**Self 3D mannequin dev plan and tech specs**

Nowadays people prefer to shop online at Amazon, Best buy, eBay, etc.

However, the most faced problem is that the customers do not know the size and quality of the garments in the shopping cart.

Only several photos of the garments in the showcase could be misleading.

Sometimes the received items are different when we see it in real life.

Though some e-Commerce platforms display the clothes dressed by the real models, the garments can still be the wrong size for the customers. To alleviate the anxiety of customers, almost all top fashion e-Commerce provide the free return service for customers.

So virtual dressing is very important for e-Commerce shopping fashion.

The scope of this proposal is as following:

• 3D human reconstruction

– Body reconstruction

– Face reconstruction

– Hand reconstruction

• 3D garments reconstruction

– Garments reconstruction

– Cloth simulation

1. 3D human full body mesh reconstruction(estimated timeline - 15days)

Image-based 3D human reconstruction is an important topic in virtual dressing, VR/AR tech, image and video editing . It’s a hot topic starting from 2D pose detection , 3D pose detection and model-based full reconstruction . However, due to the ambiguity of the 3D information, it is still challenging to recover an accurate human model from a single RGB image.

Even worse, multiple variations in-the-wild images, including human body shapes, clothes, environment, and viewpoints, gives this inverse problem multiple solutions.

The optimal representation of the 3D object remains the open question in the research field.

I’m going to use Pyramidal Mesh Alignment Feedback-X (PyMAF-X) deep learning model to achieve new state-of-the-art results both qualitatively and quantitatively, contributing novel solutions towards the well-aligned and natural recovery of full-body models from monocular images. The structure of PyMAF is as following.

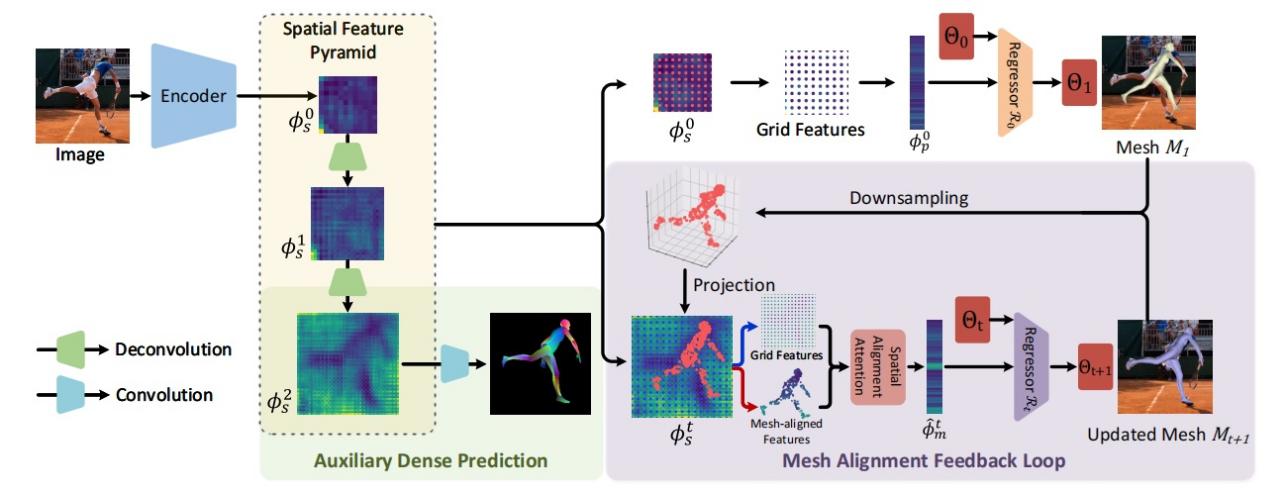


Fig.1 Illustration of the proposed Pyramidal Mesh Alignment Feedback (PyMAF) for human mesh recovery

As you can see, PyMAF leverages a feature pyramid and enables an alignment feedback loop in my network. Given a coarse-aligned model prediction, mesh-aligned evidence is extracted from finer-resolution features accordingly and fed back to a regressor for parameter rectification.

I’m going to implement PyMAF, a powerful model for regression-based human mesh recovery, then extend it to PyMAF-X for full-body mesh recovery.

The overal pipeline of PyMAF-X is as following.

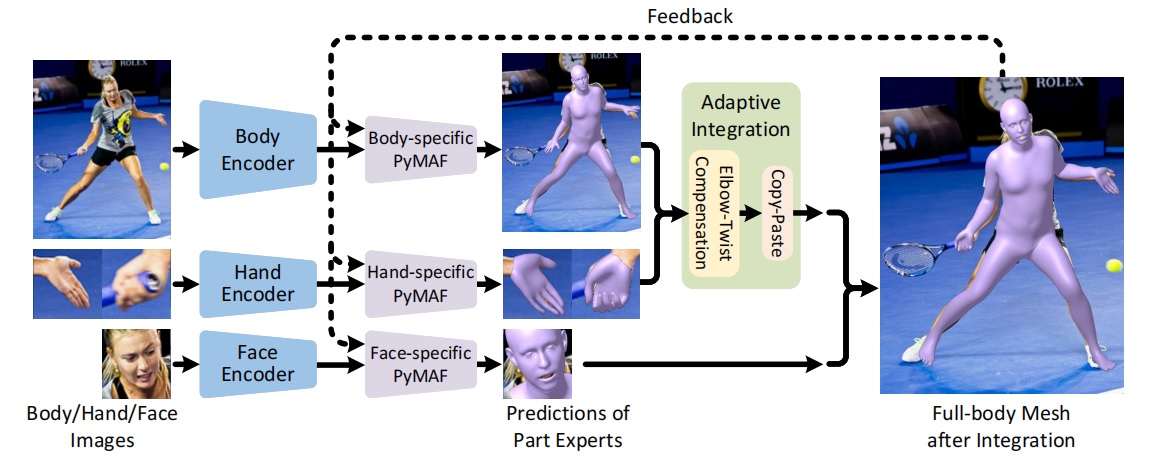


Fig 2. The overall pipeline of PyMAF-X for full-body mesh recovery

As you can see, PyMAF-X consists of three part-specific PyMAFs for part mesh prediction and integrates them together via the proposed adaptive integration strategy.

PyMAF-X consists of three experts, three part-specific PyMAFs, to predict the parameters of body, hand, and face, as illustrated in Fig 2.

To ensure high-resolution observations of part regions, part experts perform individual predictions on the body, hand, and face images cropped from the original inputs.

At each iteration of the mesh alignment feedback loop, the predictions of the body, hand, and face-specific PyMAF are collected and integrated as the parameters of the full-body model SMPL-X.

These parameters are the pose, shape, and facial expression parameters, respectively.

The pose parameters consist of the rotational poses of 55 joints in total, including 22 joints for the body, 30 finger joints for the hands, and 3 jaw joints for the face.

The camera parameters are taken from the predictions of the bodyspecific PyMAF and used to project body, hand, and face vertices on the image plane.

Moreover, considering that the positions of hand and face are susceptible to inaccurate body pose estimations, I’m going to align the center of their re-projected points to the image center of hand and face to ensure their mesh-aligned features are meaningful.

**Implement details of PyMAF-x:**

The part-specific PyMAF primarily adopts ResNet-50 as the backbone of the image encoder.

For each part-specific PyMAF, the image encoder takes a 224 × 224 image as input and produces spatial feature maps with resolutions of {14 × 14, 28 × 28, 56 × 56}.

I’m going to implement PyTorch deep learning framework for full-body mesh recovery.

In my estimation, the PyTorch implementation of the body-only PyMAF will be taken about 22 ms to process one sample on the machine with an NVIDIA RTX 3090 GPU.

For full-body mesh recovery, PyMAF-X will be taken about 80 ms to process one sample, which is on par with existing regression-based approaches.

**Training approaches of PyMAF-x:**

the body expert is trained on a mixture of data from several datasets with 3D and 2D annotations, including Human3.6M, and COCO.

For the hand expert, I’m going to use images from COCO-Wholebody for training.

For the face expert, I’m going to use the images from VGGFace2 for training.

--Pseudo Ground-truth

the SMPL/SMPL-X models are used as pseudo ground-truth annotations for the training of body and full-body model regressors.

For the training of the face expert, I’m going to use DECA and a face alignment algorithm FAN to generate pseudo groundtruth FLAME models and facial landmarks on the training set of VGGFace2.

--Dense Correspondence.

I’m going to not use the DensePose annotations in COCO for auxiliary supervision but render dense correspondence maps based on the pseudo ground-truth meshes.

**Testing approaches(Evaluation Metrics) of PyMAF-x:**

To quantitatively evaluate the performance of the 3D pose estimation, PVE, MPJPE, PA-PVE, and PA-MPJPE are adopted as the primary evaluation metrics.

They are all reported in millimeters (mm) by default.

Among these metrics, PVE denotes the mean Per-vertex Error, defined as the average point-to-point Euclidean distance between the predicted and ground truth mesh vertices, while MPJPE denotes the Mean Per Joint Position Error.

on fullbody mesh recovery, I’m going to evaluate the performance of PyMAF-X on two benchmark datasets such as EHF and AGORA.

And also I’m going to test the PyMAF-X model on the datasets provided from your company.

The test result for full body mesh using PyMAF-x is as following.

After getting full body mesh accurately, I can get measurements of the body from the body mesh using python library such as body-measurements.

Link of python library for body measurements based on body full mesh is as following.

https://pypi.org/project/body-measurements/



Fig 3. The test result of PyMAF-X

1. 3D garments reconstruction

* 1. Garments reconstruction (estimated timeline will be taken 10 days)

The problem of garment reconstruction from RGB images into separated human body

shape and clothing can be partially addressed by the multi-garment net (MGN).

The multi-garment net could predict the body shape with the PyMAF-X model and the

clothing it covers from several images.

Besides, this model could be transferred to different people with different poses.

To train the multi-garment net, I’m going to use a digital wardrobe containing 712 digital garments.

The structure of MGN is as following.

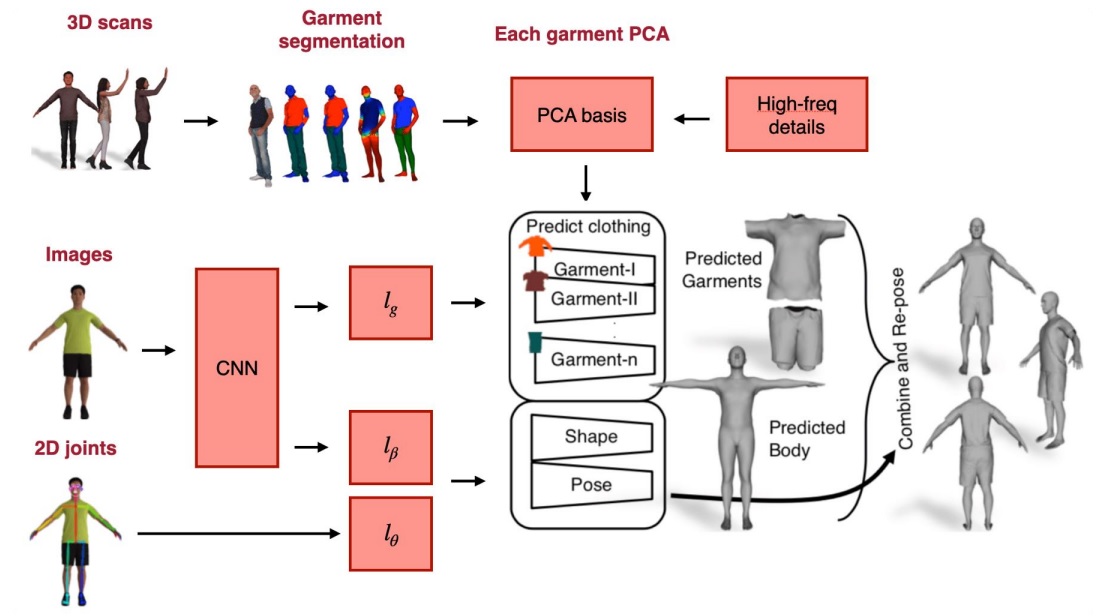


Fig 4. Detailed architecture of MGN

As you can see, CNN is used to encode image and 2D joint information. The garment network decoded the garment parameters to predict the garments parameters with PCA and added high-frequency details to the garment mesh.

Within each template, different clothes still possess diverse 3D shape. I’m going to a linear system to minimize the distance between the template and the 3D scanning, and keep the laplacian on the surface of the template.

In the registration process, I could get the vertex-based PCA for each garment.

MGN can be trained with multiple images, body pose and shape, PCA components of each garment. This method is better compared with the silhouette matching.

The base network worked as CNN to map the datasets into the body shape, pose, and garment latent spaces.

Each category of the garments could be trained in separate garment networks.

Two branch was contained in the garment network. The first one predicted the mesh shape, and the second work added the high-frequency details.

Evaluation of the remapped garments of the digital wardrobe into different human body shape and poses.

Multi-garment networks samples is as following.

Garments from digital wardrobes remapped into SMPL models.

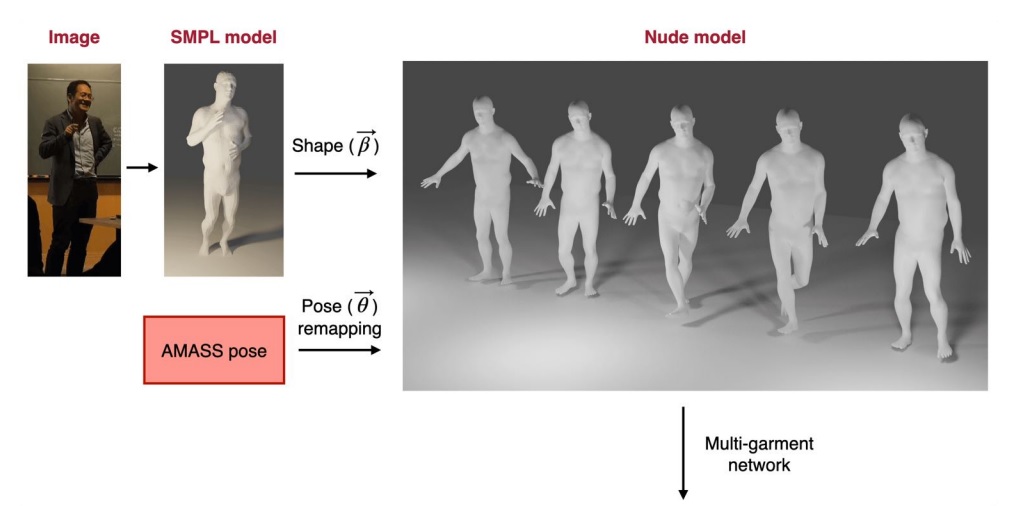


Fig 5(a). SMPL model with no clothing

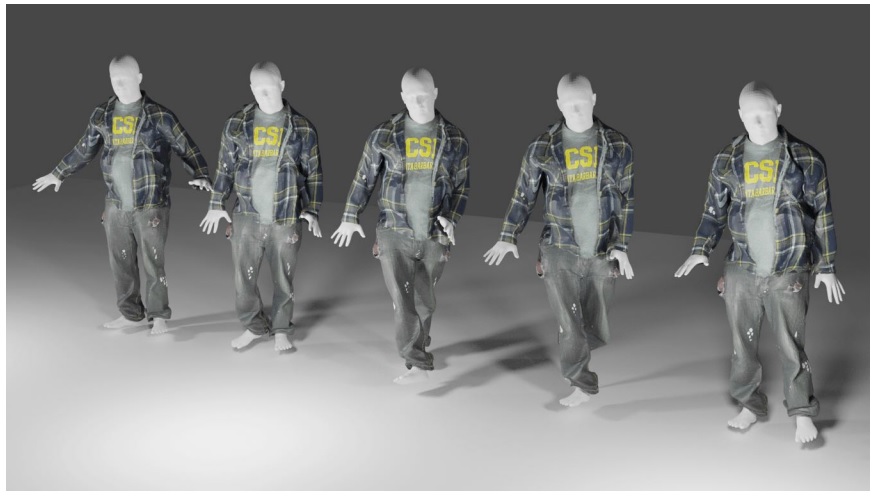


Fig 5(b). Long coat with pants

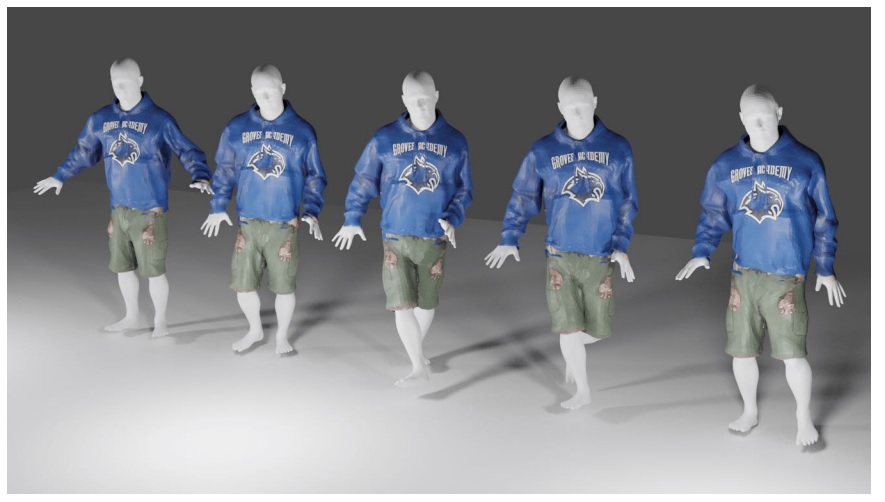


Fig 5(c). Shirt with short pants

* 1. Cloth simulation(estimated timeline will be 10 days)

A physical engine is the computer software that provides a realistic simulation of certain physical systems, e.g., rigid body dynamics, clothes, soft tissues, fluid dynamics,etc.

The simulation in computer graphics is usually different from the one in engineering.

The latter always requires extraordinary high accuracy, and the algorithm needs to be convergent in the finer mesh.

However, the physical engine here does not need to achieve the best accuracy.

However, the real-time speed is required, especially in the application.

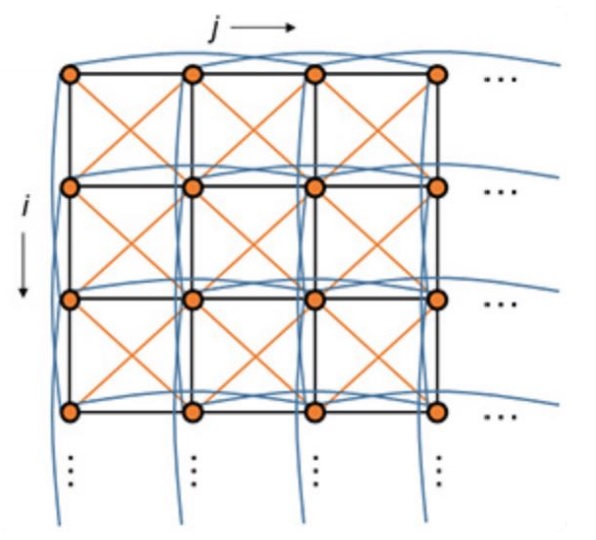


Fig 6. Illustration of the mass-spring system in the cloth simulation.

In the clothing simulation, I’m going to simplify the meshes of cloth into the simple

spring-mass system.

The cloth could be considered as a collection of particles interconnected with three types of springs:

• Structural spring: each particle [i, j] is connected to four particles via structural connections: [i, j + 1], [i, j − 1], [i + 1, j], [i − 1, j];

• Shear spring: each particle [i, j] is connected to four particles via shear connections:

[i + 1, j + 1], [i + 1, j − 1], [i − 1, j − 1], [i − 1, j + 1];

• Flexion spring: each particle [i,j] is connected to four particles via flexion connections:

[i, j + 1], [i, j − 2], [i + 2, j], [i − 2, j]

The force can be classified into types in the cloth simulation:

• Spring force: constrain the distance of each particle in the structural mesh;

• Gravity force: the major force to actively drag the cloth;

• Damping force: constrain the infinitesimal vibration of the mass particles;

• Collision force: constrain the self-penetration of the mesh and the penetration of the human body

To effectively animate the movement of the clothing, we utilize the extended position-based dynamics (XPBD) method. The difference between the XPBD method and the traditional one is that there is no explicit contact force in the calculation.

The constraints of position determine the trajectory of the particles.

The evaluation of clothing simulation utilizes the Marvelous Designer to match the patterns and Blender cycle for realistic rendering.

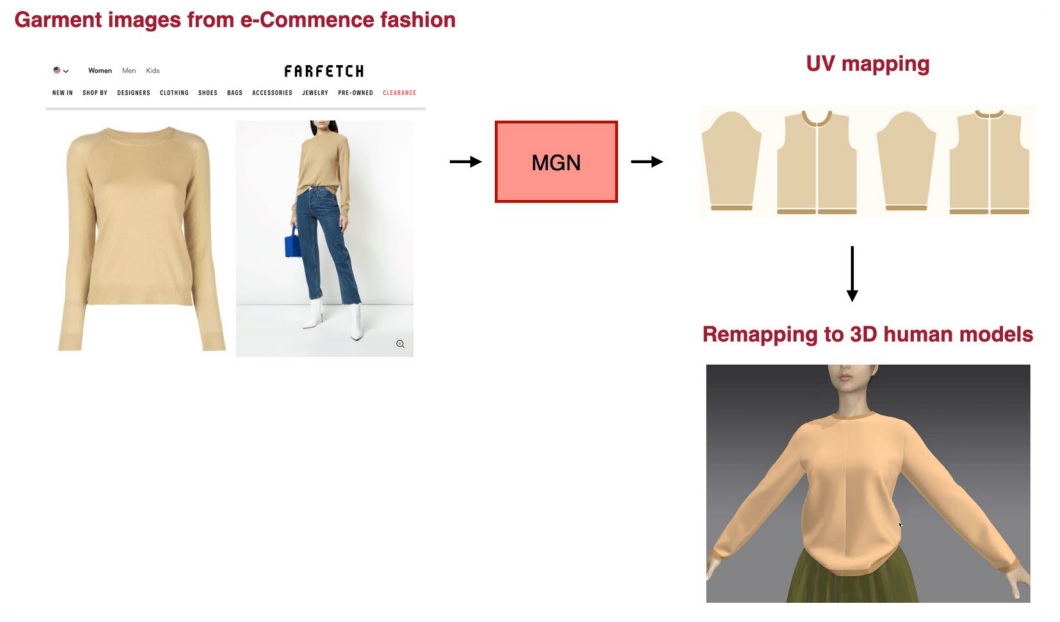


Fig 7. Multi-garment network based architecture



Fig 8. Cloth modeling with moving models inside

As you can see, the multi-garment network generates the UV mapping from the garments

images on fashion e-Commence and remapped it into the 3D human model.

The clothing can be modeled as a mass-spring system.

Thanks for your consideration.