Intro. to Geometry Processing
600.756

http://www.cs.jhu.edu/~misha/Spring17a/
Assignment 2

✓ Computed mean/Gaussian/principal curvatures!
✗ Did not compute the principal directions correctly
  • As a sanity check, validate that the principal directions are orthogonal
Assignment 3 (smoothing)

• Smooth a 3D mesh using the gradient-domain formulation:

\[
\tilde{c}_\alpha = \arg\min_{c \in \mathbb{R}^n} \|c - c_\alpha\|^2 + \varepsilon \|\nabla c\|^2
\]

• \(\{c_\alpha\}_{\alpha \in \{x, y, z\}}\) are the input \(x-, y-, z\)-coordinates

• \(\{\tilde{c}_\alpha\}_{\alpha \in \{x, y, z\}}\) are the output \(x-, y-, z\)-coordinates

• \(\varepsilon\) is the smoothing weight

• \(n\) is the number of vertices
Assignment 3 (smoothing)

• Smooth a 3D mesh using the gradient-domain formulation:
  \[ \hat{c}_\alpha = \arg\min_{c \in \mathbb{R}^n} \|c - c_\alpha\|^2 + \epsilon \|\nabla c\|^2 \]

smooth --out o.ply --smooth 1e-4 --in armadillo.bad.ply
Assignment 3 (smoothing)

• Smooth a 3D mesh using the gradient-domain formulation:
  \[ \tilde{c}_\alpha = \arg\min_{c \in \mathbb{R}^n} \|c - c_\alpha\|^2 + \varepsilon \|\nabla c\|^2 \]

Recall:

• If we set the \( M \) and \( S \) to be the stiffness matrices and \( c \) to be the input vector of coefficients, then the solution \( \tilde{c} \) is the minimizer of:
  \[ E(\tilde{c}) = (c - \tilde{c})^t \cdot M \cdot (c - \tilde{c}) + \varepsilon \cdot \tilde{c}^t \cdot S \cdot \tilde{c} \]
  \[ = \tilde{c}^t \cdot (M + \varepsilon \cdot S) \cdot \tilde{c} - 2\tilde{c}^t \cdot M \cdot c + c^t \cdot M \cdot c \]

• Taking the gradient with respect to \( \tilde{c} \) and setting to zero, we get:
  \[ 0 = 2(M + \varepsilon \cdot S) \cdot \tilde{c} - 2M \cdot c \]
  \[ \tilde{c} = (M + \varepsilon \cdot S)^{-1} \cdot M \cdot c \]
Assignment 3 (smoothing)

• Smooth a 3D mesh using the gradient-domain formulation:
  \[ \tilde{c}_\alpha = \arg\min_{c \in \mathbb{R}^n} \|c - c_\alpha\|^2 + \varepsilon \|\nabla c\|^2 \]

Use the **Eigen** package:

  • **Eigen::VectorXd**
    - Vectors (with double values)
  
  • **Eigen::SparseMatrix< double >**
    - Sparse matrices
  
  • **Eigen::Triplet< double >**
    - A triplet of values storing the column/row index and the associated matrix entry
  
  • **Eigen::SimplicialLLt< Eigen::SparseMatrix< double >>**
    - Solver for sparse symmetric positive definite linear systems
Assignment 3 (smoothing)

• Smooth a 3D mesh using the gradient-domain formulation:
  \[ \tilde{c}_\alpha = \arg\min_{c \in \mathbb{R}^n} \| c - c_\alpha \|^2 + \varepsilon \| \nabla c \|^2 \]

Use the Eigen package:
  • Note that standard algebraic operators are defined for matrices and vectors:
    ```cpp
double s;
Eigen::SparseMatrix< double > A, B, C;
Eigen::VectorXd x, b;
...
C = A + B * s;
b = C * x
```
Assignment 3 (smoothing)

• Smooth a 3D mesh using the gradient-domain formulation:
  $$\tilde{c}_\alpha = \arg\min_{c \in \mathbb{R}^n} \|c - c_\alpha\|^2 + \varepsilon \|\nabla c\|^2$$

1. Normalize the input so that it has unit area
   (The energy is scale-dependent so normalize so that $\varepsilon$ is consistent)
Assignment 3 (smoothing)

• Smooth a 3D mesh using the gradient-domain formulation:
\[
\tilde{c}_\alpha = \arg\min_{c \in \mathbb{R}^n} \| c - c_\alpha \|^2 + \varepsilon \| \nabla c \|^2
\]

1. Normalize the input so that it has unit area

2. Compute the mass and stiffness matrices
   • Set the entries using the \texttt{Eigen::SparseMatrix::setFromTriplets} method which takes a \texttt{std::vector< Eigen::Triplet >} object
     • If the same entry index pair appears multiple times in the \texttt{std::vector}, the corresponding entries are summed
Assignment 3 (smoothing)

• Smooth a 3D mesh using the gradient-domain formulation:

\[
\tilde{c}_\alpha = \arg\min_{c \in \mathbb{R}^n} \|c - c_\alpha\|^2 + \varepsilon \|
\nabla c\|^2
\]

1. Normalize the input so that it has unit area
2. Compute the mass and stiffness matrices
3. Compute the right-hand-side for each coefficient
Assignment 3 (smoothing)

• Smooth a 3D mesh using the gradient-domain formulation:
  \[ \tilde{c}_\alpha = \arg\min_{c \in \mathbb{R}^n} \|c - c_\alpha\|^2 + \varepsilon \|\nabla c\|^2 \]

1. Normalize the input so that it has unit area
2. Compute the mass and stiffness matrices
3. Compute the right-hand-side for each coefficient
4. Solve the linear system to get the new coefficient values
   • The `Eigen::SimplicialLLt` constructor takes the matrix to invert
   • The `Eigen::SimplicialLLt::solve` method takes a RHS `Eigen::VectorXd` as its argument and returns the solution
Assignment 3 (smoothing)

• Smooth a 3D mesh using the gradient-domain formulation:
  \[ \tilde{c}_\alpha = \arg\min_{c \in \mathbb{R}^n} \|c - c_\alpha\|^2 + \varepsilon \|\nabla c\|^2 \]

--smooth:
What happens as the smoothing value gets larger?

--iters:
What happens if you run multiple smoothing iterations, updating the mass and stiffness matrices at each iteration using the new geometry?

--cmcf:
What happens if you run multiple smoothing iterations, but only update the mass matrix at each iteration using the new geometry (keeping the stiffness from the first iteration)?