Point Cloud Resampling
Using
Centroidal Voronoi Tessellation Methods

Zhonggui Chen$^{1,2}$
Tieyi Zhang$^1$
Juan Cao$^{1,2}$
Yongjie Jessica Zhang$^2$
Cheng Wang$^1$

Graphics@XMU  http://graphics.xmu.edu.cn/
Motivation

- raw input
Motivation

- input
- uniform
Motivation

- input
- weighted
Motivation

- input
- anisotropic
Motivation

- input
- anisotropic
Related Work

- **LOP**
  - [Y. Lipman et al. TOG ’07]

- **WLOP**
  - [H. Huang et al. TOG ’09]

- **graph Laplacian**
  - [C. Luo et al. CGF ’18]
Background

- Surface remeshing
  - Centroidal Voronoi Tessellation, CVT

- CVT
  - [Q. Du et al. ’99]

- $L_p$ CVT
  - [B. Le´vy & Y. Liu ’10]
Background

- Centroidal Voronoi Tessellation
  - Restricted Voronoi Diagram, RVD

- input mesh

- RVD
Background

● Restricted Voronoi Diagram
  - Tangent planes
Background

- Restricted Voronoi Diagram
  - Tangent planes
Background

- Restricted Voronoi Diagram
  - Tangent planes
Background

• Restricted Voronoi Diagram
  - Tangent planes
Background

- Restricted Voronoi Diagram
  - Tangent planes
Point Cloud Resampling

Overview

1. $P = \{p_i\}_{i=1}^m$
2. $\rho(x)$
3. $n$

Input: an unstructured point cloud

Initial sampling

Compute tangent planes

Construct RVC

Optimize point positions

Pull back to the point cloud

Output: resampling points

$\{x_i\}_{i=1}^n$
Point Cloud Resampling

1. Initialization - sampling

- randomly sample \( n \) points

- input

- initial sampling
Point Cloud Resampling

1. Initialization - tangent planes

\[ \tau_i \]

\[ X_i \]

\[ \{ p_{ij} \}_{j=1}^k \]
Point Cloud Resampling

1. Initialization - tangent planes
Point Cloud Resampling

1. Initialization - tangent planes

\[ \mathbf{X}_i \]
Point Cloud Resampling

1. Initialization - tangent planes

\[ \delta_i^r \]

\[ X_i \]

\[ r \]
Point Cloud Resampling

1. Initialization-tangent planes

\[ r = \frac{1}{6} \sum_{i=1, x_i \in N_x}^{6} \sqrt{(x - x_i)^2} \]
Point Cloud Resampling

2. RVC computation

- clippling method

[B. Le´vy & N. Bonneel’13]
Point Cloud Resampling

2. RVC computation

\[ \delta_i^r \]

\[ X_i \]

\[ X_j \text{ to be considered} \]

\[ X_j \text{ drop} \]

\[ r \]

\[ \leq 2r \]

\[ > 2r \]
Point Cloud Resampling

2. RVC computation

\( X_i \) bisector \( X_j \) to be considered \(< 2r \)

\( \delta_i^r \) drop \( > 2r \)
2. RVC computation

\[ \delta_i^r \]

\[ X_i \]

\[ X_j \text{ to be considered} (<2r) \]

\[ X_j \text{ drop} (>2r) \]

bisector

bisector
Point Cloud Resampling

3. Optimization

\[ X = \{x_i\}_{i=1}^n \] be a set of seed points on a given domain \( R^d \).

\[
E(X) = \sum_{i=1}^{n} \int_{V(x_i) \cap \tau_i} \rho(x)\psi(x, x_i) \, d\sigma \quad \text{Lloyd & BFGS}
\]

<table>
<thead>
<tr>
<th>( V(x_i) \cap \tau_i )</th>
<th>Voronoi cell ( V(x_i) ) of ( x_i ) restricted to tangent plane ( \tau_i ) (mesh ( \delta_i^\tau ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \rho(x) )</td>
<td>User-defined density function</td>
</tr>
<tr>
<td>( \psi(x, x_i) )</td>
<td>Metric defining distance between points ( x ) and ( x_i )</td>
</tr>
</tbody>
</table>
Point Cloud Resampling

3. Optimization - Lloyd's method

- Lloyd
  - [S. Lloyd ’82]

\[ E(X) = \sum_{i=1}^{n} \int_{V(x_i) \cap \tau_i} \rho(x) \psi(x, x_i) d\sigma \]

\[ o_i = \frac{\int_{V(x_i) \cap \tau_i} \rho(x) x d\sigma}{\int_{V(x_i) \cap \tau_i} \rho(x) d\sigma} \]
3. Optimization - pulling back

\[ \tau_i \]

\[ X_i \]
Point Cloud Resampling

3. Optimization - pulling back

(a), (b) The RVCs before and after Lloyd's relaxation
Point Cloud Resampling

3. Optimization - pulling back

- (a)
- (b)
- (c) The plot of CVT energy function versus the iteration #
Point Cloud Resampling

3. Optimization - BFGS

\[ E(X) = \sum_{i=1}^{n} \int_{V(x_i) \cap \tau_i} \rho(x) \psi(x, x_i) d\sigma \]

Integral domains \( V(x_i) \cap \tau_i \) change discontinuously.
Point Cloud Resampling

3. Optimization - BFGS

\[ E(X) = \sum_{i=1}^{n} \int_{V(x_i) \cap \tau_i} \rho(x) \psi(x, x_i) d\sigma \]

\[ \rho(x) \equiv 1 \]

\[ \psi(x, x_i) = \| M(x)(x - x_i) \|_p^p \]

<table>
<thead>
<tr>
<th>| \cdot |_p^p</th>
<th>L_p norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>M(x)</td>
<td>Tensor field ( G(x) ), ( G(x) = M(x)^T M(x) )</td>
</tr>
</tbody>
</table>
### Point Cloud Resampling

#### 3. Optimization - BFGS

<table>
<thead>
<tr>
<th>Iteration #</th>
<th>Lloyd</th>
<th>BFGS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Points</td>
<td>Planes</td>
<td>Points</td>
</tr>
<tr>
<td>1</td>
<td>update</td>
<td>update</td>
<td>update</td>
</tr>
<tr>
<td>2</td>
<td>update</td>
<td>update</td>
<td>update</td>
</tr>
<tr>
<td>3</td>
<td>update</td>
<td>update</td>
<td>update</td>
</tr>
<tr>
<td>4</td>
<td>update</td>
<td>update</td>
<td>update</td>
</tr>
<tr>
<td>$J_{max} = 5$</td>
<td>update</td>
<td>update</td>
<td>update</td>
</tr>
<tr>
<td>6</td>
<td>update</td>
<td>update</td>
<td>update</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>n</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>
3. Optimization - BFGS

- Computing $E(x)$ & $\nabla E(x)$


  - [G. Parigi, M. Piastra, Gradient of the objective function for an anisotropic centroidal Voronoi tessellation (CVT)-a revised, detailed derivation, 721 arXiv:1408.5622 (2014).]
Point Cloud Resampling

3. Optimization - BFGS

- Computing $E(x)$ & $\nabla E(x)$

  - [B. Lévy, Y. Liu, Lp centroidal Voronoi applications, ACM Transactions on Graphics 29]

\[ E(x) = \sum_{i=1}^{n} d^p(x_i, y) \]
\[ \nabla E(x) = \sum_{i=1}^{n} \nabla d^p(x_i, y) \]

\[ d^p(x_i, y) = \left\{ \begin{array}{ll} 0 & \text{if } d(x_i, y) \leq r \\|x_i\|_p \\|y\|_p^{1/p} \\ \|x_i\|_p \\|y\|_p^{1/p} - d(x_i, y) & \text{otherwise} \end{array} \right. \]

\[ \nabla d^p(x_i, y) = \left\{ \begin{array}{ll} 0 & \text{if } d(x_i, y) \leq r \\|x_i\|_p \\|y\|_p^{1/p} \\ \frac{x_i - y}{d(x_i, y)} & \text{otherwise} \end{array} \right. \]

Acknowledgments

This project is supported by the European Research Council (GOODSHAPE ERC-SIG-205693). The authors want to warmly thank Richard Ziegler and Wapeng Wang for many discussions. Jean-François Remacle for help with GMiB, Nicola Saffogari for packing, Dong-Ming Yan for the implementation of BVOD, the CGAL project for Delaunay triangulation, Loic Marchal and Pierre Alliez for data and discussions, Trimul Dey for data and help with Delaunay, Daniel Bonnoum, Leif Kobbelt and AutoShape for data, and the anonymous reviewers for their suggestions that helped improving the paper.

B. Expression of $\nabla F$

We first derive $\nabla F(x) = \nabla F(x)$ and then $\nabla F(x)$. Recalling that $F(x) = \sum \nabla F(x)$, where $\nabla F(x)$ is given Equation 4 in Appendix A, we obtain:

\[ \nabla F(x) = \sum \nabla F(x) \]

In surface meshing, $\sum \nabla F(x) = \sum \nabla F(x)$.

In volume meshing, $\sum \nabla F(x) = \sum \nabla F(x)$.

Finally, the derivatives of $\nabla F(x)$, with respect to $x_i$, $C_1$, $C_2$, and $C_3$ are given by:

\[ \frac{\partial \nabla F(x)}{\partial x_i} = \frac{\partial \nabla F(x)}{\partial C_1} = \frac{\partial \nabla F(x)}{\partial C_2} = \frac{\partial \nabla F(x)}{\partial C_3} \]
Point Cloud Resampling

3. Optimization - BFGS - computing $E(x)$ and $\nabla E(x)$

$$E_T(X) = \int_T \|M_T(x - x_i)\|^p \, d\sigma$$

$$= \frac{|T|}{\binom{2+p}{2}} \sum_{\alpha+\beta+\gamma=p} u_1^\alpha \ast u_2^\beta \ast u_3^\gamma,$$

where

$$\begin{align*}
  u_j &= M_T(c_j - x_i), \\
  u_1 \ast u_2 &= [x_1, x_2, y_1, y_2, z_1, z_2]^T, \\
  u^\alpha &= u \ast u \ast \cdots \ast u \ (\alpha \ \text{times}), \\
  \bar{u} &= x + y + z.
\end{align*}$$

$$\frac{dE_T(x_i, c_1, c_2, c_3)}{dX} = \frac{dE_T}{dx_i} + \frac{dE_T}{dc_1} \frac{dc_1}{dX} + \frac{dE_T}{dc_2} \frac{dc_2}{dX} + \frac{dE_T}{dc_3} \frac{dc_3}{dX}$$
Initial sampling is resampled again based on the RVCs.
Point Cloud Resampling

4. Initialization-improved sampling

\[ A_w(\tau_i) : \text{weighted area for each RVC } \tau_i \]

\[ A_w(\tau_i) = |\tau_i| \sum_{j=1}^{k} \rho(p_{ij})/k \]

\[ |\tau_i| : \text{area of RVC } \tau_i \]

with the probability of selecting a RVC \( \tau_i \) proportional to \( \frac{A_w(\tau_i)}{\sum_{i=1}^{n} A_w(\tau_i)} \)
Point Cloud Resampling

4. Initialization-improved sampling

initial sampling

resampling

10 itr.

10 itr.
Results

1. Uniform resampling

(a) input

- [H. Huang et al. TOG ’09]

(b) WLOP

- [C. Luo et al. CGF ’18]

(c) graph Laplacian

(d) ours

\[
\psi(x, x_i) = \| (x - x_i) \|^2
\]

(e) k-means

<table>
<thead>
<tr>
<th>input #</th>
<th>110K</th>
</tr>
</thead>
<tbody>
<tr>
<td>output #</td>
<td>5K</td>
</tr>
</tbody>
</table>
Results

1. Uniform resampling

(a) input

- [H. Huang et al. TOG ’09]
- [C. Luo et al. CGF ’18]

(b) WLOP

(c) graph Laplacian

(d) ours

(e) k-means

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>input #</td>
<td>110K</td>
</tr>
<tr>
<td>output #</td>
<td>5K</td>
</tr>
</tbody>
</table>
Results

1. Uniform resampling

<table>
<thead>
<tr>
<th>method</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(b) WLOP</td>
<td>0.182</td>
</tr>
<tr>
<td>(c) graph Laplacian</td>
<td>0.037</td>
</tr>
<tr>
<td>(d) ours</td>
<td>0.052</td>
</tr>
<tr>
<td>(e) k-means</td>
<td>0.124</td>
</tr>
</tbody>
</table>

-[H. Huang et al. TOG '09]

-[C. Luo et al. CGF '18]
Results

1. Uniform

- **Running time** against the number of input points ranging from 10K to 10M, with a fixed output point number (m = 10K)
Results

2. Weighted resampling

(a) input (3M)  
(b) weighted resampling (50k)

- Adaptive resampling result of scan data of a dragon model in 16.4s
Results

3. Anisotropic resampling

(a) anisotropic

(b) $L_8$
Results

4. Noise depression

(a) input  (b) graph Laplacian  (c) ours
Results

5. Hole filling

(a) input

(b) output

by surface extrapolation
Results

6. Boundary handling

(a) input

(b) drift away
Results

6. Boundary handling

(a) input

(c) boundary

Point Cloud Library
Results

6. Boundary handling

(a) input

(d) resample the boundary
6. Boundary handling

(a) input

(e) output
Results

7. Surface reconstruction

- Surface reconstruction using the duality between RVCs and restricted Delaunay triangulation
Contributions

 ✓ Extend the CVT (Centroidal Voronoi Tessellation) energy function defined on point clouds;

 ✓ High-quality resampling results with isotropic or anisotropic distribution;

 ✓ Effectively remove noise and fill holes or preserve boundaries of the point cloud.
Limitation

- When the number of resampling points is too small;
- When the data points contain sharp edges.
Limitation

(a) input

(b) output
THANKS

Q&A