DeepCap: Monocular Human Performance Capture Using Weak Supervision

Objective

- A learning-based 3D human performance capture approach that jointly tracks the skeletal pose and the non- rigid surface deformations from monocular images.

- A new differentiable representation of deforming human surfaces which enables training from multi-view video footage directly.

- The core of the method is a CNN model which integrates a fully differentiable *mesh* template parameterized with *pose* and an *embedded deformation graph*.

.

Workflow

- Template Acquisition
- Training Data
- Pose Network
  - Kinematics Layer
  - Global Alignment layer
- DefNet

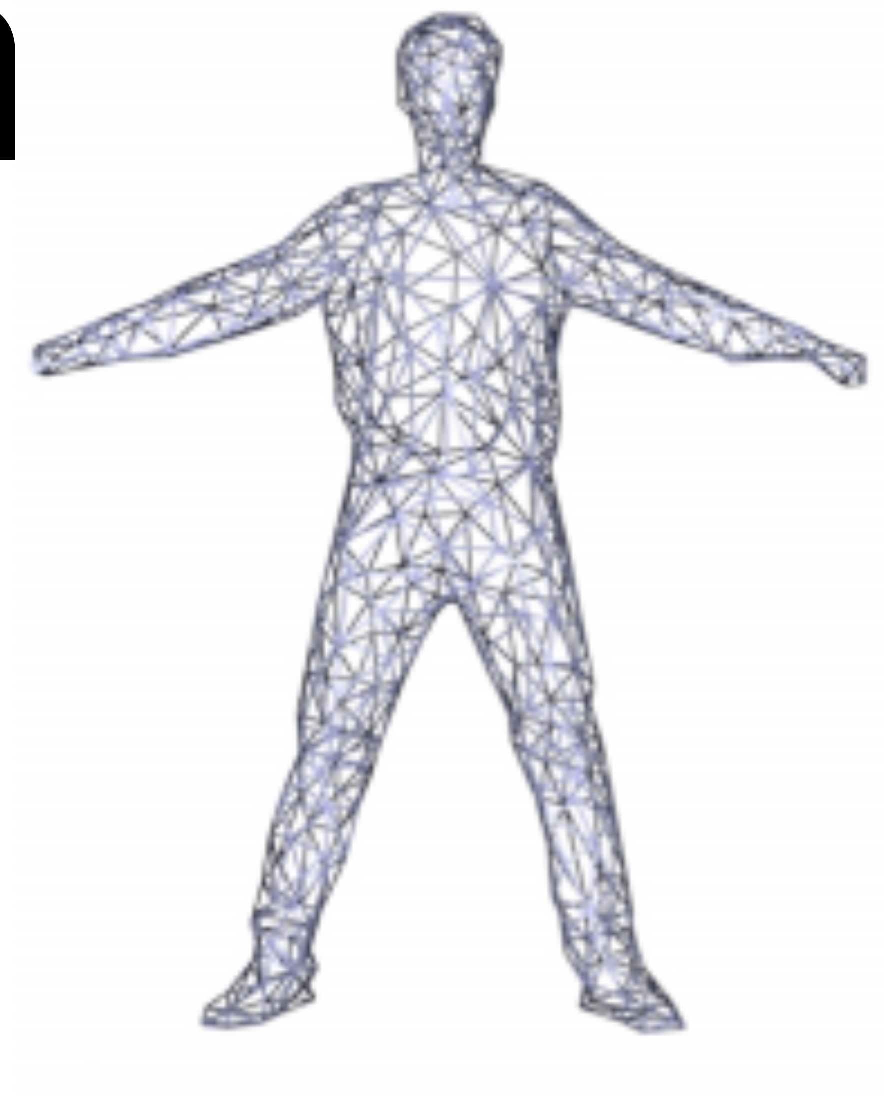
Architecture

Template Acquisition

- Person in static T-pose is captured in 134 RGB images
- From these RGB images, textured 3D model is made out using commercial softwares
  - https://www.treedys.com/
  - http://www.agisoft.com
  - http://www.meshmixer.com/

Mesh Mesh (Textured)

Template Acquisition



- **Embedded Deformation graph:**

- Decimate the template mesh to around 500 vertices.
  parametrised by {A,T}. A -> Rotation and T -> Translation

- The connections of a node k to neighboring nodes are given by the vertex connections of the decimated mesh and are denoted as the set $N_n(k)$.

- For each vertex of the graph, nearest node on template vertices is searched. The positions of graph nodes is the nearest vertex’s position

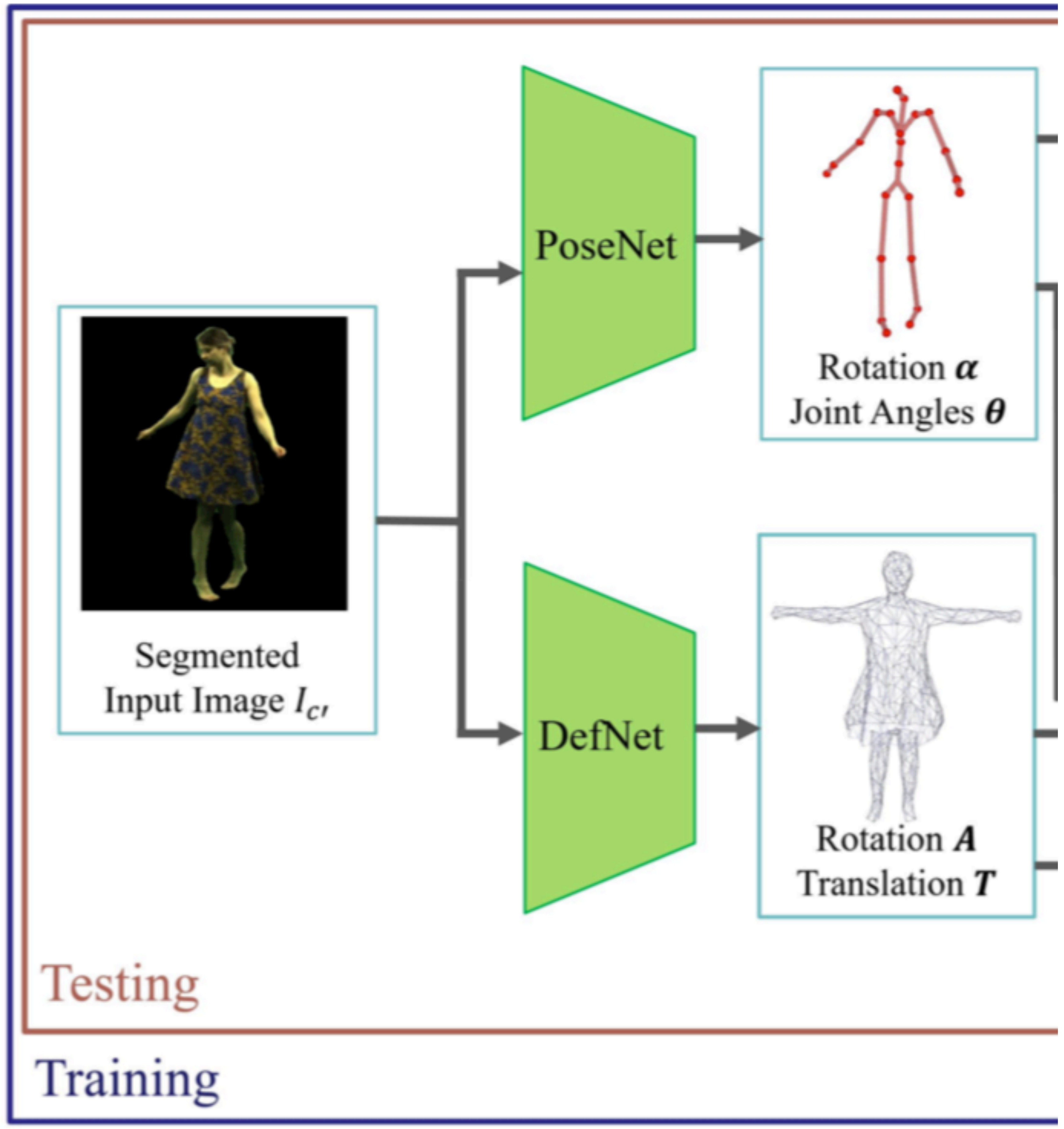
- Each vertex in template vertex is assigned $w_{i,k}$ which is distance between node k and vertex i

Training Data

- Record multi-view video of the actor in calibrated multi-camera studio
- Apply open pose to detect 2D joint locations and apply temporal filtering
- Generate fore-ground mask using color-keying
- A random camera view *v'* is chosen
- Final input is $256 \times 256 \times 3$ image which is back ground filtered. The image is augmented with random brightness, hue, contrast and saturation changes.

Pose Network

- Used ResNet50 pre-trained on Imagenet
- Last fully connected layer is modified to detect $\theta \in \mathscr{R}^{27}$ joint angles, camera root relative rotation $\alpha \in \mathscr{R}^{3}$ given input image
- Since ground truth for these parameters are non-trivial, weakly supervised setup is considered



PoseNetwork: Kinematics Layer

- A differentiable function that takes θ and α to produce the positions $P_c \in R^{M \times 3}$ of the *M 3D* land marks attached to 3D skeleton
- 17 body joints and 4 face landmarks
- P_c is in camera-root-relative coordinate system

PoseNetwork: Global Alignment Layer

- In order to project the landmarks on other camera views we need to set everything in global coordinate system

- This layer transforms into world coordinate system $P_m = R_{c'}^T P_c + t$ by estimating Rotation and translation parameters

$$\sum_c \sum_m \sigma_{c,m} \|(\mathbf{R}_{c'}^T \mathbf{P}_{c',m} + \mathbf{t} - \mathbf{o}_c) \times \mathbf{d}_{c,m}\|^2$$

where $\mathbf{d}_{c,m}$ is the direction of a ray from camera c to the 2D joint detection $\mathbf{p}_{c,m}$ corresponding to landmark m:

$$\mathbf{d}_{c,m} = \frac{(\mathbf{E}_c^{-1} \tilde{\mathbf{p}}_{c,m})_{xyz} - \mathbf{o}_c}{\|(\mathbf{E}_c^{-1} \tilde{\mathbf{p}}_{c,m})_{xyz} - \mathbf{o}_c\|}. \quad (3)$$

PoseNetwork: Losses

- Sparse Key point Loss: Ensures each land mark projects on to corresponding 2D joint locations where λ_m is weight of m[th] joint in kinematic tree and $\sigma_{c,m}$ is confidence of the joint location predicted.

$$\mathcal{L}_{\mathrm{kp}}(\mathbf{P}) = \sum_{c} \sum_{m} \lambda_m \sigma_{c,m} \| \pi_c(\mathbf{P}_m) - \mathbf{p}_{c,m} \|^2$$

- Pose Prior Loss: Used to avoid unnatural poses

$$\mathcal{L}_{\mathrm{limit}}(\boldsymbol{\theta}) = \sum_{i=1}^{27} \Psi(\boldsymbol{\theta}_i)$$

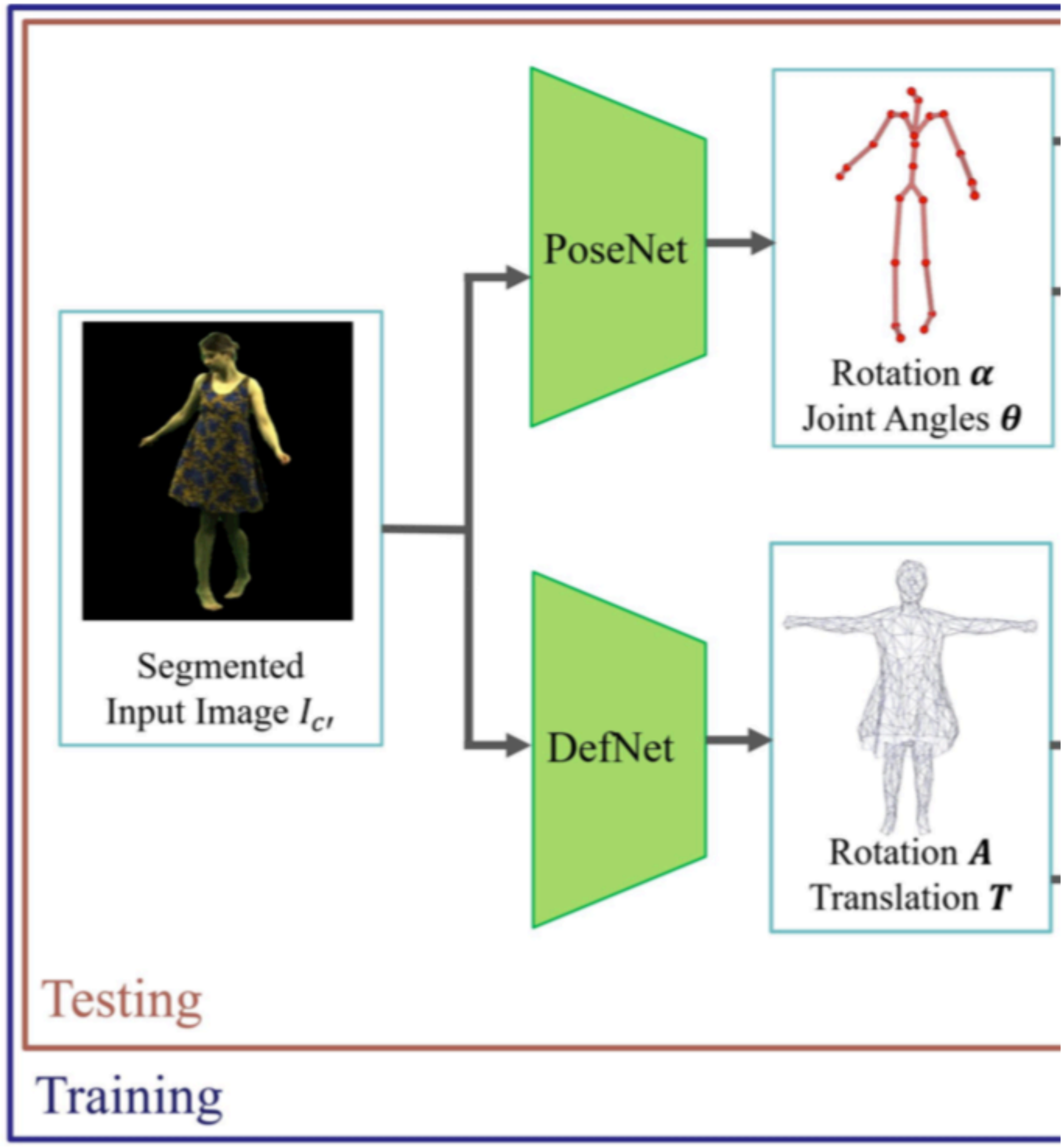
$$\Psi(x) = \begin{cases} (x - \boldsymbol{\theta}_{\max,i})^2, & \text{if } x > \boldsymbol{\theta}_{\max,i} \\ (\boldsymbol{\theta}_{\min,i} - x)^2, & \text{if } x < \boldsymbol{\theta}_{\min,i} \\ 0, & \text{otherwise} \end{cases}$$

Deformation Layer

- Using skeletal pose alone, the non-rigid deformations cannot be explained

- Regresses to A and T params. Uses ResNet like architecture except for final layer. Outputs 6K dim vec

- Differentiable rendering and multi-view sillhouette loss



Deformation Layer

- Deformed template vertices: $$\mathbf{Y}_i = \sum_{k \in \mathcal{N}_{\mathrm{vn}}(i)} w_{i,k}(R(\mathbf{A}_k)(\hat{\mathbf{V}}_i - \mathbf{G}_k) + \mathbf{G}_k + \mathbf{T}_k).$$
- Deformation from Skeletal Pose: $$\mathbf{V}_{\mathrm{c}',i} = \sum_{k \in \mathcal{N}_{\mathrm{vn}}(i)} w_{i,k}(R_{\mathrm{sk},k}(\boldsymbol{\theta}, \boldsymbol{\alpha})\mathbf{Y}_i + t_{\mathrm{sk},k}(\boldsymbol{\theta}, \boldsymbol{\alpha}))$$
- Sparse Keypoint loss: $$\mathcal{L}_{\mathrm{kp}}(\mathbf{P}) = \sum_c \sum_m \lambda_m \sigma_{c,m} \|\pi_c(\mathbf{P}_m) - \mathbf{p}_{c,m}\|^2$$
- As-rigid-as-possible:

$$\mathcal{L}_{\mathrm{arap}}(\mathbf{A}, \mathbf{T}) = \sum_k \sum_{l \in \mathcal{N}_{\mathrm{n}}(k)} u_{k,l} \|d_{k,l}(\mathbf{A}, \mathbf{T})\|_1, \qquad (13)$$

where

$$d_{k,l}(\mathbf{A}, \mathbf{T}) = R(\mathbf{A}_k)(\mathbf{G}_l - \mathbf{G}_k) + \mathbf{T}_k + \mathbf{G}_k - (\mathbf{G}_l + \mathbf{T}_l).$$