Pose2Mesh

Graph Convolutional Network for 3D Human Pose and Mesh Recovery from 2D Human Pose

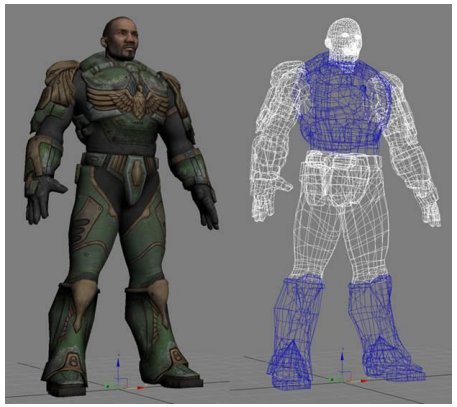
made by 최홍석 (Hongsuk Choi)
2020.06.19

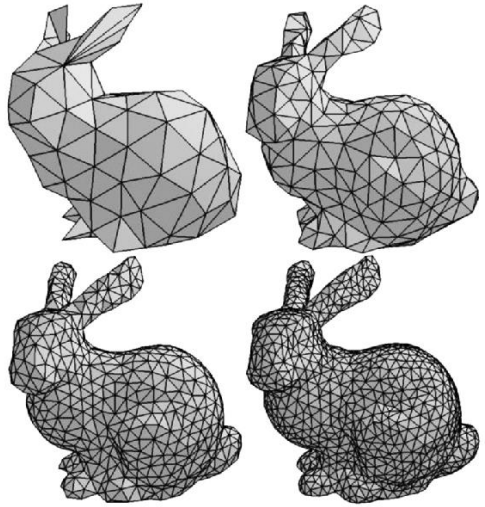
Introduction

- The goal of this paper is to estimate a 3D human pose and a naked human mesh from a single-view image

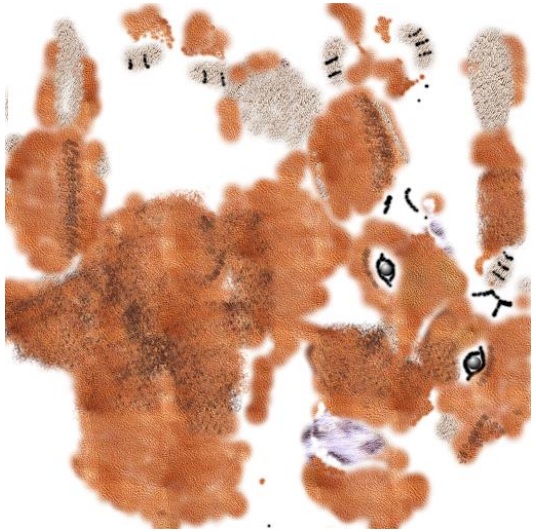
Introduction

- Mesh
  - Compact with a pre-defined topology
    - 3D xyz coordinates of vertices: R^{V x 3}
    - Vertex indices of each xyz triangle face: R^{F x 3}
    - 2D uv coordinates: R^{V` x 2}
    - Vertex indices of each uv triangle face: R^{F x 3}
    - Texture: R^{H x W}
  - Standard in computer game and movie industry





Mesh geometry (3D xyz)



Mesh texture (2D uv)

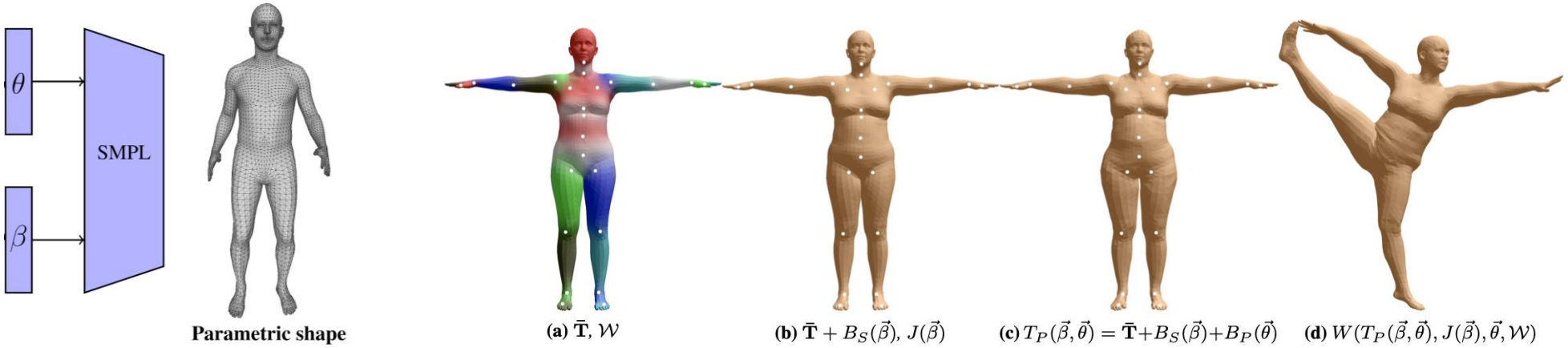
Introduction

- Applications
  - VR/AR games, virtual communication etc.

Deep Appearance Models for Face Rendering (2018 SIGGRAPH)

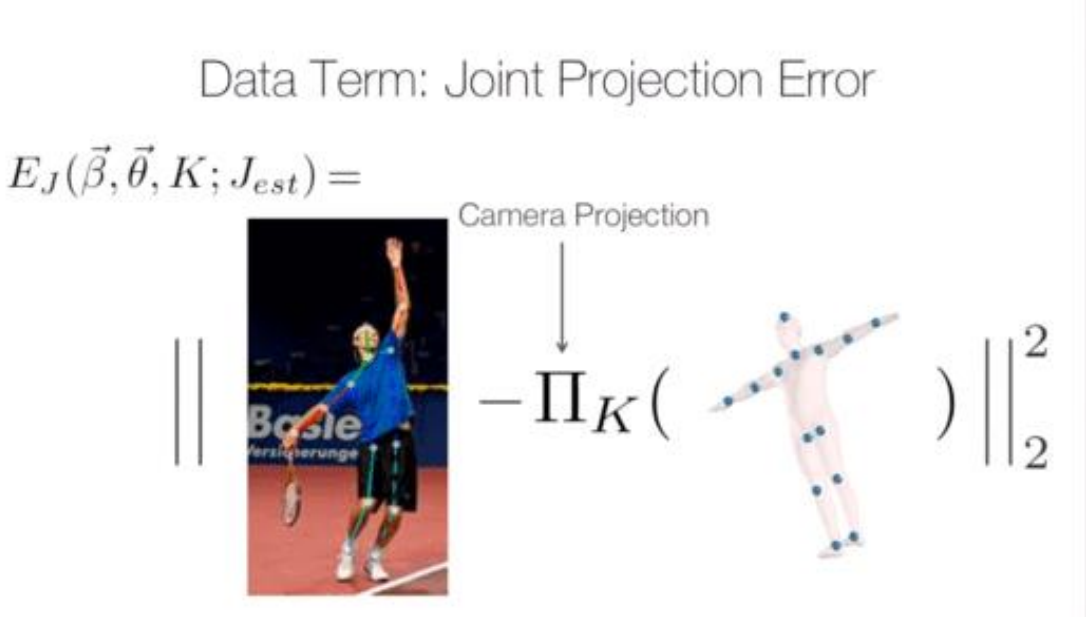
Related works

- SMPL: a skinned multi-person linear model (SIGGRAPH 2015)
- SMPL is parameterized with pose and identity parameters
- It takes pose parameters (72 dimensional 3D rotations of keypoints) and identity parameters (10 dimensional PCA coefficients) and outputs corresponding mesh vertices (consistent topology)
- SMPL supports various features such as correctives, but I'll skip them



Related works

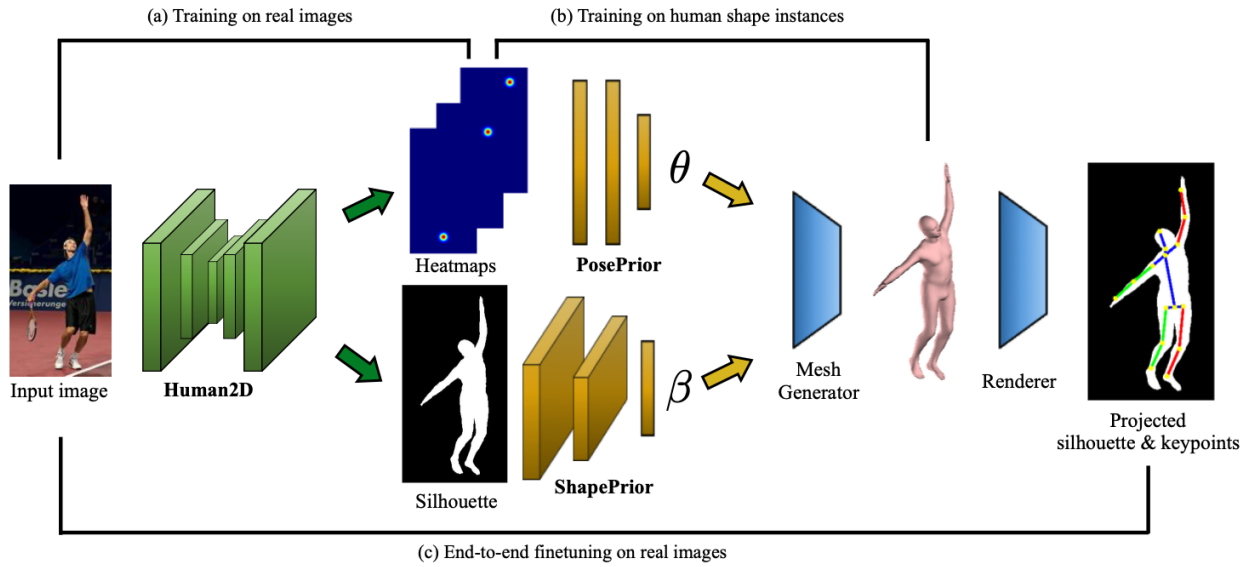
- Optimization-based approach
  - Given a parametric model of the human body model (ex. SMPL), an iterative fitting approach attempts to estimate the body pose and shape that best explains 2D observations, most typically 2D joint locations
  - Output is well-aligned with image / Very slow and sensitive to initialization



SMPLify: Automatic Estimation of 3D Human Pose and Shape from a Single Image (2016 ECCV)

Related works

- Learning-based approach
  - Most of the works estimate SMPL parameters (model-based)
  - Can be weakly-supervised by pose reprojection loss or fully-supervised by parameter supervision



Learning to Estimate 3D Human Pose and Shape from a Single Color Image (2018 CVPR)

Challenges

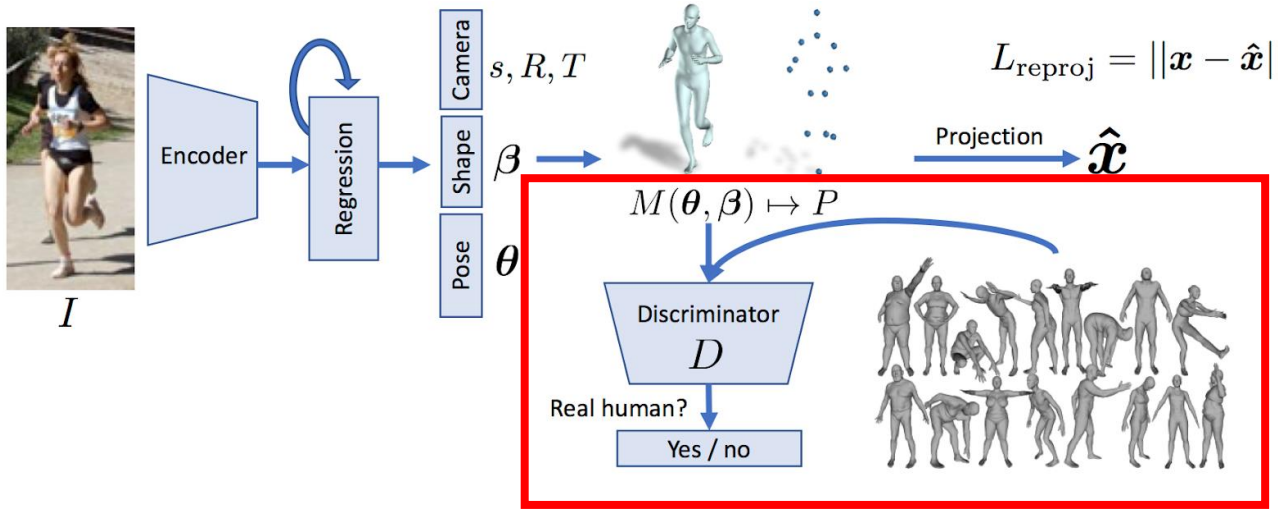
- Optimization-based approach is very slow
- Learning-based approach suffers from lack of 3D mesh GT from in-the-wild dataset
- Image appearance of controlled environment data and in-the-wild data are very different

Controlled environment dataset image

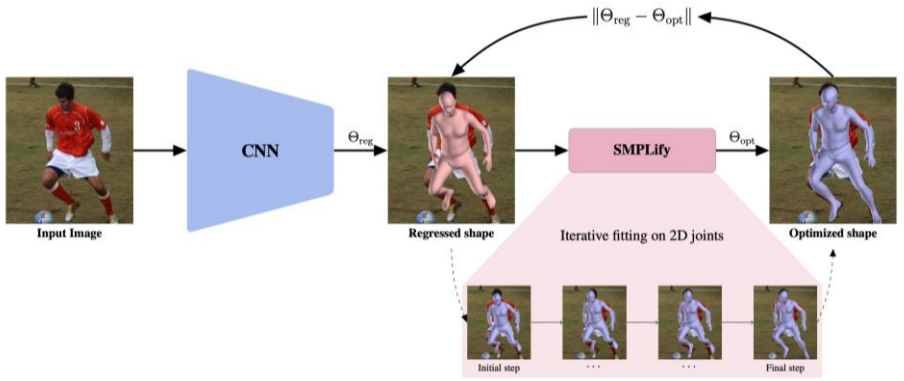
In-the-wild image

Related works

- Adversarial learning
  - Utilize unpaired 3D mesh GT from mocap dataset



- SPIN: optimization + learning
  - initialize an optimization with a learning-based method output, and supervise the learning-based method with the optimization output



Motivation

- Still there are not enough in-the-wild data with 3D mesh GT
- Overfitting issue from image appearance difference between in-the-wild data and controlled environment data remains
- Only mixed-batch training is used to alleviate it

- Motivation1:

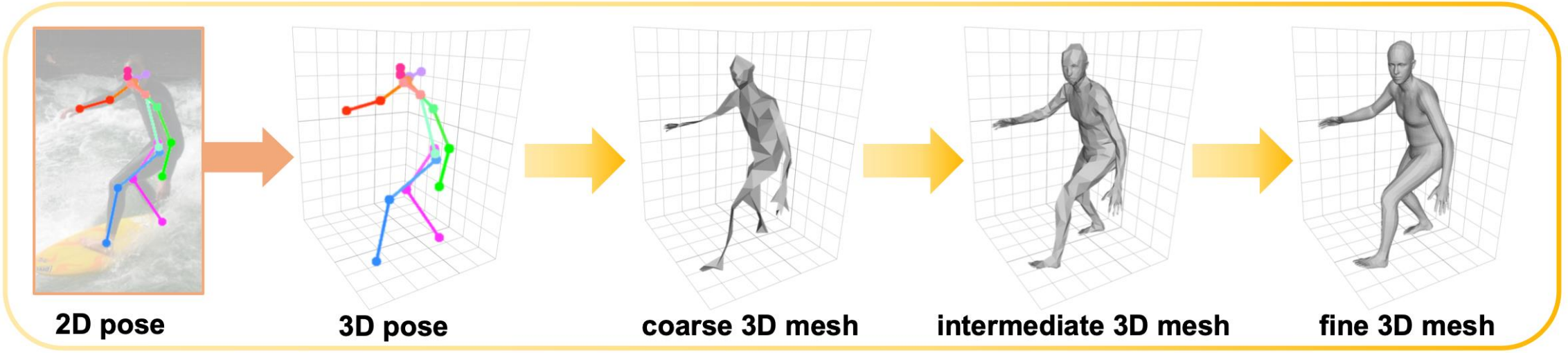
SOTA 2D pose estimators are trained on large-scale in-the-wild dataset

- Motivation2:

2D pose is free from image appearance difference, but contain essential human geometry information

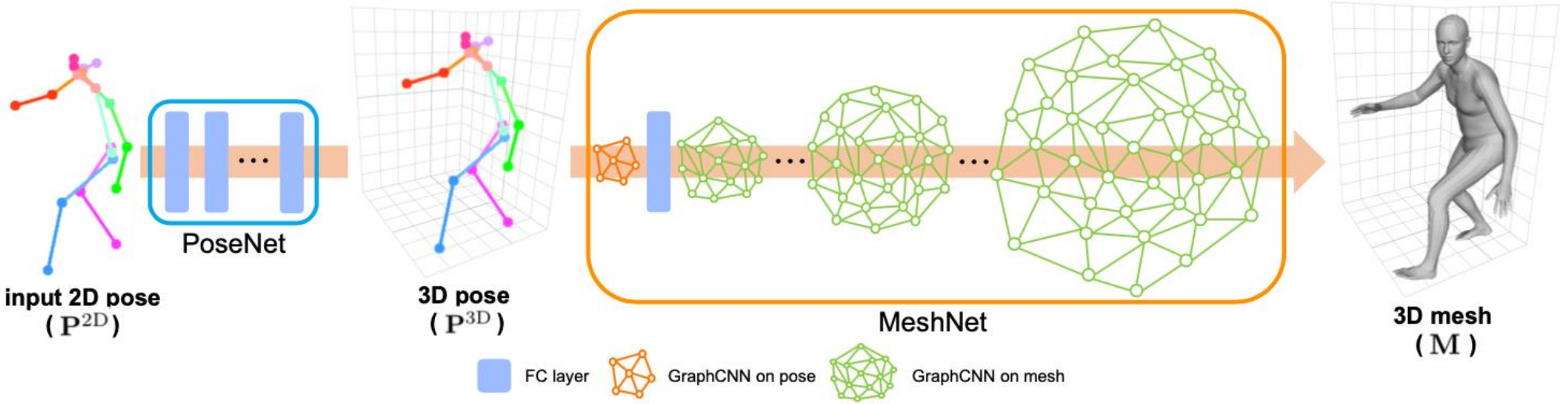
Proposed method - concept

- We decouple the task into image ⇨ 2D pose and 2D pose ⇨ 3D mesh
- The proposed Pose2Mesh first lifts a 2D human pose to a 3D human pose, and estimates a 3D human mesh from the 3D human pose in a coarse-to-fine manner.



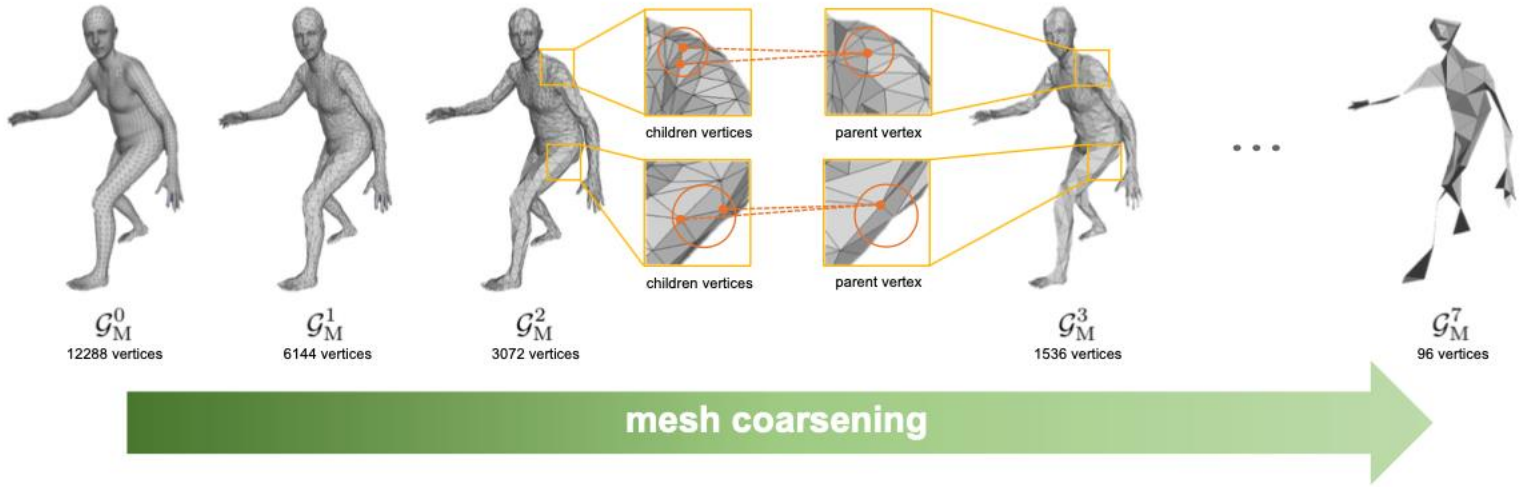
Proposed method - overview

- The input 2D pose from in-the-wild image can be acquired with high accuracy using SOTA 2D pose estimators
- The output 3D mesh coordinates are free from the rotation representation problem in SMPL pose parameters, and MeshNet, which is based on Chebysev GCN (2016 NIPS), can leverage the mesh topology



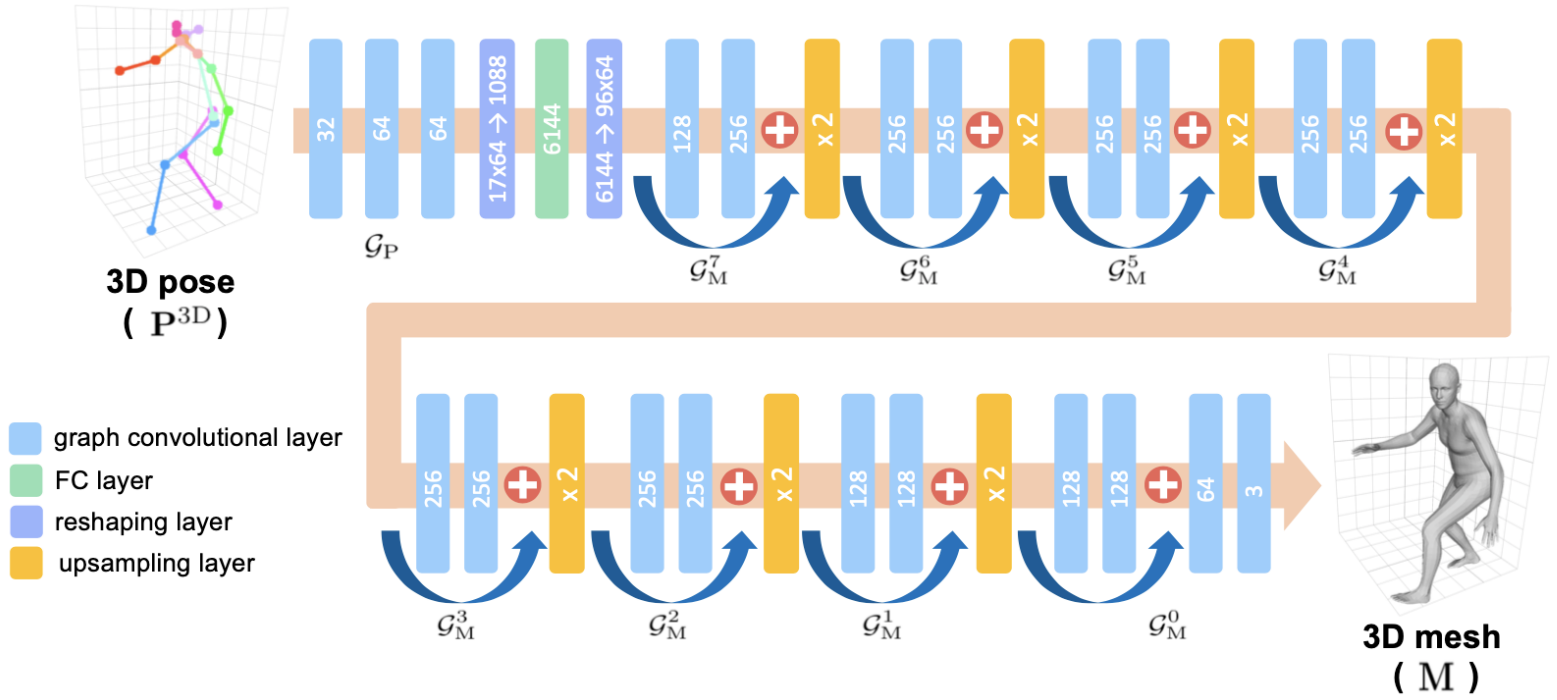
Proposed method

- Coarse-to-fine approach consumes less GPU memory and give higher fps in inference time
- For this, we generate coarse-level mesh topologies using the pooling technique introduced in Chebysev GCN



Proposed method

- Detailed architecture of MeshNet



Proposed method

- Overall loss function

$$L_{\text{mesh}} = \lambda_v L_{\text{vertex}} + \lambda_j L_{\text{joint}} + \lambda_n L_{\text{normal}} + \lambda_e L_{\text{edge}}$$

Primary data term

Regularization for visual quality
(ex. Smoothness)

Experiments

- Datasets
  - Train data

  SMPL-fits on COCO dataset (in-the-wild) and Human3.6M dataset (MoCap)
  AMASS dataset which is a synthetic dataset with GT SMPL parameters

  - Test benchmark

  Human3.6M dataset: 3D joint GT annotations
  3DPW dataset (in-the-wild): 3D joint and vertex GT annotations
  FreiHAND dataset: hand dataset with real images and synthetic images

- Evaluation
  - Mean Per Joint Position Error (MPJPE)
  - Mean Per Vertex Position Error (MPVPE)

Experiments

- Ablation studies
  - Trained and tested on Human3.6M dataset
  - Target representation and network design

target\network	FC			GraphCNN		
	MPJPE	PA-MPJPE	no. of param.	MPJPE	PA-MPJPE	no. of 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**	**48.6**	**8.8M**

  - Coarse-to-fine vs. direct approach

methods	GPU mem.	fps	MPJPE	PA-MPJPE
direct	10G	24	65.3	49.2
coarse-to-fine (Ours)	**6G**	**37**	**64.9**	**48.6**

Experiments

- Comparison with SOTAs
  - Tested on Human3.6M (MoCap)

methods	Human3.6M		Human3.6M + COCO	
	MPJPE	PA-MPJPE	MPJPE	PA-MPJPE
HMR [28]	184.7	88.4	153.2	85.5
GraphCMR [32]	148.0	104.6	78.3	59.5
SPIN [31]	85.6	55.6	72.9	**51.9**
Pose2Mesh (Ours)	**64.9**	**48.6**	**69.6**	52.1

  - Tested on 3DPW (in-the-wild)

methods	Human3.6M		Human3.6M + COCO	
	MPJPE	MPVPE	MPJPE	MPVPE
HMR [28]	377.3	481.0	300.4	406.8
GraphCMR [32]	332.5	380.8	126.5	144.8
SPIN [31]	313.8	344.3	113.1	122.8
Pose2Mesh (Simple [71])	108.1	126.2	105.2	122.7
Pose2Mesh (HR [63])	**105.7**	**123.7**	**103.1**	**120.4**

train sets	MPJPE	MPVPE
Human3.6M	105.7	123.7
+ COCO	103.1	120.4
+ AMASS	**101.4**	**118.4**

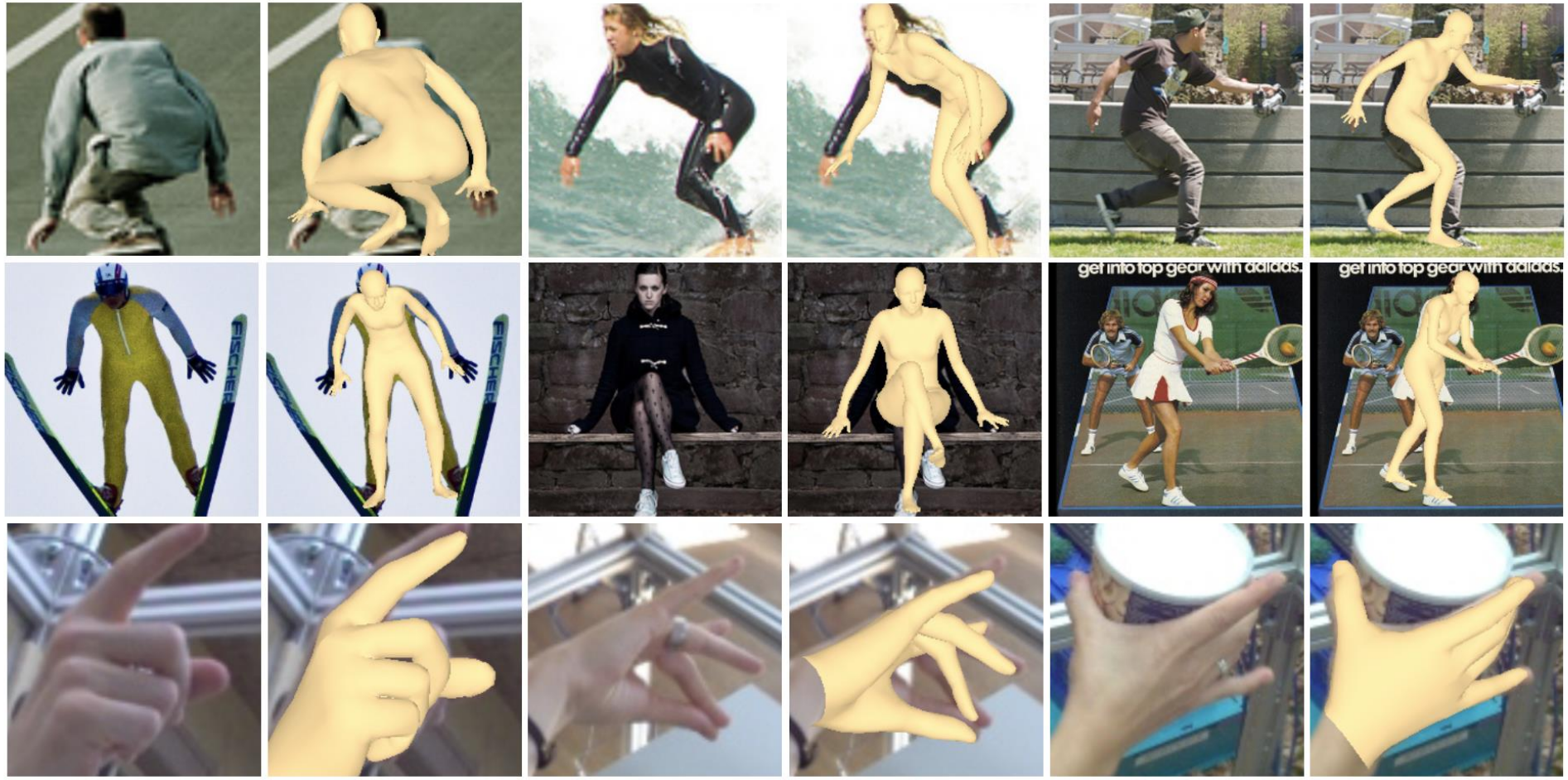
Experiments

- Comparison with SOTAs
  - Tested on FreiHAND

methods	PA-MPVPE	PA-MPJPE	F@5 mm	F@15 mm
Hasson et al. [20]	13.2	-	0.436	0.908
Boukhayma et al. [6]	13.0	-	0.435	0.898
FreiHAND [76]	10.7	-	0.529	0.935
Pose2Mesh (Ours)	**7.8**	**7.7**	**0.674**	**0.969**

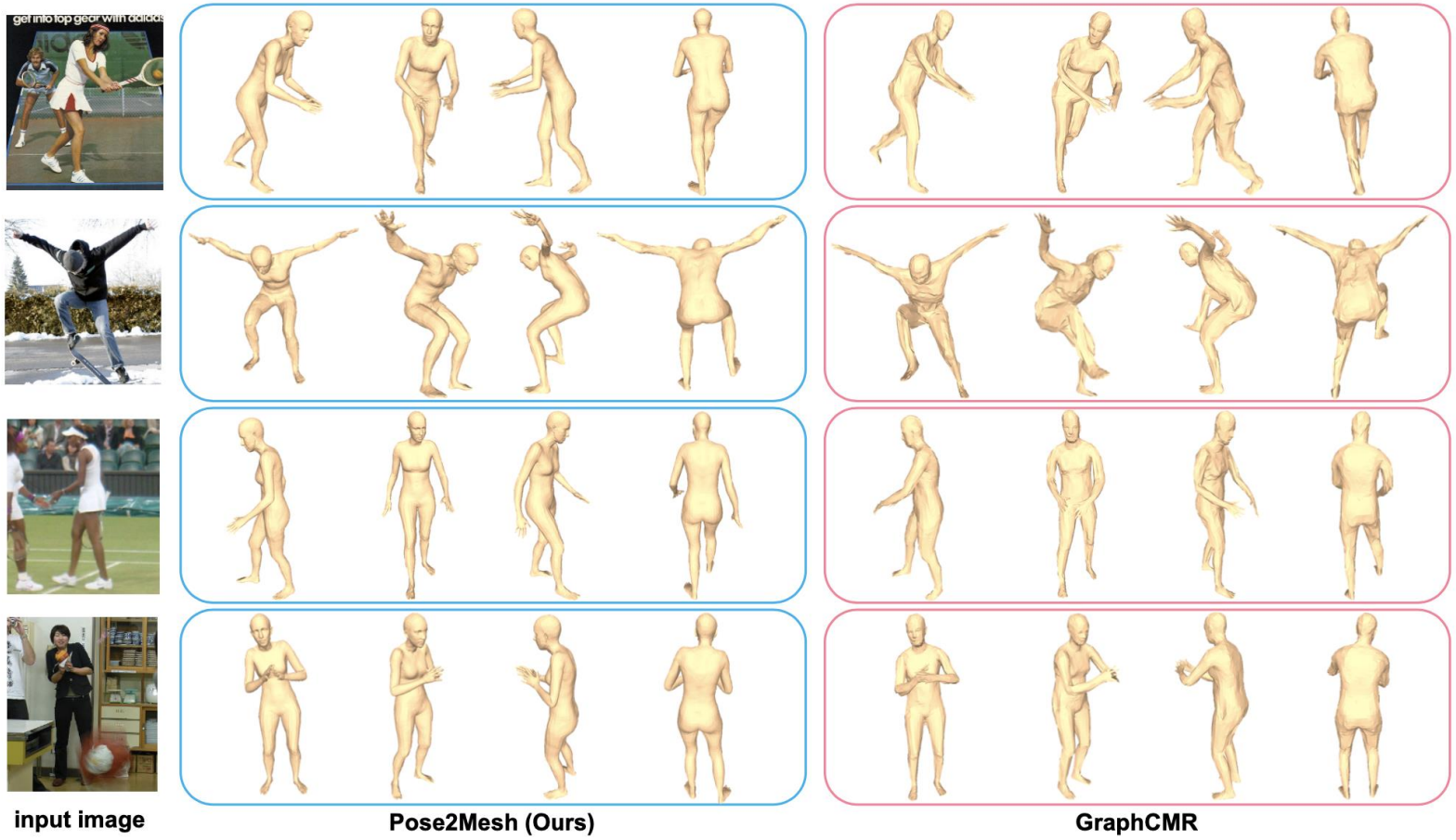
Experiments

- Qualitative results



Experiments

- Qualitative results comparison with other model-free method



Conclusions

- Contributions
  - The 2D pose input can avoid image appearance overfitting
  - The 2D pose input can be generated synthetically by projecting synthetic 3D human meshes, and thus Pose2Mesh can potentially benefit from unlimited training data
- Limitation: less shape information in input, high frequency noise due to high dimensionality of target
- Future works: incorporate other shape input without image (ex. part segmentation), explore other regularization methods, explore other graph convolution on mesh