Adaptive Manifold Fitting

Lecture 4 - February 3, 2009 - 2-3 PM

Outline

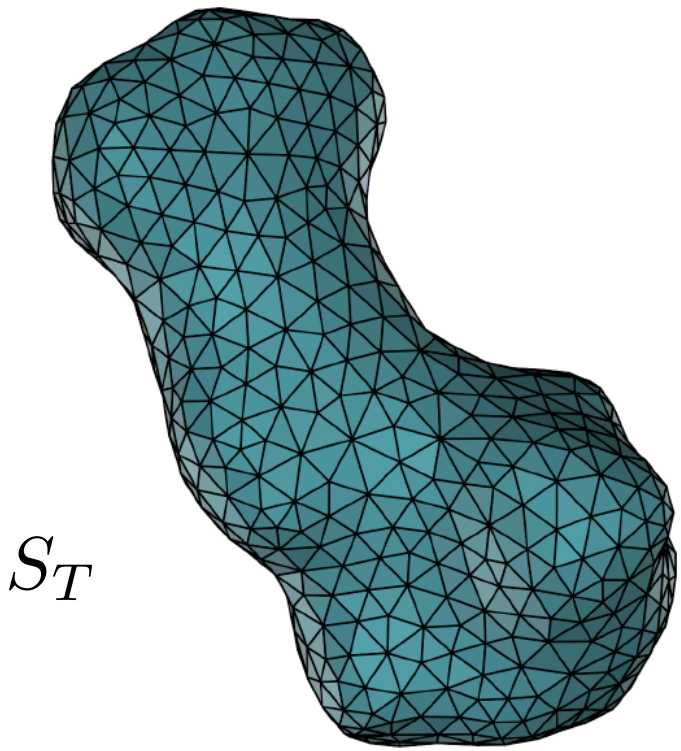
Outline

- Fitting Surfaces to Very Large Meshes
- Multiresolution Operators
- Building Base Meshes
- Mesh Refinement
- Adaptive Manifold Fitting
- Conclusions

The Surface Fitting Problem

The Surface Fitting Problem

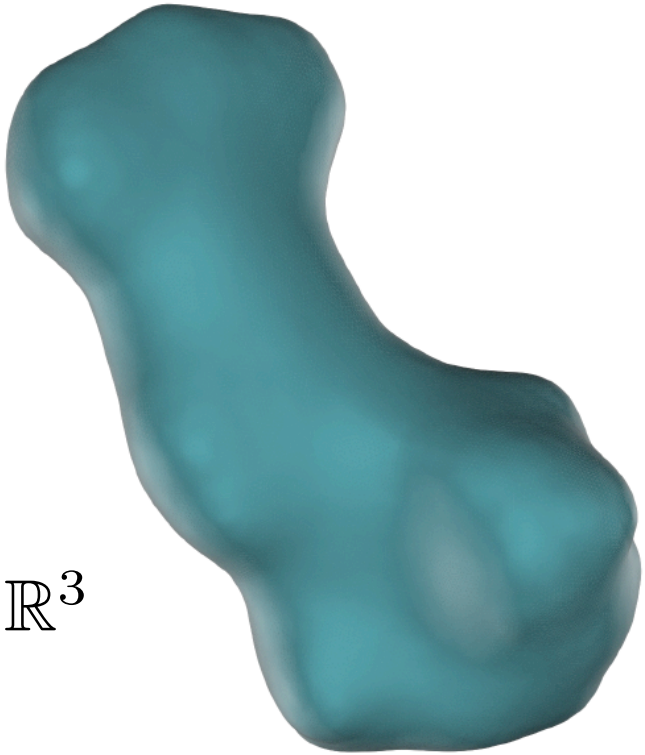
We are a given a piecewise-linear surface, S_T, in $\mathbb{R}^3$, with an empty boundary, a positive integer k, and a positive number ϵ, ...



The Surface Fitting Problem

The Surface Fitting Problem

We want to find a C^k surface $S \subset \mathbb{R}^3$ ...



$S \subset \mathbb{R}^3$

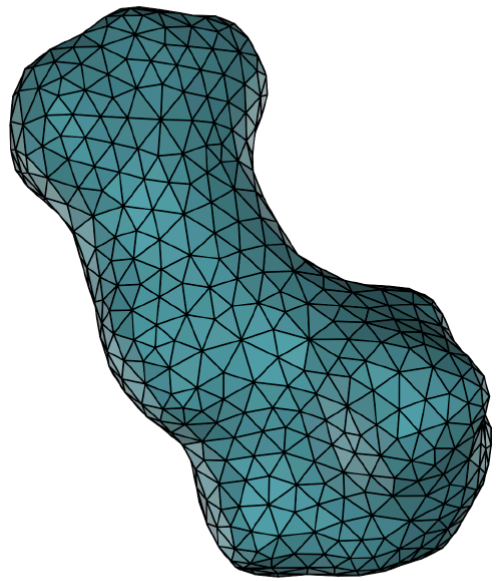
The Surface Fitting Problem

The Surface Fitting Problem

such that there exists a homeomorphism, $h : S \to |S_T|$, satisfying

$$\|h(v) - v\| \leq \epsilon \,,$$

for every vertex v of S_T.

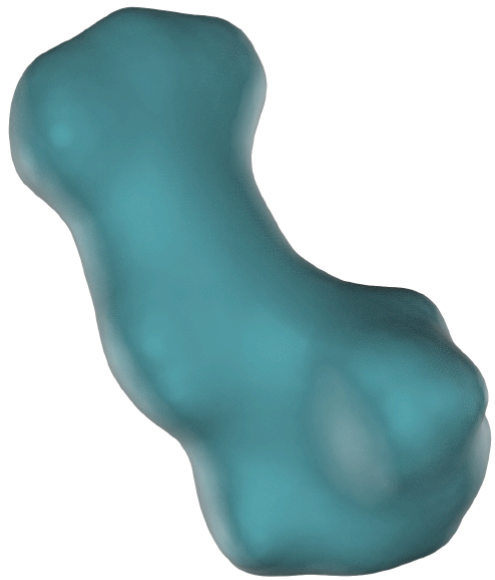


The Surface Fitting Problem

such that there exists a homeomorphism, $h : S \to |S_T|$, satisfying

$$\|h(v) - v\| \leq \epsilon \,,$$

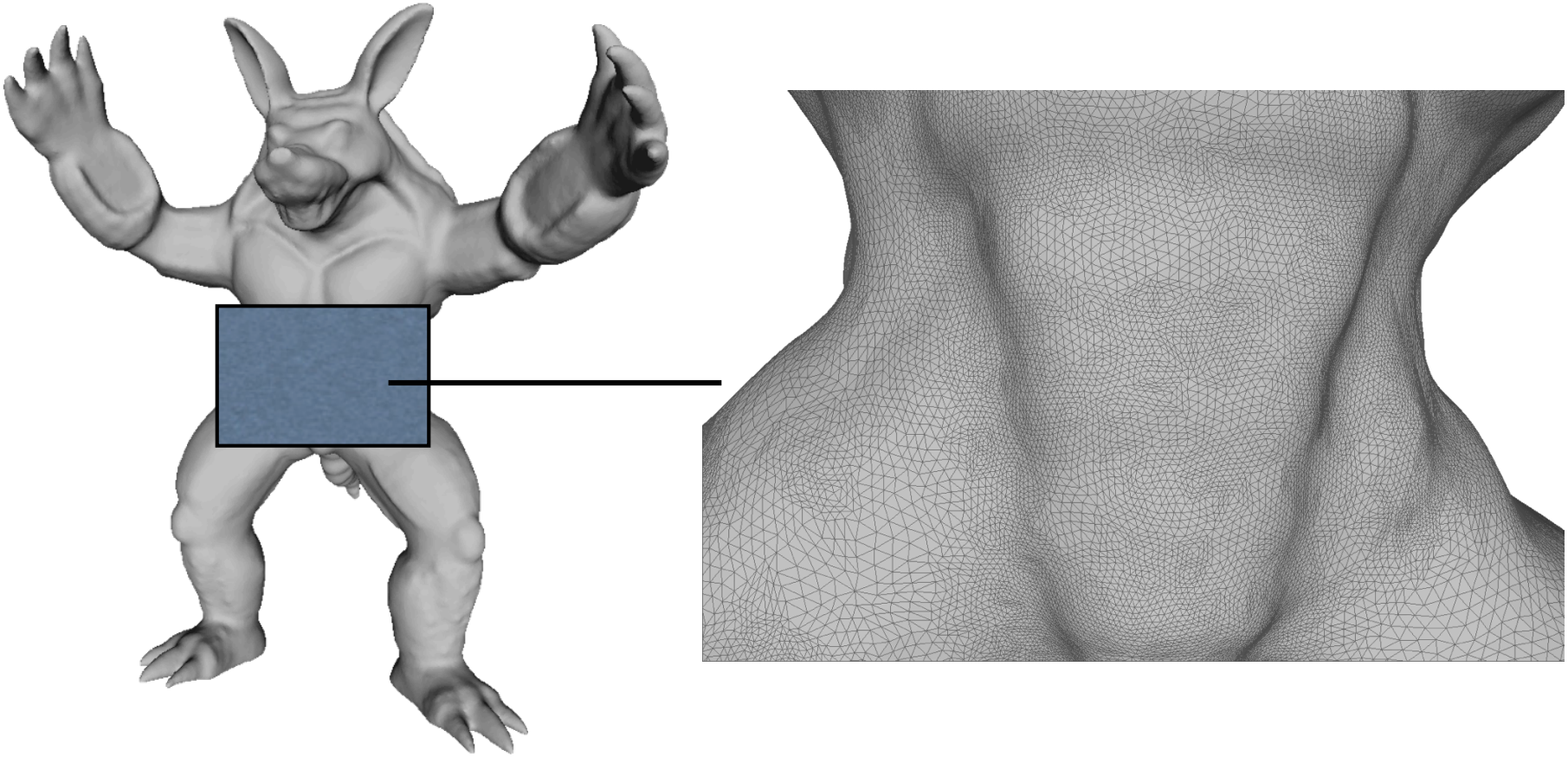
for every vertex v of S_T.



Surface Fitting

Surface Fitting

- Very Large Meshes (10^6 vertices)
  - Challenging Problem!



Manifolds and Fitting

Manifolds and Fitting

- Basic Setting
  - Gluing Data proportional to Mesh Size

- Problem: *Very Large Meshes*
  - Computationally Inefficient
  - Do not Exploit Approximation Power
- Solution:
  - Adaptation

Adaptive Fitting

Adaptive Fitting

- Optimization Formulation:
  - Given an Approximation Error ϵ
  - Find $\mathcal{M}$ with Smallest Number of Charts

- Strategy:
  - Combine
    - Multiresolution Structure
    - Manifold Surface Approximation

Multiresolution Framework

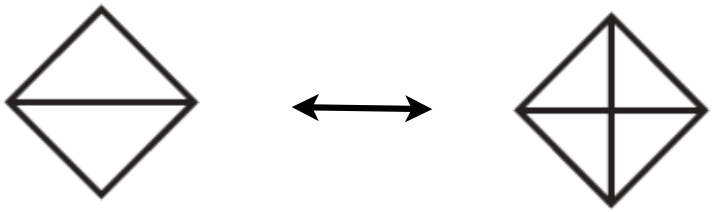
Multiresolution Framework

- Simplicial Multi-triangulation
  - Stellar Theory
- Building Base Meshes
  - Surface Simplification
- Adaptive Fitting
  - *4-8* Refinement

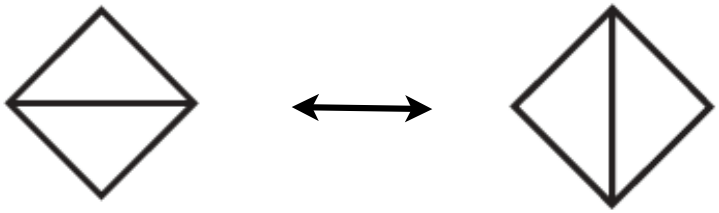
Stellar Theory

Stellar Theory

- Topological Operators
- Edge Split and Weld
  - Change Mesh Resolution



- Edge Flip
  - Change Mesh Connectivity



Stellar Simplification

Stellar Simplification

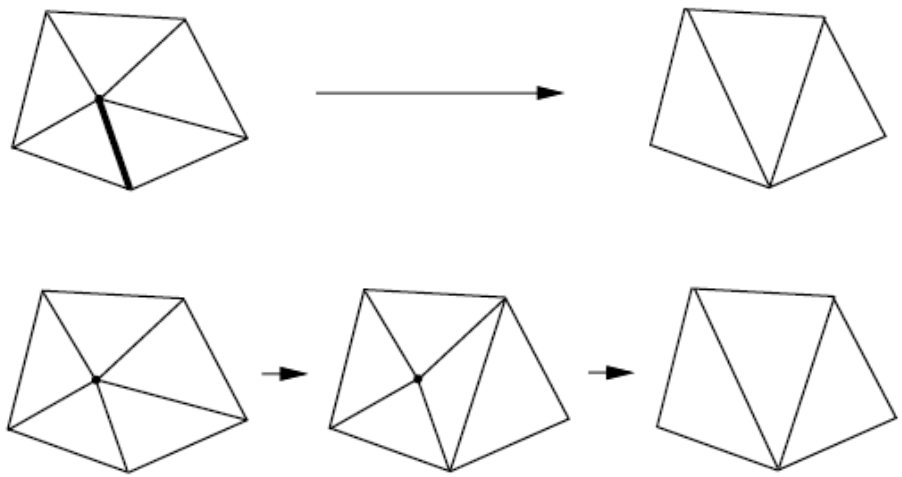
- Basic Elements:

I. Operator Factorization

- Edge Collapse

↕

- Flip + Weld



II. Quadric Error Metric

Basic Algorithm

Basic Algorithm

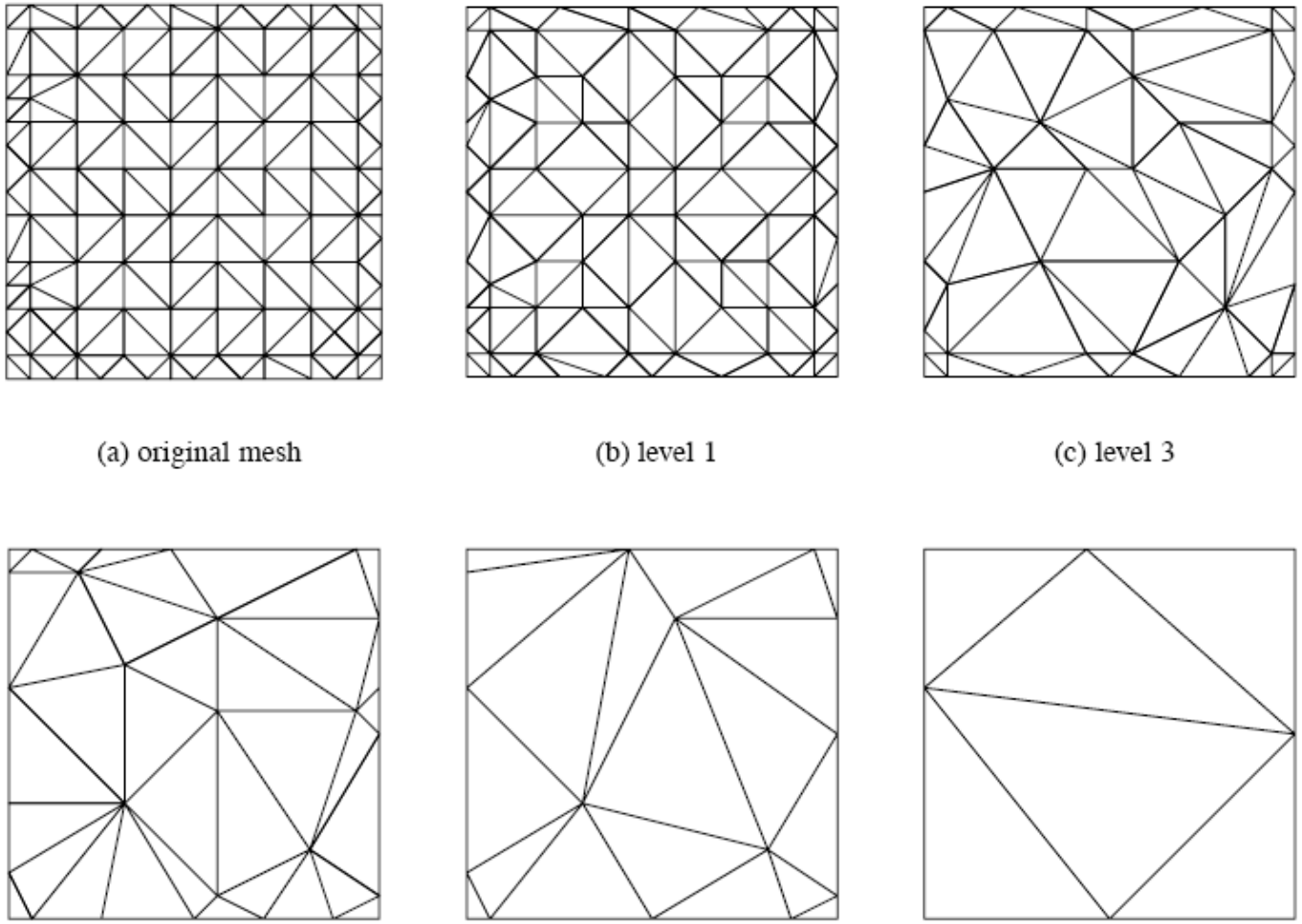
- Repeat for *N* Resolution Levels
  1. Rank Vertices Based on Quadric Error
  2. Select Independent Set of Clusters
  3. Simplify Mesh using Stellar Operators

✳ *Properties*

- *Logarithmic Height*
- *Good Aspect Ratios*

Example 1: Plane

Example 1: Plane

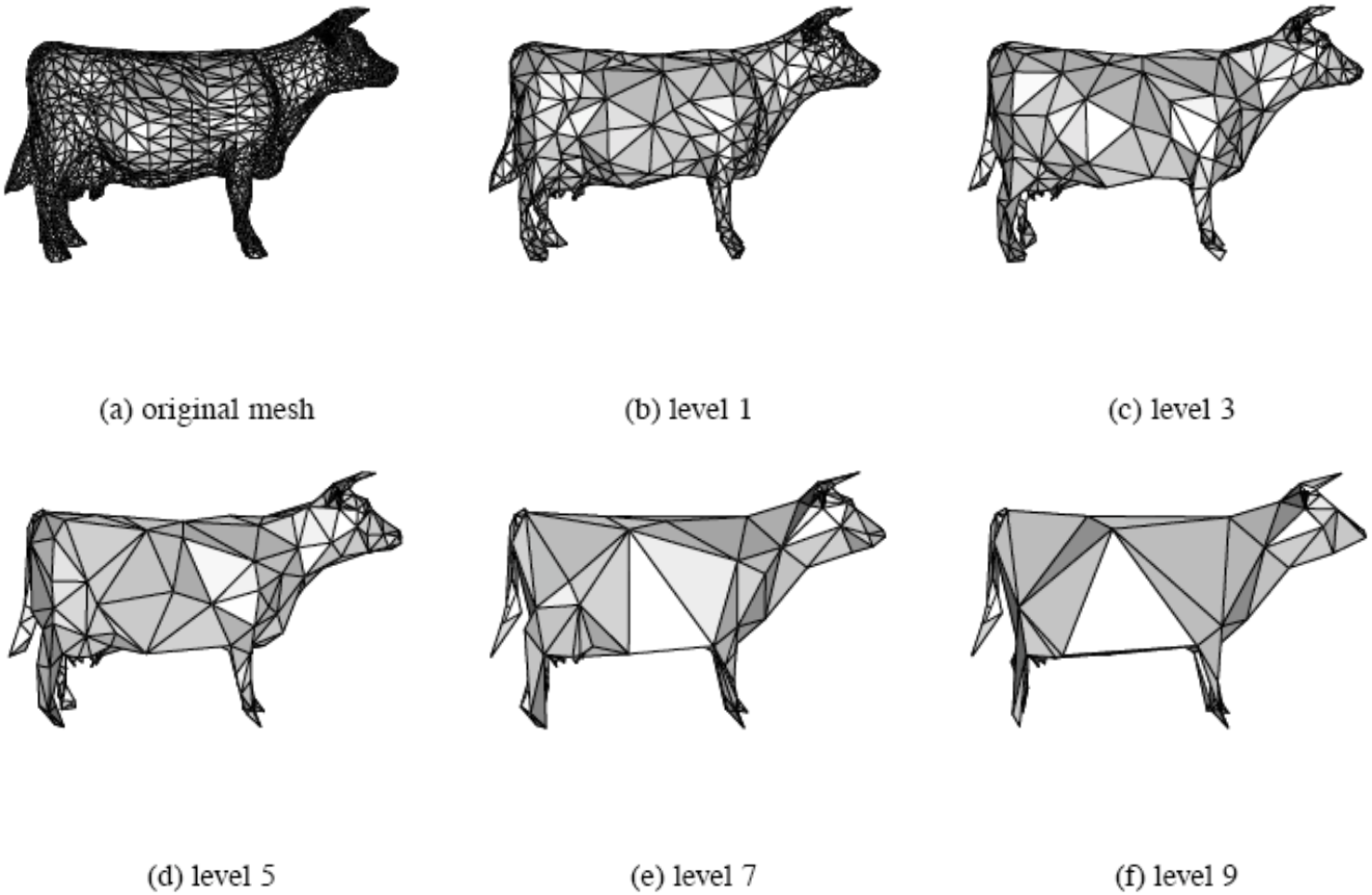


(a) original mesh (b) level 1 (c) level 3

(d) level 5 (e) level 7 (f) level 9

Example 2: Cow

Example 2: Cow



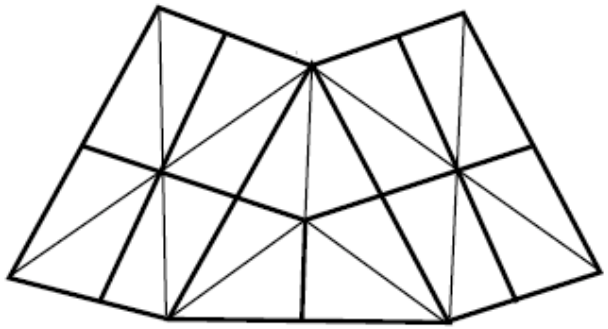
(a) original mesh (b) level 1 (c) level 3

(d) level 5 (e) level 7 (f) level 9

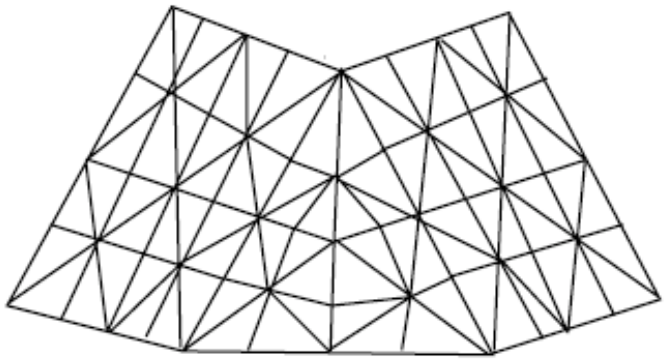
Variable Resolution Mesh

Variable Resolution Mesh

- Underlying Semi-Regular Structure
  - Tri-quad Base Mesh



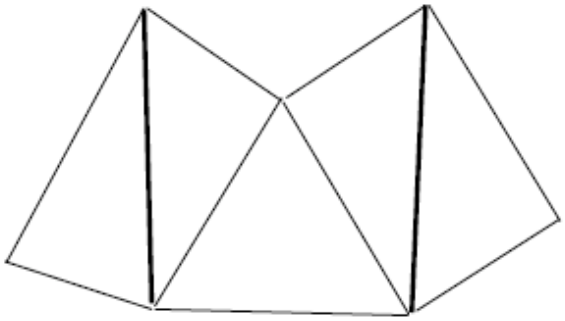
  - *4-8* Subdivision



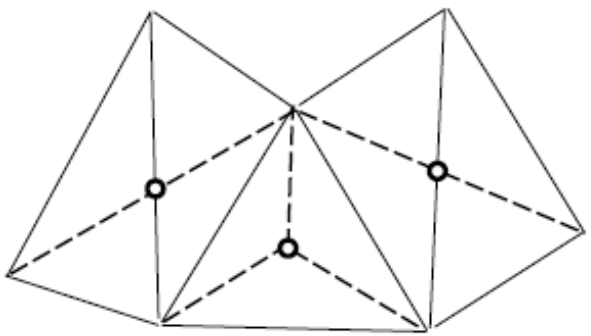
Building the Base Mesh

Building the Base Mesh

1. Two-Face Clusters + Single Triangles



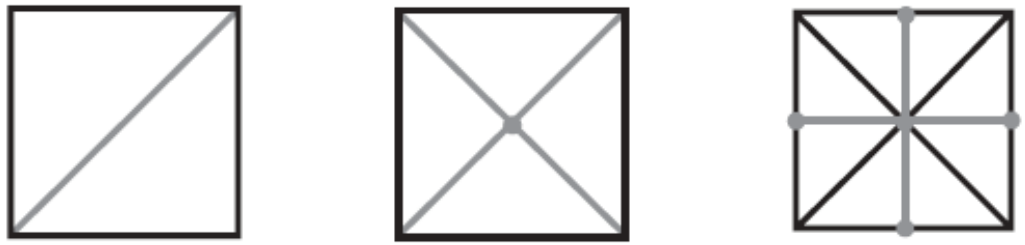
2. Barycenter Subdivision



4-8 Subdivision

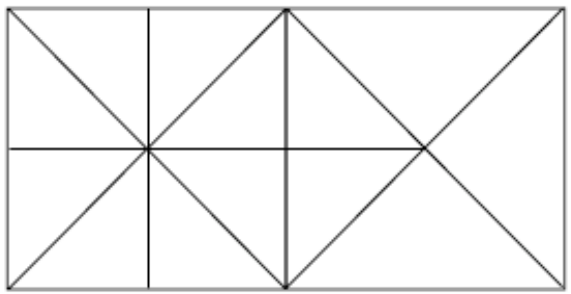
4-8 Subdivision

- Interleaved Binary Subdivision



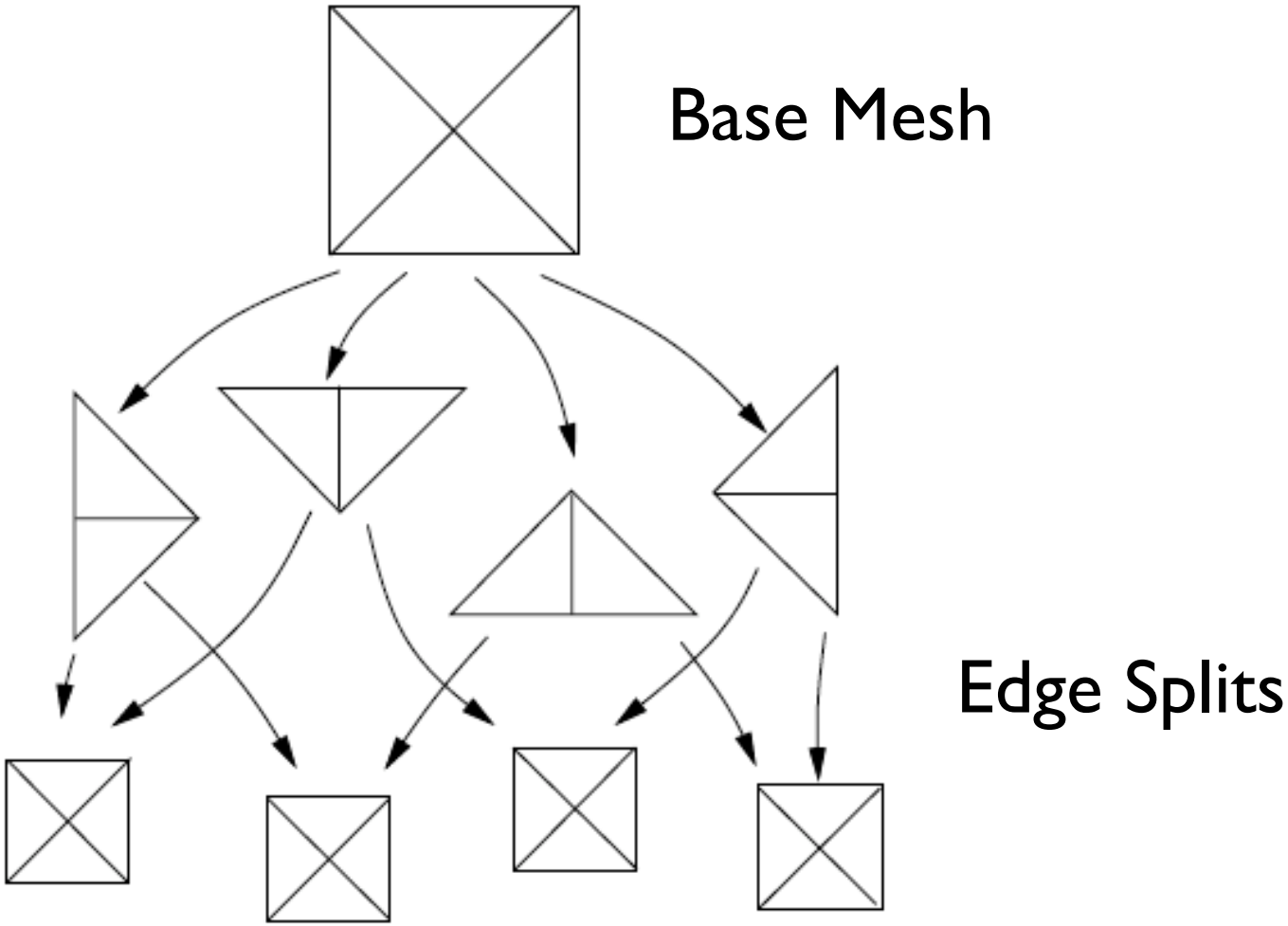
i i+1 i+2

- Non-Uniform Refinement



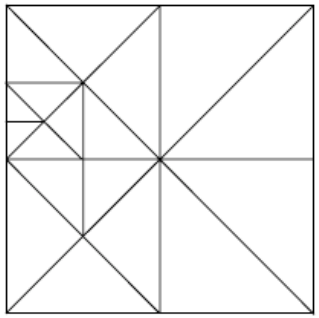
Binary Multi-Triangulation

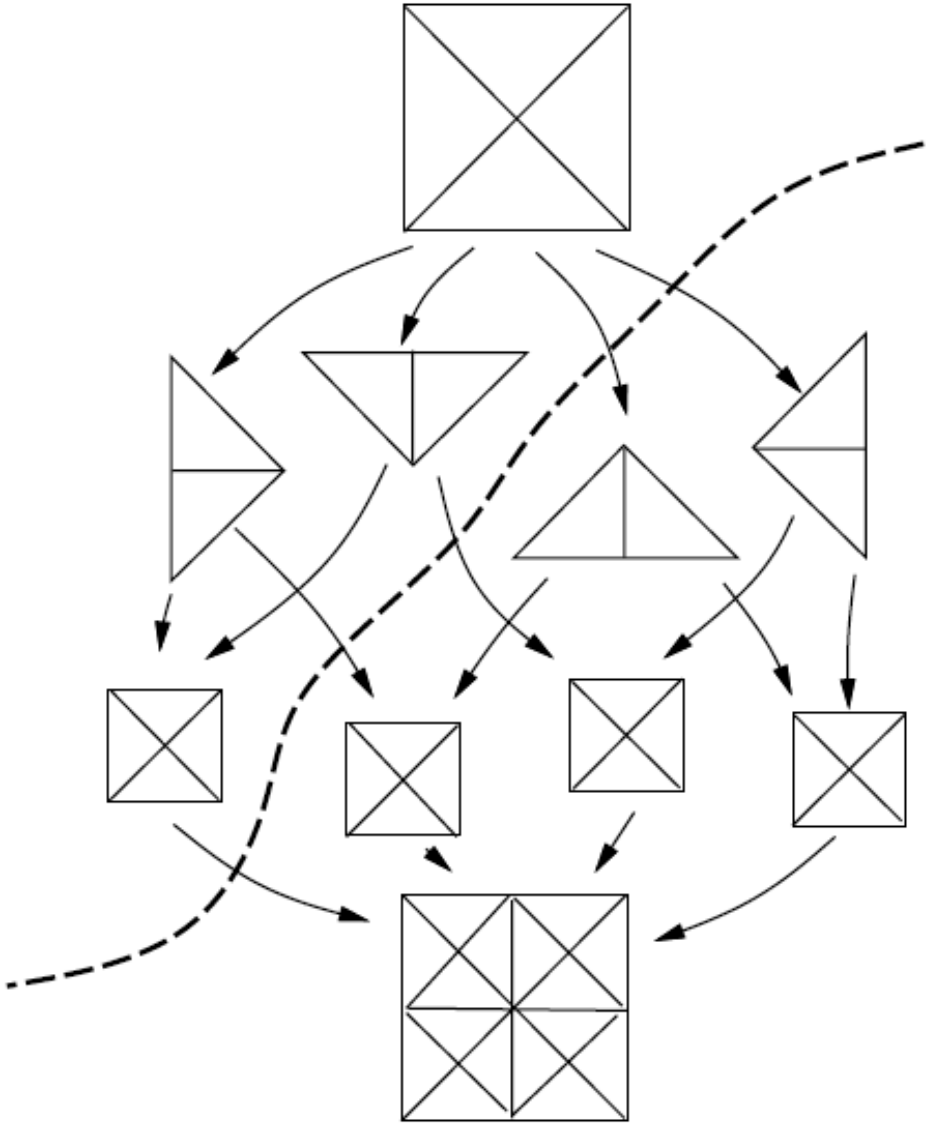
Binary Multi-Triangulation



Adaptive Refinement

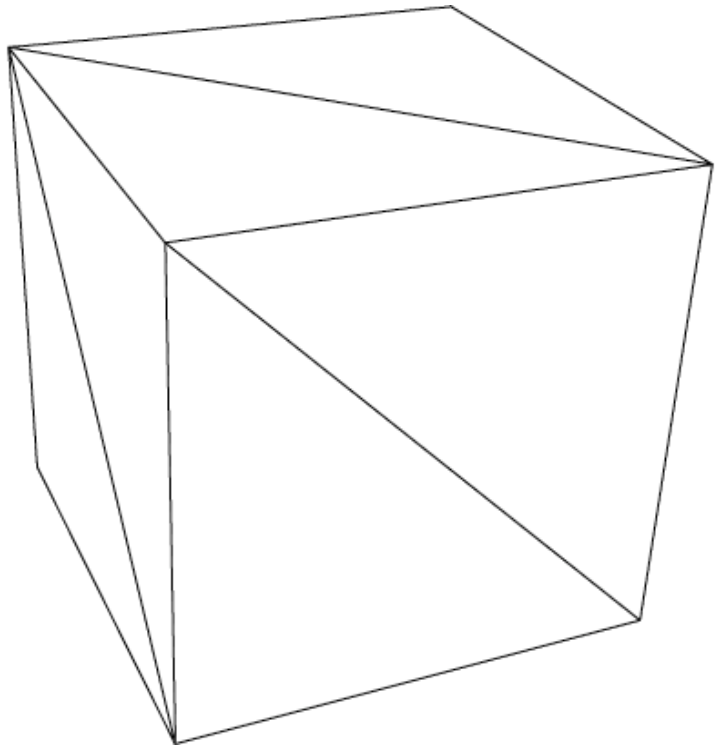
Adaptive Refinement



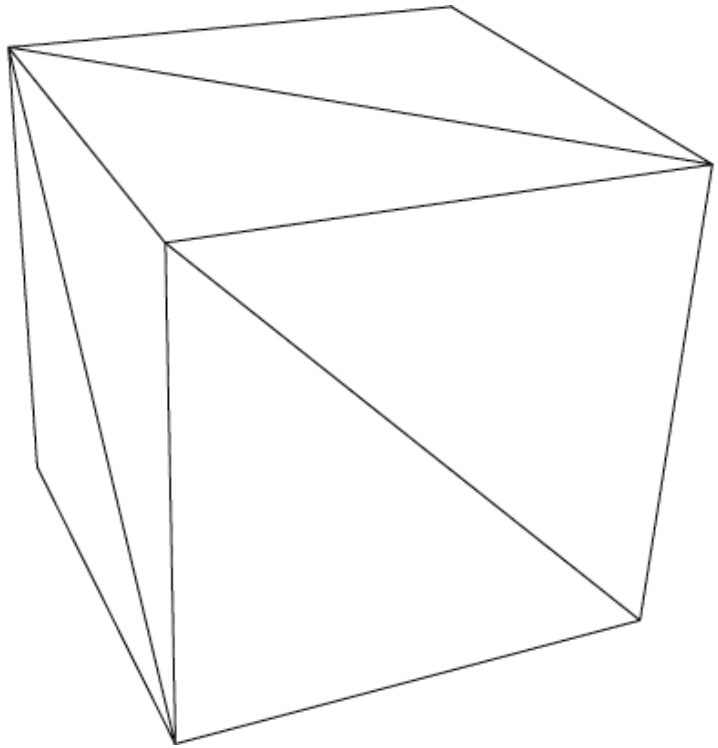


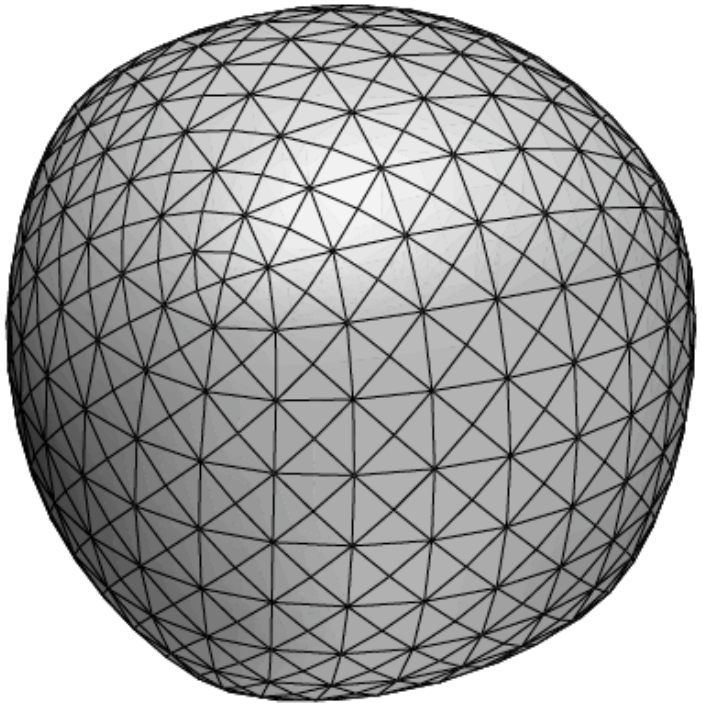
Example 1: Uniform

Example 1: Uniform



Example I: Uniform





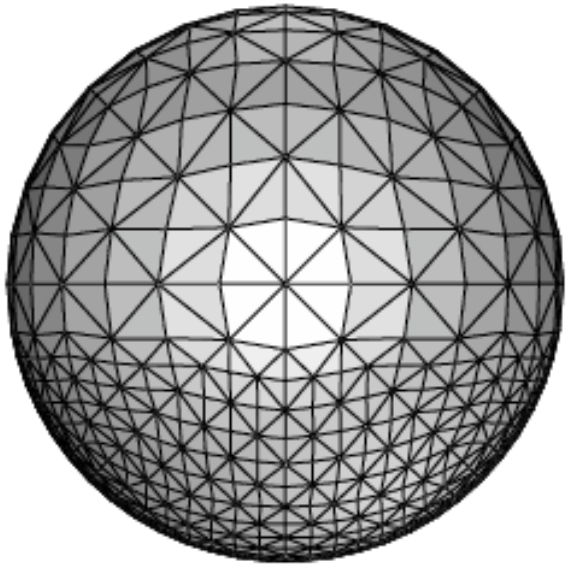
Example 2: Adaptive

Example 2: Adaptive

- Application-Dependent Criteria

Example 2: Adaptive

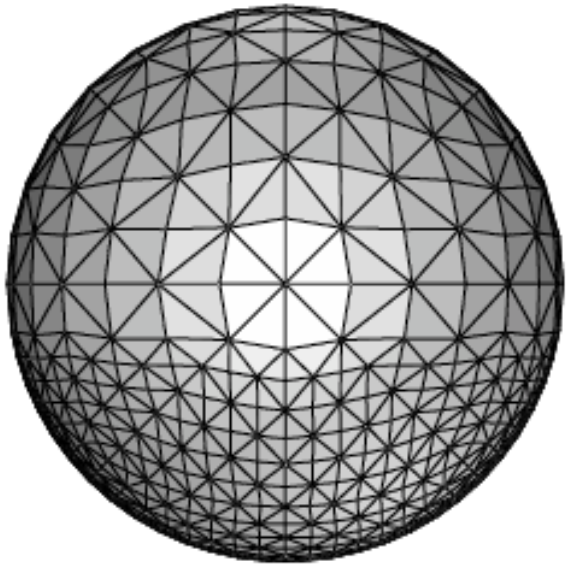
- Application-Dependent Criteria



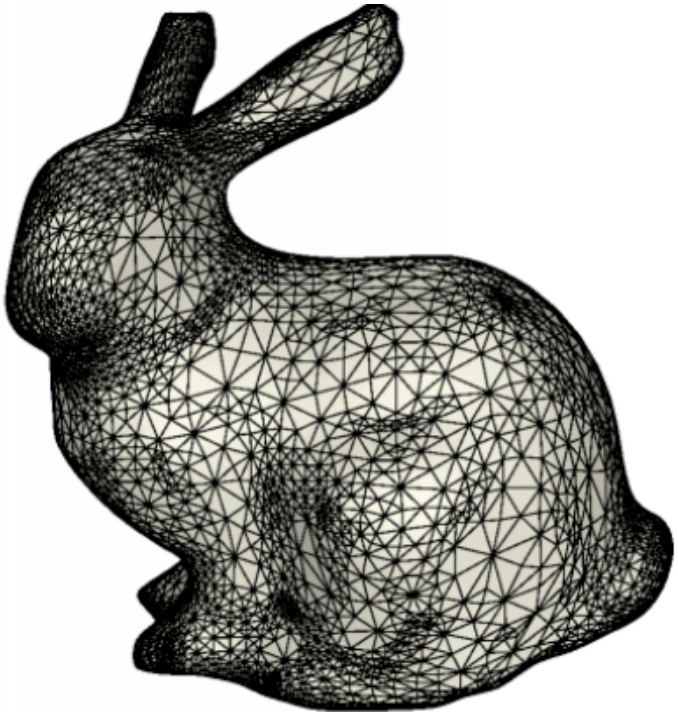
<u>Spatial Selection</u>

Example 2: Adaptive

- Application-Dependent Criteria



Spatial Selection



Curvature

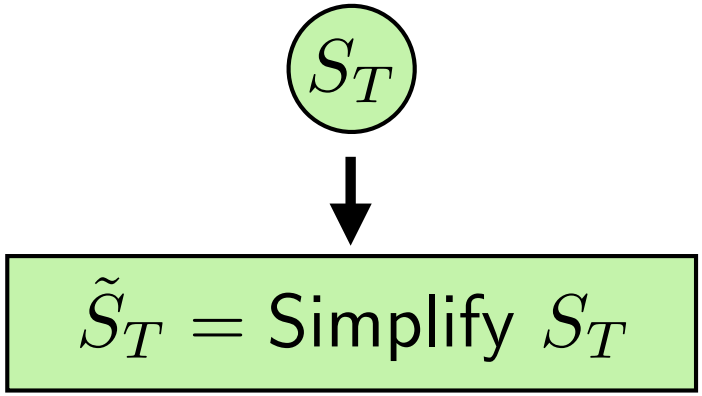
Adaptive Fitting

Adaptive Fitting

S_T

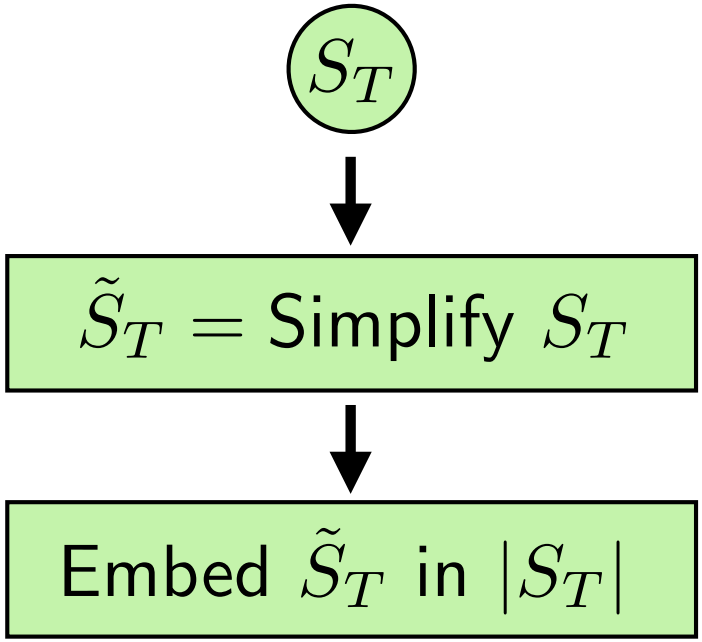
PIPELINE

Adaptive Fitting



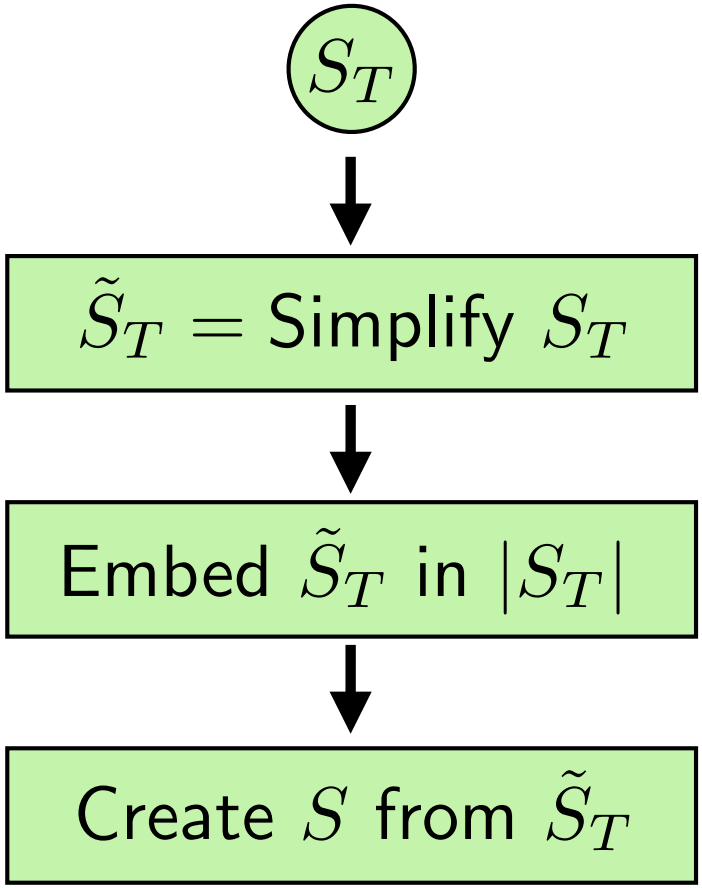
PIPELINE

Adaptive Fitting



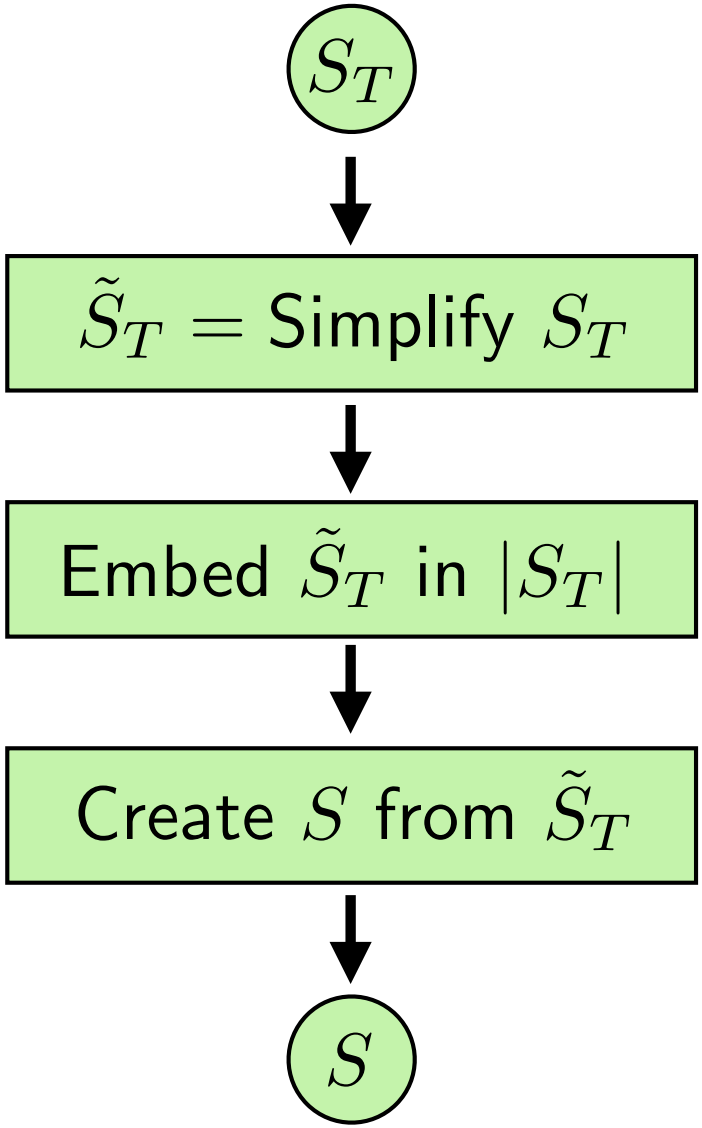
PIPELINE

Adaptive Fitting



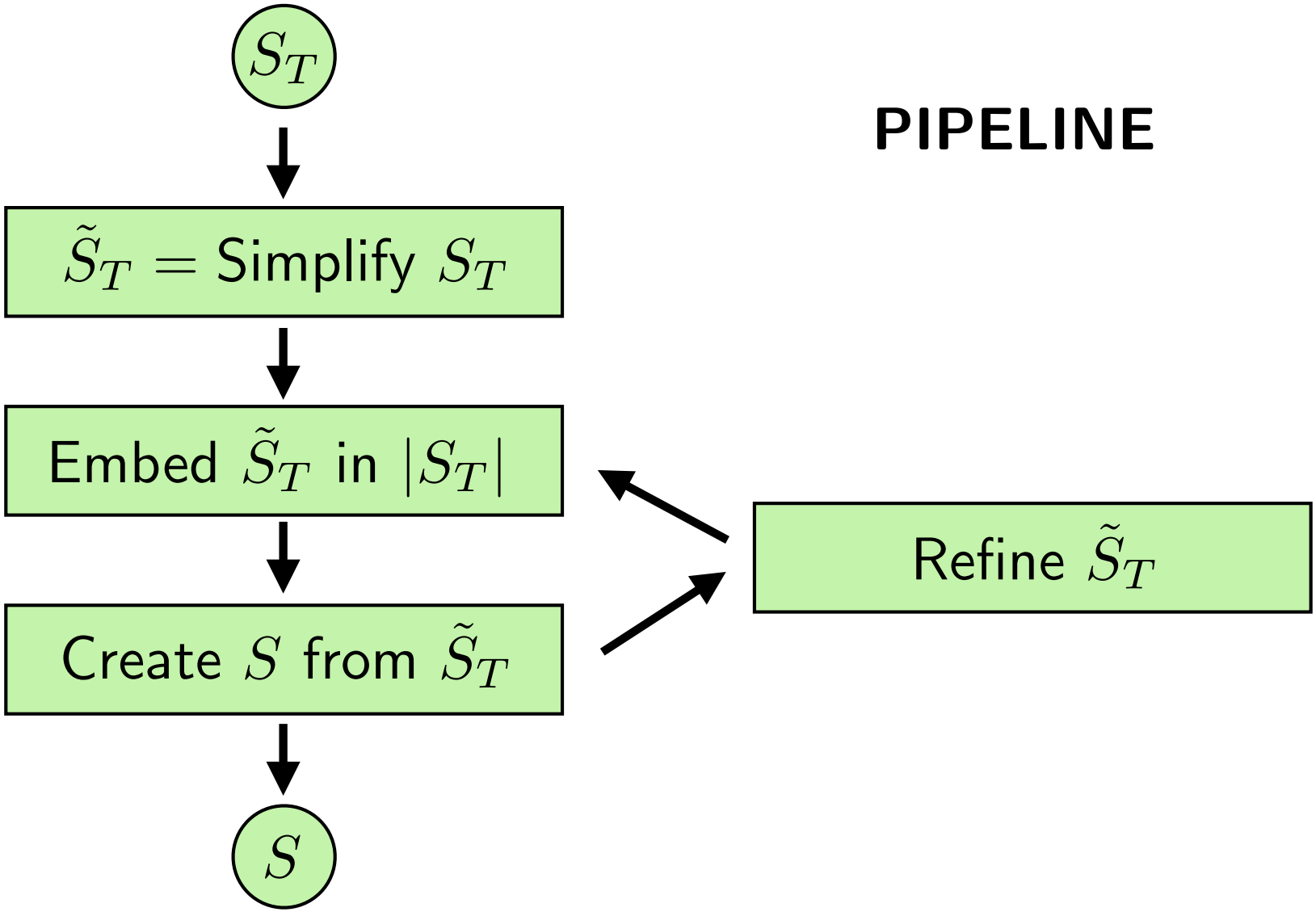
PIPELINE

Adaptive Fitting



PIPELINE

Adaptive Fitting



Adaptive Fitting

Adaptive Fitting

S_T → $\tilde{S}_T = \text{Simplify } S_T$

Adaptive Fitting

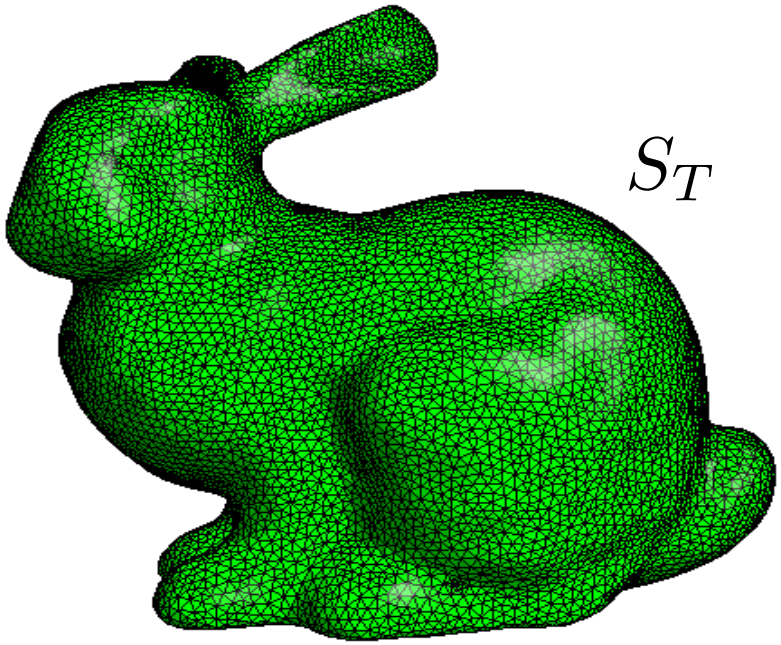
$S_T \longrightarrow \boxed{\tilde{S}_T = \text{Simplify } S_T}$

- Four-Face Clusters Algorithm

Adaptive Fitting

$S_T \longrightarrow \boxed{\tilde{S}_T = \text{Simplify } S_T}$

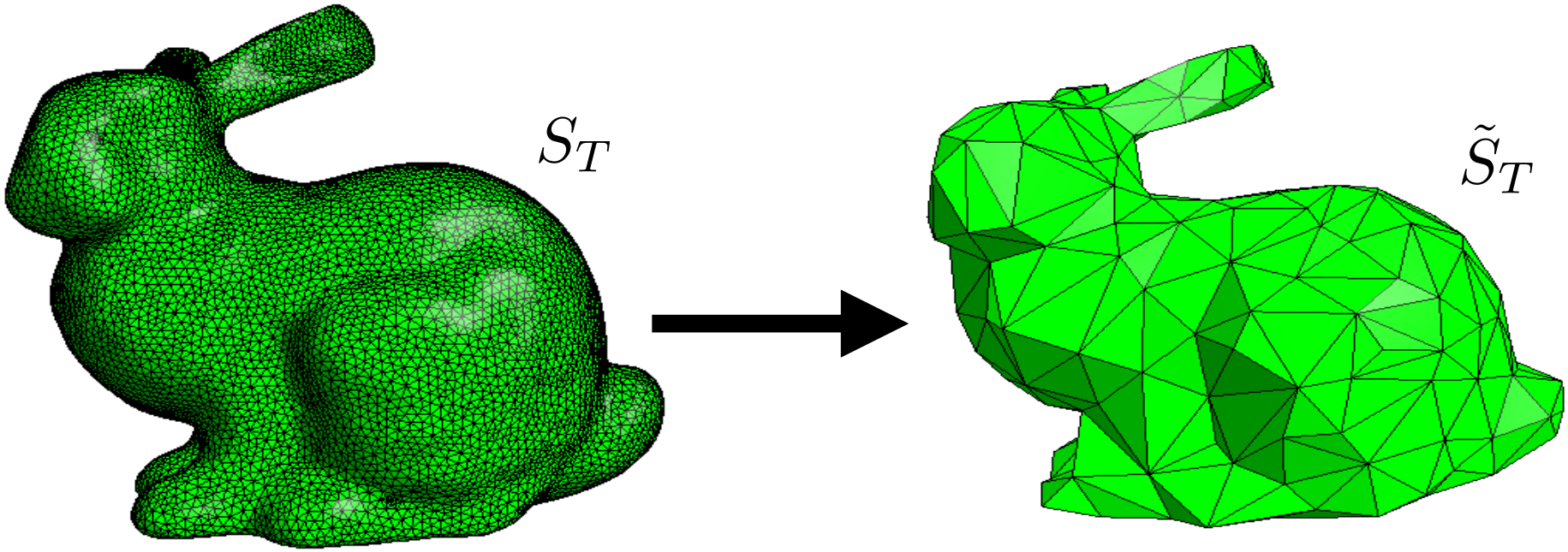
- Four-Face Clusters Algorithm



Adaptive Fitting

$S_T \longrightarrow \boxed{\tilde{S}_T = \text{Simplify } S_T}$

- Four-Face Clusters Algorithm



Adaptive Fitting

Adaptive Fitting

Embed $\tilde{S}_T$ in $|S_T|$

Adaptive Fitting

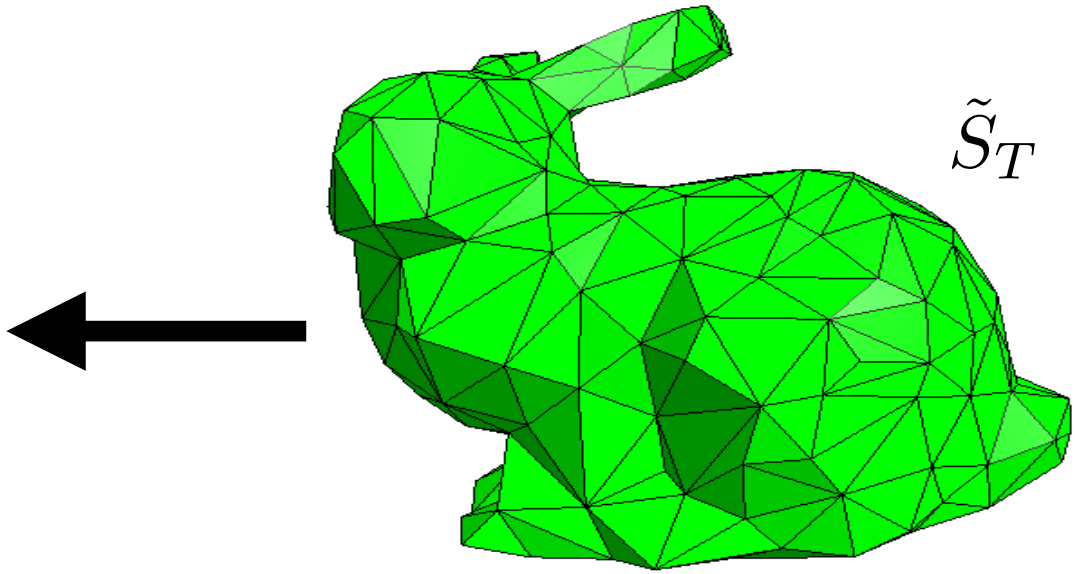
Embed $\tilde{S}_T$ in $|S_T|$

- Each edge of $\tilde{S}_T$ is embedded in $|S_T|$ as a "geodesic".

Adaptive Fitting

Embed $\tilde{S}_T$ in $|S_T|$

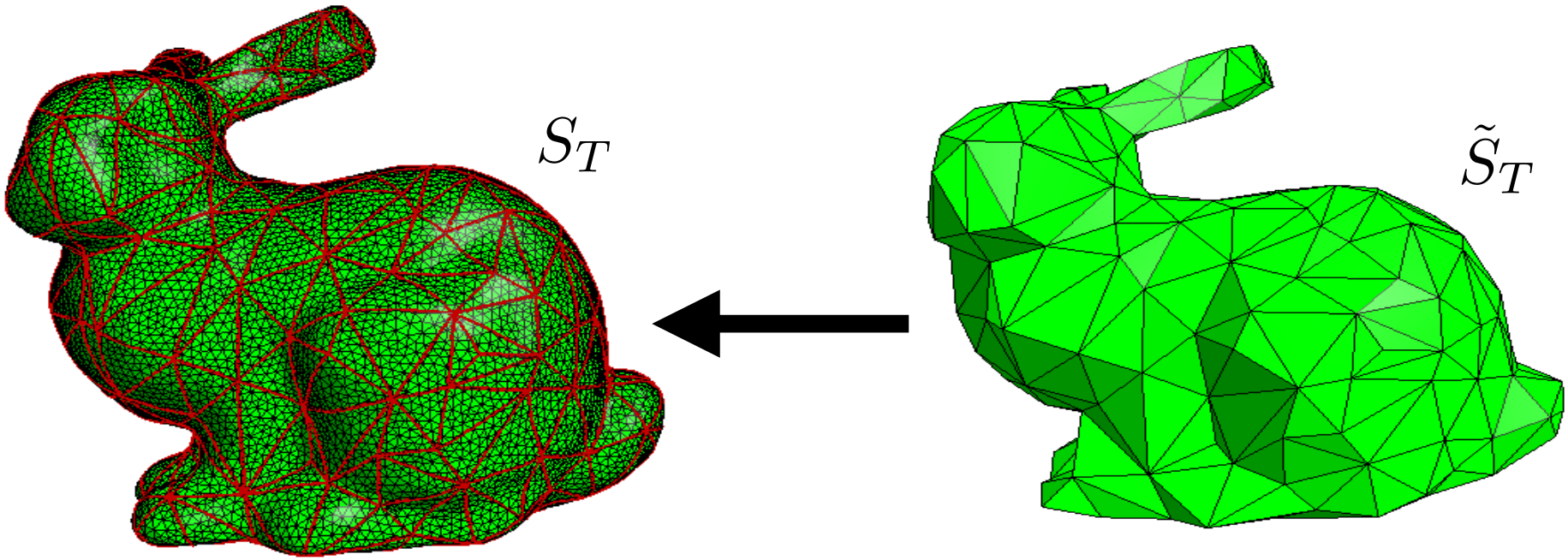
- Each edge of $\tilde{S}_T$ is embedded in $|S_T|$ as a "geodesic".



Adaptive Fitting

Embed $\tilde{S}_T$ in $|S_T|$

- Each edge of $\tilde{S}_T$ is embedded in $|S_T|$ as a "geodesic".

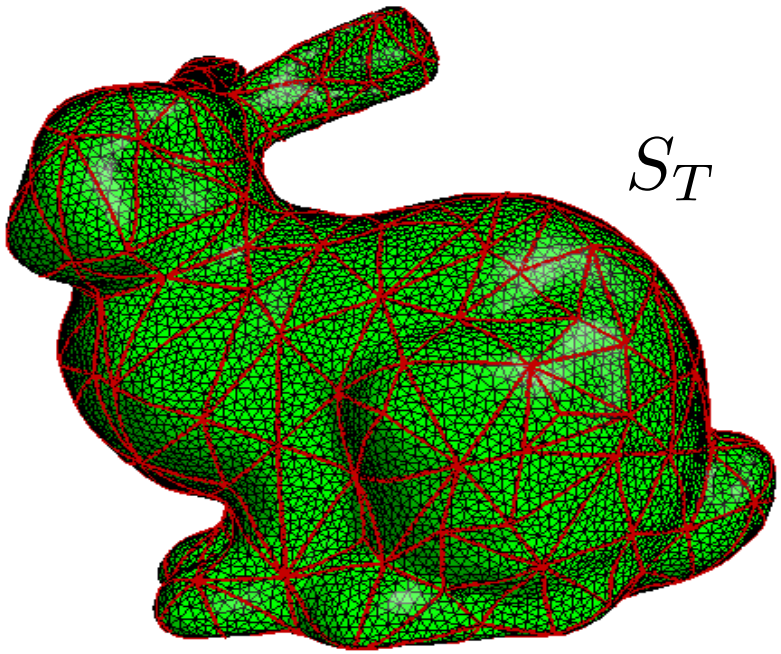


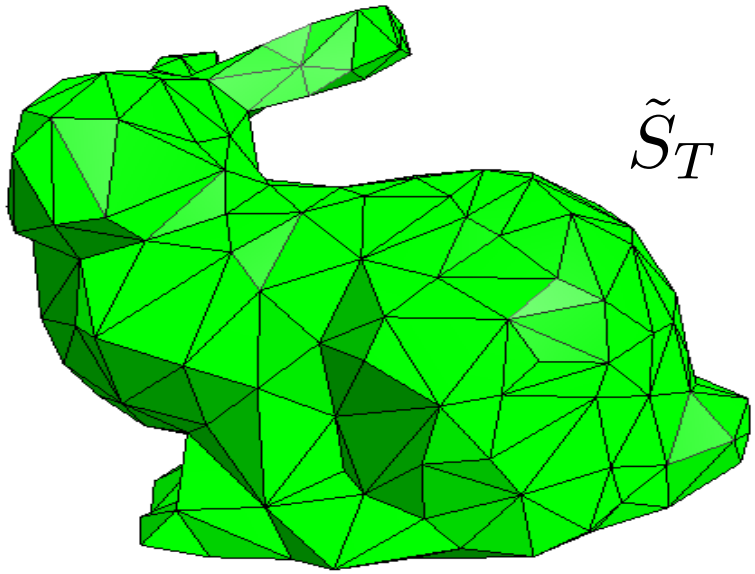
Adaptive Fitting

Adaptive Fitting

REMARK:

The vertices of $\tilde{S}_T$ ARE vertices of S_T.

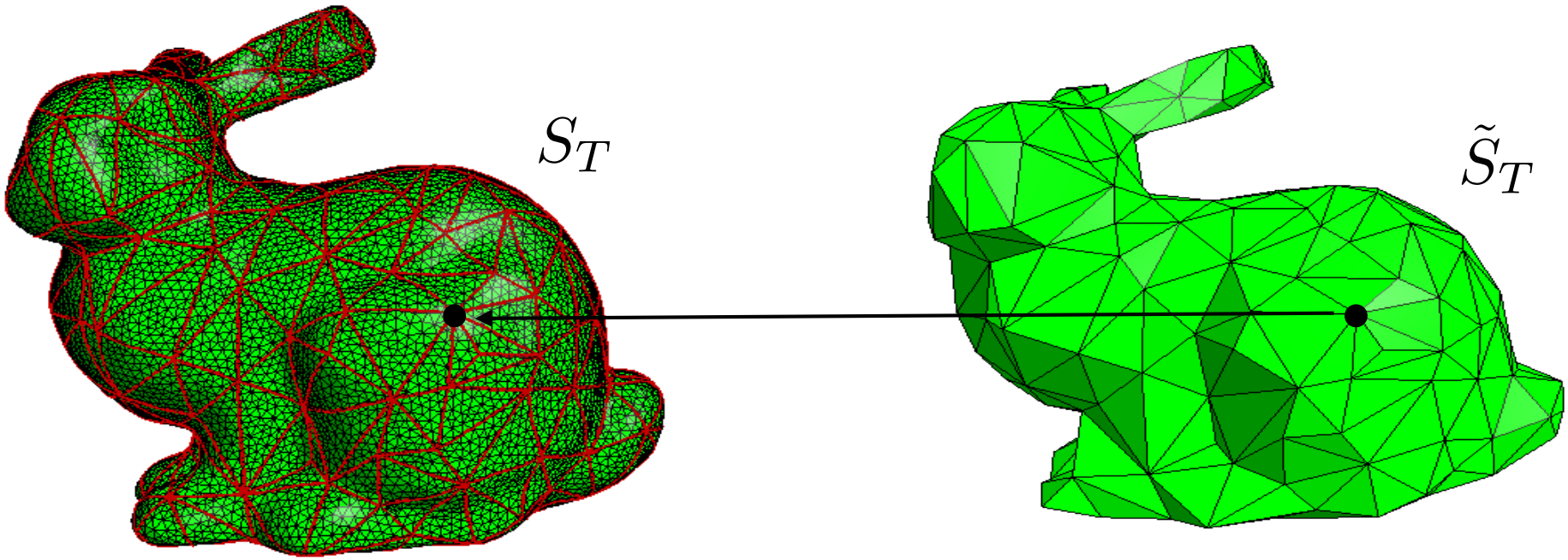




Adaptive Fitting

REMARK:

The vertices of $\tilde{S}_T$ ARE vertices of S_T.



Adaptive Fitting

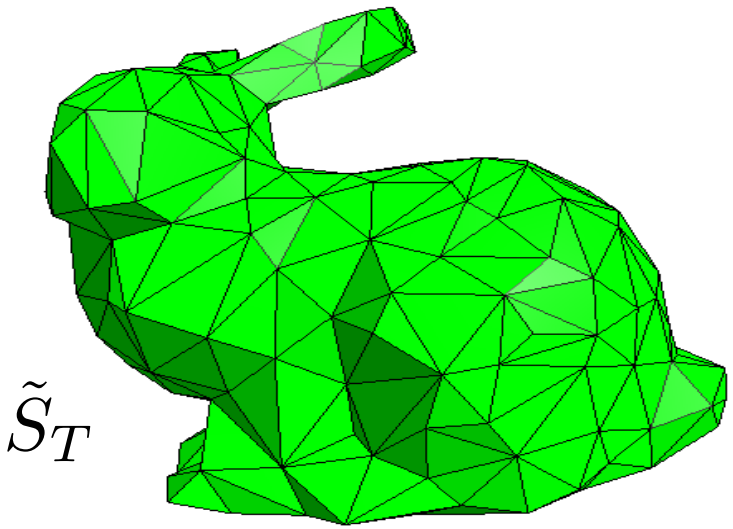
Adaptive Fitting

Create S from $\tilde{S}_T$

Adaptive Fitting

Create S from $\tilde{S}_T$

- For each vertex v of $\tilde{S}_T$, we consider the P-polygon, P_v, of v in $\mathbb{R}^2$, and the standard triangulation, T_v, of the P-polygon P_v.

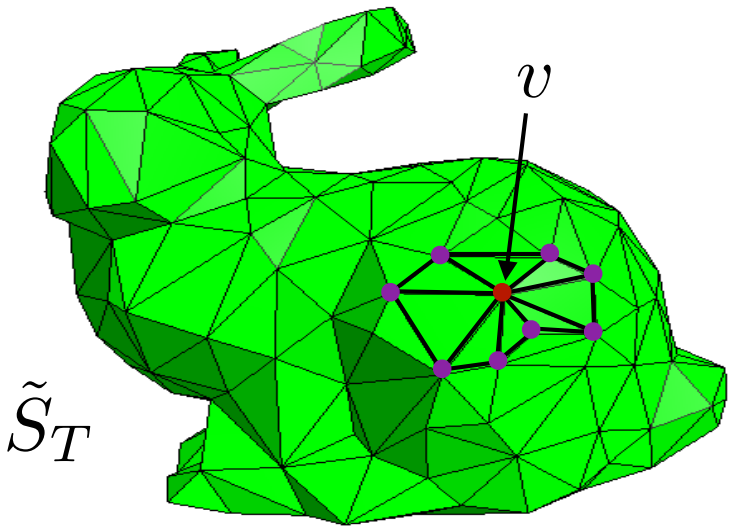


$\tilde{S}_T$

Adaptive Fitting

Create S from $\tilde{S}_T$

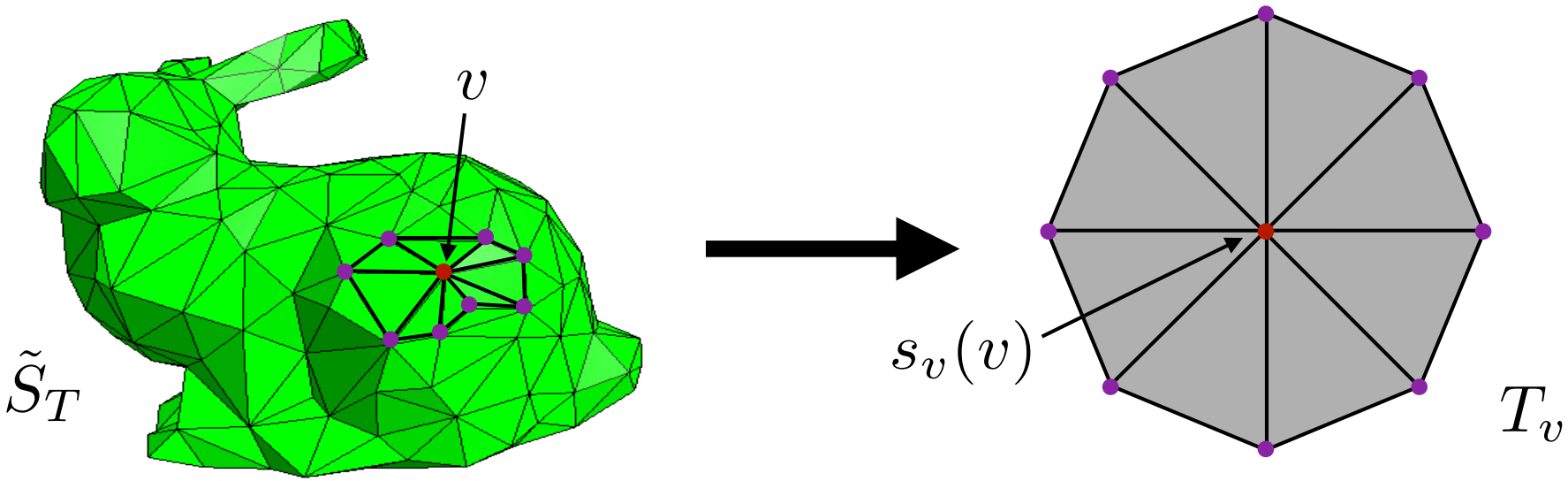
- For each vertex v of $\tilde{S}_T$, we consider the P-polygon, P_v, of v in $\mathbb{R}^2$, and the standard triangulation, T_v, of the P-polygon P_v.



Adaptive Fitting

Create S from $\tilde{S}_T$

- For each vertex v of $\tilde{S}_T$, we consider the P-polygon, P_v, of v in $\mathbb{R}^2$, and the standard triangulation, T_v, of the P-polygon P_v.



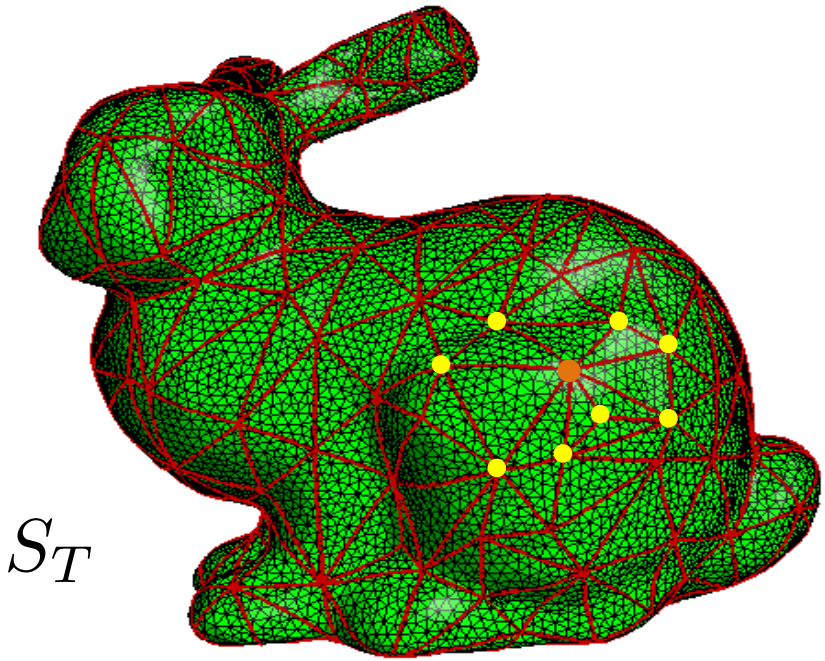
Adaptive Fitting

Create S from $\tilde{S}_T$

Adaptive Fitting

Create S from $\tilde{S}_T$

- Consider the embedding of the star, $st(v, \tilde{S}_T)$, of v in S_T.



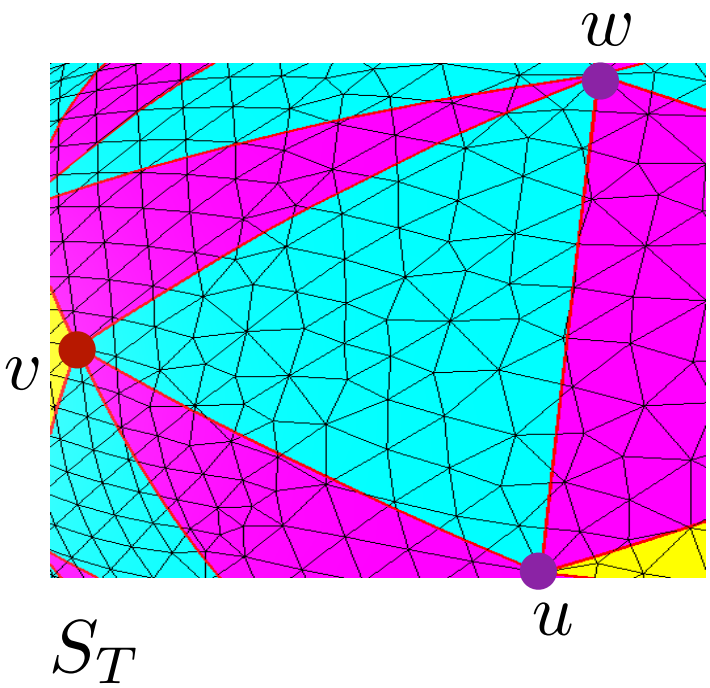
Adaptive Fitting

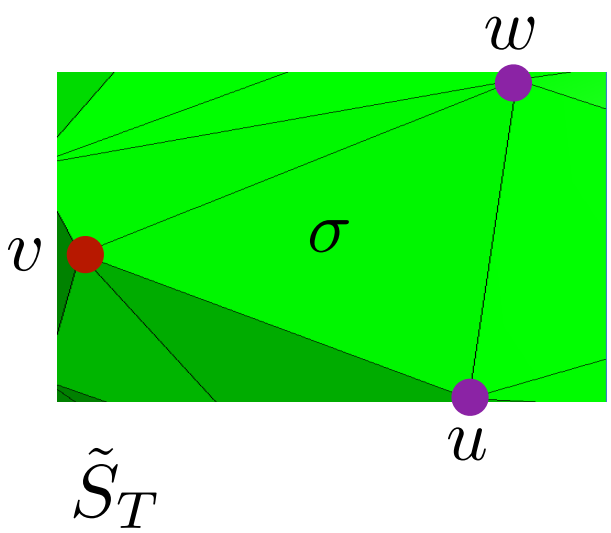
Create S from $\tilde{S}_T$

Adaptive Fitting

Create S from $\tilde{S}_T$

- Map the vertices of S_T bounded by the embedding of $st(v, \tilde{S}_T)$ to T_v.





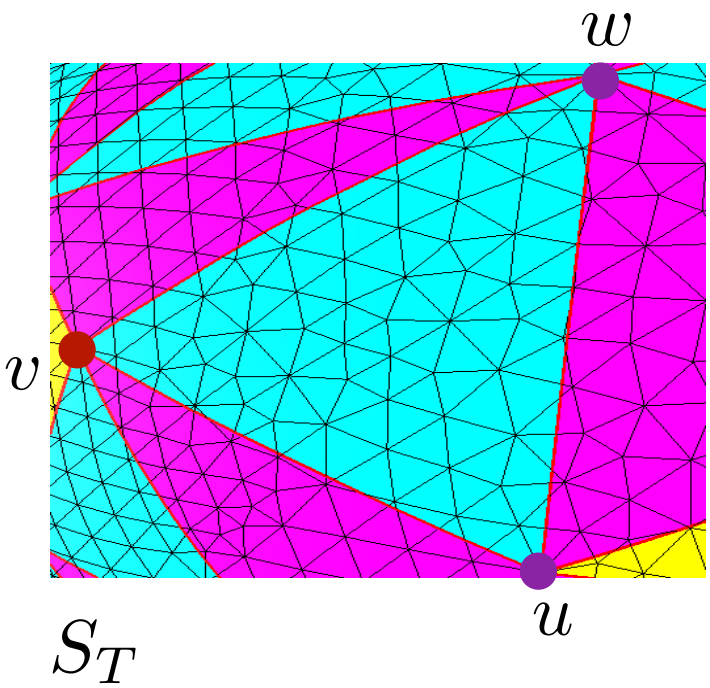
Adaptive Fitting

Create S from $\tilde{S}_T$

Adaptive Fitting

Create S from $\tilde{S}_T$

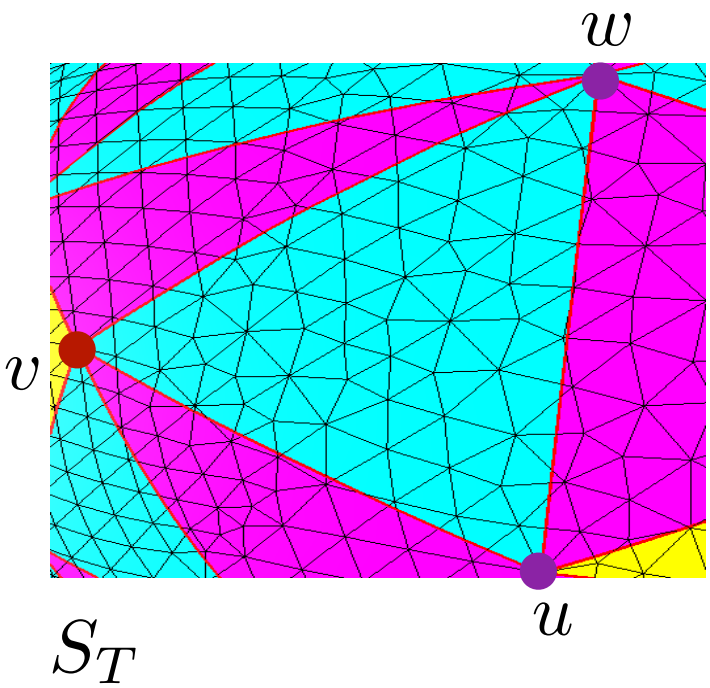
- Map the vertices of S_T bounded by the embedding of $st(v, \tilde{S}_T)$ to T_v.

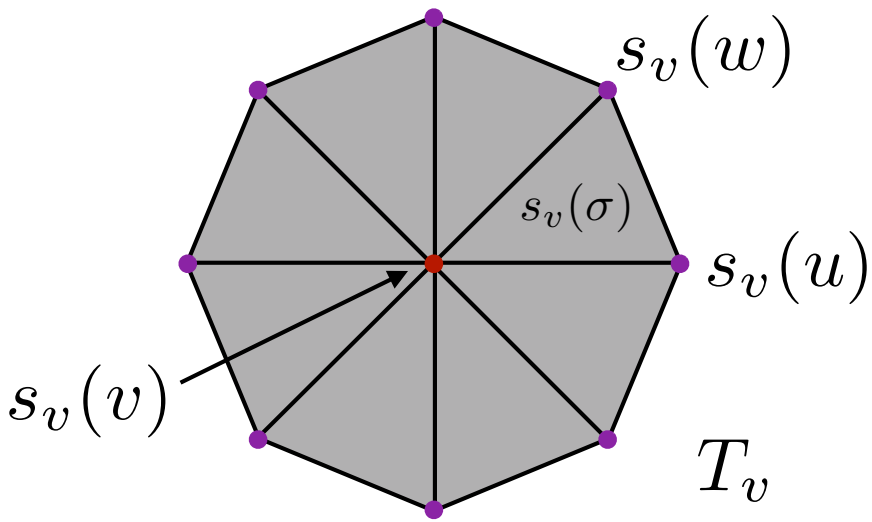


Adaptive Fitting

Create S from $\tilde{S}_T$

- Map the vertices of S_T bounded by the embedding of $st(v, \tilde{S}_T)$ to T_v.

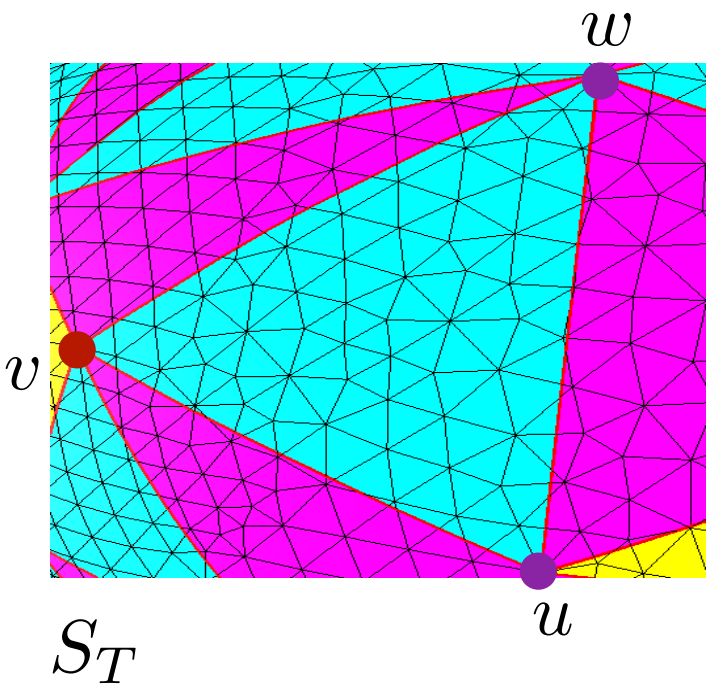


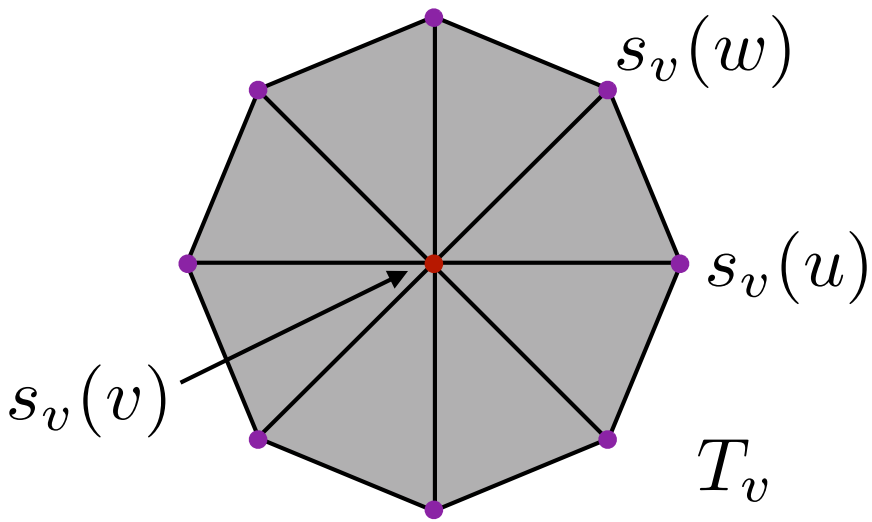


Adaptive Fitting

Create S from $\tilde{S}_T$

- Map the vertices of S_T bounded by the embedding of $st(v, \tilde{S}_T)$ to T_v.

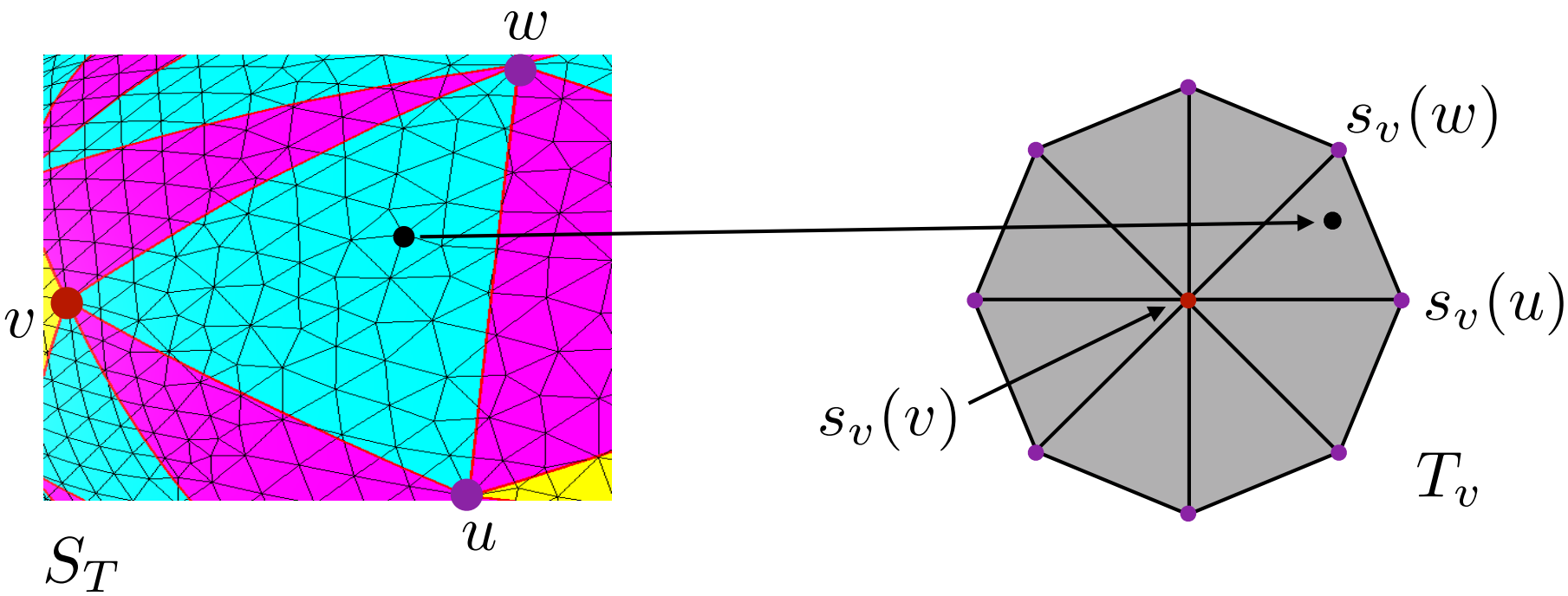




Adaptive Fitting

Create S from $\tilde{S}_T$

- Map the vertices of S_T bounded by the embedding of $st(v, \tilde{S}_T)$ to T_v.



Adaptive Fitting

Create S from $\tilde{S}_T$

Adaptive Fitting

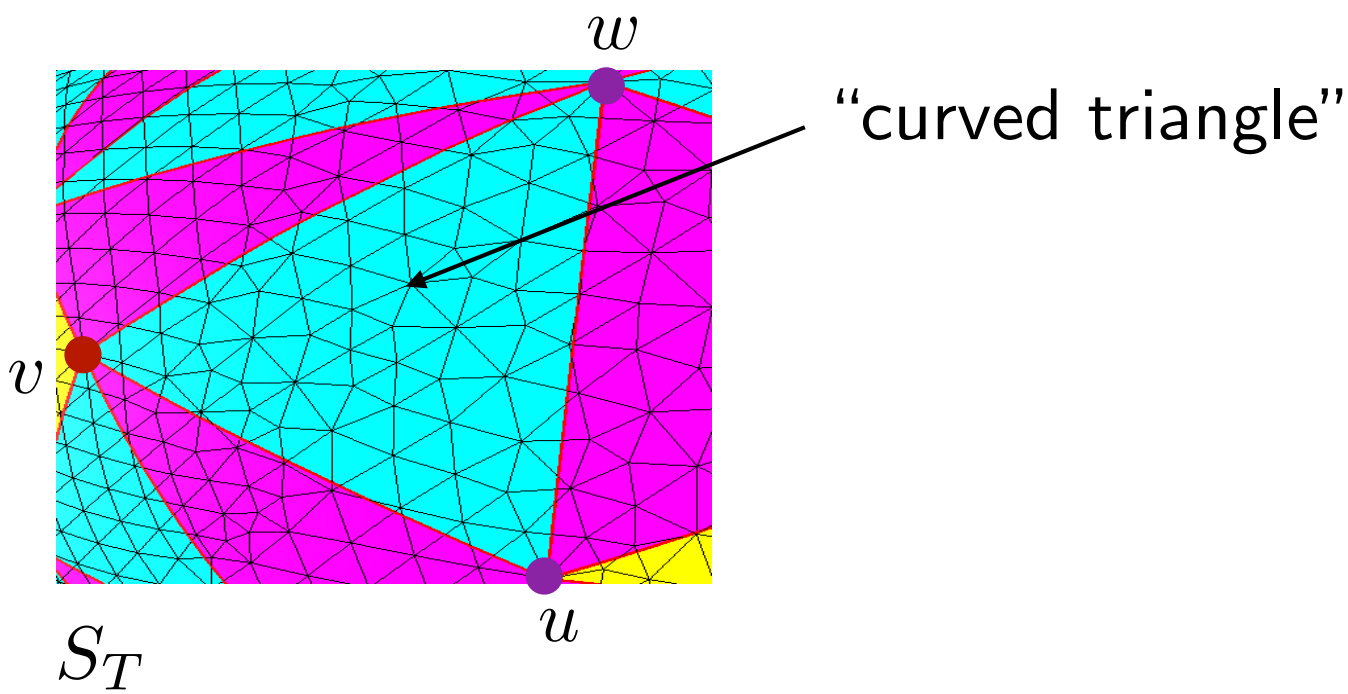
Create S from $\tilde{S}_T$

- We map the vertices in each "curved" triangle separately.

Adaptive Fitting

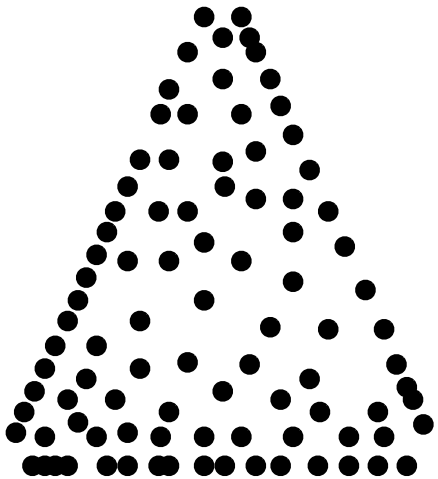
Create S from $\tilde{S}_T$

- We map the vertices in each "curved" triangle separately.



Adaptive Fitting

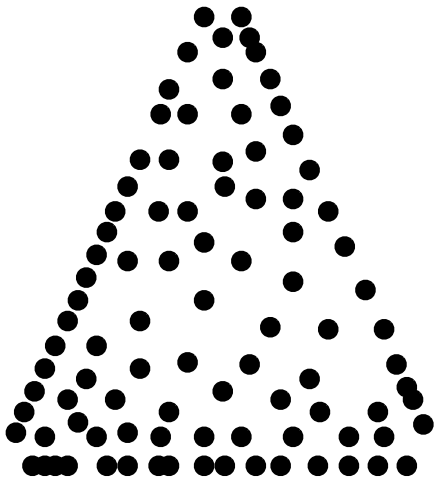
Create S from $\tilde{S}_T$



Adaptive Fitting

Create S from $\tilde{S}_T$

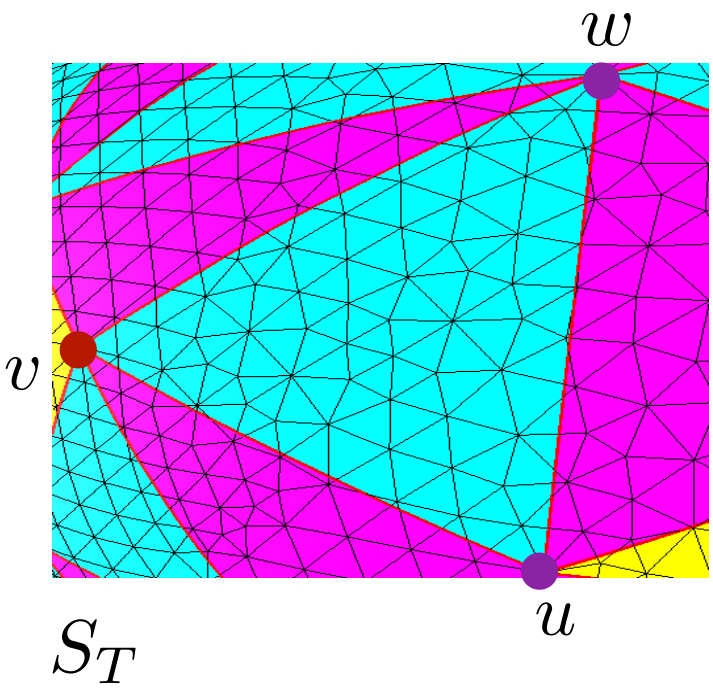
- We use Floater's parametrization to build the map for each "curved" triangle.

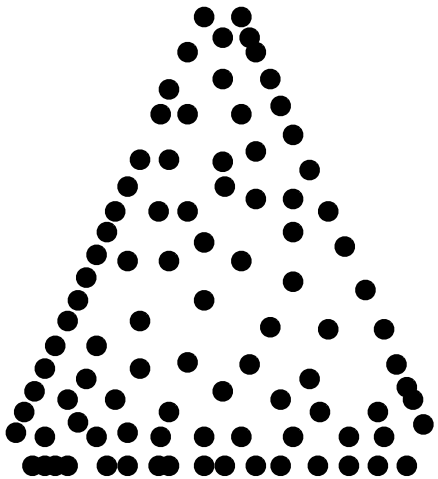


Adaptive Fitting

Create S from $\tilde{S}_T$

- We use Floater's parametrization to build the map for each "curved" triangle.

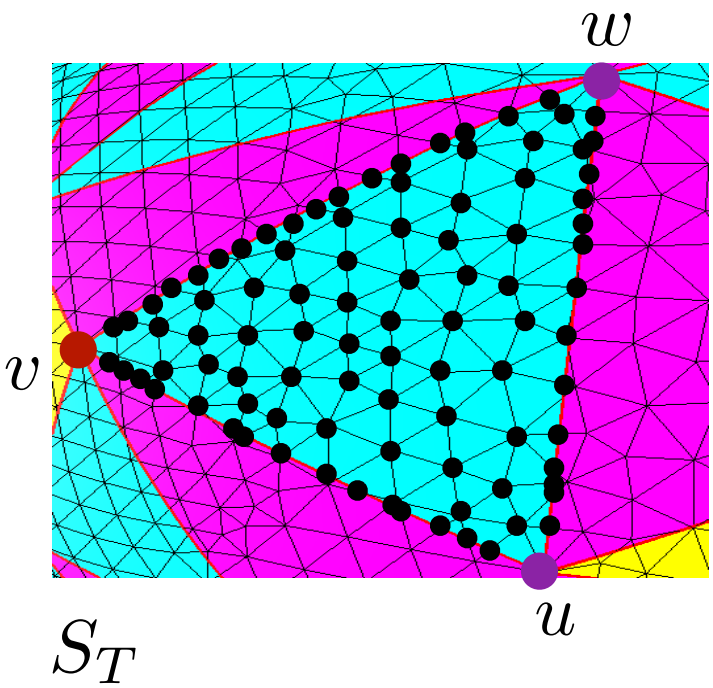


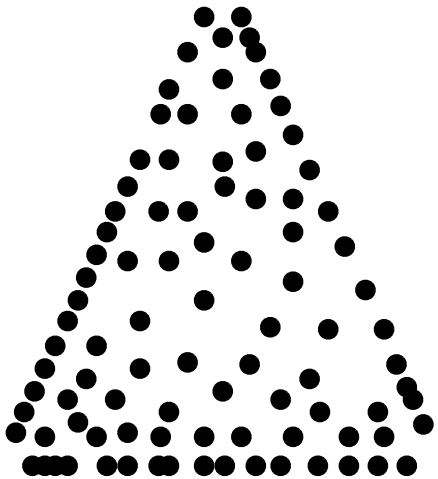


Adaptive Fitting

Create S from $\tilde{S}_T$

- We use Floater's parametrization to build the map for each "curved" triangle.

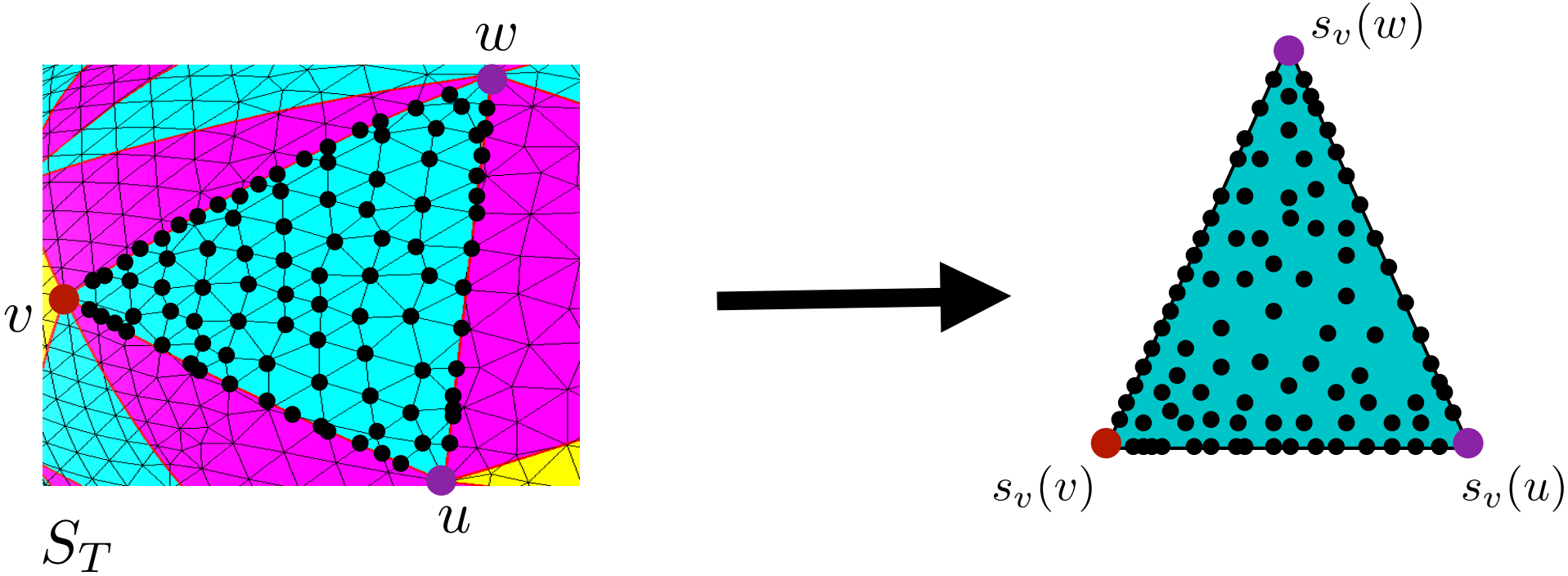




Adaptive Fitting

Create S from $\tilde{S}_T$

- We use Floater's parametrization to build the map for each "curved" triangle.



Adaptive Fitting

Create S from $\tilde{S}_T$

Adaptive Fitting

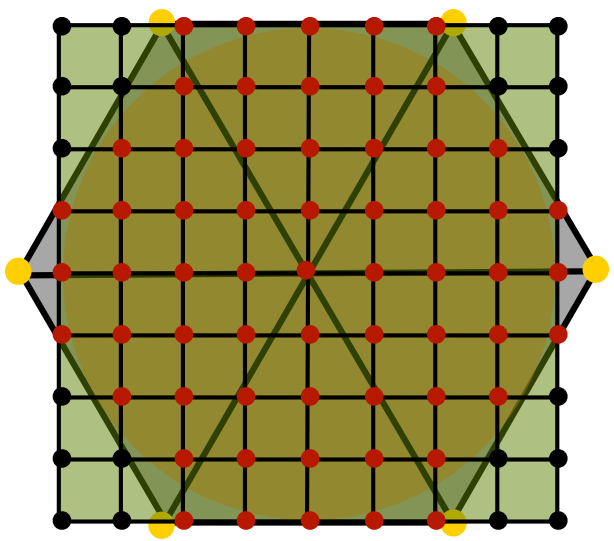
Create S from $\tilde{S}_T$

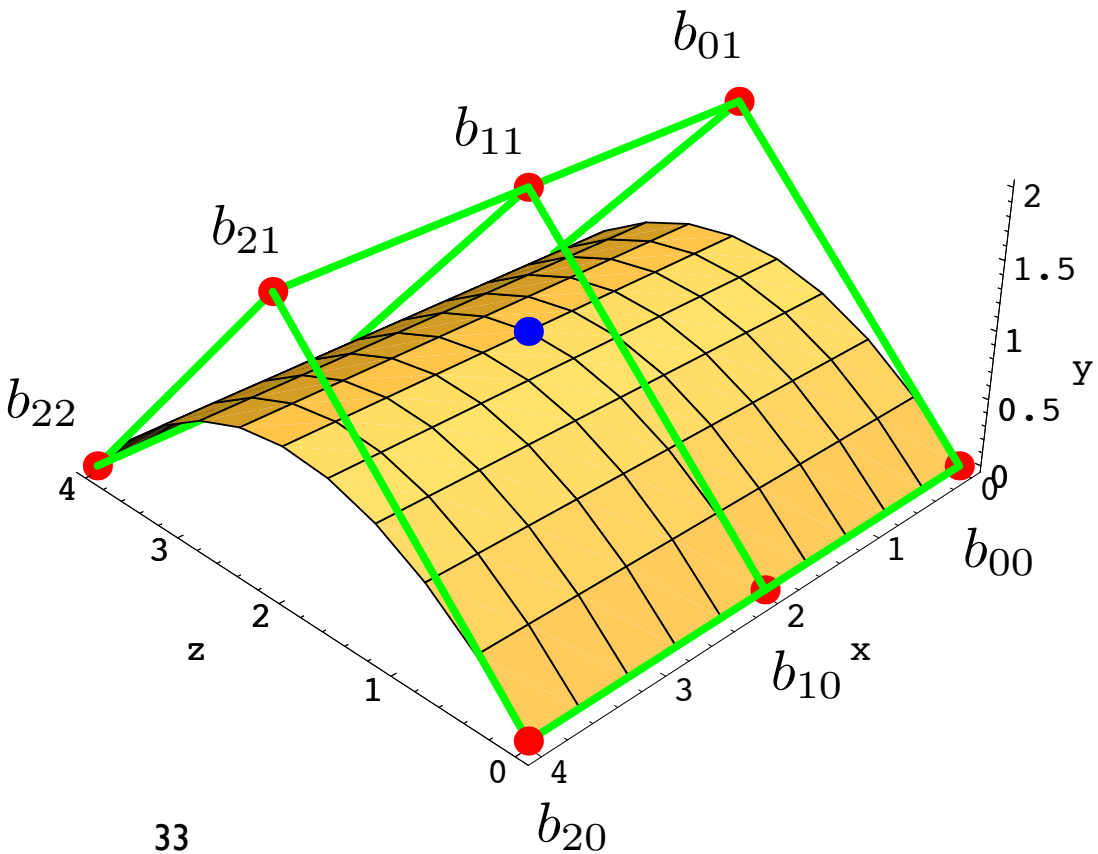
- For each triangle in $st(v, \tilde{S}_T)$, compute the shape function ψ_v.

Adaptive Fitting

Create S from $\tilde{S}_T$

- For each triangle in $st(v, \tilde{S}_T)$, compute the shape function ψ_v.





Adaptive Fitting

Create S from $\tilde{S}_T$

Adaptive Fitting

Create S from $\tilde{S}_T$

- Control points of ψ_v are computed by a least squares procedure.

Adaptive Fitting

Create S from $\tilde{S}_T$

- Control points of ψ_v are computed by a least squares procedure.

- But, this time, the sample points are the vertices of S_T that correspond to the points in T_v through Floater's parametrization!

Adaptive Fitting

Create S from $\tilde{S}_T$

Adaptive Fitting

Create S from $\tilde{S}_T$

- For each point $p \in T_v$, we compute the **approximation error**,

$$\|q - \psi_v(p)\|,$$

where q is the vertex of S_T corresponding to p through Floater's parametrization.

Adaptive Fitting

Create S from $\tilde{S}_T$

- For each point $p \in T_v$, we compute the **approximation error**,

$$\|q - \psi_v(p)\|,$$

where q is the vertex of S_T corresponding to p through Floater's parametrization.

- If the above error is smaller than the given number ϵ, we keep computing ψ_u, for each $u \in I$. Otherwise, we stop this process and go to the refinement step.

Adaptive Fitting

Adaptive Fitting

Refine $\tilde{S}_T$

Adaptive Fitting

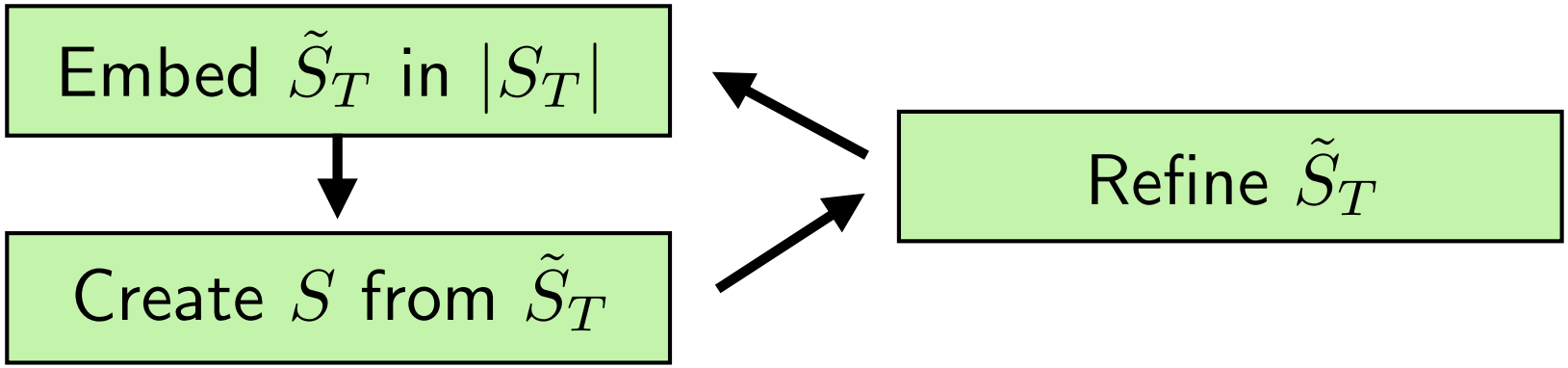
Refine $\tilde{S}_T$

- We locally refine $\tilde{S}_T$ using the stellar operations and the 4-8 refinement, and then embed the resulting $\tilde{S}_T$ in $|S_T|$ again.

Adaptive Fitting

Refine $\tilde{S}_T$

- We locally refine $\tilde{S}_T$ using the stellar operations and the 4-8 refinement, and then embed the resulting $\tilde{S}_T$ in $|S_T|$ again.



Conclusions

Conclusions

- Simplicial Multiresolution
  - Powerful Mechanism for Adaptation
- First Part
  - Simplification
  - Adaptive Refinement
- Second Part
  - Geodesic Parametrization
  - Bezier Approximation