Geometric Modeling

Assignment sheet #4

"PCA & Splines"

(due May 30th/June 1st 2012 during the interviews)

Silke Jansen, Ruxandra Lasowski,
Avinash Sharma, **Art Tevs**, Michael Wand

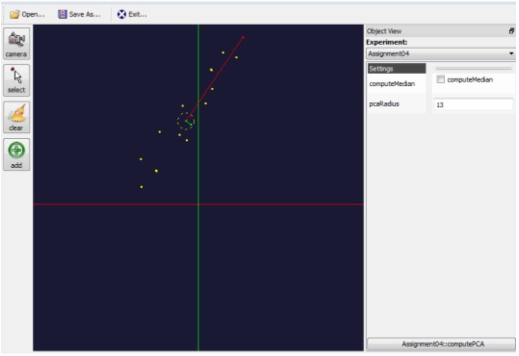
UNIVERSITÄT DES SAARLANDES

M2CI CLUSTER OF EXCELLENCE

mpi

For the following exercise you ***can*** patch GeoX with an upgrade. The upgrade features an easy way of accessing selected points. In order to patch it, just download the **patch.zip** file from the course website and put it over your GeoX installation. Patching is not mandatory and the following exercises can be also implemented without this upgrade!

(1) Principal Component Analysis [6+2 points]

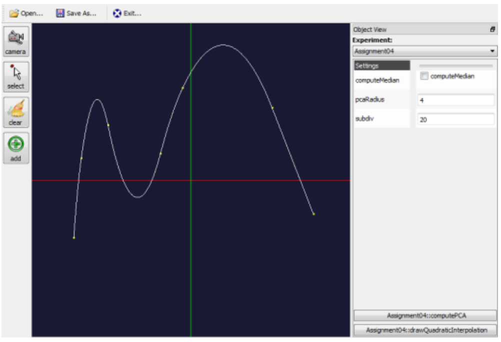


For this exercise you would need to solve an Eigenproblem. In order to do this you can either use the ***StaticMatrix::computeEigenStructure*** method or work with ***powerIteration*** found in IterativeSolvers.h

a) Create a new 2D experiment . Add a button that computes the two principal components of the point distribution. Draw the two axis of the distribution.

b) Do the same as in (a) however compute now the normal and the tangent on any selected points using the PCA (*on the image above the tangent and the normal for the selected point are indicated)*. Consider only points in a specified neighborhood, i.e. neighbor points which distance to the selected point is below given threshold. Move the selected point around and look how normal and the tangent changes, especially if you don't find enough points in the selected range.

(2) Quadratic Spline Interpolation [7 points]



a) A quadratic spline interpolation curve of n data points consists of n-1 segments made of quadratic polynomials ($a_ix^2 + b_ix + c_i = y$) connecting two consecutive data points ($[x_{i-1},y_{i-1}]$ and $[x_i,y_i]$). The tangents of two neighbor segment are equal on every point y_i. Implement a method which computes a_i, b_i and c_i and draws this piecewise polynomial function. For simplicity you can assume, that last segment is linear. Experiment with different point constellations. Consider also to experiment with the same points using polynomial interpolation from the assignment sheet #2.

(3) Uniform B-Splines [3+2 Points]

In this exercise you are asked to draw uniform B-splines controlled by a set of points, i.e., the splines approximate the control points.

a) Implement a method which draws the quadratic uniform B-spline curve modeled by a *n* control points $p_i: 0 \le i < n$. Every spline segment S_i is controlled by three consecutive points in an interval $t \in [0;1]$. The curve must interpolate the first and the last control points. Draw the line in green color.
The piecewise polynomials $S_i(t)$ describing the quadratic B-spline can be computed as:

$$\mathbf{S}_i(t) = \begin{bmatrix} t^2 & t & 1 \end{bmatrix} \frac{1}{2} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_{i-1} \\ \mathbf{p}_i \\ \mathbf{p}_{i+1} \end{bmatrix}$$

Hint: Interpolation of points can be achieved by doubling or tripling the control points.

b) Using the previous control points draw now a cubic B-spline controlled by these points. The spline must interpolate the first and the last points. Draw the spline in red. The corresponding piecewise polynomials $S_i(t)$ are computed as:

$$\mathbf{S}_i(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_{i-1} \\ \mathbf{p}_i \\ \mathbf{p}_{i+1} \\ \mathbf{p}_{i+2} \end{bmatrix}$$