Direct Resampling for Isotropic Surface Remeshing

Simon Fuhrmann, Jens Ackermann, Thomas Kalbe and Michael Goesele

Technische Universität Darmstadt
Interactive Graphics Systems Group
Introduction

- High-quality remeshing
  - For a given input mesh, compute a new mesh, change props
  - Strive for equilateral triangles, isotropic vertex distribution

Original mesh, ~35k vertices

Remeshing

Remesh, 5k vertices
Introduction

- Relaxation-based remeshing
  - Centroidal Voronoi tessellation (CVT) for isotopic remeshing
  - Lloyd relaxation to compute the CVT
  - Other relaxation schemes applicable
Lloyd relaxation

Lloyd's algorithm

- Build Voronoi diagram
- Move samples to cell centroid
- Repeat until convergence

Lloyd relaxation on random samples
Lloyd relaxation

Lloyd's algorithm

- Build Voronoi diagram
- Move samples to cell centroid
- Repeat until convergence

Lloyd relaxation on random samples
Lloyd relaxation

Lloyd's algorithm

- Build Voronoi diagram
- Move samples to cell centroid
- Repeat until convergence

Lloyd relaxation on random samples
Lloyd relaxation

Lloyd's algorithm

- Build Voronoi diagram
- Move samples to cell centroid
- Repeat until convergence

Lloyd relaxation on random samples

Animation missing!
Motivation

- Lloyd relaxation yields **uniform distribution**
- Relaxation requires good initialization
  - **Local** randomness quickly removed
  - **Global** mismatch between initial and desired distribution leads to slow convergence

**Goal:** Provide good initial vertex distribution

Animation missing!
Contribution

• Practical method for isotropic remeshing
  • Exact vertex budget as user parameter
  • Final vertex distribution controlled by density field
  • Preservation of tagged feature edges

• Main contributions
  • Fast direct resampling according to density field
  • Robust meshing in 3D using mutual tessellation
  • Beneficial sample distribution leads to fast CVT computation
Related Work (1)

- **Isotropic Remeshing of Surfaces** [Alliez et al. '03]
  - Uses global parametrization
  - Sampling, Lloyd relaxation & meshing in 2D

[Diagram showing the process of isotropic remeshing: Original mesh → Global parametrization → Lifting to 3D → Remeshed model]
Related Work (2)

- **A Local Parametrization Approach** [Surazhsky et al. '03]
  - Vertex-split and edge-collapse to control complexity
  - Small, overlapping parametrizations
  - Possibly “wrong” global vertex distribution

- **Direct sampling on surfaces for HQ remeshing** [Fu et al. '09]
  - Poisson-disc sampling for initial vertex distribution
  - Requires geodesics (slow)
  - Little control over the result (no vertex budget)
Algorithm Overview

- **Preprocessing**
  - Curvature estimation
  - Feature extraction

- **Resampling**
  - Surface/feature resampling
    - Mutual tessellation
  - Vertex decimation
    - Cleans original vertices

- **Mesh Improvement**
  - Lloyd relaxation to build WCVT
Algorithm Overview

- **Preprocessing**
  - Curvature estimation
  - Feature extraction

- **Resampling**
  - Surface/feature resampling
    - Mutual tessellation
  - Vertex decimation
    - Cleans original vertices

- **Mesh Improvement**
  - Lloyd relaxation to build WCVT
Algorithm Overview

- **Preprocessing**
  - Curvature estimation
  - Feature extraction

- **Resampling**
  - Surface/feature resampling
    - Mutual tessellation
  - Vertex decimation
    - Cleans original vertices

- **Mesh Improvement**
  - Lloyd relaxation to build WCVT
Resampling algorithm

- Resampling with exact number of vertices (budget)
- Partition budget between surface and features
  - Integrate surface density (2D quantity)
  - Integrate feature density (1D quantity)
  - Formulas from [Alliez et al. '03] to get
    - number of surface samples
    - number of feature samples
- Resample smooth surface (random placement)
- Resample feature skeleton (exact placement)
Surface resampling (1)

- Given: Number of samples for whole surface

- For each triangle $T$
  - Calculate optimal number of samples for $T$
  - Round to nearest integer, sample $T$
  - Pass error to next triangle
    - Similar to dithering, but no error diffusion (saves memory)

- Problem: Error is teleported to arbitrary locations!
Surface resampling (2)

- Re-order triangles for spatial locality [Sander et al. '07]
Feature resampling (1)

- Build skeleton by chaining feature edges
Feature resampling (2)

- For each backbone \( B \)
  - Compute optimal number of samples for \( B \)
  - Round number to nearest integer
    \( \rightarrow \) pass error to next backbone
  - Deduce optimal “density spacing” \( S \)
  - Place a sample every \( S \) density

![Diagram showing feature resampling process](attachment:image.png)
Decimation and Relaxation

- Samples are inserted into the original mesh (mutual tessellation) [Turk '92]
- Old vertices are deleted afterwards [Schroeder et al. '92]
- Lloyd relaxation using local parametrization framework [Surazhsky et al. '03]
Results – Horse model

Original model, ~50k vertices
Remeshed model, 6k vertices

7 secs with 100 Lloyd iterations
Results – Hygieia

Original mesh, ~8k vertices  After resampling  Remesh, 10k vertices

3.5 secs
Results – Comparison

* Our results use CPU parallelization

<table>
<thead>
<tr>
<th></th>
<th>Our work*</th>
<th>Surazhsky et al. '03</th>
<th>Fu et al. '09</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 sec</td>
<td>28 sec</td>
<td>103 sec</td>
<td></td>
</tr>
<tr>
<td>3.5 sec</td>
<td>17 sec</td>
<td>113 sec</td>
<td></td>
</tr>
</tbody>
</table>

* Our results use CPU parallelization
Results – Beethoven

Original model, ~1.5M vertices

Remeshed model, 500k vertices

5 mins with 100 Lloyd iterations
Limitations & Future Work

- Vertex relocation most fragile operation [Surazhsky et al. 03]
  - Involves (local) parametrization
  - Distortion in the patches is problematic
- **Future Work:** Relaxation without parametrization

Degenerated faces
~3k vertices

Mesh Slicing
[Botsch & Kobbelt ’01]

Remeshed Model
5k vertices
Conclusion

- We presented a remeshing approach that is
  - fast and efficient
    beneficial initial vertex distribution
    CPU parallelization in relaxation framework
  - simple and robust
    straight-forward implementation
    well-known, robust techniques (mutual tessellation, ...)
  - general and accurate
    2-manifolds with arbitrary genus
    relocation on reference surface
Thanks for your attendance!

Source code is available at http://tinyurl.com/remesher
Thanks for your attendance!

Source code is available at
http://tinyurl.com/remesher