GEOMetrics

Exploiting Geometric Structure for Graph-Encoded Objects

Edward Smith, Scott Fujimoto, Adriana Romero, David Meger
Topic: Mesh Object Generation
What is a Mesh?

3D surface representation
- Collection of connected triangular faces

Defined by a graph $G = \{V, A\}$
- $V =$ collection of vertices
- $A =$ Adjacency Matrix
  - $A[i,j] \neq 0$ if and only if there exist a face $f$, such that $\{i,j\}$ is in $f$
Why Choose Meshes?

Voxels  Point cloud

Uniform mesh  Adaptive mesh
Mesh Generation

How do you predict a complicated graph structure?
  • You don’t

Deform a predefined mesh
  • Assume initial graph structure
  • Predict updates to the structure
  • How do we make these updates?
  • How do we compare to know mesh ground truth?
Deform: Graph Convolutional Network

Input: - graph \{V, A\} - features over V, \{H\} - weight and bias \{W, b\}

Apply the following operation:

\[ H' = \sigma (AHW + b) \]

Problem:
- Vertex smoothing
- Each vertex in an mesh is important
- Exacerbated in adaptive mesh

Reference for figure: https://tkipf.github.io/graph-convolutional-networks/
Solution: Zero Neighbor GCN

Basic formulation: \[ H' = \sigma(AHW + b) \]

Higher order: \[ H' = \sigma([AH_1||A^2H_2||...||A^kH_k]W + b) \]

0N-GCN: \[ H' = \sigma([A^0H_0||AH_1||]W + b) \]
\[ H' = \sigma([H_0||AH_1||]W + b) \]

- Soft middle ground between neighbor update and none
- Adaptive meshes should emerge more easily
Compare: Chamfer Distance

\[ \mathcal{L}_{\text{Chamfer}} = \sum_{p \in S} \min_{q \in \hat{S}} \| p - q \|_2^2 + \sum_{q \in \hat{S}} \min_{p \in S} \| p - q \|_2^2 \]

Problem with naïve mesh application:
- Arbitrary vertex placement
- No consideration of the faces they define
Vertex-to-Point

\[ \mathcal{L}_{\text{Chamfer}} = \sum_{p \in \mathcal{S}} \min_{q \in \hat{\mathcal{S}}} \| p - q \|_2^2 + \sum_{q \in \hat{\mathcal{S}}} \min_{p \in \mathcal{S}} \| p - q \|_2^2 \]

Past attempt to solve issue:
- Make ground truth and predicted meshes huge
- Definitely not going to get adaptive mesh
Solution: Sample using Reparameterization

Sample both meshes uniformly
- Given: \( F = \{v_1, v_2, v_3\} \), \( U \) & \( W \) are Uniform(0,1)
- Sample: \( u \sim U \), \( w \sim W \)
- Sample projected onto triangle:
  \[
  r = (1 - \sqrt{u})v_1 + \sqrt{u}(1 - w)v_2 + \sqrt{uw}v_3
  \]
- Select faces at rate proportional to relative surface area
Point-to-Point Loss

$$\mathcal{L}_{PtP} = \sum_{p \in S} \min_{q \in \hat{S}} \|p - q\|_2^2 + \sum_{a \in \hat{S}} \min_{p \in S} \|p - q\|_2^2$$

Can sample independent of vertex position
- Removes ambiguity of the target placement
- Do not have to match vertex placement

Face information is now taken into account
Vertices can be placed optimally
Point-to-surface Loss

Can do even better still: compare to surfaces instead of points

\[ L_{PtS} = \sum_{p \in S} \min_{f \in \hat{M}} \text{dist}(p, f) + \sum_{q \in \hat{S}} \min_{f \in M} \text{dist}(q, f) \]

• Function dist() is the minimum distance from a point to a triangle in 3D space
• More accurate to the previous functions
Toy Example

- Initial
- Target

Point Correspondence
- Vertex to Point
- Point to Point
- Point to Surface

Result
Comparison of Loss Functions

- **Vertex to Point**
- **Point to Point**
- **Point to Surface**

IoU

Toy Example
Latent Loss

Train an encoder decoder system from mesh to voxel space
- Using 0N-GCN networks followed by 3D convolutional network

• The latent encoding should poses all info on passed object
Latent Loss

Use the difference between latent encodings of GT and predicted objects as a loss signal:
Mesh Generation Pipeline

**Input:** Image & initial mesh

**Output:** Mesh reconstruction

1. Pass image through CNN
2. Project image features onto initial mesh as feature vectors
3. Pass through graph through multiple ON-GCN layers
4. Train using: PtP loss, PtS loss, latent loss
Face Splitting

Analyse local curvature of the mesh
  • At each face calculate average change in normal

Every face over a given threshold is split into three

Repeat the pipeline with new initial mesh
  • End to end, fully differentiable
  • Encourages the generation of adaptive meshes
Full Mesh Generation Pipeline

Feature extraction

Mesh deformation

Face splitting
Quantitative Results

![Graph showing the relationship between F1 score improvement and number of vertices compression rate.](image-url)
Quantitative Results

Ablation study:

<table>
<thead>
<tr>
<th>Ours</th>
<th>GCN</th>
<th>Unif. Split.</th>
<th>No $\mathcal{L}_{\text{latent}}$</th>
<th>$\mathcal{L}_{\text{VtP}}$</th>
<th>Pixel2Mesh</th>
</tr>
</thead>
<tbody>
<tr>
<td>56.61</td>
<td>54.57</td>
<td>50.33</td>
<td>55.59</td>
<td>52.92</td>
<td>38.13</td>
</tr>
</tbody>
</table>
Qualitative Results
Qualitative Results
GEOMetric: Exploiting Geometric Structure for Graph-Encoded Objects

Visit our poster: 06:30 -- 09:00 PM @ Pacific Ballroom #145

Email us at: edward.smith@mail.mcgill.ca

Source code: https://github.com/EdwardSmith1884/GEOMetrics

Thank you for listening.