Pose Space Deformation

Yifan Wang & Veronica Lee

Skinning

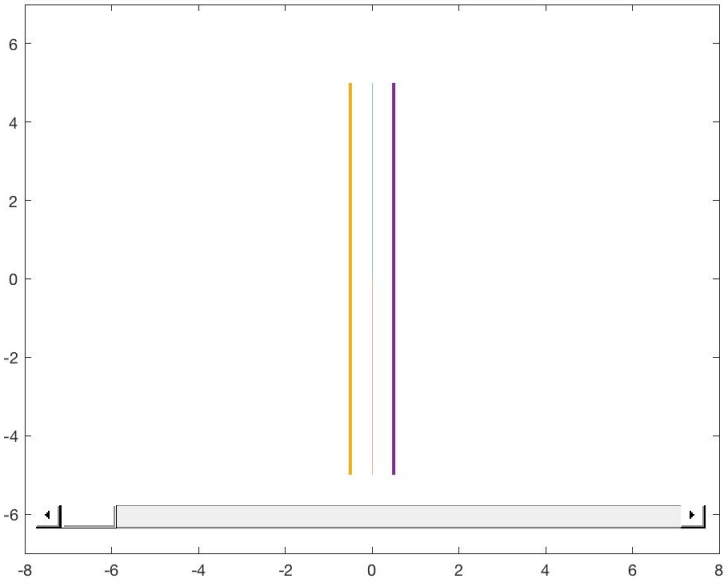
- Skeletal subspace deformation (SSD) [MTLT 1988]
- Pose space deformation (PSD) [LCF 2000]
- Weighted pose space deformation (WPSD) [KM 2004]

Skeletal Subspace Deformation (SSD)

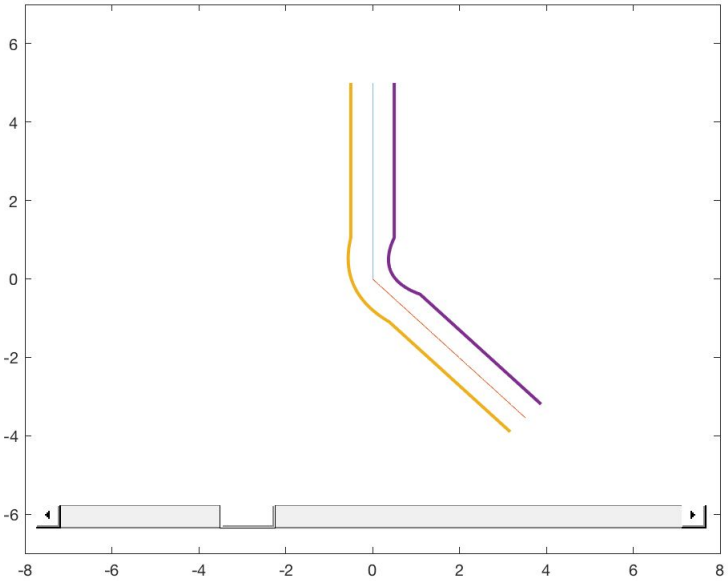
General equation: $v(p_a) = S(v_0)$

- p_a is an arbitrary pose
- $v(p_a)$ is a vertex of a deformed target of the arbitrary pose
- v_0 is the vertex in bind pose
- S is the SSD function

Results of SSD

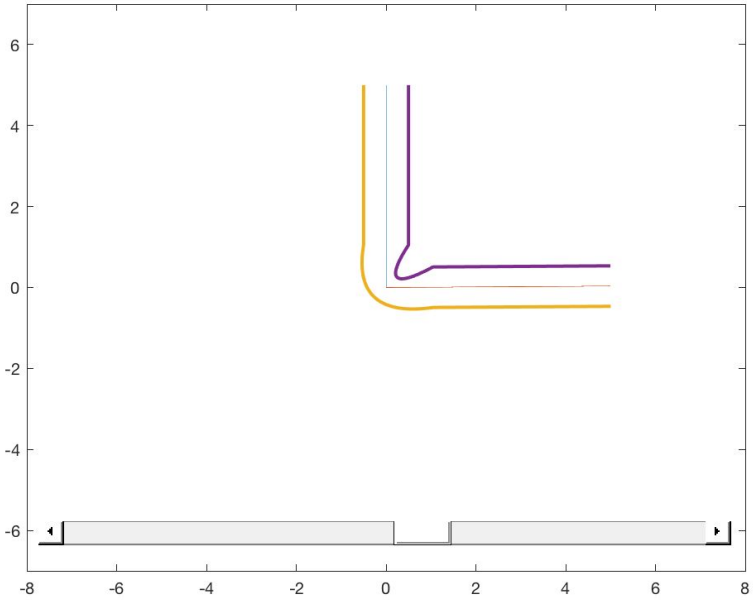


180 degrees (bind pose)

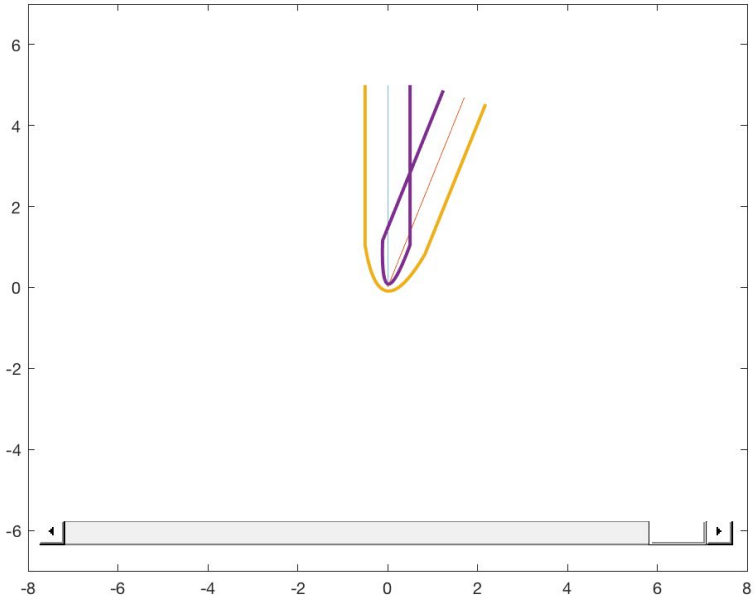


135 degrees

Results of SSD



90 degrees



20 degrees

Shortcomings of SSD

- Cannot handle sharp turns
- Corners do not look good
- Cannot handle rotation in 3D

Solution 1: Define poses and calculate weights

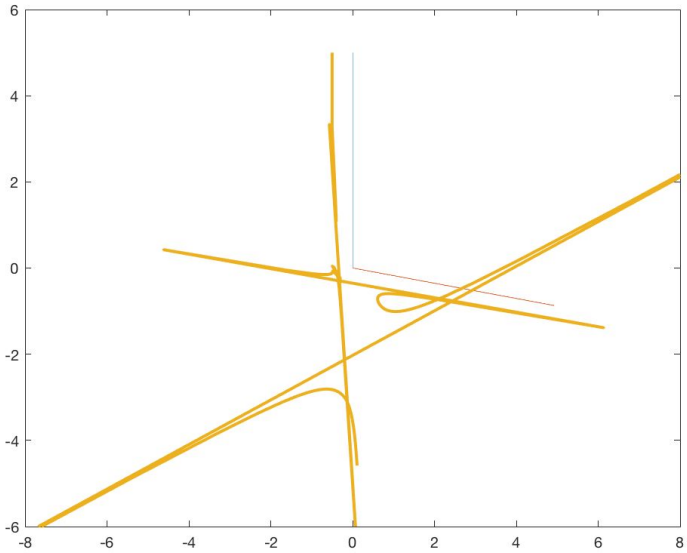
We defined poses for 180, 140, 100, 60, and 20 degrees, and used least squares to calculate weights for each joint:

$$\widetilde{v_k} - e_k = (\sum_{j=1}^{n_{joint}} v_j w_j)$$

where v_j is v_0 transformed by T_j.

Results of computed joint weights

Doesn’t look good because least squares may produce negative weights so the outcome will not converge.



Solution 2: Pose Space Deformation (PSD)

General equation: $v(p_a) = S(v_0 + D(p_a))$

where $D(p_a)$ is a displacement as a function of the arbitrary pose.

Process of PSD

1. Define poses for 180, 140, 100, 60, and 20 degrees.

2. Calculate displacements for those angles.

3. Use the Radial Basis Function to interpolate displacement for an arbitrary pose.

Details of PSD

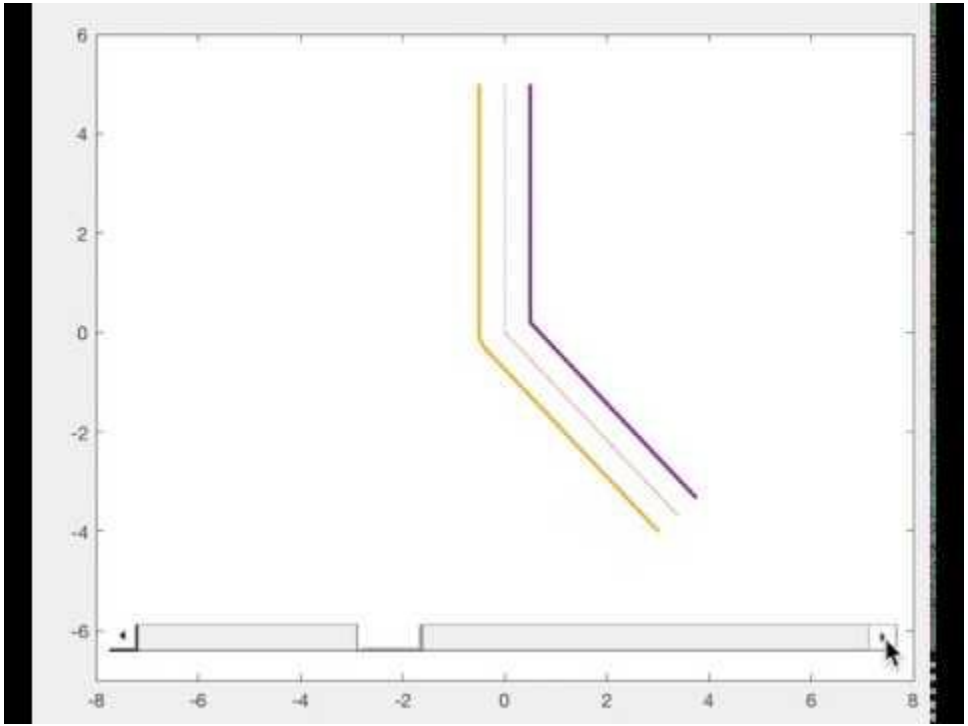
Displacement of each pose can be calculated as

$$d_k = (\sum_{j=1}^{n_{joint}} w_j T_j)^{-1} v_k - v_0$$

Using RBF, we have $D(x) = \sum_{k=1}^{n_{pose}} \lambda_k \phi(\|\mathbf{x} - \mathbf{x_i}\|)$

where we choose $\phi(x) = e^{-(\varepsilon x)^2}$

Results of PSD



Conclusion

- PSD can do the skinning smoothly in 3D
- By defining poses, PSD can also handle the rotation in 3D
- We can add DOF in our system and do the WPSD in future

Thank you!