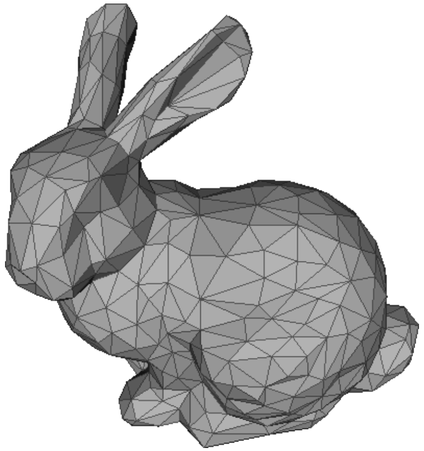
FeaStNet: Feature-Steered Graph Convolutions for 3D Shape Analysis

Tae Gyun Ahn

Seoul National University

Motivation



- 3D shape models can be represented using **mesh**
- **Mesh** is a graph structured data:

  usually consists of vertices, edge, and face data
- Mesh is **irregular** structure: vertices can have a varying number of neighbors
- **CANNOT** use CNN
- **Need GCN!**

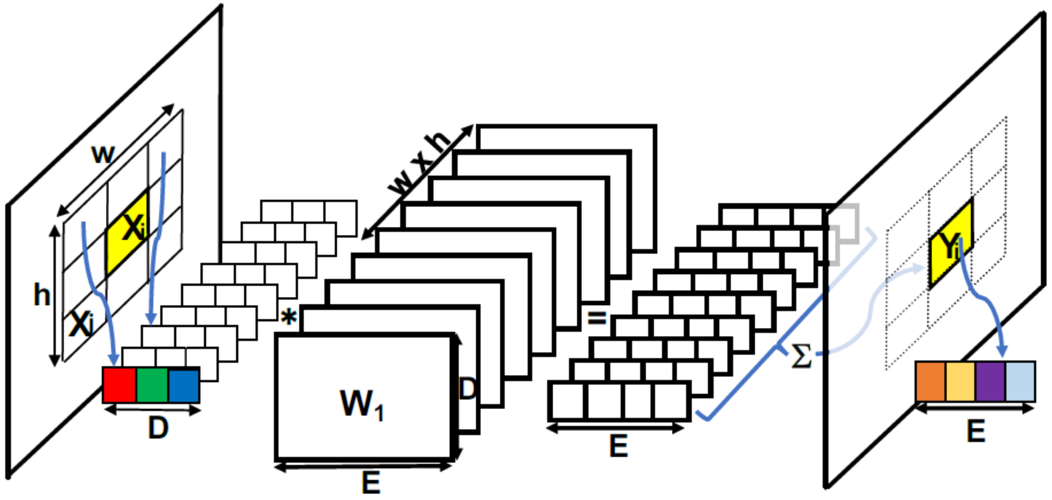
Contribution

- 1. **Dynamically** determine the **association** between **filer weights** and the **nodes**, using learned features of the preceding network layer
- 2. Can learn correspondences using raw 3D shape coordinates instead of 3D shape descriptors
- 3. Can be generalized to 3D data without explicit surface information

Related Works: Problems with previous GCN

- **1. Spectral Filtering**
  - Successful with synthetic 3D shape model (noise free data)
  - Not suitable for real shape models
  - Since global decompositions are unstable across different graphs
- **2. Local Filtering**
  - rely on sub optimal hard-coded local pseudo coordinates to define filters

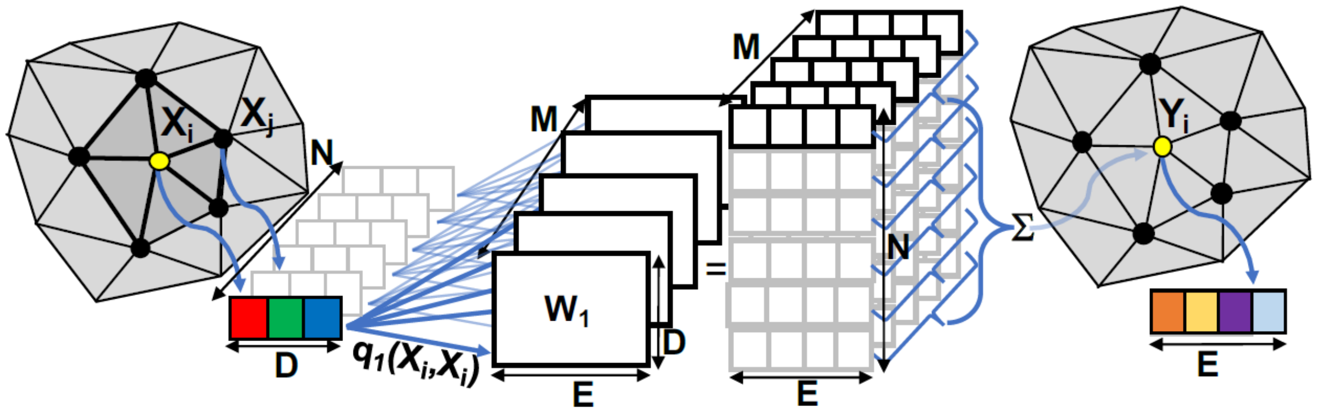
Method: CNN



$$y_i = b + \sum_{m=1}^{M} W_m x_{n(m,i)}$$

- $x_{n(m,i)} \in \mathbb{R}^D$, $y_i \in \mathbb{R}^E$, where D and E are number of channels
- $W_m \in \mathbb{R}^{E \times D}$: weight matrix of mth neighbor, $b \in \mathbb{R}^E$: bias
- $n(m,i)$: global index of mth neighbor

Method: GCN using node to weight association

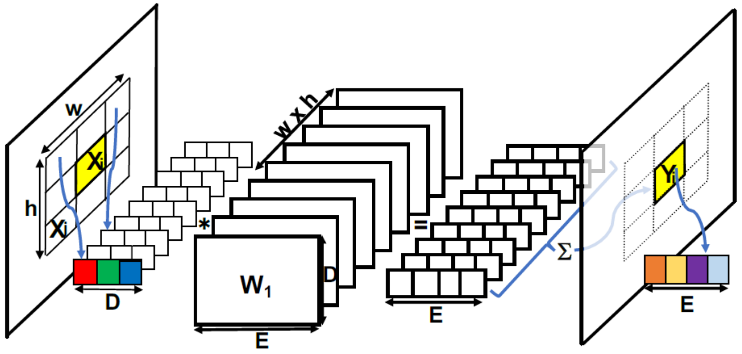


$$y_i = b + \sum_{m=1}^{M} \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} q_m(\mathrm{x}_i, \mathrm{x}_j) \mathrm{W}_m \mathrm{x}_j$$

- $q_m(\mathrm{x}_i, \mathrm{x}_j)$: assignment of x_j to W_m, $\sum_{\mathrm{m}=1}^{\mathrm{M}} q_m(\mathrm{x}_i, \mathrm{x}_j) = 1$
- $q_m(\mathrm{x}_i, \mathrm{x}_j) \propto \exp(\mathrm{u}_m^T \mathrm{x}_j + \mathrm{v}_m^T \mathrm{x}_\mathrm{i} + c_m)$, $\mathrm{u}_m, \mathrm{v}_m, c_m$: parameters of linear transformation.
- Translation invariant in feature space: $q_m(\mathrm{x}_i, \mathrm{x}_j) \propto \exp(\mathrm{u}_m^T(\mathrm{x}_j - \mathrm{x}_\mathrm{i}) + c_m)$
- Robust to variations in the degree of the nodes: $\sum_{j \in \mathcal{N}_i} \frac{1}{|\mathcal{N}_i|} \sum_{\mathrm{m}=1}^{\mathrm{M}} q_m(\mathrm{x}_i, \mathrm{x}_j) = 1$
- $\mathcal{N}_i$ can be expended to higher degree neighbors

Comparison

CNN



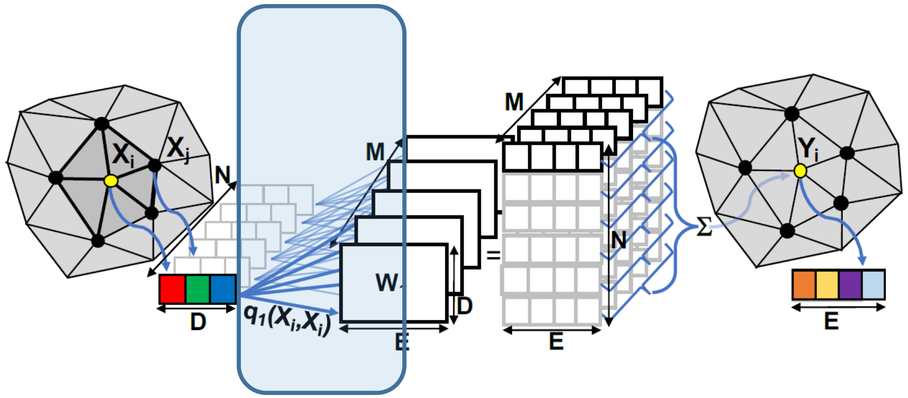
$$y_i = b + \sum_{m=1}^{M} \mathrm{W}_m \mathrm{x}_{n(m,i)}$$

- single node x_j - single weight matrix W_j

- Cost of computation: $\mathcal{O}(NMED)$

GCN

multiple filter calculation



$$y_i = b + \sum_{m=1}^{M} \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} q_m(\mathrm{x}_i, \mathrm{x}_j) \mathrm{W}_m \mathrm{x}_j$$

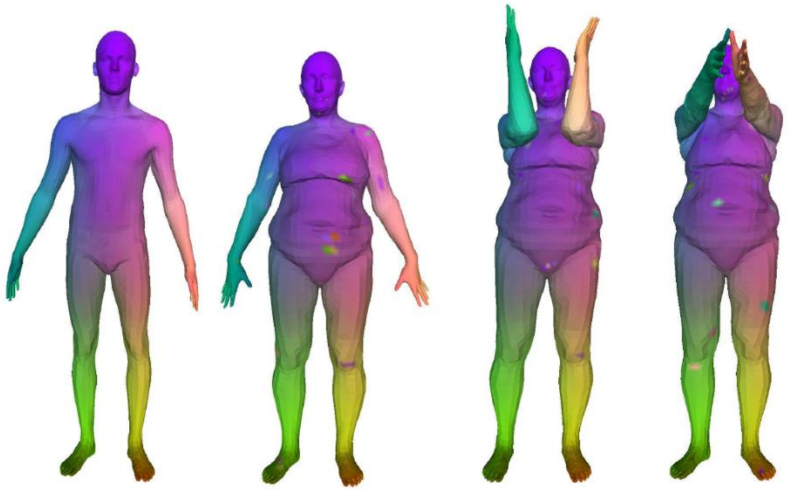
- single node x_j - multiple weight matrix W_m, $m = 1, \dots, |\mathcal{N}_i|$ (enough with 8)
- Cost of computation : $\mathcal{O}\big(NME(K+D)\big)$, K : average number of neighbors

Experiments

3D shape correspondence between 3D meshes using FAUST human shape dataset (dataset consists of 100 watertight meshes with 6,890 vertices each, corresponding to 10 shapes in 10 different poses each)

Architecture: Lin16+Conv32+Conv64+Conv128+Lin256+Lin6890
Loss: cross-entropy classification loss

Method	Input	Accuracy
Logistic Regr.	SHOT	39.9%
PointNet [19]	SHOT	49.7%
GCNN [14], w/o refinement	SHOT	42.3%
GCNN [14], w/ refinement	SHOT	65.4%
ACNN [2], w/o refinement	SHOT	60.6%
ACNN [2], w/ refinement [17]	SHOT	62.4%
MoNet [15], w/o refinement	SHOT	73.8%
MoNet [15], w/ refinement [29]	SHOT	88.2%
FeaStNet, w/o refinement	XYZ	88.1%
FeaStNet, w/ refinement [29]	XYZ	92.2%
FeaStNet, multi scale, w/o refinement	XYZ	98.6%
FeaStNet, multi scale, w/ refinement [29]	XYZ	98.7%
FeaStNet, multi scale, w/o refinement	SHOT	90.9%



Does not need descriptors!