Predict 3D Cloth Shape from Body Pose

Goal: learn a function that maps from input skeletal pose to output 3D cloth shapes
Predict 3D Cloth Shape from Body Pose

- Input: pose parameters $\theta$ represented as joint rotation matrices
  - E.g., 10 joints in upper body with 3x3 rotation matrix for each joint leads to a 90 dimensional pose vector (could be 30 dimensional using quaternions)
  - global translation/rotation of root frame is ignored

- Output: 3D cloth shape $\varphi$
  - E.g., 3,000 vertices in a cloth triangle mesh leads to a 9,000 dimensional shape vector

- One needs to learn a function that maps from a relatively low-dimensional space to a very high-dimensional space
  - E.g., $f: \Omega \subset \mathbb{R}^{90} \rightarrow \mathbb{R}^{9000}$, where $\Omega$ represents the valid subset of poses
Predict 3D Cloth Shape from Body Pose

- Given: $n$ training data points $(\theta_i, \varphi_i), i = 1, 2, ..., n$ generated from the true/approximated function $\varphi_i = f(\theta_i)$
  - E.g. using simulation or capture

- Goal: learn a function $\hat{f}$ that approximates $f: \hat{f}(\theta) = \hat{\varphi} \approx \varphi = f(\theta)$

- Question 1: what basis to represent the output cloth shape $\varphi$ in?
- Question 2: how do we learn the function using that representation?
Predict 3D Cloth Shape from Body Pose

• Formulation 1: learn per-vertex absolute 3D positions in the root frame
  • i.e., directly learn $\hat{\phi} = \{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_m\}$, where $\hat{v}_j \in \mathbb{R}^3$ is the predicted position of vertex $j$, and $m$ is the number of vertices in the cloth mesh

• Issues:
  • complex nonlinear function to learn
Predict 3D Cloth Shape from Body Pose

• Solution 1: train a fully connected network that takes the pose parameters $\theta$ and outputs the predicted 3D vertex positions $\hat{\phi}$

• Rationale: joints have non-local influence, and thus it makes sense to model global correlation of the whole garment with a fully connected network
  • E.g., lifting an arm can induce non-local shape change in the waist, back, etc.

• Challenges:
  • Hard to train a deep fully connected net
    • Too many parameters -> need lots and lots of training data (expensive to obtain)
  • A shallower network may not have enough capacity for the function complexity
Change the function!
Aside: Procedural Skinning

• Skinning is a standard method to deform a character’s surface skin given its skeletal pose
  • widely used in character animation

• In the rest pose, each vertex on the body surface is associated with a few joints/bones, where the weight of each joint/bone dictates how much impact its pose has on the vertex

• As the pose changes, the joint angles change, and the vertex positions are transformed using the precomputed weights

Picture from Blender website [link]
Change the function!

• Formulation 2: learn per-vertex offsets from a procedurally skinned surface
  • first, apply a standard procedural approach to skin the body from the joint angles/pose, then learn offsets of the cloth from this procedurally skinned body
  • i.e., represent $\hat{\phi} = S(\theta) + D(\theta)$, where $S(\theta)$ is the procedural skinning function and $D(\theta)$ is the offset function to be learned

• Basically building a procedural prior into the learning

• Pros: easier, smoother function to learn

• How do we do this?
Pixel Based Cloth

• Assign texture coordinates to a triangle cloth mesh
  • transfer mesh into pattern space (texture space)
• Store \((u, v, n)\) offsets in the pattern space
• Convert \((u, v, n)\) offsets to RGB color values
Body Skinning of Cloth Pixels

• Shrink-wrap the cloth pixels to the body mesh
  • barycentrically embed cloth pixels into body triangles

• As the body deforms, cloth pixels move with their parent triangles

• This makes the offset functions simpler to learn, since they don’t account for identity, pose, or skinning deformations
RGB Values of Cloth Pixels Correspond to Offset Vectors
Better Solution

• Solution 2: train a fully connected network that takes as input the pose parameters $\theta$ and outputs the colors of the cloth pixels $D(\theta)$
  • and then compute $\hat{\phi} = S(\theta) + D(\theta)$.

• Improvements:
  Since the function to learn is simpler, the same network architecture (same capacity) can be more easily trained to achieve better performance
Change the function again!
Image Based Cloth

• Rasterize triangle vertex colors to standard pixels
• Function outputs become standard 2D images
• More continuous than the cloth pixels (which are more discrete)
• Now, can learn with a Convolutional Neural Network (CNN)
Encode 3D Cloth Shapes as 2D Images

- Absolute per-vertex positions
- Colors of cloth pixels (per-vertex offset vectors)
- Cloth offset images (in pattern space)
Even better solution

• Formulation 3: learn cloth offset images $I(\theta)$
  • Interpolate the offsets $D(\theta)$ to an RGB image $I(\theta) = g(D(\theta))$, and learn a function that predicts the cloth offset images $I(\theta)$ from input pose
  • i.e., represent $\hat{\phi} = S(\theta) + h(I(\theta))$, where $S(\theta)$ is the procedural skinning function, and $h$ is the function that interpolates from cloth images back to cloth pixels

• Pros:
  • more natural and smoother function to learn,
  • can exploit spatial coherency and leverage large amount of techniques developed for learning on images, e.g. CNNs
Even better solution

• Solution 3: train a convolutional network that takes the pose parameters $\theta$ and outputs the cloth offset images $I(\theta)$
  • and then compute $\hat{\phi} = S(\theta) + h(I(\theta))$.

• Benefits:
  • smaller, deeper network that is easier to train and achieve better performance
  • Can use all sorts of pre-existing CNN networks/technology
Example: Predict 3D Cloth Shape from Body Pose

• Takeaways:
  • Problem formulation/representation matters for learning
  • Same functions represented in different spaces, or as perturbations of procedural functions, can have varying degree of difficulty for learning
  • Careful choice of basis/priors can lead to easier and more successful learning