Pose2Mesh: GCN for 3D Human Pose and Mesh Recovery from a 2D Human Pose (ECCV2020)

윤 현 석

Vision & Display Systems Lab.

Dept. of Electronic Engineering, Sogang University

Outline

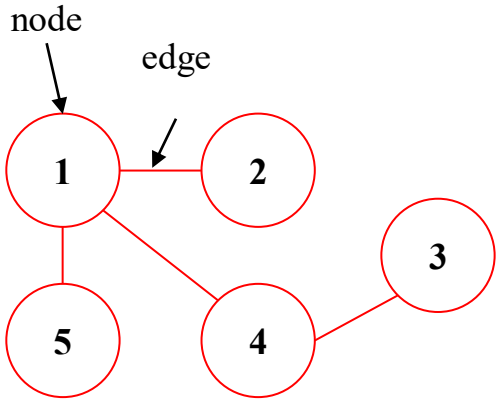
- Previous GCN works
- Major Architectures
  - PoseNet
  - MeshNet
- Implementation
- Discussion and Conclusions
- References

본 논문 reference:

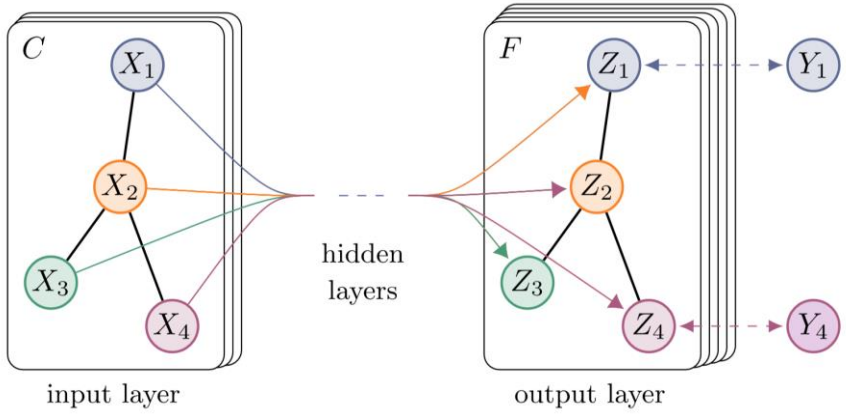
Choi H, Moon G, Lee KM. Pose2Mesh: Graph convolutional network for 3D human pose and mesh recovery from a 2D human pose. InEuropean Conference on Computer Vision 2020 Aug 23 (pp. 769-787). Springer, Cham.

Previous GCN Works

- Kipf의 연구 [1]
  - 수식
    - A: adjacency matrix, D: degree matrix
    - $A = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$, $D = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$
    - Laplacian matrix 와 유사한 행렬 이용 => convolution 효과
    - $H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)})$
    - $H^{(l)}, W^{(l)}$: feature map, weights of layer l
    - $\sigma(\cdot)$: activation function
    - $\tilde{A} = A + I_N, \tilde{D} = D + I_N$



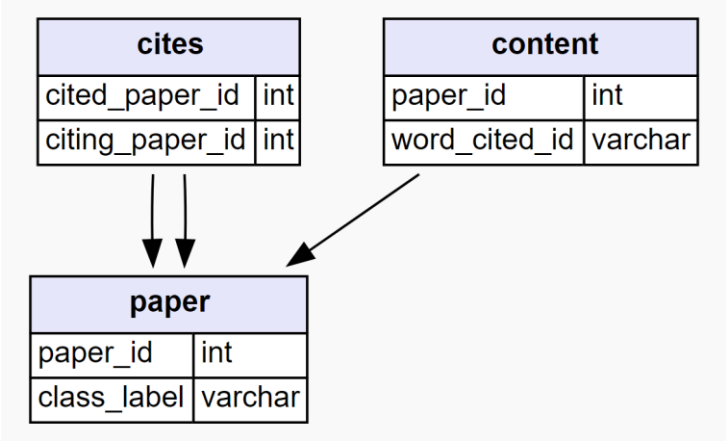
Nodes with edges



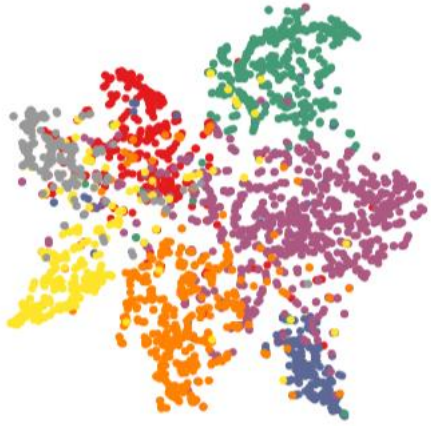
Schematic of depiction of multi-layer GCN

Previous GCN Works

- Kipf의 연구 [1]
  - 결과물
    - 각 node가 어떠한 class에 속하는지 확인
    - 해당 논문에서는 5%의 node에 label을 붙인 후 training
    - 나머지 node가 어떠한 label을 가지고 있는지 feature와 edge를 기반으로 예측
  - 의의 및 한계점
    - GCN을 CNN의 대체제로써 제시
    - GNN의 node간 데이터 흐름을 convolution의 영역에 들여놓음
    - Node 자체의 특성을 찾아내느라, node의 형태를 inference 끝까지 유지하는 수 밖에 없음
    - Undirected graph에 한정
      - 현재는 작가 스스로 간단한 해결책을 제안한 상황



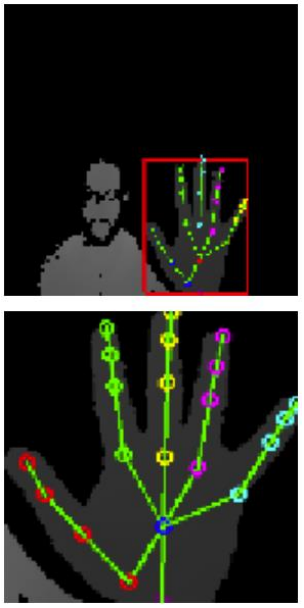
Structure of dataset "*Cora*" [2]



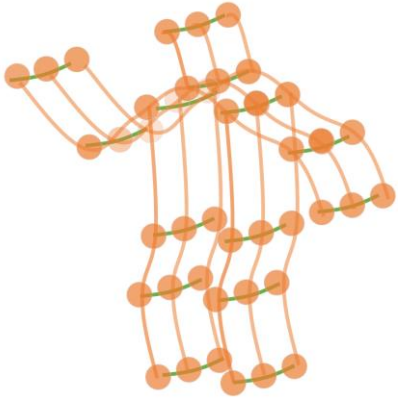
Visualization of hidden layer activations of two-layer GCN trained on the Cora dataset using 5% of labels

Previous GCN Works

- Gesture recognition
  - 사용 예
    - 머리, 몸, 팔, 그리고 다리 관절의 구분을 통해 body gesture recognition
    - 손바닥과 손가락의 각 관절의 구분을 통해 hand gesture recognition
  - 원리
    - Image로 부터 body 및 hand detect
    - Body 및 hand의 관절을 node, 뼈를 edge로 하는 skeleton 구성
    - 하나의 GCN의 channel에 같은 시간에 수집 된 skeleton 정보를 입력
  - Skeleton action recognition 관련 best method 사이트
    - https://paperswithcode.com/task/skeleton-based-action-recognition



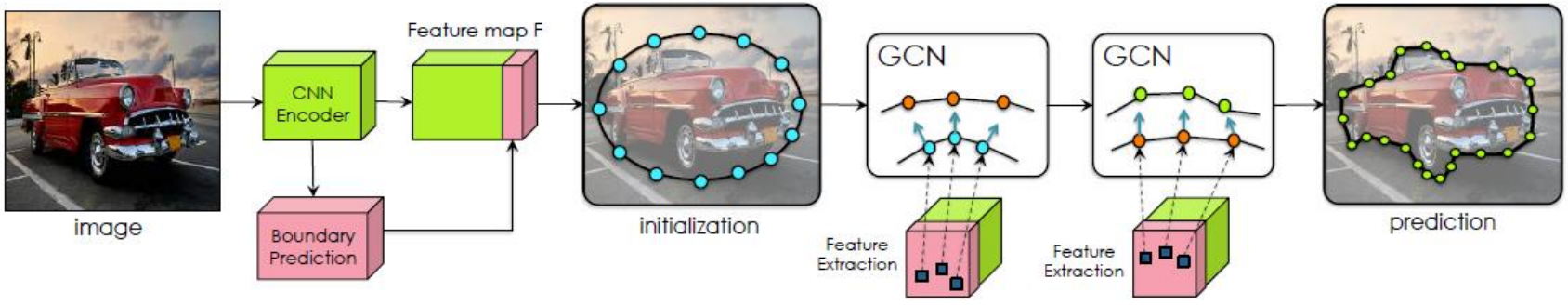
Hand detection and skeleton extraction [3]



GCN on body gesture classification [4]

Previous GCN Works

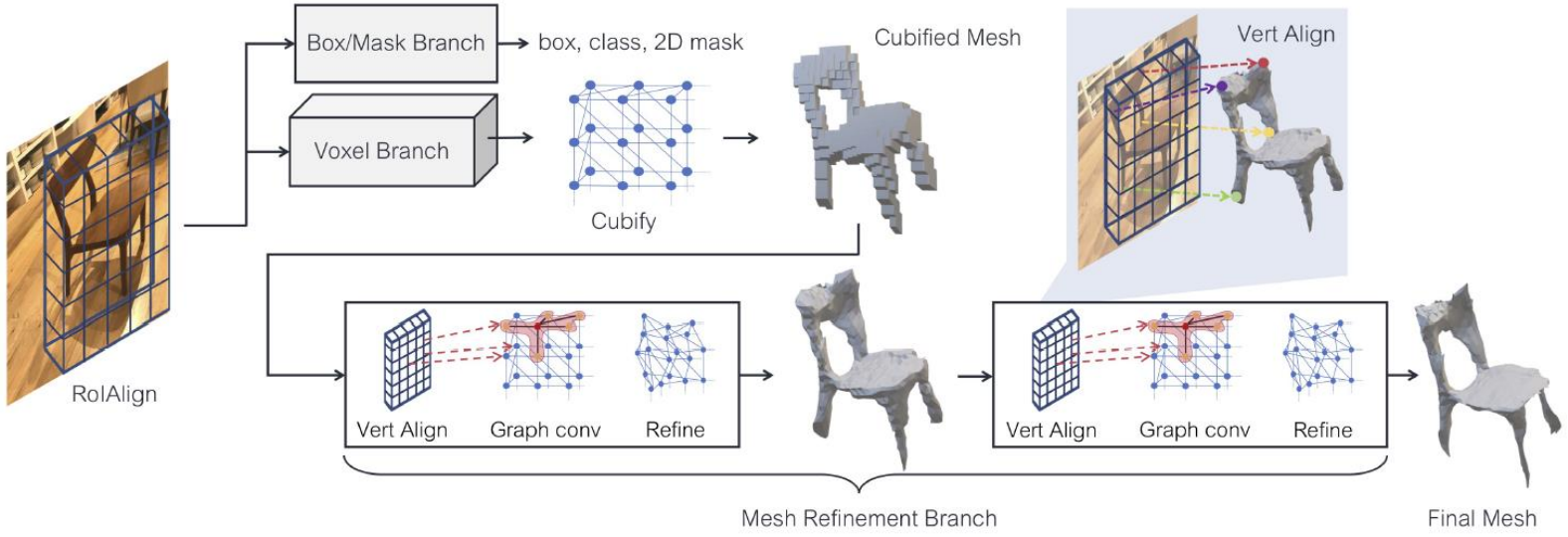
- 2D segmentation [5]
  - Initialization
    - 원형 배치된 initial node 형성 후, 정확한 형태를 찾아 감
  - 연산
    - Image를 encode하고, 이것을 predict된 boundary와 concatenate 시켜 진행
    - Boundary prediction을 진행하는 channel을 분류하여 진행



GCN을 이용한 2D segmentation

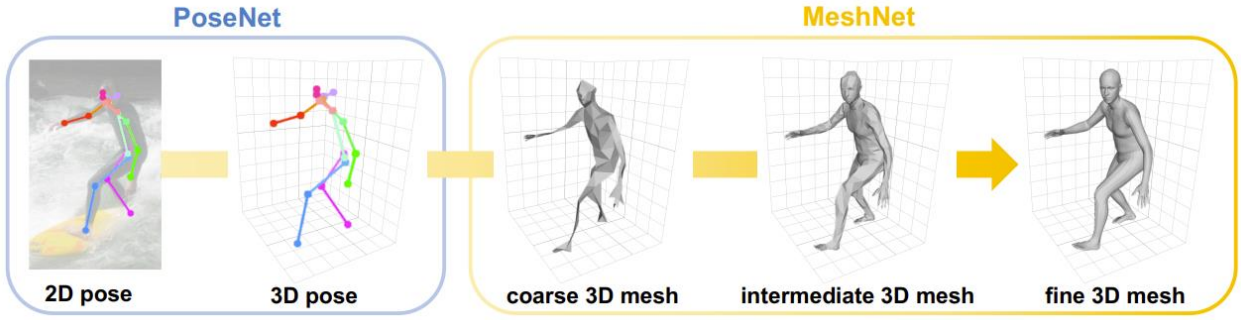
Previous GCN Works

- Mesh-RCNN [6, 7]
  - 구형의 mesh를 입력된 image에 맞게 조각
  - Image의 align을 통해 initialization 진행
  - Align된 graph를 GCN을 이용해 refine진행



Mesh-RCNN의 진행도

Major Architectures - PoseNet



Pose2Mesh Network 진행도

- 기초
  - PoseNet은 3D pose $\boldsymbol{P}^{3D} \in \boldsymbol{R}^{J\times 3}$를 2D pose로 부터 생성
    - J : human joint(node)들의 개수
  - 몸과 손의 joint를 pelvis와 wrist로 정의
- 2D pose input에서의 error synthesize
  - 추출된 2D pose는 occlusion과 challenging pose에 의해 error를 함유할 가능성 상승
  - Realistic error ground truth에 더하여 2D input pose를 synthesize [8, 9]
    - Training stage에 진행
    - Synthesize 된 2D pose를 $\boldsymbol{P}^{2D} \in \boldsymbol{R}^{J\times 2}$로 표현

Major Architectures - PoseNet

- 2D input pose normalization [8, 10]
  - $\boldsymbol{P}^{2D}$에서 mean subtraction 및 standard deviation division 진행 => $\bar{\boldsymbol{P}}^{2D}$ 생성
  - $\boldsymbol{P}^{2D}$의 mean, standard deviation은 subject의 2D location 나타냄
  - $\boldsymbol{P}^{3D}$의 $\boldsymbol{P}^{2D}$로 부터의 scale 및 location 독립성으로 인해 해당 과정은 필수적
- Network Architecture [8, 11]
  - Normalized 2D input pose $\bar{\boldsymbol{P}}^{2D}$ 가 fully-connected layer를 통해 4096-dimension feature vector로 변환
  - 이후 2개의 residual block들에 입력 [12]
  - Residual block들의 output feature vector들이 fully-connected layer로 인해 (3J)-dimensional vector $\boldsymbol{P}^{3D}$로 변환
- Loss function
  - $\boldsymbol{P}^{3D}$의 ground truth 와 비교
  - $L_{pose} = \left\| \boldsymbol{P}^{3D} - \boldsymbol{P}^{3D^*} \right\|_1$

Major Architectures - MeshNet

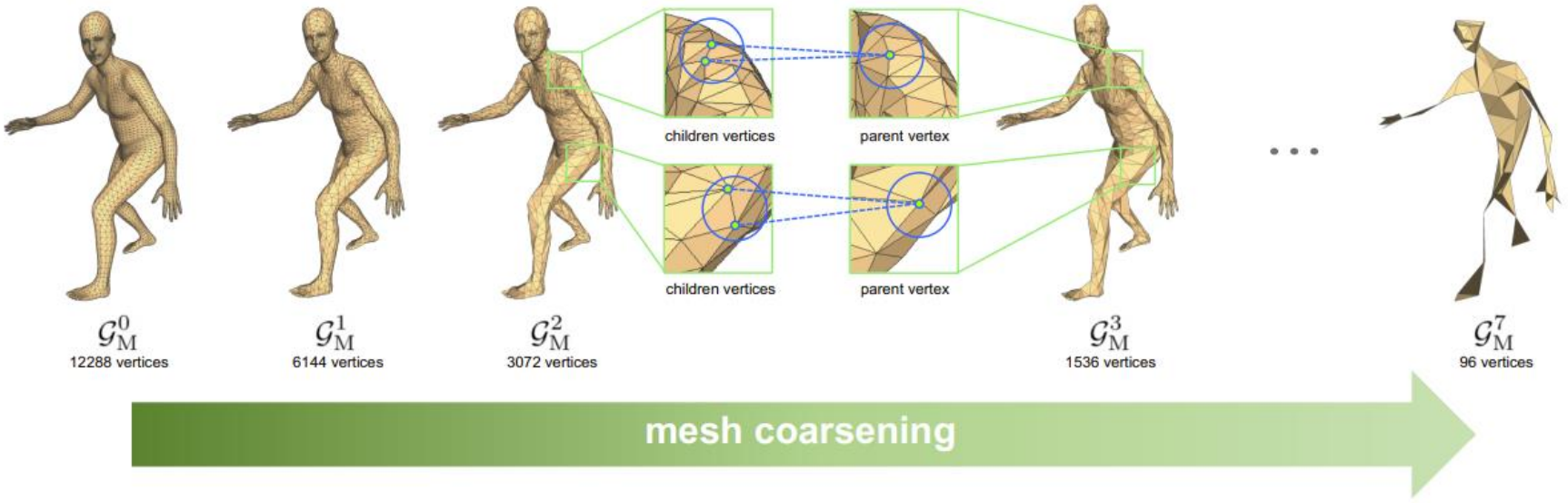
- Graph convolution on pose
  - 기초
    - $\bar{\boldsymbol{P}}^{2D}$와 $\boldsymbol{P}^{3D}$ concatenate 시켜 $\boldsymbol{P} \in R^{J\times 5}$ 형성
    - 3D mesh $\boldsymbol{M} \in R^{V\times 3}$을 $\boldsymbol{P}$로 부터 estimate
      - V: human mesh vertex들 의미
    - Spectral graph convolution을 이용하여 수행 [13, 14]
  - Graph construction
    - $\boldsymbol{P}$를 근거로 형성되는 graph, $\mathcal{G}_P = (\mathcal{V}_P, A_P)$를 construct
      - $\mathcal{V}_P = \boldsymbol{P} = \{\boldsymbol{p_i}\}_{i=1}^{J}$: Set of J human joints
      - $A_P \in \{\mathbf{0}, \mathbf{1}\}^{J\times J}$: Adjacency matrix
    - Normalized Laplacian: $L_P = I_J - D_P^{-1/2} A_P D_P^{-1/2}$
      - I_J: identity matrix with J dimensions
      - D_P: diagonal matrix which $(D_P)_{ij} = \sum_j (A_P)_{ij}$
    - Scaled Laplacian: $\widetilde{L_P} = \frac{2L_P}{\lambda_{max}} - I_J$

Major Architectures - MeshNet

- Graph convolution on pose
  - Spectral convolution on graph: $F_{out} = \sum_{k=0}^{K-1} T_k(\widetilde{L_P}) F_{in} \theta_k$
    - $F_{in} \in R^{J \times f_{in}}$, $F_{out} \in R^{J \times f_{out}}$: Input, output feature maps
    - f_{in}, f_{out}: Input, output feature의 dimension
    - $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$: kth Chebyshev polynomial [15]
      - Graph coarsening을 위한 장치
    - $\theta_k \in R^{f_{in} \times f_{out}}$: kth Chebyshev coefficient matrix
      - Element들이 학습 가능한 parameter들임
  - Convolution in detail
    - Initial input feature map $F_{in} = \boldsymbol{P}$, $f_{in} = 5$
    - K-hop neighbor까지 영향을 받음 (K-localized) [12, 15]
      - Laplacian의 K-order polynomial이기 때문
    - MeshNet에서는 $K = 3$으로 설정

Major Architectures - MeshNet

- Coarse-to-fine mesh upsampling
  - Upsampling 과정



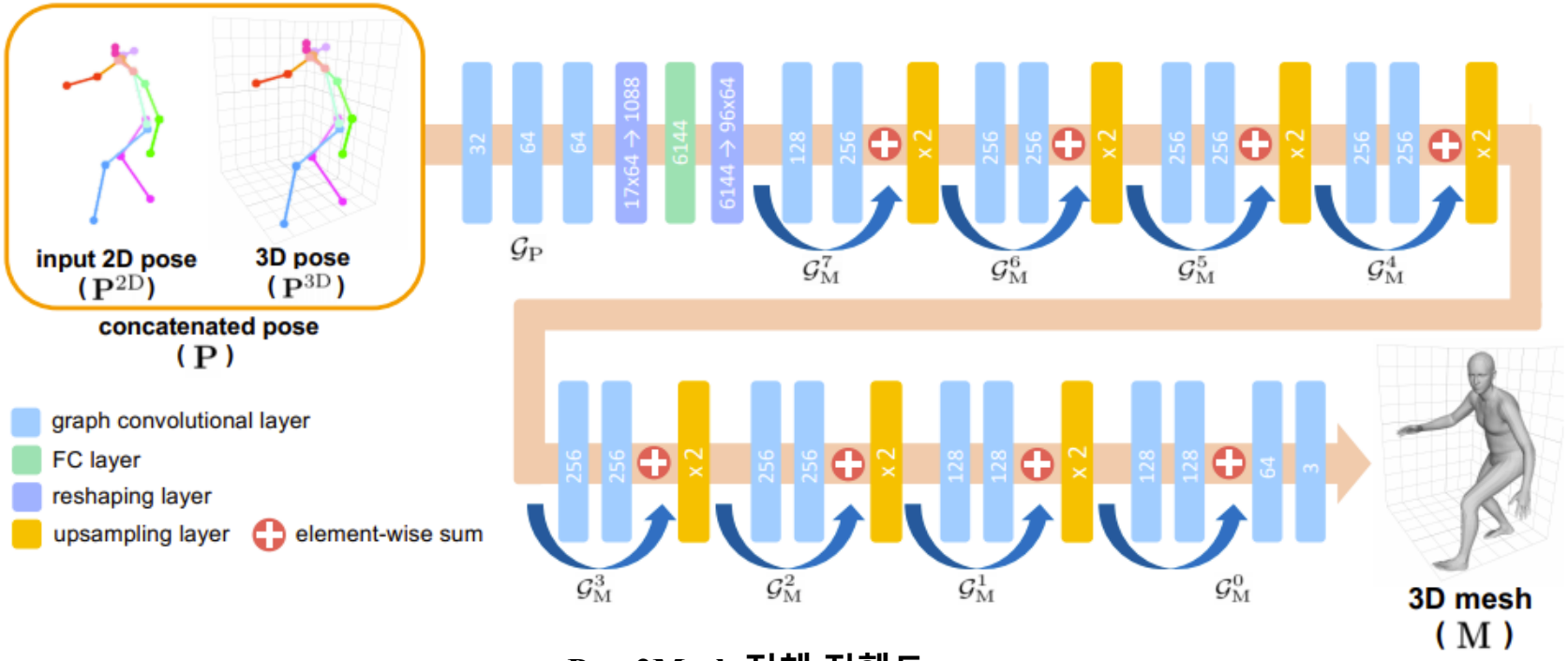
Upsampling 의 진행도

Major Architectures - MeshNet

- Coarse-to-fine mesh upsampling
  - Upsampling를 위한 정의
    - $\mathcal{G}_{\boldsymbol{M}} = (\mathcal{V}_{\boldsymbol{M}}, A_{\boldsymbol{M}})$를 construct
      - $\mathcal{V}_{\boldsymbol{M}} = \boldsymbol{M} = \{\boldsymbol{m_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{V}}$: Set of $\boldsymbol{V}$ human mesh vertices
      - $A_{\boldsymbol{M}} \in \{\boldsymbol{0}, \boldsymbol{1}\}^{V \times V}$: Adjacency matrix defining edges of the human mesh
    - 그래프의 resolution에 따른 수식: $\{\mathcal{G}_M{}^{c} = (\mathcal{V}_{\boldsymbol{M}}{}^{c}, A_{\boldsymbol{M}}{}^{c})\}_{c=0}^{C}$
      - C : coarsening step 개수
      - $\mathcal{G}_M{}^{c+1}$의 ith vertex : $\mathcal{G}_M{}^{c}$의 $2i-1$th, $2i$th vertex들의 parent node
      - $2|\mathcal{V}_{\boldsymbol{M}}{}^{c+1}| = |\mathcal{V}_{\boldsymbol{M}}{}^{c}|$
      - i는 1에서 부터 시작
    - MeshNet의 최종 output: $\mathcal{V}_{\boldsymbol{M}}$
      - Reshaping과 fully-connected layer 에 의함

Major Architectures - MeshNet

- Coarse-to-fine mesh upsampling
  - 수반되는 연산 전개도



Pose2Mesh 전체 진행도

Major Architectures - MeshNet

- Coarse-to-fine mesh upsampling
  - Upsampling에 수반되는 연산
    - 각 단계의 mesh마다 graph convolution 수행
      - $F_{out} = \sum_{k=0}^{K-1} T_k(\widetilde{L_M}^c) F_{in} \theta_k$ 연산
      - $\widetilde{L_M}^c$: ${\mathcal{G}_M}^c$의 scaled Laplacian
    - Upsampling, $F_c = \varphi({F_{c+1}}^T)^T$
      - Parent vertex ${\mathcal{G}_M}^{c+1}$의 feature를 대응되는 children ${\mathcal{G}_M}^c$의 vertex들에 복사
        - ✓${\mathcal{G}_M}^{c+1}$의 ith vertex feature -> ${\mathcal{G}_M}^c$의 $2i-1$th, $2i$th vertex들로 복사
      - $F_c \in \boldsymbol{R}^{{\mathcal{V}_M}^c \times f_c}$: ${\mathcal{G}_M}^c$의 첫 feature map, $F_{c+1} \in \boldsymbol{R}^{{\mathcal{V}_M}^{c+1} \times f_{c+1}}$: ${\mathcal{G}_M}^{c+1}$의 마지막 feature map
      - φ: $\boldsymbol{R}^{f_{c+1} \times {\mathcal{V}_M}^{c+1}} \rightarrow \boldsymbol{R}^{f_c \times {\mathcal{V}_M}^c}$, nearest-neighbor upsampling function 의미
      - f_c, f_{c+1}: vertex들의 feature dimension들

Major Architectures - MeshNet

- Loss functions, 4가지 loss function들의 이용
  - Vertex coordinate loss
    - 3D mesh coordinate들인 $\boldsymbol{M}$의 ground truth간 $L1$ distance를 최소화
    - $L_{vertex} = \|\boldsymbol{M} - \boldsymbol{M}^*\|_1$
  - Joint coordinate loss
    - 3D pose ground truth와, $\boldsymbol{M}$으로부터의 regression으로 형성된 3D pose간 비교
    - $L_{joint} = \left\|\mathcal{J}\boldsymbol{M} - \boldsymbol{P}^{3D^*}\right\|_1$
    - $\mathcal{J} \in \boldsymbol{R}^{J \times V}$: joint regression matrix (defined in SMPL(몸체), MANO(손) model)
  - Surface normal loss
    - 출력 mesh 표면의 normal vector들이 ground truth에 대해 consistent 하도록 supervise
    - Surface smoothness와 local detail들을 개선 [16]
    - $L_{normal} = \sum_f \sum_{\{i,j\} \subset f} \left| \left\langle \frac{m_i - m_j}{\|m_i - m_j\|_2}, n_f{}^* \right\rangle \right|$
    - f, $n_f{}^*$: Human mesh의 triangle face, ground truth unit normal vector of f
    - m_i, m_j: f의 ith, jth vertex

Major Architectures - MeshNet

- Loss functions, 4가지 loss function들의 이용
  - Surface edge loss
    - Predicted edge와 ground truth edge간 비교 [16]
    - Vertex가 촘촘하게 위치해 있는 손, 발, 입의 smoothness를 recover하는데 effective함
    - $L_{edge} = \sum_f \sum_{\{i,j\} \subset f} \left| \|m_i - m_j\|_2 - \|m_i^* - m_j^*\|_2 \right|$
    - $f, *$: Human mesh의 triangle face, ground truth unit normal vector of f
    - m_i, m_j: f의 ith, jth vertex
  - Total loss function
    - $L_{mesh} = \lambda_v L_{vertex} + \lambda_j L_{joint} + \lambda_n L_{normal} + \lambda_e L_{edge}$
    - $\lambda_v = 1, \lambda_j = 1, \lambda_n = 0.1, \lambda_e = 20$

Implementation

- Datasets and evaluation metrics
  - Human3.6M
    - Large-scale indoor 3D body pose benchmark [17]
    - Ground truth가 제공되지 않으므로, SMPLify-X [18]를 이용하여 pseudo ground truth 생성
    - 두가지 metric을 이용하여 3D pose의 performance evaluate
      - Mean per joint position error (MPJPE) [17]: Estimated 수치와 ground truth의 joint coordinate들간 Euclidean distance (milimeter단위)
      - PA-MPJPE [19]: $\mathcal{JM}$이 이용되며, 총 17개의 joint중 14개 이용
        - ✓PA: Procrustes Analysis

Implementation

- Ablation study
  - Regression target and network design
    - Settings
      - Fully-connected(FC) 및 GraphCNN 두 네트워크와 SMPL parameter (joint coordinate loss) 및 vertex coordintates 두 regression target에 따른 결과
    - 결과 분석
      - Vertex-FC와 vertex-GraphCNN간 성능의 차이: 3D vertex estimation에서 human mesh topology exploitation의 중요성 보여 줌
      - Vertex-GraphCNN이 두 SMPL parameter estimation 진행 network보다 성능이 높고, parameter 개수 적음: 제안 loss function의 타당성 보여 줌

Regression target and network design result

network /target	FC			GraphCNN		
	MPJPE	PA-MPJPE	no. param.	MPJPE	PA-MPJPE	no. param.
SMPL param.	72.8	55.5	17.3M	79.1	59.1	13.5M
vertex coord.	119.6	95.1	37.5M	**64.9**	**48.0**	**8.8M**

Implementation

- Ablation study
  - Coarse-to-fine mesh upsampling
    - Direct upsampling
      - middle layer까지 graph convolution 수행 후 최고 mesh까지 바로 upsampling 진행
      - Graph convolution layer 개수는 coarse-to-fine method와 일치
    - 결과 분석
      - GPU memory: 절반 가량 이용, fps: 1.5배 빠름
      - 연산에 성능 향상 이유: High resolution의 graph convolution이 시간과 memory를 더 필요로 하기 때문
  - Cascaded architecture analysis
    - 3D pose만을 이용하는 것은 오히려 performance가 떨어짐
    - 2D pose에 3D pose를 concatenate 시켜 씀으로써 geometry information이 추가되어 연산이 진행 된 것으로 볼 수 있음

Upsampling 방법에 따른 결과

method	GPU mem.	fps	MPJPE
direct	10G	24	65.3
coarse-to-fine	**6G**	**37**	**64.9**

Mesh 형성에 쓰이는 pose data에 따른 결과

architecture	MPJPE
2D -> mesh	101.1
2D -> 3D -> mesh	103.2
2D -> 3D+2D -> mesh	**100.5**

Implementation

- 정성적 결과

Pose2Mesh를 이용하여 얻은 정성적 결과

Discussion and Conclusions

- Discussion
  - 제안된 system이 입력된 2D pose로 부터 다양한 domain을 가진 geometric property들을 이용하지만, 다양하게 존재하는 3D shape를 pose만으로 부터 복원하는 것은 쉽지 않음
  - 그럼에도 불구하고, 2D pose는 이에 상응하는 3D shape를 예측하는데 필수적인 정보를 가지고 있음
- Conclusion
  - Input 2D pose는 시스템이 appearance domain gap issue에 의한 영향 없이 3D data를 얻을 수 있도록 해 줌
  - GraphCNN을 이용한 model-free approach는 mesh topology를 완전히 exploit할 수 있게 해주고, 3D rotation parameter들로 인한 representation issue를 피할 수 있게 함
  - 추후 더 조밀한 key point들과 part segmentation을 Pose2Mesh에 적용하여 shape를 enhance할 예정

References

[1] Kipf TN, Welling M. Semi-Supervised Classification with Graph Convolutional Networks.
[2] Sen P, Namata G, Bilgic M, Getoor L, Gallagher B, Eliassi-Rad T. Collective Classification in Network Data. AI Magazine. 2008 Oct 1;29(3):93.
[3] Do NT, Kim SH, Yang HJ, Lee GS. Robust Hand Shape Features for Dynamic Hand Gesture Recognition Using Multi-Level Feature LSTM. Applied Sciences. 2020 Jan;10(18):6293.
[4] Shi L, Zhang Y, Cheng J, Lu H. Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. IEEE Transactions on Image Processing. 2020 Oct 9;29:9532-45.
[5] Ling H, Gao J, Kar A, Chen W, Fidler S. Fast Interactive Object Annotation With Curve-GCN. In2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2019 Jun 1 (pp. 5252-5261).
[6] Smith E, Fujimoto S, Romero A, Meger D. GEOMetrics: Exploiting Geometric Structure for Graph-Encoded Objects. InInternational Conference on Machine Learning (ICML) 2019 May 24 (pp. 5866-5876).
[7] Gkioxari G, Johnson J, Malik J. Mesh R-CNN. In2019 IEEE/CVF International Conference on Computer Vision (ICCV) 2019 Jun 6 (pp. 9784-9794).
[8] Chang JY, Moon G, Lee KM. AbsPoseLifter: Absolute 3D Human Pose Lifting Network from a Single Noisy 2D Human Pose.
[9] Moon G, Chang JY, Lee KM. PoseFix: Model-Agnostic General Human Pose Refinement Network. In2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2019 Jun 15 (pp. 7765-7773).

References

[10] Wandt B, Rosenhahn B. RepNet: Weakly Supervised Training of an Adversarial Reprojection Network for 3D Human Pose Estimation. In2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2019 Jun 15 (pp. 7774-7783).
[11] Martinez J, Hossain R, Romero J, Little JJ. A Simple Yet Effective Baseline for 3d Human Pose Estimation. In2017 IEEE International Conference on Computer Vision (ICCV) 2017 Oct 22 (pp. 2659-2668).
[12] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition.
[13] Bruna J, Zaremba W, Szlam A, Lecun Y. Spectral networks and locally connected networks on graphs. InInternational Conference on Learning Representations (ICLR), 2014 Apr.
[14] Shuman DI, Narang SK, Frossard P, Ortega A, Vandergheynst P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE signal processing magazine. 2013 Apr 5;30(3):83-98.
[15] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. InProceedings of the 30th International Conference on Neural Information Processing Systems (NIPS), 2016 Dec 5 (pp. 3844-3852).
[16] Wang N, Zhang Y, Li Z, Fu Y, Liu W, Jiang YG. Pixel2mesh: Generating 3d mesh models from single rgb images. InProceedings of the European Conference on Computer Vision (ECCV) 2018 (pp. 52-67).
[17] Ionescu C, Papava D, Olaru V, Sminchisescu C. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE transactions on pattern analysis and machine intelligence. 2013 Dec 12;36(7):1325-39.

References

[18] Pavlakos G, Choutas V, Ghorbani N, Bolkart T, Osman AA, Tzionas D, Black MJ. Expressive Body Capture: 3D Hands, Face, and Body From a Single Image. In2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2019 Jun 1 (pp. 10967-10977).
[19] Gower JC. Generalized procrustes analysis. Psychometrika. 1975 Mar 1;40(1):33-51.