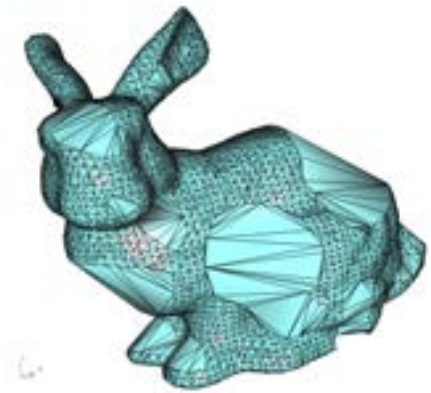
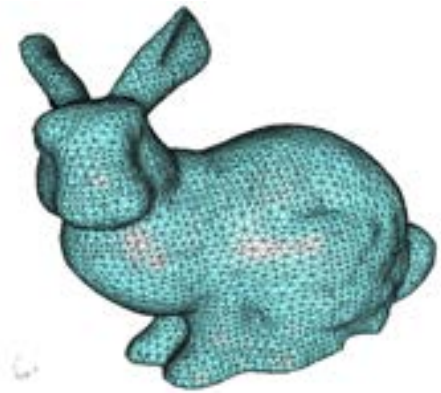
Auffüllen von Löchern in Polygonnetzen

Eine Methode für unstrukturierte Dreiecksnetze nach P. Liepa

Benjamin Willenberg

Seminar "Computergrafik" an der Georg-August-Universität Göttingen,
geleitet durch Dipl.-Inf. Reinhard Hemmerling und Prof. Dr. Winfried Kurth

10. Januar 2012

Einleitung und Motivation

Polygonnetze in der Computergrafik

- Mit Kanten untereinander verbundene Punkte
- Derzeit sehr populäre mathematische Beschreibung geometrischer Daten
- Basis vieler CAD-Anwendungen mit besonderen Anforderungen (z.B. wasserdichtes und/oder mannigfaltiges Netz für 3D-Drucke (SLT) oder Finite-Elemente-Methode)
- Generiert aus Punktwolken verschiedenster Eingabedaten
  $\Rightarrow$ oft Artefakte wie Überdeckungen, Löcher, Spalte, Selbstschnitte usw. im primären Polygonnetz

$\Rightarrow$ zuverlässige Algorithmen zum Entfernen der Artefakte benötigt

Auffüllen von Löchern in Polygonnetzen

Oberflächenbasierte Verfahren

Generelle Funktionsweise der Verfahren

- Direkte Verarbeitung des Inputs (Defekte werden auf Oberfläche lokalisiert und repariert)
- Beseitigt Spalten, Löcher und kleinere Überlappungen
- Methoden: *snapping*, *stitching*, *Triangulation*, Aufspalten von Kanten und Dreiecken

Vorteile

- Struktur des Originalnetzes bleibt erhalten
  $\Rightarrow$ mit Netz verknüpfte lokale Eigenschaften weiter verfügbar
- nur wenige neue Dreiecke

Nachteile

- Gültige Ausgabe nur bei erfüllten Voraussetzungen (Mannigfaltigkeit, Selbstschnittfreiheit) an Input garantiert
- Anfällig für numerische Ungenauigkeiten (kleine Defekte werden nicht gefunden)

Volumenbasierte Verfahren

Generelle Funktionsweise der Verfahren

- Konvertierung des Inputs (Polygonnetz) in Volumenmodell
- Zwischenmodell $\Leftrightarrow$ Unterteilung des Raumes in Volumenelemente (für jedes festlegen, ob Inneres, Äußeres oder Rand des Modells)

Vorteile

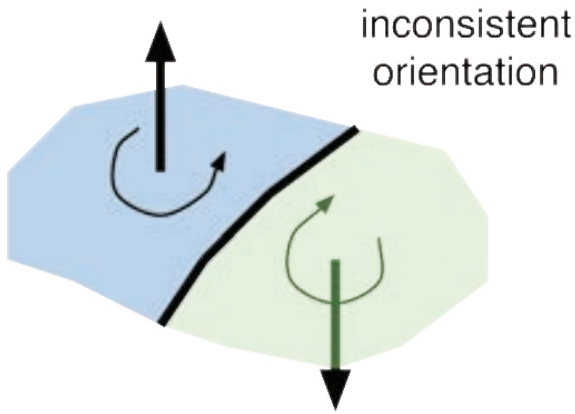
- keine Selbstschnitte, Löcher, Lücken oder Überlappungen
- vollautomatisch
- garantiert wasserdichte Ausgabe

Nachteile

- Downsampling des Modells zur Umwandlung in Volumenmodell $\Rightarrow$ Alias Artefakte, Detailverlust
- Verlust jeglicher Strukturinformationen des Originals
- Dreiecksanzahl der Ausgabe i.d.R. viel Größer
  $\Rightarrow$ sehr speicherlastig, Nachberarbeitung notwendig

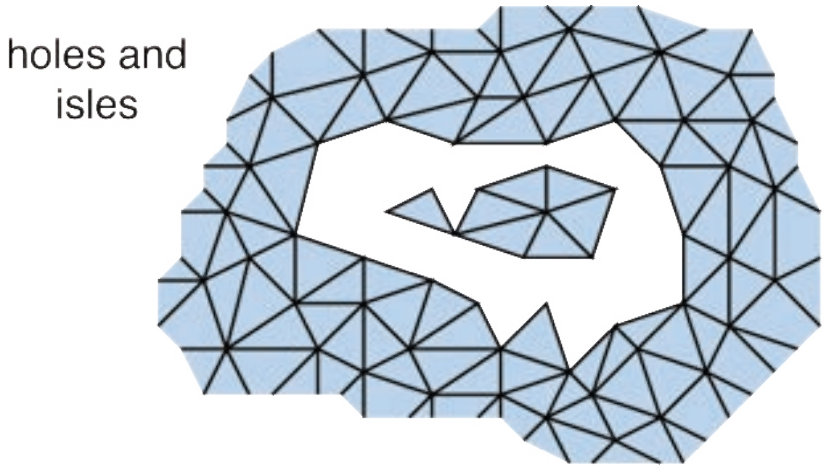
Definitionen und Begriffe

- Dreiecksnetz: Polygonnetz aus orientierten Dreiecken
- benachbarte Dreiecke: eine gemeine Kante
- konsistent orientierte Dreiecke: Dreiecke besitzen gleichen Umlaufsinn, d.h. gemeinsame Kanten werden gegenläufig durchlaufen
- Orientiertes Dreiecksnetz: alle Dreiecke sind gleich orientiert



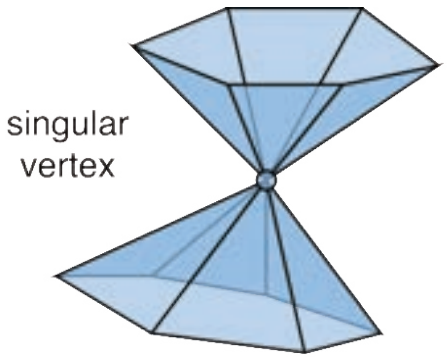
Definitionen und Begriffe

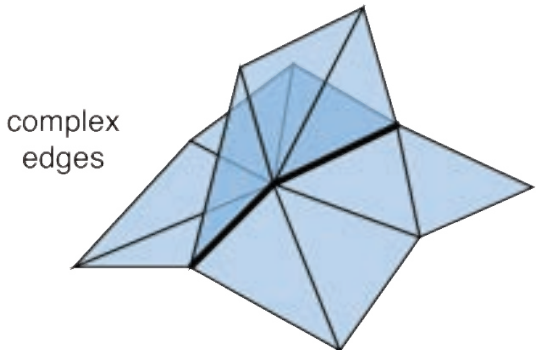
- Randkante: Kante, die genau an ein Dreieck angrenzt
- Innere Kante: Kante, die keine Randkante ist
- Loch: geschlossene Kurve aus Randkanten
- Insel: nicht mit dem Rest des Polygonnetzes verbundenes Netzstück

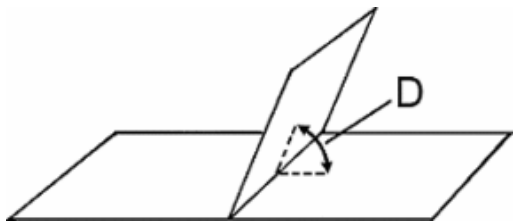


Definitionen und Begriffe

- Singulärer (Eck-)Punkt: mehr als zwei assoziierte Randkanten
- Nicht-mannigfaltige Kante: > 2 angrenzende Dreiecke
- Mannigfaltiges Polygonnetz: nur mannigfaltige Kanten
- Diederwinkel: Winkel zwischen den Normalen zweier gleich orientierter Dreiecke







Oberflächenbasiertes Verfahren zum Ausfüllen von Löchern in Dreiecksnetzen nach Liepa

Berechnet zuverlässig Flicken für Löcher in orientierten, mannigfaltigen und zusammenhängenden Dreiecksnetzen.

Methode arbeitet in 4 Schritten

1. Lokalisierung der Löcher auf der Oberfläche
2. Triangulation der gefunden Löcher
3. Verfeinerung des berechneten Flickens für das Polygonnetz
4. Glätten und Modellieren des Polygonnetz-Flickens

Anforderungen an den Algorithmus

1. möglichst automatisch, ohne eingreifen des Benutzers
   (wenn Nutzeraktion notwendig, dann so einfach wie möglich)
2. geringe Laufzeit
   (idealerweise gering genug für interaktiven Einsatz)
3. große Ähnlichkeit des Flickens für das Loch mit umliegendem Polygonnetz, d.h. Übereinstimmung von Dichte und Struktur des Flickens mit umgebendem Netz
4. keine Probleme beim Umgang mit diversen, insbesondere extrem unregelmäßigen und nicht planaren Löchern

Klassifizierung und Lokalisierung der Löcher

Möglichkeiten zur Bestimmung von Löchern

1. Gebe eine Randkante vor und laufe so lange weiter, bis sich eine geschlossene Kurve aus Randkanten ergibt (automatisch ohne Benutzereingabe).
2. Lassoauswahl eines Bereichs mit Loch (benutzerinteraktiv)

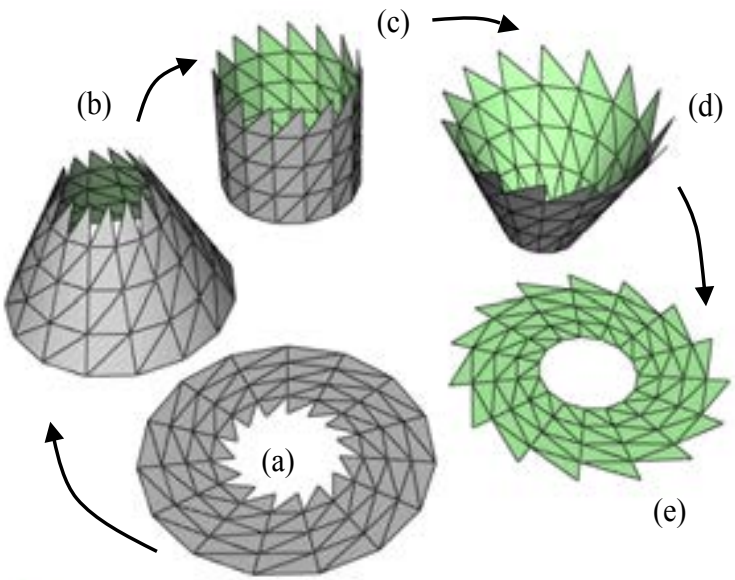
Problem:
Topologische Definition eines Loches zu allgemein,
inneres Loch sollte eher gefüllt werden.

Klassifizierung und Lokalisierung der Löcher

Lösung: Klassifizierung der Löcher durch Oberflächenwinkel

$$\alpha(v) = \sum_{0<i<n} \angle(e_i, e_{i+1}),$$

$\{e_1, \ldots, e_n\}$ sind die an Ecke v angrenzenden Kanten



Gesamtoberflächenwinkel für gezacktes Loch ($\alpha_{tot} = \sum_i \alpha(v_i)$)

$$\alpha_{tot}(a) = (n+2) \cdot \pi$$
$$\alpha_{tot}(c) = n \cdot \pi$$
$$\alpha_{tot}(e) = (n-2) \cdot \pi$$

$\Rightarrow$ größerer Winkel im Vergleich zur Eckenzahl n spricht für 'echtes' Loch

Triangulation von Löchern in Polygonnetzen

Zerlegung des Lochpolygons in Dreiecksnetz unter Minimierung eines Gewichts (z.B. Fläche). (nach Barequet et al., 1995)

Definitionen

Gewichtungsmenge L: geordnete Menge mit Add. und Nullelm. 0_L

Gewicht: Element dieser Menge

$V = \{v_0, v_1, ..., v_{n-1}\}, v_i \in \mathbb{R}^3$ sei Menge von Punkten, dann weist die *Gewichtungsfunktion* $\Omega : V^3 \to L$ jedem Dreieck mit Punkten aus V ein Gewicht aus L zu.

Die v_i spannen des Polygon $P = (v_0, v_1, ..., v_{n-1})$ auf.

Für $0 \leq i < j < n$ sei $W_{i,j}$ das min. *Gesamtgewicht* für die Triangulation des Teilpolygons $P_{i,j} = (v_i, ..., v_j)$.

Triangulation von Löchern in Polygonnetzen

Algoithmus

Starte mit kleinsten Polygoneinheiten

1. Kanten: $W_{i,i+1} = 0_L, i = 0, 1, ..., n-2$
   Dreiecke: $W_{i,i+2} = \Omega(v_i, v_{i+1}, v_{i+2}), i = 0, 1, ..., n-3$
   Setze $j = 2$.
2. $j = j + 1$
   $k = i + j, i = 0, 1, ..., n - j - 1$
   $W_{i,k} = \min_{i<m<k}(W_{i,m} + W_{m,k} + \Omega(v_i, v_m, v_k))$
   Setze $\lambda_{i,k} = m$, wobei m der Index des Minimums.
3. Für $j < (n-1)$ zurück zu 2, sonst berechne Gewicht $W_{0,n-1}$ der gewichtsminimierenden Triangulation
4. Bestimme beste Triangulation aus den $\lambda_{i,k}$.

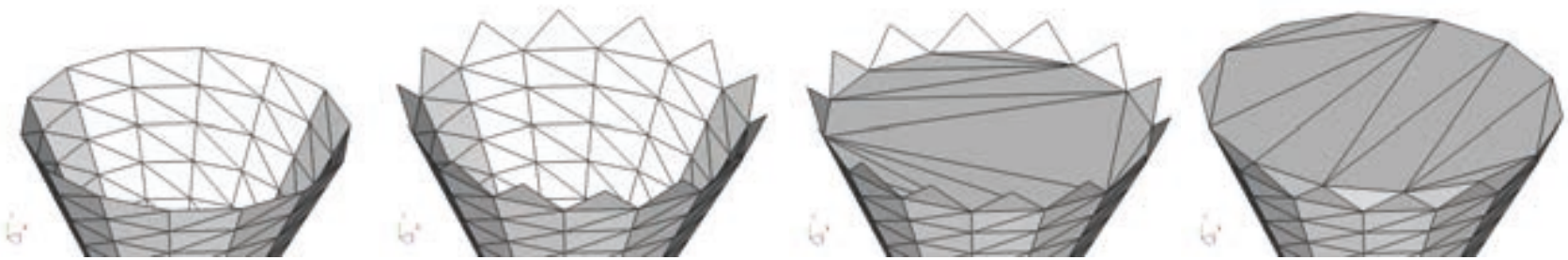
Triangulation von Löchern in Polygonnetzen

Fläche als Gewichtung

Idee: $L_{area} = [0, \infty)$, $0_{L_{area}} = 0$,

$$\Omega(v_i, v_m, v_k) = \text{Fläche}(\triangle(v_i, v_m, v_k))$$

Ergebnis: Triangulation mit min. Fläche
Beobachtung: gute Ergebnisse für relativ *planare* Löcher, Probleme bei *Zinnen*, d.h. zur Triangulationsfläche senkrecht stehenden Randstücken



Triangulation von Löchern in Polygonnetzen

Tupel (Diederwinkel, Fläche) als Gewichtung

Idee: $L_{angle} = [0, \pi] \times [0, \infty)$, $0_{L_{angle}} = (0, 0)$
Ordnung: $(a, b) < (c, d) \Leftrightarrow (a < c) \vee (a = c \wedge b < d)$
Addition: $(a, b) + (c, d) := (\max(a, c), b + d)$
Gewichtungsfkt.:

$$\Omega_{angle}(v_i, v_m, v_k) = (\mu(v_i, v_m, v_k), \Omega_{area}(v_i, v_m, v_k))$$

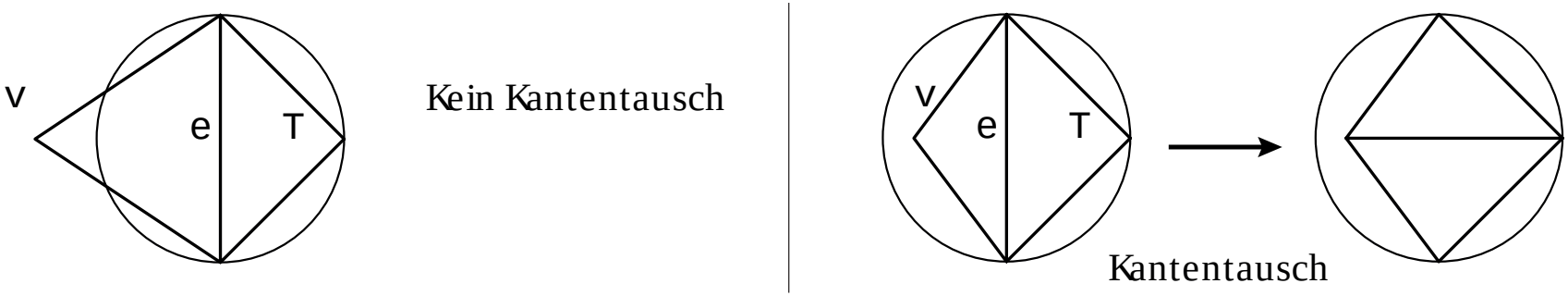
$\mu(v_i, v_m, v_k)$: max. Diederwinkel zw. $\triangle(v_i, v_m, v_k)$ u. angrenzenden
Beobachtung: große Diederwinkel werden bestraft, Aufbau von konvexer Hülle um den Lochrand vor Triangulation mit min. Fläche

Verfeinerung und Entspannung des Polygonnetz-Flickens

Ziel: Übertragung der Dichte des Polygonnetzes auf Flicken
Algorithmus (Idee nach Pfeifle und Seidel, 1996):
1. Verkürzen der Kanten im Flicken durch neue Eckpunkte
2. Entspannen innerer Kanten

$\Rightarrow$ Delaunay Triangulation (kein Punkt in Umkreis jedes Dreiecks)

Entspannung innerer Kanten (Flip-Algorithmus)



Keine spitzen Winkel $\Rightarrow$ weniger Rundungsfehler

Verfeinerung und Entspannung des Polygonnetz-Flickens

Verfeinerungs-Algorithmus (nach Pfeifle und Seidel)
Dichte Kontrollparameter $\alpha = \sqrt{2}$ ist empirische gegeben.

1. Für Ecken v_i auf dem Lochrand: Kantenlängenattribut:
   $\sigma(v_i) = \frac{1}{N} \sum_{j=1}^{N} ||v_i - v_j||$ mit v_j benachbarte Eckpunkte
2. Für jedes $\triangle(v_i, v_j, v_k)$ in Flicken berechne Schwerpunkt v_c,
   setze $\sigma(v_c) = \frac{1}{3}(\sigma(v_i) + \sigma(v_j) + \sigma(v_k))$.
   Wenn $\alpha||v_c - v_m|| > \sigma(v_m)$ und $\alpha||v_c - v_m|| > \sigma(v_c)$
   $(m = i, j, k)$ teile Dreieck in (v_i, v_j, v_c), (v_i, v_c, v_k), (v_c, v_j, v_k)
   und entspanne Kanten (v_i, v_j), (v_j, v_k), (v_k, v_i)
3. Wenn keine neuen Dreiecke in 2, dann fertig
4. Entspannen aller inneren Kanten des Flickens
5. Wenn kein Kantentausch in Schritt 4,
   dann zurück zu Schritt 2, sonst wiederhole Schritt 4.

Glätten und Modellieren des Polygonnetz-Flickens

Ziel: Glättung der Oberfläche und Anpassung der Wölbung des Flickens an den Rand
Algorithmus aufbauend auf dem von Kobelt et al. (1998)
$\omega : V^2 \to \mathbb{R}$ sie Gewichtungsfunktion für jede Kante

gewichteter Umbrella-Operator
$U_\omega(v) = -v + \frac{1}{\omega(v)} \sum_i \omega(v, v_i) v_i$ mit $\omega(v) = \sum_i \omega(v, v_i)$

Glättung durch Ersetzen jedes Punktes v durch $v + U_\omega(v)$.

gleichmäßiger Umbrella-Operator
mit $\omega(v_i, v_j) = 1 \forall v_i, v_j \Rightarrow$ zu globales Glätten

Abstandsabhängiger Umbrella-Operator
mit $\omega(v_i, v_j) = 1/||v_i - v_j|| \Rightarrow$ lokales Glätten

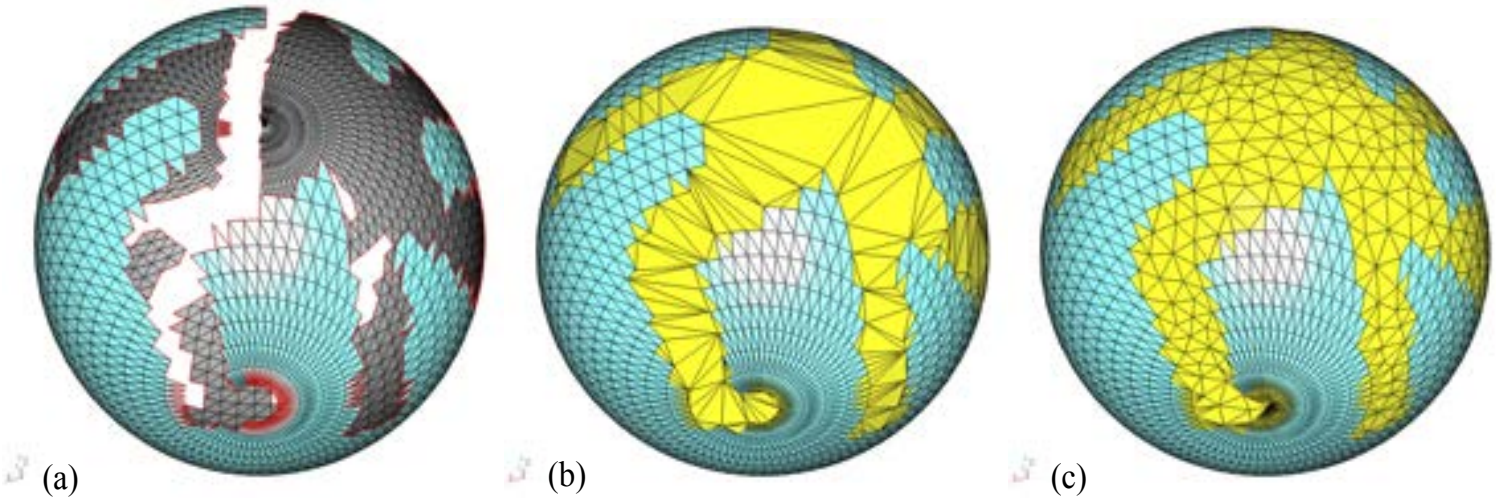
Glätten und Modellieren des Polygonnetz-Flickens

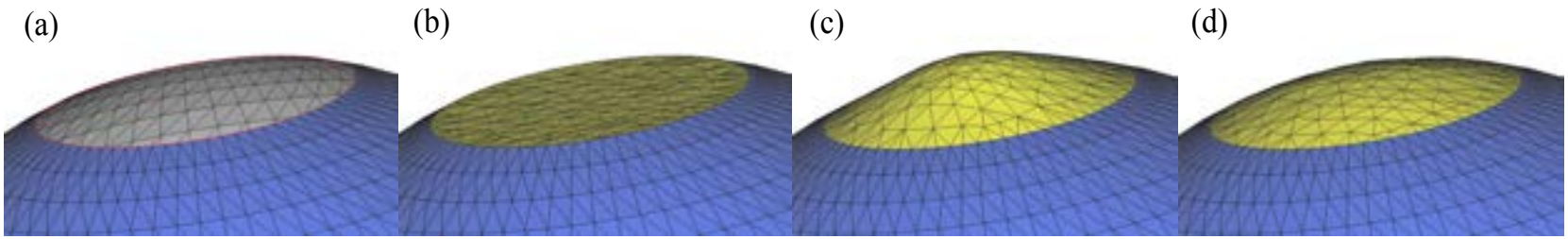
Zweite Ordnung für mannigfaltige Flächen
$U^2_\omega(v) = -U_\omega(v) + \frac{1}{\omega(v)} \sum_i \omega(v, v_i) U_\omega(v_i)$

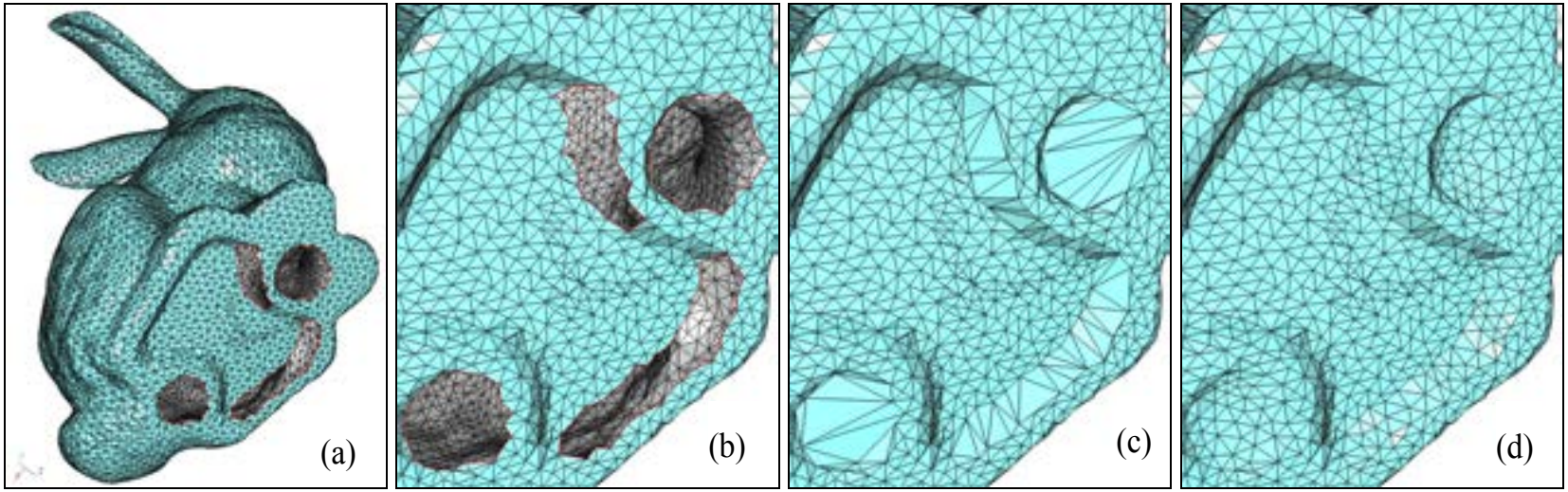
$U^2_\omega(v) = 0$ passt Krümmung in Punkt v an Umgebung an

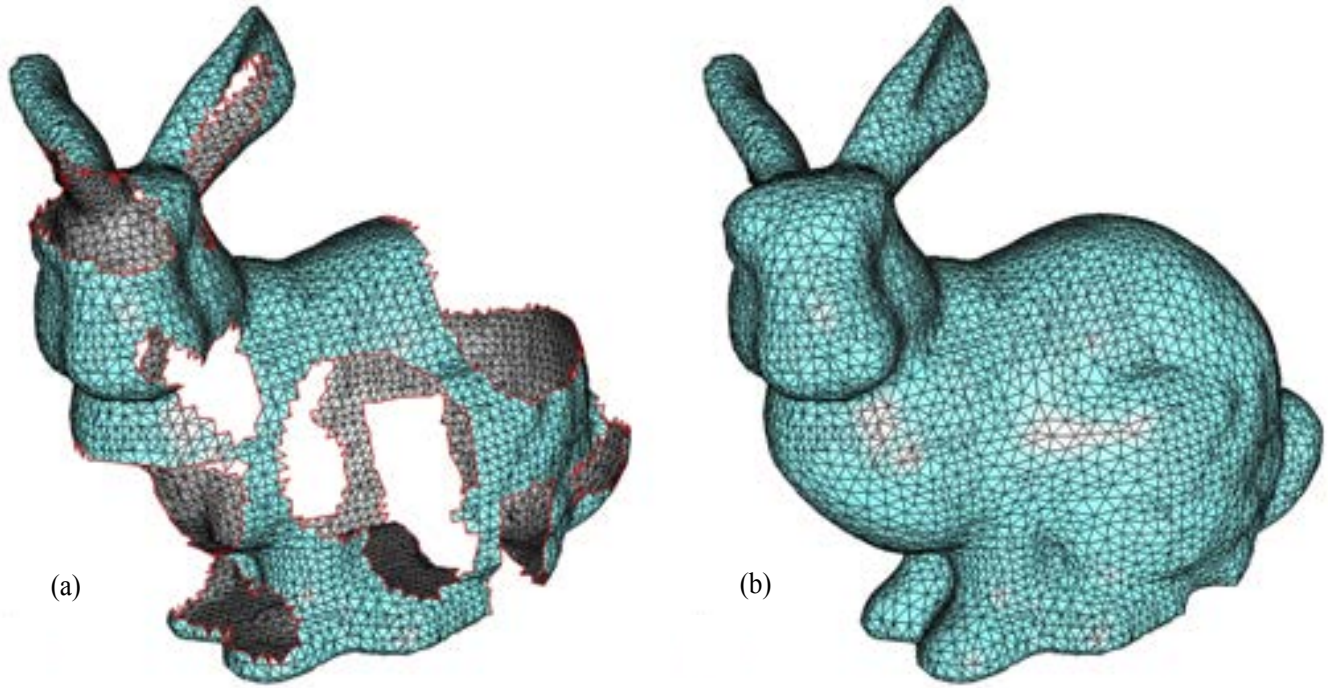
Verschiebung der v_i im Flicken, sodass $U^2_\omega(v_i) = 0$, führt auf lin. Gleichungssystem (Lösung durch konjugierte Gradienten Methode)

Anwendungsbeispiele des Verfahrens nach Liepa






(a)
(b)
(c)
(d)


(a)
(b)

Diskussion und Potential der Methode nach Liepa

- schlechte Laufzeit ($\mathcal{O}(n^3)$) bedingt durch Triangulation
  $\Rightarrow$ neuere Ansätze von Wei Zhao, Shuming Gao und Hongwei Lin (2007) verzichten auf Neuberechnung der Triangulation in jedem Schritt und sind deutlich schneller
- Die Methode von Liepa ist einfach auf beliebige Polygonnetze erweiterbar.
- Behandlung nicht zusammenhängender Polygonnetze mit Inseln ist durch vorheriges erstellen von Brücken möglich.
- Einsatz sowohl bei der Oberflächenrekonstruktion als auch bei der Beseitigung von Rauschen oder Bereichsersetzungen
- Einsatz in der Komprimierung, da Polygonnetz aus teilweise bekanntem Netz rekonstruiert werden kann. Dabei werden komplizierte Löcher weniger gut rekonstruiert.

Quellen

- Peter Liepa. 2003. Filling holes in meshes. In Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing (SGP '03). Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 200-205.
- Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. Polygon Mesh Processing. AK Peters, 2010.
- Gill Barequet and Micha Sharir. Filling Gaps in the Boundary of a Polyhedron. Computer-Aided Geometric Design, 12(2):207-229, March 1995.
- G. Barequet, M. Dickerson, and D. Eppstein. On triangulating three-dimensional polygons. Computational Geometry: Theory and Applications, 10(3):155-170, June 1998.
- ...

Alle Grafiken wurden den ersten beiden Quellen entnommen.