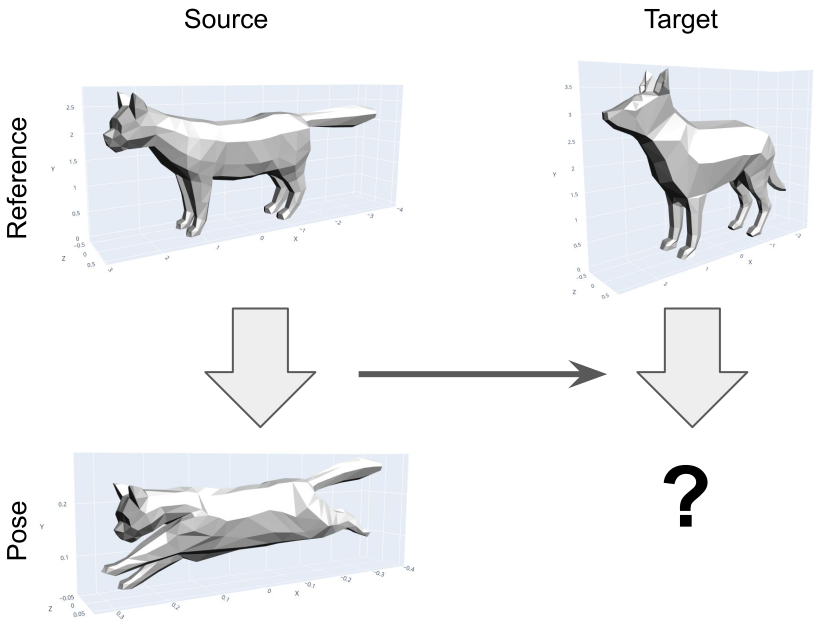
Implementation of
“Deformation Transfer for Triangle Meshes”

By

Jasmin Hoffmann
Michael Käser

Sumner, R.W. and Popović, J., 2004. Deformation transfer for triangle meshes. *ACM Transactions on graphics (TOG)*, *23*(3), pp.399-405.

Objective



Given

- Source and target mesh
- Pose(s) of the source mesh

Goal

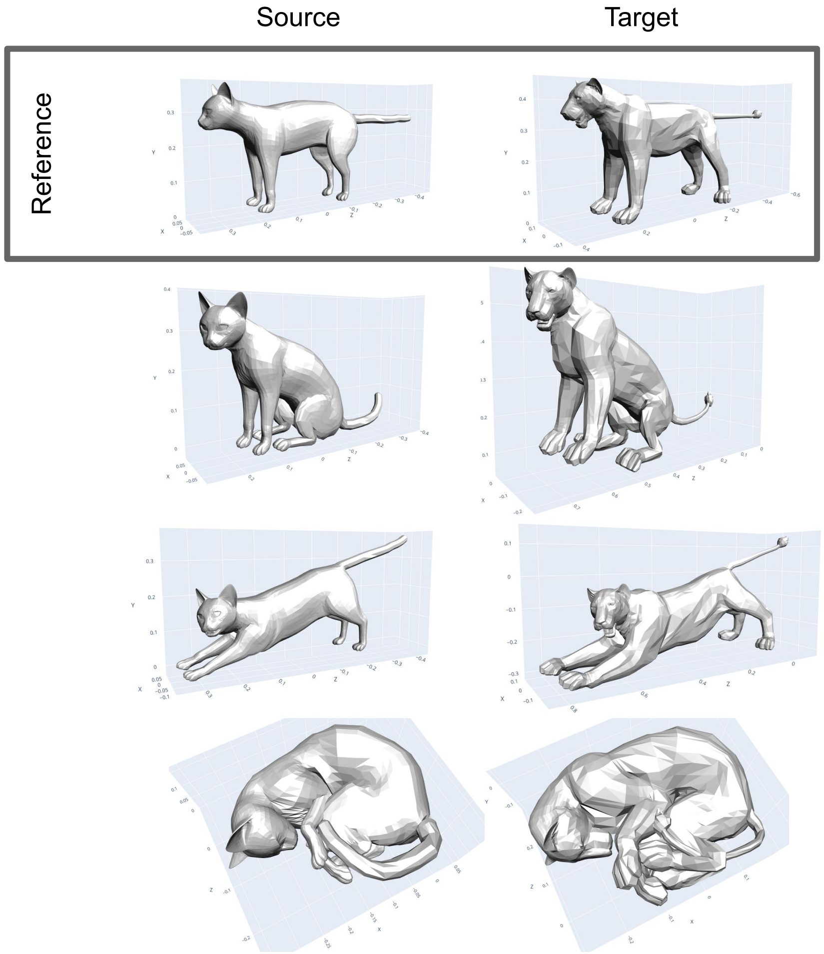
- Transfer the source's pose to the target mesh

General approach

Starting from **user provided markers**, a **correspondence mapping** between the source and target triangles is created.

This mapping is used to **transfer the deformation** of the source mesh to the target mesh.

Both steps are formulated as a **minimization problem**.



Models from https://people.csail.mit.edu/sumner/research/deftransfer/data.html#download

Core Idea: Position-independ Triangle Deformation

Triangle face of vertices ***a,b,c*** described without position as matrix $\mathbf{V}_i \in \mathbf{R}^{3x3}$.

The normal vertex ***d*** enables lateral movement of the face.

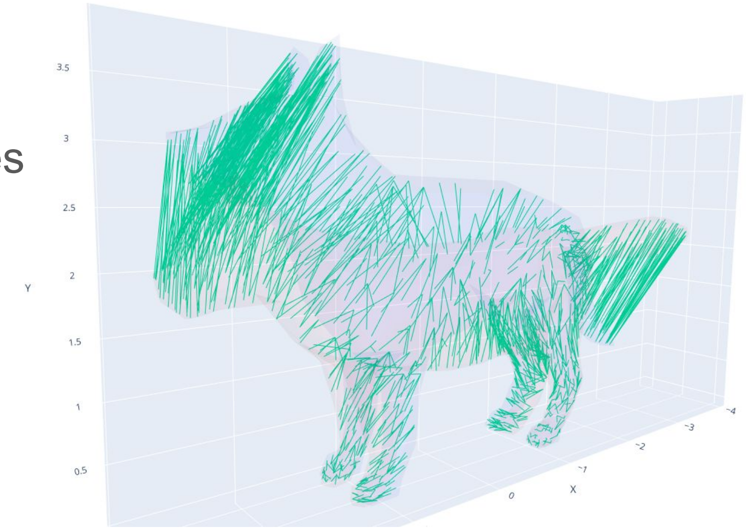
$$V_i = \left[\vec{b} - \vec{a}, \vec{c} - \vec{a}, \vec{d} - \vec{a}\right] \qquad \vec{d} = \vec{a} + \frac{\left(\vec{b} - \vec{a}\right) \times (\vec{c} - \vec{a})}{\sqrt{\left|\left(\vec{b} - \vec{a}\right) \times (\vec{c} - \vec{a})\right|}}$$

$\mathbf{T}_i$ is the triangle transformation from $\mathbf{V}_i$ to $\tilde{\mathbf{V}}_i$

$$\widetilde{V}_i = T_i \cdot V_i \longrightarrow T_i = \widetilde{V}_i V_i^{-1}$$

Correspondence

- **Mapping** between source and target triangle faces
- User-defined markers fixate the mapping
- Transforms source **into a shape like** the target
- **Minimization problem** of cost functions, solved for deformed vertices ṽ



$$\tilde{v} = \min_{\tilde{v}_1 \ldots \tilde{v}_n} \left(w_s E_s + w_I E_I + w_C E_C \right)$$

subject to m-markers $\tilde{v}_{s_k} = m_k, \; k \in 1 \ldots m$

Correspondence: Cost Components

Smoothness Cost	Transformation difference between adjacent triangles should be small	$E_S(v_1 \ldots v_n) = \sum_{i=1}^{\lvert T \rvert} \sum_{j \in adj(i))} \lvert\lvert T_i - T_j \rvert\rvert_F^2$
Identity Cost	Try to retain the source shape	$E_I(v_1 \ldots v_n) = \sum_{i=1}^{\lvert T \rvert} \lvert\lvert T_i - I \rvert\rvert_F^2$
Closest Points Cost	Add pulling force towards target shape	$E_C(v_1 \ldots v_n, c_1 \ldots c_n) = \sum_{i=1}^{n} \lvert\lvert v_i - c_i \rvert\rvert^2$

Correspondence: Progressive Closest Point

Problem:

- Initially no closest points available (or no "good guess")

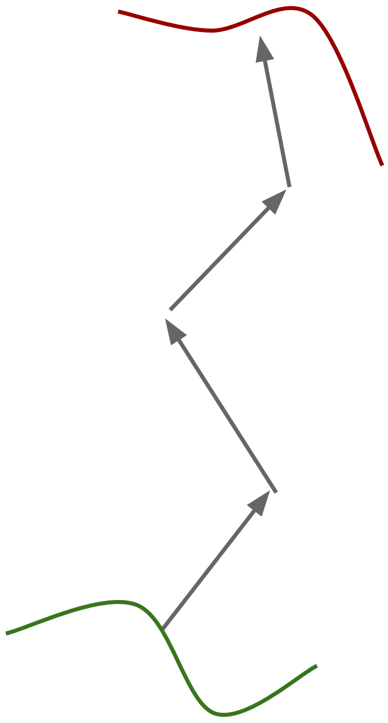
Solution:

1. First iteration ignores closest points
2. Progressively update closest points and increase the weight $\boldsymbol{w_c}$

$$\tilde{v} = \min_{\tilde{v_1} \ldots \tilde{v_n}} (w_s E_s + w_I E_I + w_C E_C)$$

Steps:

- Paper (5 steps): w_C = [0, 1….5000]
- We (8 steps): w_C = [0, 10, 50, 250, 1000, 2000, 3000, 5000]



Sparse Vector Expansion from Triangle Transformation to Mesh Transformation

Sparse vector expansion:

- Expand T_i with $\tilde{\mathbf{V}}_i \in \mathbf{R}^{3x3}$ to sparse mesh equation $\hat{T}_i$ with x $\in \mathbf{R}^{3xN}$.

$$T_i = \widetilde{V}V^{-1} \xrightarrow{\text{Expand to } x} \hat{T}_i = x\hat{V}^{-1}$$

Rearrange cost function to be able to be solved with a **least-squares approach**.

$$\sum_{i=1}^{m} \left\| x\hat{V}^{-1} - C_i \right\|_F^2 = \left\| Ax^T - b \right\|_F^2$$

Solving for Deformation

Deformed mesh is the least square solution of the weighted combination of each cost matrix A_I,A_S,A_C.

$$E = w_S E_S + w_I E_I + w_C E_C$$

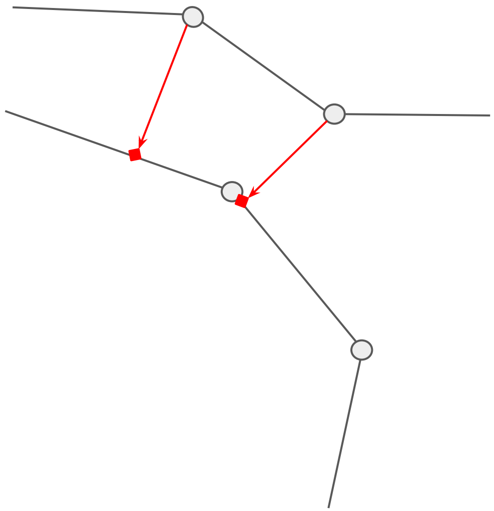
$$\widetilde{x} = \arg\min_x \left\| \begin{pmatrix} w_I A_I \\ w_S A_S \\ w_C A_C \end{pmatrix} x^T - \begin{pmatrix} w_I b_I \\ w_S b_S \\ w_C b_C \end{pmatrix} \right\|_F^2$$

Least-Squares solved with LU-Decomposition of:

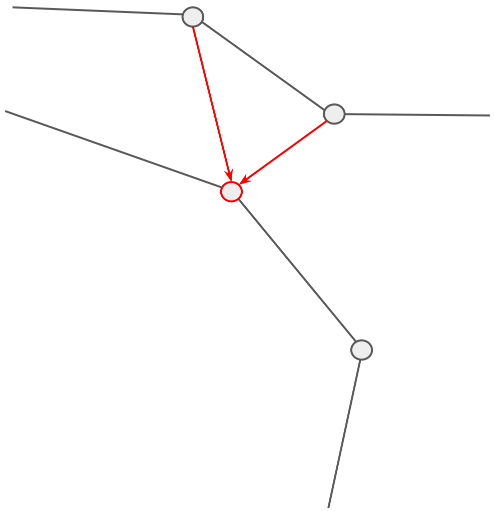
$$A^\top A \tilde{x} = A^\top c$$

Compromise: Closest points

Paper



Our approximation



Triangle Matching

We now have the deformed source mesh and the target mesh.

To map source to target triangles, we

- **Compare the centroids** of deformed source and target triangles
- Match **compatible** triangles with closest centroids
    - Compatible if angle between their normals is < 90°
- Iterate over **both** source and target mesh to find a match for each triangle
    - Triangle mapping is many-to-many

Deformation Transfer

We have

- a **mapping** between source and target triangles.

Now we want to

- **transfer a deformation** from source to target mesh.

The deformation of the target triangles should be similar to the deformation of the mapped source triangles.

Problem statement

$$\min_{\tilde{\mathbf{v}}_1 \ldots \tilde{\mathbf{v}}_n} \sum_{j=1}^{|M|} \left\| \mathbf{S}_{s_j} - \mathbf{T}_{t_j} \right\|_F^2$$

Runtime

Horse/Camel ~ 1 min

- Horse 8431 Vertices 16843 faces
- Camel 21887 Vertices 43814 Faces

Cat/Dog < 15 seconds

- Cat 398 Vertices 794 faces
- Dog 485 Vertices 960 faces

Processor Intel(R) Core(TM) i7-9700KF CPU @ 3.60GHz, 3600 Mhz, 8 Core(s), 8 Logical Processor(s)
RAM 32.0 GB

Original paper

- Sumner, R.W. and Popović, J., 2004. Deformation transfer for triangle meshes. *ACM Transactions on graphics (TOG)*, *23*(3), pp.399-405.

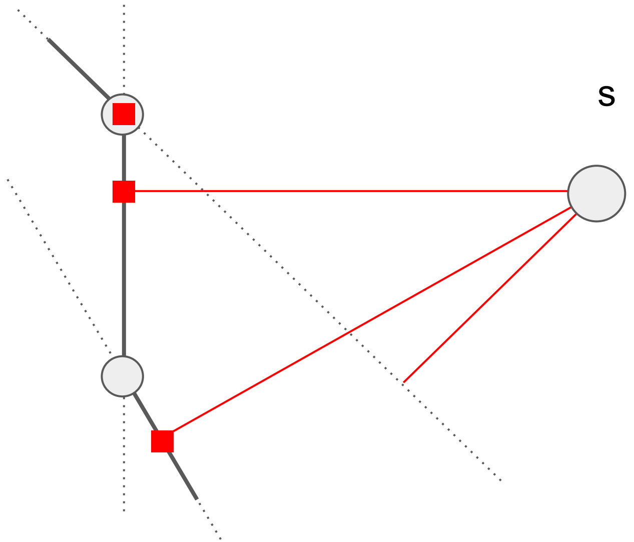
Interactive Models

https://mickare.de/dff/deformation/

Appendix

How to improve Closest Point

1. Get k closest target vertices ***p*** via KD-Tree from source vertex ***s***
2. For each triangle face ***f*** at ***p***
    a. Check if angle to face normal $\boldsymbol{n_f}$ < 90°
    b. Compute intersection ***p'*** of face normal $\boldsymbol{n_f}$ from ***s*** to the plane of ***f***
    c. Clip point ***p'*** to plane boundaries of ***f***
3. Choose closest ***p'***



Rearranging the Frobenius Norm

Sum of Frobenius Norms

$$\sum_k \|Z_k\|_F^2 = \sum_k z_{k1}^2 + z_{k2}^2 + \ldots + z_{kl}^2$$
$$= (z_{11}^2 + z_{12}^2 + \ldots + z_{1l}^2) + (z_{21}^2 + z_{22}^2 + \ldots + z_{2l}^2) + \ldots + (z_{k1}^2 + z_{k2}^2 + \ldots + z_{kl}^2)$$
$$= \left\| \sum_k Z_k \right\|_F^2$$

Transposing the Frobenius Norm

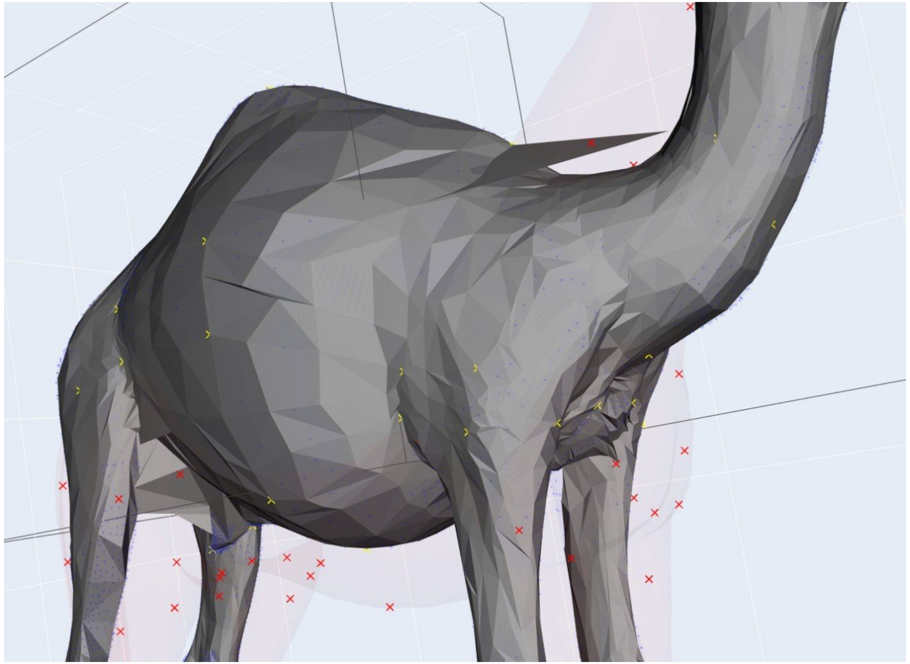
$$\|Z\|_F^2 = z_1^2 + z_2^2 + \ldots + z_l^2 = \|Z_k^T\|_F^2$$

Libraries

- Numpy - Matrix & Math
- Scipy - Sparse Matrices, LU decomposition, KD Tree
- Plotly - Plotting

- PyWavefront - Model Loading
- PyYAML - Markers and Poses
- tqdm - Progress bar (printing in terminal)

Increasing w_C to quickly



Problem Restatement

Rewrite as system of **linear equations**

Solve via **LU decomposition** (scipy.sparse.linalg.splu)

$$\min_{\tilde{v}_1 \dots \tilde{v}_n} \sum_{j=1}^{|M|} ||S_{s_j} - T_{t_j}||_F^2$$

$$\Downarrow$$

$$\tilde{x} = \arg\min_x ||Ax - b||_F^2$$

$$\Downarrow$$

$$A^\top A\tilde{x} = A^\top c$$

Problem Restatement

Merge separate terms to a **single** Frobenius norm

Move **marker vertices** as constants to b

Rewrite as system of **linear equations**

Solve via **LU decomposition** (scipy.sparse.linalg.splu).

$$\tilde{v} = \min_{\tilde{v_1} \dots \tilde{v_n}} (w_s E_s + w_I E_I + w_C E_C)$$

$$\Downarrow$$

$$\tilde{x} = \arg\min_x ||Ax - b||_F^2$$

$$=$$

$$\tilde{x} = \arg\min_x \left\| \begin{pmatrix} w_S A_S \\ w_I A_I \\ w_C A_C \end{pmatrix} x^\top - \begin{pmatrix} w_S b_S \\ w_I b_I \\ w_C b_C \end{pmatrix} \right\|_F^2$$

$$\Downarrow$$

$$A^\top A \tilde{x} = A^\top c$$

Correspondence

1. Transform source mesh to match user-defined markers in target mesh
    a. Use identity cost and smoothness cost
2. Search for closest points (vertices) in the target
3. Transform source mesh again from start
    a. Additionally use closest points cost
4. Repeat from step 2