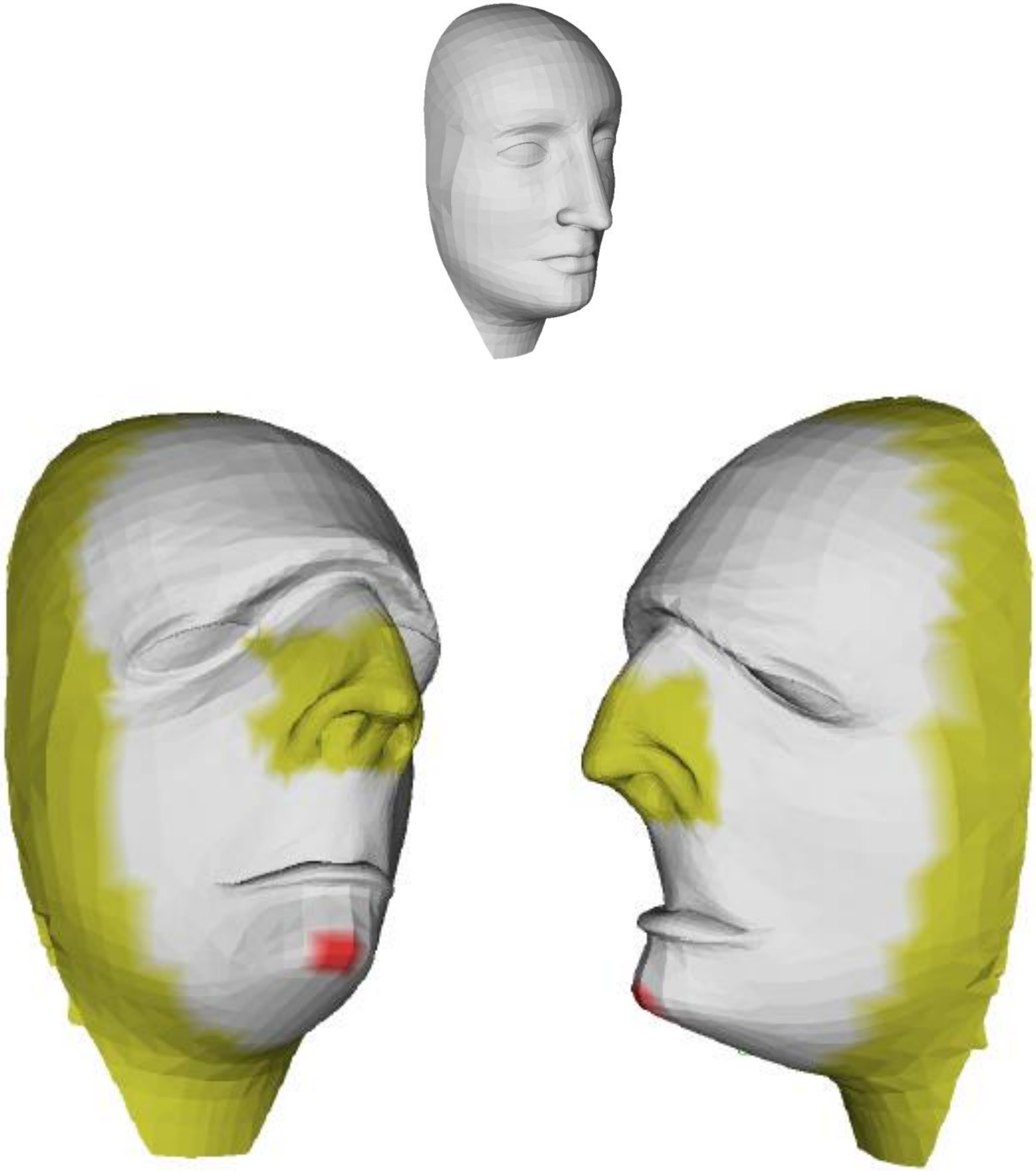
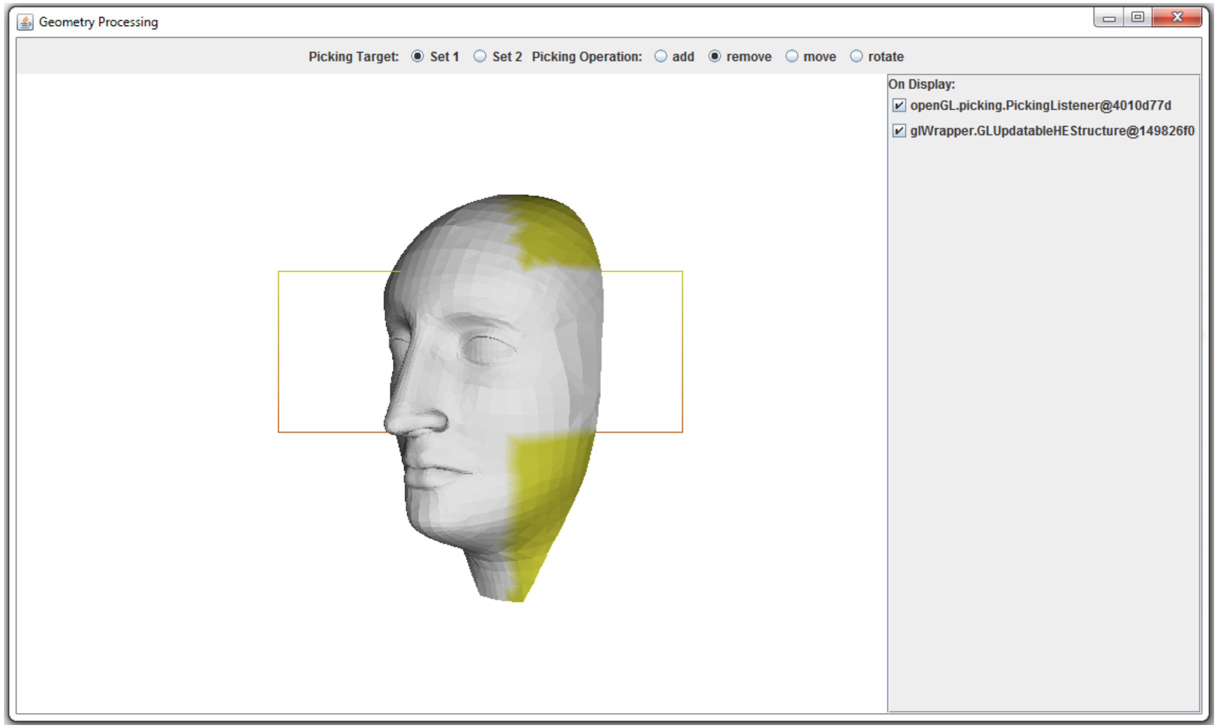
Assignment 6

RAPS: As Rigid as Possible Surface Modeling



Provided GUI



- Selection of picking mode
- Picking: ctrl + mouse dragging

Algorithm overview

- Minimizing the Energy

$$E(S', R) = \sum_i \sum_{j \in N_i} w_{ij} ||(p'_i - p'_j) - R_i(p_i - p_j)||^2.$$

Under (user) constraint

$$p' = p_{constr}$$

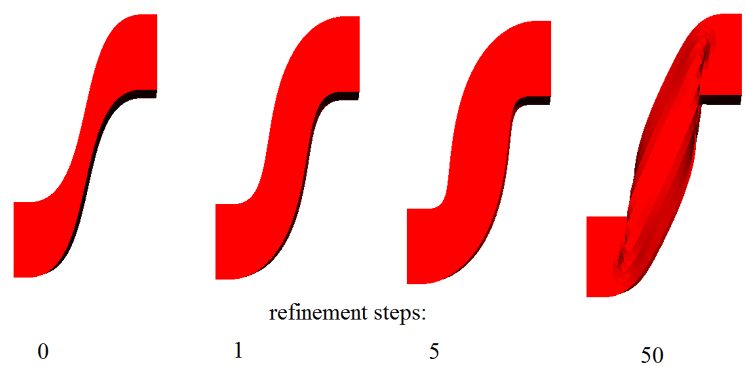
S‘: unknown deformed surface, positions p‘

S: original surface, positions p

Ri: unknown per 1-neighborhood rotations.

Algorithm overview

- Given: input mesh + user constraint
- R_i = id, solve for optimal positions
- Do n refinement iterations:
  - Optimize Rotations
  - OptimizerPositions



Tasks

- Exercise 1: position optimization for fixed rotations (R_i = Id)

  – Amounts to solving

$$(L^T L + w^2 I|_{constr})p' \; = L^T b + w^2 I|_{constr}\, p_{constr}$$

L: unnormalized cotan Laplacian, with zero rows on boundary

Wij: cotangent edge weights

W: constraint weight (e.g. = 100)

Demo

Tasks

- Exercise 2: rotation optimization, iterative refinement.

- Rotations can be found using a SVD decomposition http://igl.ethz.ch/projects/ARAP/svd_rot.pdf
  - Described also on the exercise sheet.

Demo

Implementation Issues

- Experience is not very interactive....

- Bottle Necks;
  - SVD decomposition
    - Paralelize/ implement on the GPU
  - Solution of linear system
    - Multiresolution approach
    - Cholesky factorization with optimal fill in
      - Can be good to ‚invert‘ small sparse matrices (<5000x5000)
- In Java:
  - Only CPU paralelization simple
  - Did not find Cholesky implementation with fill in reduction

Cholesky Factorization

- Idea: Linear systems with triangular matrices are easy to solve

Forward substitution

$$\begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

algorithm:

$$\begin{aligned} x_1 &:= b_1/a_{11} \\ x_2 &:= (b_2 - a_{21}x_1)/a_{22} \\ x_3 &:= (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33} \\ &\vdots \\ x_n &:= (b_n - a_{n1}x_1 - a_{n2}x_2 - \cdots - a_{n,n-1}x_{n-1})/a_{nn} \end{aligned}$$

Back substitution

$$\begin{bmatrix} a_{11} & \cdots & a_{1,n-1} & a_{1n} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & a_{n-1,n-1} & a_{n-1,n} \\ 0 & \cdots & 0 & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$

algorithm:

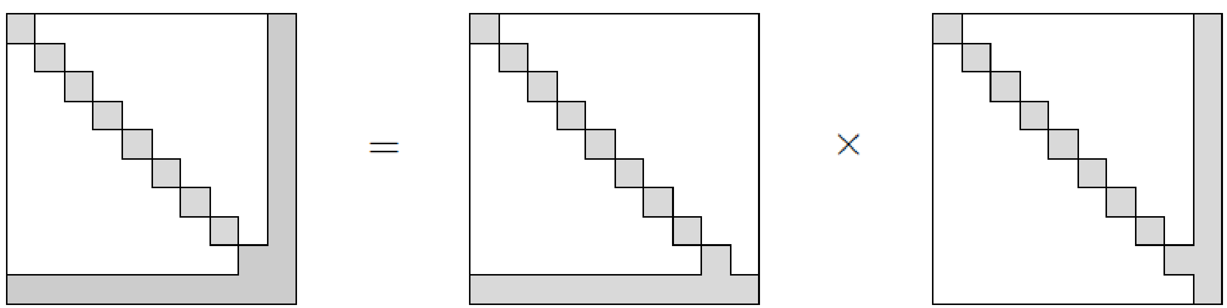
$$\begin{aligned} x_n &:= b_n/a_{nn} \\ x_{n-1} &:= (b_{n-1} - a_{n-1,n}x_n)/a_{n-1,n-1} \\ x_{n-2} &:= (b_{n-2} - a_{n-2,n-1}x_{n-1} - a_{n-2,n}x_n)/a_{n-2,n-2} \\ &\vdots \\ x_1 &:= (b_1 - a_{12}x_2 - a_{13}x_3 - \cdots - a_{1n}x_n)/a_{11} \end{aligned}$$

Cholesky Factorization

- If a matrix is symmetric and positive definite

$$A = D\ D^T, \quad \text{where } D \text{ lower diag matrix}$$

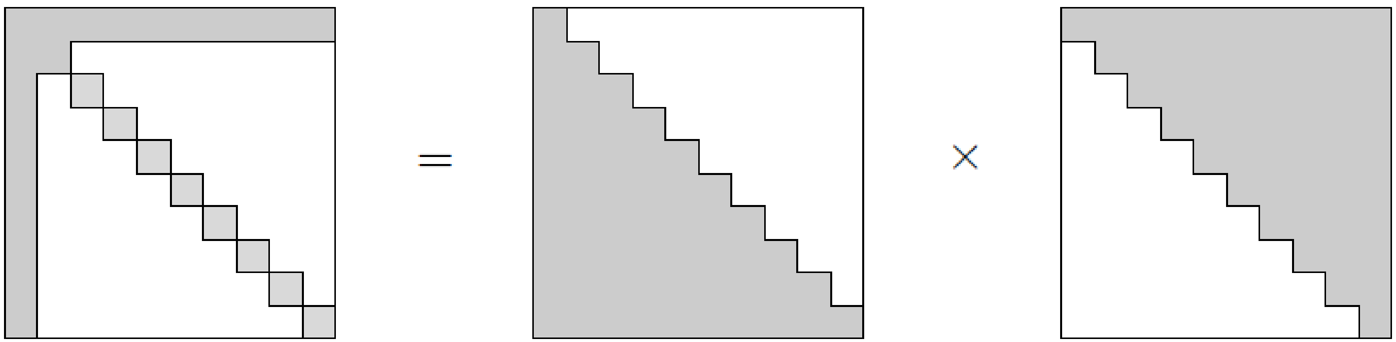
- If A is sparse, D tends to be sparse



- Forward & Backward substitution are fast!

Cholesky Factorization

- But fill in depends on ordering....



- Fill-in reduction = Find an ordering that produces a sparse Cholesky Factorization.
- Optimal ordering: NP hard.

Cholesky Factorization

- Provided in the class Cholesky.java
  - Without fill-in reduction => Provides only small speed up relative to iterative JMTSolver.
  - Is more relieable than the JMTSolver!
- Standard c++ libs: TAUCS/CHOLMOD

Questions?

Basecode will be online in a jiffy (-:

Cholesky Factorization Algorithm

partition matrices in $A = LL^T$ as

$$\begin{bmatrix} a_{11} & A_{21}^T \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} l_{11} & L_{21}^T \\ 0 & L_{22}^T \end{bmatrix}$$

$$= \begin{bmatrix} l_{11}^2 & l_{11}L_{21}^T \\ l_{11}L_{21} & L_{21}L_{21}^T + L_{22}L_{22}^T \end{bmatrix}$$

algorithm

1. determine l_{11} and L_{21}:

$$l_{11} = \sqrt{a_{11}}, \qquad L_{21} = \frac{1}{l_{11}}A_{21}$$

2. compute L_{22} from

$$A_{22} - L_{21}L_{21}^T = L_{22}L_{22}^T$$

   this is a Cholesky factorization of order $n-1$