_i liegen diskretisiert. Alle Würfel die Punkte X_i enthalten werden gespeichert um anschliessend von dem Konturalgorithmus «marching cubes» bearbeitet zu werden.

C: Der «marching cubes»-Algorithmus geht nun alle diese Würfel durch und schneidet sie mit der Oberfläche des Objetes. Die Schnittpunkte mit der Oberfläche liegen auf den Würfelkanten an den Stellen, wo die weiter oben berechnete Abstandsfunktion f gerade Null ist. Die so entstehenden Schnittflächen in den Würfeln werden durch vordefinierte Triangulationen dargestellt. Als letzter Schritt müssen nur noch die Kanten der Dreieckspatches zusammengefasst werden.

A



B



C

Optimierung des Dreiecksgitter

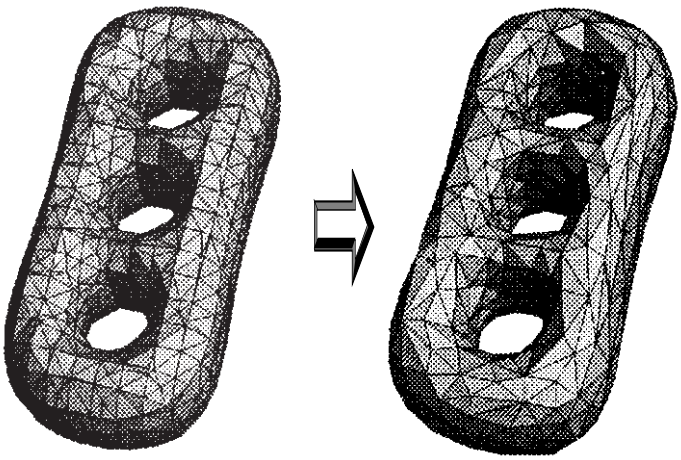
Wir haben nun das Dreiecksgitter M = (K, V). K entspricht der Konnektivität der Punkte, Kanten und Flächen und beschreibt somit die Topologie des Gitters. V ist die Menge der Knotenpunkte des Gitters.
Das erzeugte Gitter M ist wegen der Diskretisierung, der durch den «marching cubes»-Algorithmus eingeführten Würfel, noch nicht die optimale Fläche für eine Parametrisierung. Durch Verschieben der Position der Punkte V lässt sich ein Dreiecksgitter bestimmen, dessen Oberfläche näher an der durch die Punkte X_i beschriebenen Ausgangsfläche liegt. Ein in diesem Sinne optimales Dreiecksgitter lässt sich durch Minimierung der Energiefunktion E(V, K) berechnen.

$$E(V,K) = E_{dist}(K,V) + E_{spring}(K,V)$$

$$E_{dist}(K,V) = d^2(x_i, \varphi_V(K))$$

$$E_{spring}(K,V) = \kappa \, ||v_j - v_k||^2$$

$E_{dist}(K, V)$ ist die Summe des quadrierten Abstandes aller Punkte X_i zu ihrer Projektion $\varphi_V(K)$ auf das nächstgelegene Dreieckspatch. $E_{spring}(K, V)$ ist die Summe von Federkräften zwischen zwei benachbarten Gitterpunkten v_j und v_k mit der Federkonstante κ. Mit Hilfe der Federkräfte wird erreicht, dass das Dreiecksgittter lokal nicht allzustark auseinandergezogen werden kann.

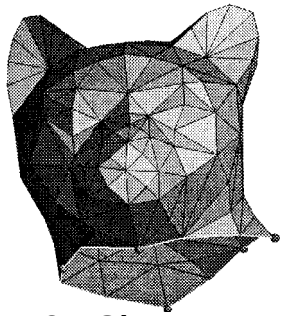


2. Reparametrisierung über einer einfachen dreiecksbasierten Fläche

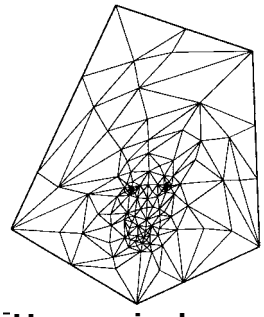
Der zweite Schritt vereinfacht die Approximationsfläche unter Beibehaltung der Topologie soweit, dass wir nur noch wenige dreieckige Basisflächen haben. Unser Ziel ist es ja, mehrere Dreiecksflächen durch ein B-Spline Patch zu ersetzen. B-Spline Patches brauchen jedoch viereckige Grundflächen. Wir werden in einem späteren Schritt jeweils zwei benachbarte dreieckige Grundflächen zu einer viereckigen Grundfläche kombinieren und brauchen deshalb eine gerade Anzahl von Dreiecksbasisflächen.

Harmonische Einbettung

Da die Dreiecksbasis natürlicherweise an den Kanten der Dreiecke Unstetigkeitsstellen aufweist, entsteht bei der Parametrisierung der Fläche metrischer Verzerrung. Während im allgemeinen nicht klar ist wie diese metrische Verzerrung möglichst klein gehalten werden kann, verweist Matthias Eck auf ein nahe verwandtes Problem mit einer einheitlichen Lösung. Er zeigt den Homomorphismus zwischen dem ursprünglichen 3D-Gitter und der Harmonischen Einbettung auf. Man stelle sich das Dreiecksgitter als ein Konstrukt von Federn vor, so dass auf jede Dreieckskante des Gitters eine Feder zu liegen kommt, deren Länge gerade gleich der Länge der Dreieckskante ist. Die Harmonische Einbettung ist nun die Lage der Federn die sich einstellt, wenn das Federgitter auseinander gezogen und auf ein konvexes Polygon in der Ebene aufgespannt wird. Die Parametrisierungen, wie auch die gleich folgende Delaunay Triangulation werden nun auf dieser Harmonischen Eingebetteten (auch harmonic map gennant) durchgeführt und anschliessend auf das 3D-Gitter transformiert.



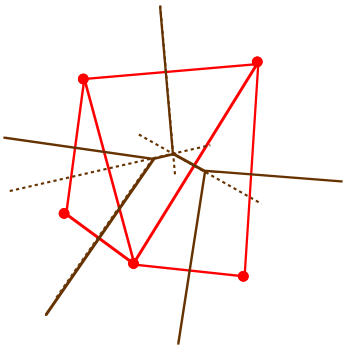
3D-Gitter



Harmonische Einbettung

Voronoi-ähnliche Zerlegung

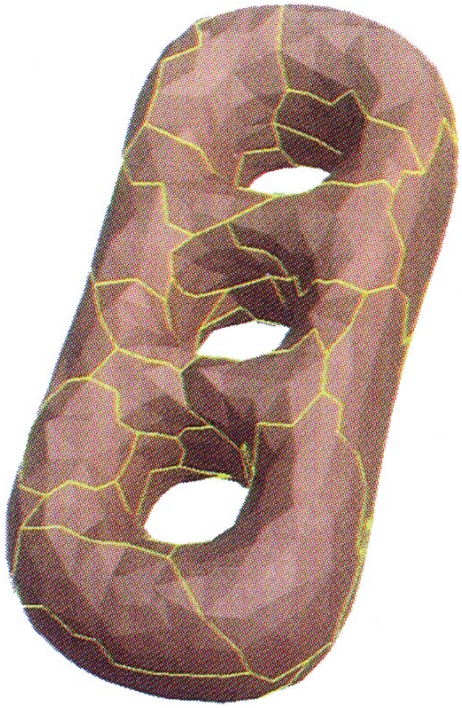
Bei der üblichen Voronoi-Zerlegung geht man von einer Menge von Punkten W_i in der Ebene aus. Die Ebene wird nun so zerlegt, dass jeweils



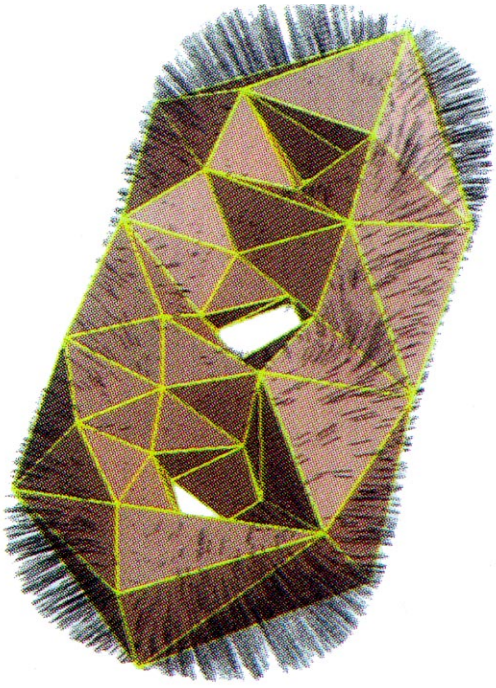
alle Punkte die näher bei einem W_X liegen als bei anderen $W_i \neq W_X$ zusammen gehören und ein Fläche V_X bilden. Berechnet werden diese Flächen, indem zwischen W_X und allen darumliegenden Punkten jeweils die Mittelsenkrechten konstruiert werden. Die gesuchte Fläche liegt (von W_X aus gesehen) innerhalb aller umgebenden Mittelsenkrechten.

Bei der Delaunay Triangulation werden nun die Punkte miteinander verbunden, deren zugehörige Flächen aneinander anliegen. Es ergeben sich der Grösse nach etwa gleich viele Delaunaydreiecke, wie es durch die Voronoi-Zerlegung Flächen gibt.

In unserem Fall möchten wir aber die Anzahl der Delaunay-Triangulations-Dreiecke drastisch reduzieren. Die Möglichkeit, die hier aufgezeigt wird, besteht darin, mehrere obige V_X zu einer neuen Fläche zu kombinieren und für diese neuen Flächen dann die Delaunay Triangulation zu berechnen. Um bei der Voronoi Zerlegung nicht alle Mittelsenkrechten berechnen zu müssen, wird das bestehende Dreiecksgitter als Approximation zu Hilfe genommen. Der diesbezügliche Algorithmus breitet die Voronoi Zerlegung, beginnend bei einem beliebigen Knoten als Startpunkt, über das ganze Dreiecksgitter aus, bis dieses vollständig zerlegt ist. Von einem bestimmten Knoten aus gesehen, der Mittelpunkt einer Voronoi-Fläche ist, kommen nur die Knoten als Zentren anliegender Voronoi-Flächen in Frage, welche eine Dreieckskante quer zwischen sich liegen haben. Diese Dreieckskante wird gerade als Näherung für die Mittelsenkrechte genommen. Zur Durchführung der Delaunay Triangulation wird für die Objektoberfläche abschnittsweise die Harmonische Einbettung berechnet. Die Triangulation wird in der Harmonischen Einbettung ausgeführt und anschliessend wieder auf die Objektoberfläche transformiert. Die so erreichte Delaunay Triangulation kann nun aber noch an den Grenzstellen zwischen zwei Voronoi Flächen Knicke aufweisen. Um die Delaunaykanten zu strecken, konstruieren wir jeweils noch eine Harmonische Einbettung, indem wir die zwei an die Kante anliegenden Dreiecke auf ein Quadrat aufspannen und als neue Delaunaykante die Diagonale des Quadrates nehmen und auf die Oberfläche zurück transformieren.



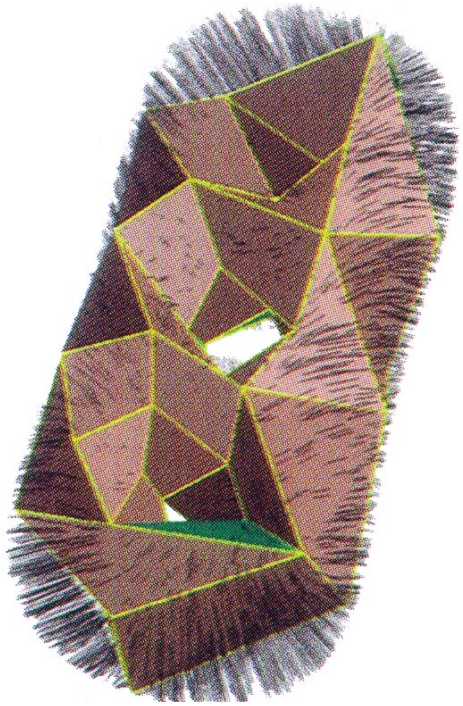
Voronoi-ähnliche Zerlegung



aus Delaunay Triangulation resultierende vereinfachter Basiskomplex

3. Reparametrisierung über einer viereckbasierten Grundfläche

Da wir für die Konstruktion von B-Splines eine Grundfläche mit viereckigen Patches benötigen, müssen wir, ausgehend von einer dreiecksbasierten, eine viereckbasierte Grundfläche erzeugen. Wir erreichen dies, indem wir jeweils zwei benachbarte dreieckige Grundflächen zu einer Viereckigen kombinieren. Um eine komplette Paarung der Patches zu ermöglichen haben wir uns ja schon weiter oben auf eine gerade Anzahl von Dreieckspatches festgelegt.
Die Aufgabe der vollständigen Kombination solcher Dreieckspatches wird auf ein Maximum-Graphen-Matching-Problem zurückgeführt. Es wird ein zum Dreiecksgitter dualer Graph $G = (V_G, E_G)$ eingeführt. Die Dreieckspatches gehen über in die Knoten des Graphen V_G und jeweils zwei aneinander anliegende Dreiecksflächen bilden eine Kante E_G des Graphen. Es sind viele verschiedene Lösungen dieses Kombinationsproblems möglich. Wir hätten gerne eine Paarung der Dreiecksflächen, welche die Verzerrung der nachfolgenden Parametrisierung minimiert. Erreicht wird dies durch die Einführung von Gewichten für die Kanten des Graphen G. Als Gewicht einer Kante nehmen wir die Engergie, der über ein Einheitsquadrat konstruierten Harmonischen Umgebenden des Viereckspatches, das bei der Kombination der zwei Dreiecksflächen entstehen würde.



Zum Schluss müssen wir die Punkte X_i wieder neu über den viereckbasierten Basiskomplex parametrisieren. Um eine minimale Verzerrung zu erreichen, machen wir nochmals Gebrauch von der bereits berechneten Harmonischen Umgebenden.

4. B-Spline Fitting

Flächenrekonstruktion

Bei der Flächenrekonstruktion möchten wir die Kontrollpunkte aller B-Splines so bestimmen, dass die Distanz der Datenpunkte X_i zu unserer Fläche S minimal wird. Wir minimieren die Distanzfunktion

$$E_{dist}(S) = \sum_{i=1}^{N} d^2(X_i, S)$$

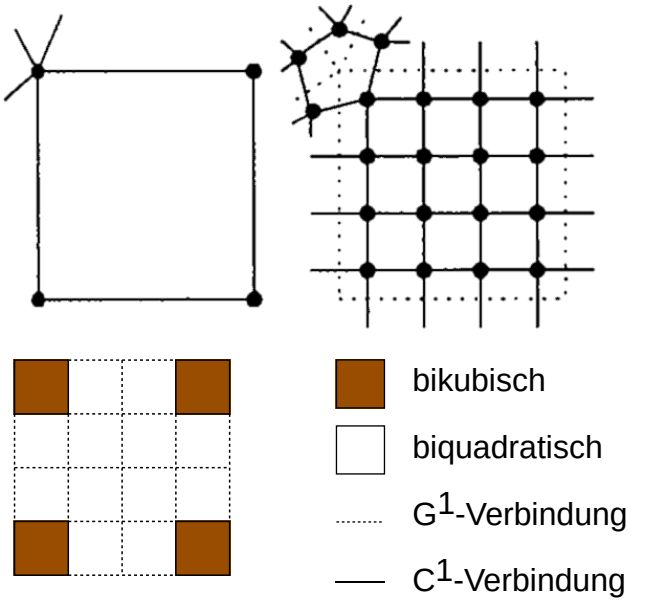
Die Schwierigkeit dabei ist, dass die Fläche S durch die Position der Kontrollpunkte und deren Parametrisierung beschrieben wird. Beides sind zu Beginn noch Unbekannte. Die gleichzeitige Berechnung der Positionen der Kontrollpunkte und deren Parametrisierung ist nicht möglich. Abhilfe schafft hierbei ein iteratives Verfahren bei dem jede Iteration jeweils aus zwei Schritten besteht:
1. Fitting-Schritt: Für eine feste Parametrisierung werden durch Lösung eines linearen Least-Squares Problems die optimalen Kontrollpunkte bestimmt.
2. Parameter-Korrektur-Schritt: Für feste Kontrollpunkte werden durch Projektion der Punkte auf S die optimalen Parameter gefunden.
Vier Durchgänge dieser Iteration genügen um die Approximationsfläche genügend genau zu bestimmen.
Ein weiteres, mit diesem Ansatz noch ungelöstes Problem ist, dass die so entstehenden Flächen noch ungewollte ziemlich spitzige Hügel auf der Oberfläche aufweisen können. Eine Variante dies zu verunmöglichen, ist die Hinzufügung eines zusätzlichen Fairnessterms zur Energiefunktion. Die zu minimierende Gleichung lautet nun neu:
$E(S) = E_{dist}(S) + \lambda * E_{fair}(S)$
Als Fairnessterm $E_{fair}(S)$ wird das «thin plate energy»-Funktional eingesetzt. Das «thin plate energy»-Funktional ist das Flächenintegral der zweiten Ableitung der Oberflächenfunktion über

die ganze Oberfläche und beschreibt somit die Krümmung der Fläche. Wird versucht die Gesamtkrümmung der Oberfläche möglichst klein zu halten, so wirkt dies der Bildung von spitzigen Stellen (starke Krümmung) entgegen.

Einpassen von Bézier-Flächen

Bézierflächen sind einfacher von Grund auf zu konstruieren als B-Splines (automatische Randpunktinterpolation, einfache Vorgabe der Steigung). Deshalb werden in die Viereckspatches der Basisfläche erst Bézierflächen eingesetzt, die dann zu einem späteren Zeitpunkt in B-Splines umgewandelt werden. Die Bézier-Flächen werden in zwei Schritten in den Basiskomplex K_4 eingepasst:

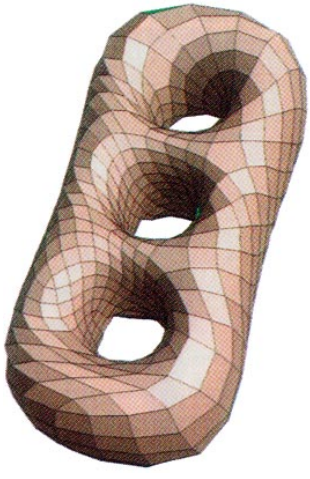
Als erstes wird K_4 entsprechend der Figur B in ein feineres Kontrollnetz M_X unterteilt. In jede Viereckspatch von K_4 kommen 4 x 4 Gitterpunkte zu liegen. Alle Gitterpunkte haben so die Wertigkeit 4. M_X besteht hauptsächlich aus vierseitigen Flächen. Die Ausnahme bilden einige wenige isolierte Flächen die mehr als vier Kanten haben.



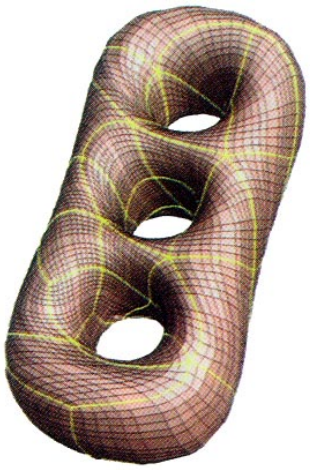
In einem zweiten Schritt wird über jedem Knoten von M_X eine Tensorprodukt Bézierfläche konstruiert. Wie in Figur C gezeigt ist, wählt man eine bikubische Bézierfläche, wenn sie an eine mehr-als-vierkantige Fläche grenzt, ansonsten eine biquadratische. Die Kontrollpunkte für ein Biquadratisches Bézier-Patch werden aus der Mittelung der Gitterpunkte M_X berechnet. Die Berechnung der biqubischen Bézier-Patches ist einiges aufwendiger und sei hier nicht aufgeführt. Die Einhaltung der C^0-Stetigkeit folgt trivialerweise daraus, dass die Kontrollpunkte entlang der Grenzen der Bézier-Patches geteilt werden. J. Peters beweist auch dass die von ihm gewählte Anordung die G^1- resp. C^1-Stetigkeit erfüllt.

Bézier-Flächen -> B-Spline-Flächen

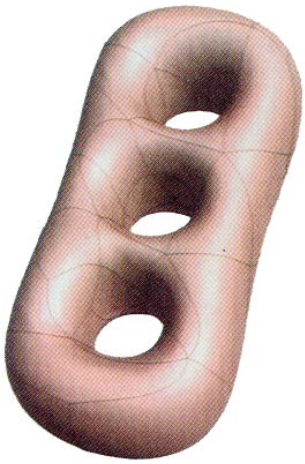
Die 4x4 Bézier-Patches über jedem Viereckspatch von K_4 können zu einem einzigen bikubischen Tensorprodukt B-Spline-Patch zusammengefasst werden. Um die G^1- und C^1-Stetigkeit zu befriedigen wird der Knotenvektor in beide Parameterrichtungen auf
(0, 0, 0, 0, 0.25, 0.25, 0.25, 0.5, 0.5, 0.75, 0.75, 0.75, 1, 1, 1, 1)
gesetzt. Da die B-Spline-Fläche bikubisch ist, wird mit der vierfachen Einsetzung der Randpukte gerade eine Randpunktinterpolation erreicht.



Kontrollgitter M_X



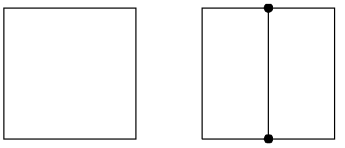
B-Spline Kontrollnetz



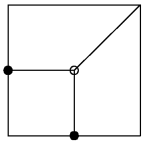
B-Spline Fläche

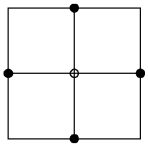
5. Anpassungsfähige Verfeinerung (adaptive refinement)

Beim Oberflächen-Fitting Algorithmus haben wir schon versucht die Summe des quadratischen Abstands $d(X_i, S)^2$ der Punkte X_i zur B-Spline-Oberfläche S möglichst klein zu halten. Nun ist es aber wünschenswert für das Fitting eine maximale Fehlertoleranz angeben zu können. Die Anpassungsfähige Verfeinerung berechnet eine Fläche für die $\max(d(X_i, S)) < \varepsilon$, wobei ε die vom Benutzer vorgegebenen Fehlertoleranz ist.
Erreicht wird dies, indem jedes Viereckspatch das einen Punkt X_i beinhaltet, der die Fehlertoleranz überschreitet, unterteilt wird. Die resultierenden Patches müssen immer noch ein gültiges Netzwerk beschreiben. Die Verfeinerung der viereckbasierten Basisfläche K_4 wird durch die Auswahl einer Untermenge von Kanten E' von allen Kanten der Basisfläche E definiert. Für jede Kante der Menge E' wird in deren Mitte ein neuer Knoten eingeführt. Wenn eine Fläche verfeinert werden muss, so werden alle ihre vier Kanten der Menge E' zugewiesen. Anschliessend werden alle Patches von K_4 unter der Benutzung eines der Verfeinerungstemplates aufgeteilt. Da für Viereckspatches mit einer ungeraden Anzahl verfeinerten Kanten keine Verfeinerungen möglich sind, muss die Menge E' noch um zusätzliche Kanten ergänzt werden. Nachdem die Basis K_4 lokal verfeinert worden ist muss die Parametrisierung der Punkte X_i stellenweise neu berechnet werden. Nach dem Schritt der Anpassungsfähigen Verfeinerung wird der B-Spline-Fitting Schritt wiederholt. Der Anpassungsfähige Verfeinerungsschritt und das B-Spline-Fitting werden im Weiteren so oft nacheinander wiederholt, bis die angegebene Fehlertoleranz ε erreicht wird.


template 1

template 2


template 3


template 4

6. Resultat, Kritik, Ausblick

Das angegebene Ziel, ein vollautomatisches Verfahren zur Rekonstruktion von B-Splines zu finden, wurde sicherlich erreicht. Gerade auch mit der Möglichkeit der Anpassungsfähigen Verfeinerung, ist das Verfahren sehr anwendungsnahe, auch wenn gerade die Verfeinerung ganz schön viel Rechenzeit in Anspruch nimmt. Die Autoren haben auf geschickte Weise Vorgängerarbeiten etwas abgeändert und aneinander angepasst. Dies ist ihnen sicherlich nicht zuletzt daher so gut gelungen, da ja die meisten dieser Arbeiten von ihnen selbst geschrieben wurden. Im allgemeinen kann man sicher auch sagen, dass die Autoren aus einem grossen Basiswissen schöpften, werden doch Ansätze und Verfahren aus ganz verschieden Bereichen zusammengetragen und angewandt.
Das Hauptproblem bei der hier gewählten Flächendarstellung ist sicherlich die Darstellung von gewollten Knicken in der Oberfläche, von scharfen Kanten. Eine Weiterführung wäre hier stückweise glatte Oberflächenrekonstruktionen zu ermöglichen. Eine andere Variante für die Darstellung von kleineren spitzigen Oberflächendetails wäre die Unterstützung von «displacement maps», die auf der glatten Oberfläche aufliegen würden.
Bei der Kombination von zwei Dreieckspatches im Bearbeitungsschritt: «Reparametrisierung über eine viereckbasierte Basisfläche» wird etwas Heuristik betrieben. Es ist nicht klar, ob es die Möglichkeit gibt, dass ein vollständiges Matching nicht existiert.

Quellen

- «Automatic Reconstruction of B-Spline Surfaces of Arbitrary Topological Type» von Matthias Eck und Hugues Hoppe, '96
- «Surface Reconstruction from Unorganized Points» von Hugues Hoppe und anderen, '92
- «Mesh Optimization» von Hugues Hoppe und anderen, '93
- «Multiresolution Analysis of Arbitrary Meshes» von Matthias Eck und anderen, '95
- Skript zur Vorlesung GDV II von Prof. Markus Gross, ETHZ