

BOUNDARY

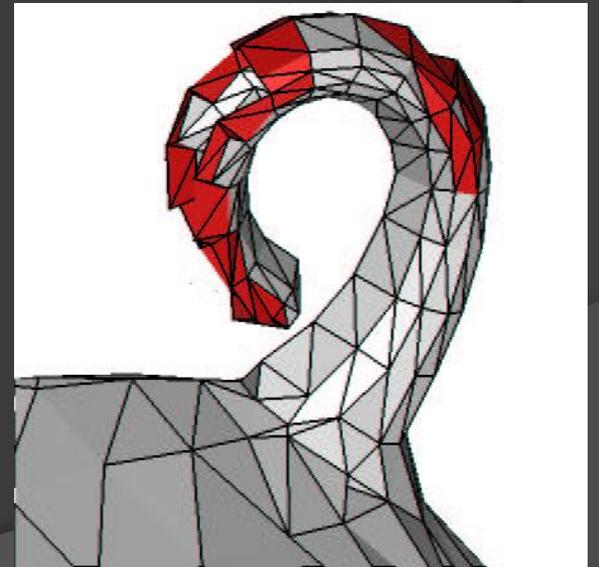
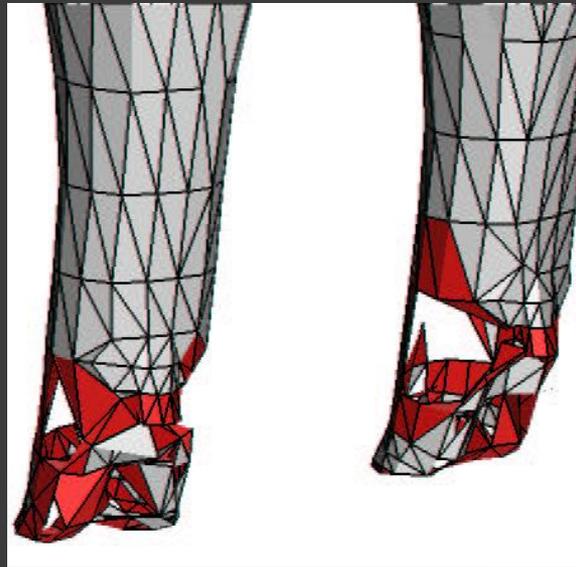
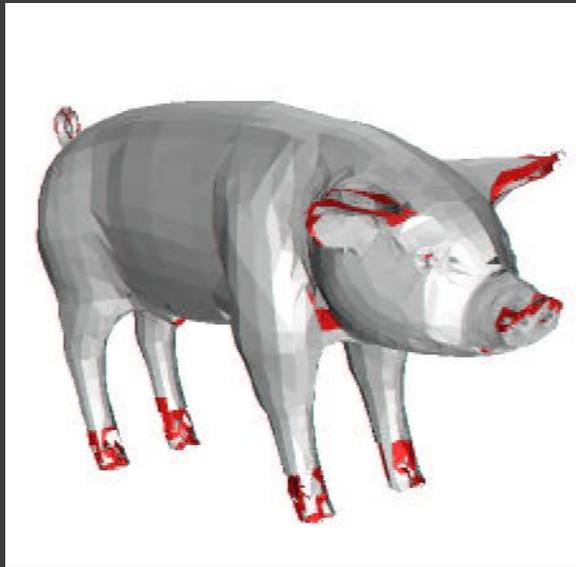
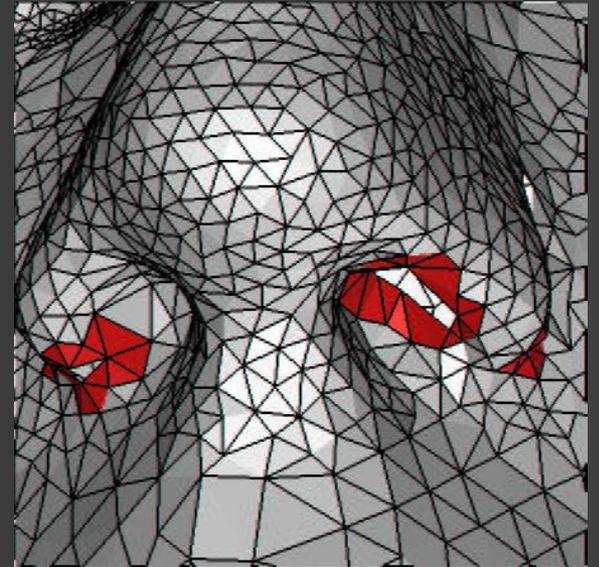
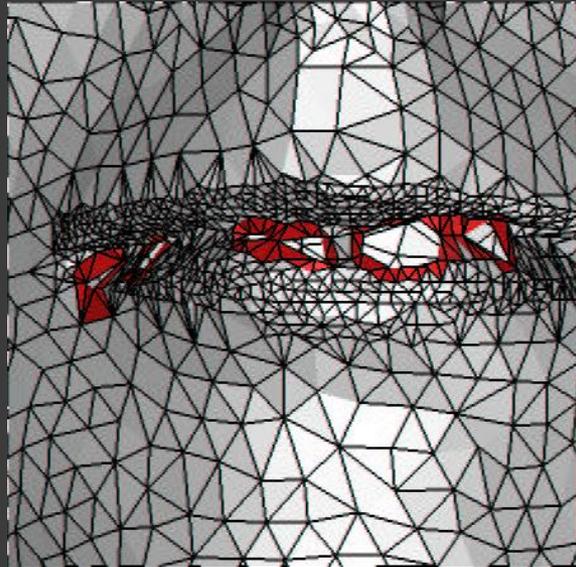
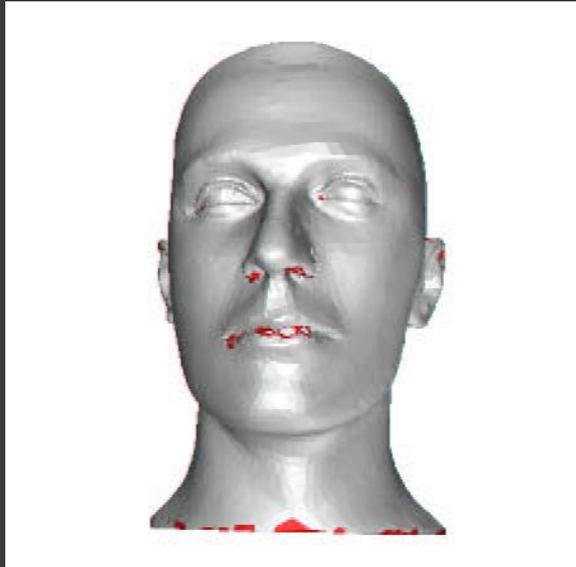
Method to detect
undersampling

Index

- ⦿ Einführung
- ⦿ Wechtige Vorkenntnisse
- ⦿ Theorie zu Algorithmus
- ⦿ Algorithmus
- ⦿ Rekonstruktion

Schwäche Cocone

- ⦿ Funktioniert gut, nur wenn Input Point Set P dicht ist. In der Praxis Input ist häufig unterabgetastet („undersample“).
- ⦿ Die Punkte in Undersampled Bereiche haben die Normalen nicht zuverlässig geschätzt.
- ⦿ Wählt deshalb unerwünschte Dreiecke in der Nähe von Undersampled Bereichen aus.



Der Algorithmus Boundary

- ⊙ Dey und Giesen: Vorschlag für eine Methode zur Feststellung der undersampled Bereiche von der Input-Punkt.
- ⊙ BOUNDARY erkennt die Punkte, die in der undersampled Bereiche liegen. Die Voronoi-Cells davon sind nicht lang und dünn entlang der Normalen an die Oberfläche. Dies wird durch zwei Condoitions „ratio“ und „normal“ festgestellt.

Verbesserte Cocone

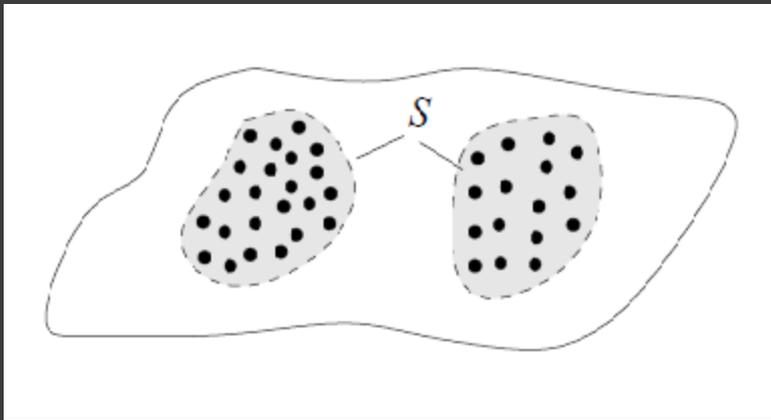
- ⦿ Die COCONE Algorithmus wird mit der Erkennung der Punkte in den undersampled Bereichen verbessert.
- ⦿ Nur die Kandidaten Dreiecke der Punkte, die nicht bei Boundary markiert sind, sind zu wählen.
- ⦿ Die Verbesserung löscht die meisten der unerwünschten Dreiecke in der Nähe von undersampled Bereichen, die die Löcher in der Oberfläche erzeugen.

Index

- Einführung
- **Wichtige Vorkenntnisse**
- Theorie zu Algorithmus
- Algorithmus
- Rekonstruktion

Well Sampled Patch

- ⦿ S : Flicken der Oberfläche (Surface Patch)
- ⦿ Jeder Punkt $p \in S$ hat mindestens einen anderen Punkt im Abstand $\varepsilon * f(p)$. Und die alle andere Punkte, die nicht in S sind, sollen die Attribute nicht habe.



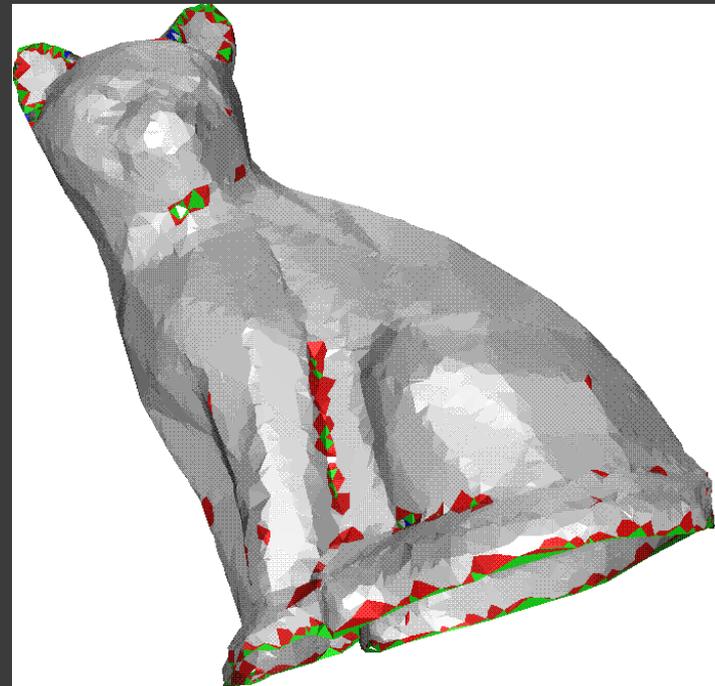
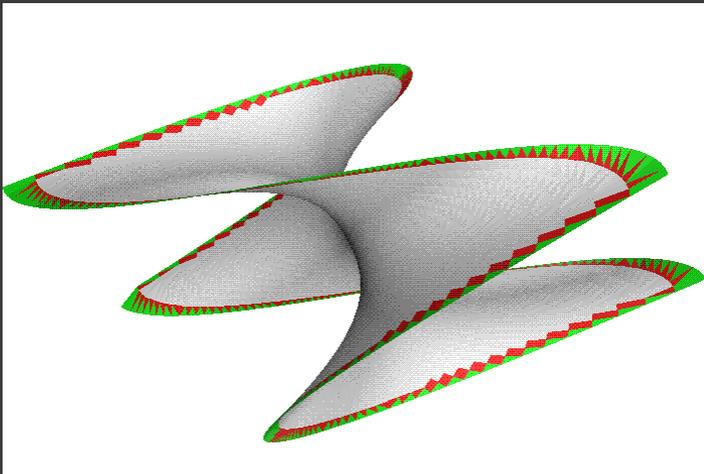
- ⦿ S kann aus mehr Komponenten bestehen.

Undersampled

- ⦿ Der Rest von der Oberfläche ist „Undersampled“ .
- ⦿ Unser Ziel ist, die Grenze(boundary) zwischen well-sampled und undersampled Patches zu finden.

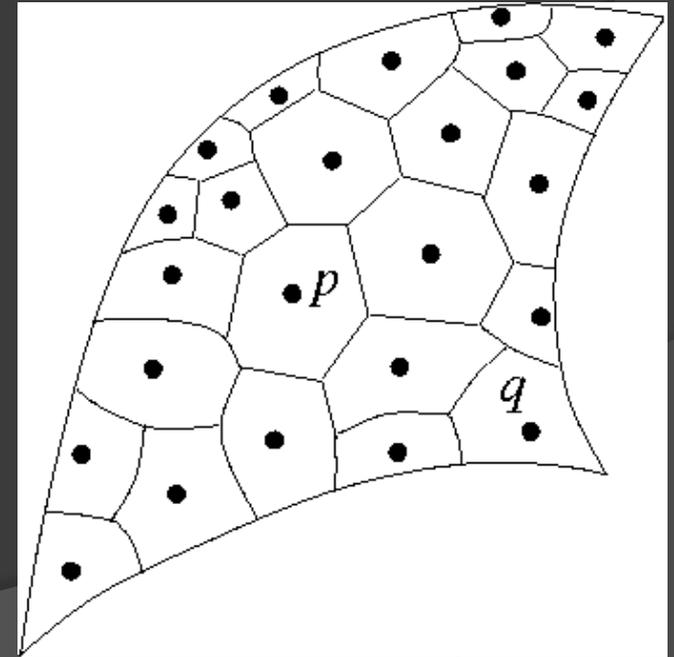
Undersampled

- ⦿ Gebiet mit Hohe Krümmung ist häufig undersampled



Interior und boundary Punkte

- ⊙ $V_{p,s}$: Restricted Voronoi cells auf Oberfläche S
- ⊙ p ist “innen”(interior), wenn $V_{p,s}$ keinen Boundary-Punkte beinhaltet.
- ⊙ Sonst ist p boundary.

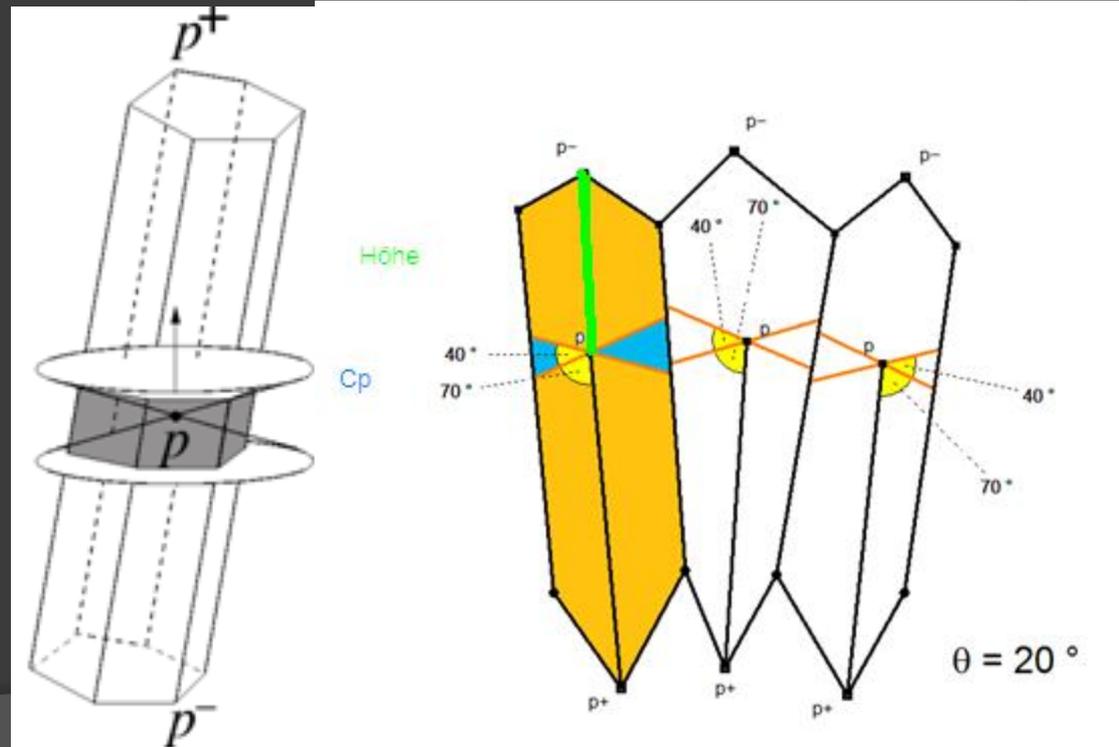


Radius und Höhe

- Der Radius r_p von Voronoi cell V_p wird als der Radius von C_p definiert,

$$\text{d.h. } r_p = \max \{ |y-p| : y \in C_p \}$$

- Höhe h_p von V_p :
Distanz
zwischen $|p^--p|$



Cocone-Neighbors

- Cocone-Nachbar: $N_p = \{q \in P: C_p \cap V_q \neq \emptyset\}$, N_p ist Menge der Cocone Nachbarn, C_p ist Cocone für p .

Flatness

- ⊙ Ein Punkt p ist flat, wenn
 - ⊙ 1. Ratio condition: $r(p) \leq \rho h(p)$
 - V_p ist lang und dünn in Richtung $v(p)$.
 - ⊙ 2. Normal condition: $\angle\{v(p), v(q)\} \leq \alpha \quad \forall q$
with $p \in N_q$
 - die Richtung der Dehnung von V_p trifft cocone Nachbar von p .

Index

- ⦿ Einführung
- ⦿ Wechtige Vorkenntnisse
- ⦿ Theorie zu Algorithmus
- ⦿ Algorithmus
- ⦿ Rekonstruktion

Theorie zu Algorithmus

Definition: Ein interior Punkt ist “*deep*”, wenn es kein Cocone-Nachbar eines Boundary Punktes ist.

Theorem 1: Alle deep interior Punkte sind flat.

Theorem 2: Boundary Punkte sind nicht flat.

Theorem 3: Boundary() entdeckt alle und nur boundary Punkte.

Index

- ⦿ Einführung
- ⦿ Wechtige Vorkenntnisse
- ⦿ Theorie zu Algorithmus
- ⦿ **Algorithmus**
- ⦿ Rekonstruktion

Boundary Detection(erste phase)

IsFlat(p, ρ, α)

check ratio and normal condition for V_p ;

if $r(p) \leq \rho h(p)$

 if $\angle v(p), v(q) \leq \alpha$

 return true

else

 return false

end

Boundary Detection(zweite phase)

```
Boundary(P,ρ,α)
  R:=∅
  for all p∈P
    if IsFlat(p)
      R:=R∪p
    endif
  endfor
  while ∃p∉R and p∈Nq with q∈R
    and r(p)≤ρh(p) and ∠v(p),v(q) ≤α
    R:=R∪p;
  endwhile
  return P\R
end
```

Index

- ⦿ Einführung
- ⦿ Wechtige Vorkenntnisse
- ⦿ Theorie zu Algorithmus
- ⦿ Algorithmus
- ⦿ **Rekonstruktion**

Reconstruction

Cocone(P, ρ, α)

 Compute V_p ;

$B := \text{Boundary}(P, \rho, \alpha)$

 for each $p \in P$

 if $p \notin B$

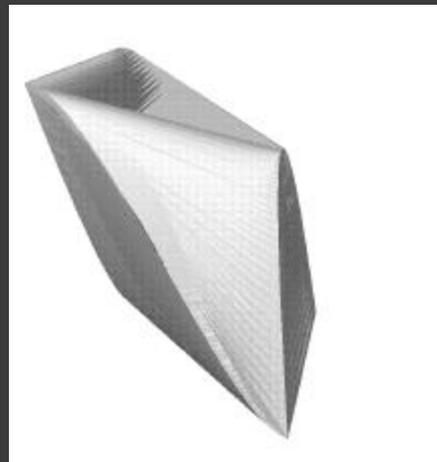
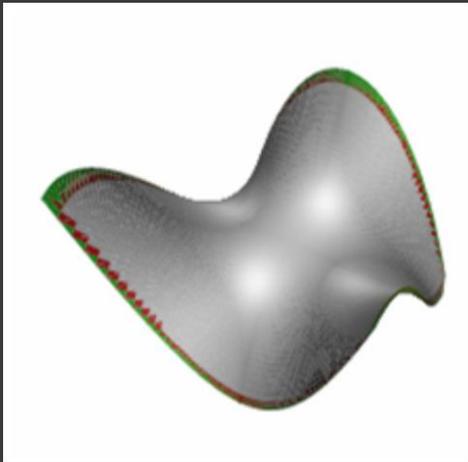
 compute T of triangles with duals intersecting C_p ;

 endif

 enfor

 Extract manifold from T (using pruning and walking as mentioned and detailed);

end



Fuß und
Reitsattel
mit Hilfe
Boundary

Timings (Exact-Double)

Name	Delaunay	Boundary	Reconstruction
Cactus	130-	40-	23-
Cat	120-50	108-58	68-51
Engine	831-405	108-58	79-59
foot	233-99	224-119	138-103
Mannequin	135-57	137-73	85-64
Oilpump	409-153	314-166	197-141
Club	171-74	179-95	112-84

Timings

Name	#points	#triangles	Reconstruction(sec.)
Halfsphere	245	486	0.58
Mannequin	12772	25339	74
Foot	20021	39995	122
Oilpump	30931	61548	194
Monkeysaddle	10000	19596	345

PIII, 933Mhz, 512MB

Umgebung

- ⦿ Die Theorie basiert auf die Annahme, dass die Oberfläche glatt ist.
- ⦿ Kann aber auch die raue Oberfläche analysieren. Es wird als eine glatte Oberfläche geschätzt, welche die fehlende Punkte hat.
- ⦿ Wie fast alle andere Cocone-Algorithmus, funktioniert nicht so gut wenn Noise stark ist.