Geometric Modeling

Assignment sheet #8

"Bezier Curves, MLS approximation"

(due June 27th/June 29th 2012 during the interviews)

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

---

(1) Bezier Spline for Approximation [3+3 points]

a) Add a button which finds the control points of a single cubic Bézier curve which approximates the user-defined points in a least squares sense. Draw the resulting curve and its control polygon.
Hint: Assign a unique parameter value t_i to each point $\mathbf{x}_i$ and minimize the squared distances between $\mathbf{x}_i$ and the points $B(t_i)$ on the curve.

b) Add a button which approximates the control points with a C^0 cubic Bézier spline. Allow the user to change the number of spline segments used for the approximation. Output the root mean square (RMS) distance between the approximated points and your curve and observe how it changes with the number of spline segments.

Hint: The RMS distance is defined as :

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} |\boldsymbol{x}_i - B(t_i)|^2}$$

(2) MLS surface approximation [0 + 4 + 10 points]

In this exercise you are asked to perform approximation of a curve by *moving least squares*. A curve is defined by any set of points you choose. In order to solve the exercise you would need to consider *k* nearest neighbors around every point. Use either your kd-tree implementation from the last practical assignment sheet or a brute force approach to gather nearest neighbors.

a) Plot a several points for between which we will reconstruct the surface. Connect those points by a simple line subdivided into *s* parts. We will use these points in order to project them on to the MLS surface.
Hint: This is just the same procedure as usual when drawing a curve in GeoX. This is here just as a reminder.

b) In order to perform MLS approximation, we need to compute a coordinate frame at each point. For this consider to take k (default k=10) nearest neighbors. Compute the local coordinate frame with a weighted PCA approach. Add a button which renders the second vector of the coordinate frame (i.e. normal) at user specified surface point. Use following weight function to compute the influence weight of a point $\boldsymbol{x_i}$ in respect to the point $\boldsymbol{x}$ (let σ be a user specified parameter, default 10):

$$w(\boldsymbol{x_i}, \boldsymbol{x}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{||\boldsymbol{x_i} - \boldsymbol{x}||}{\sigma}\right)^2\right)$$

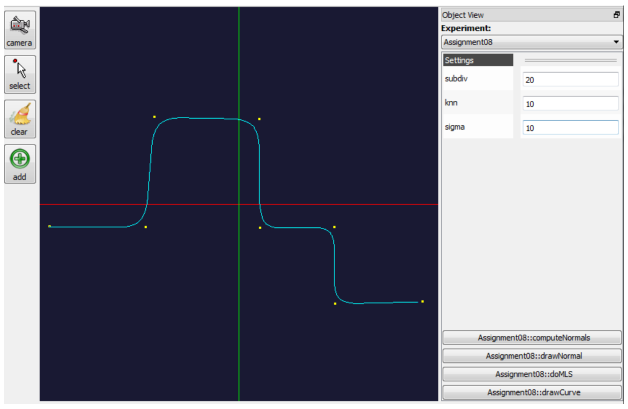
Hint: Reuse your code from the assignment sheet #4, however be careful that this time we are using weighted PCA, i.e. weighted covariance matrix as also weighted centroids.

c) Perform MLS surface reconstruction. For this take points added in (a) between the plotted control points and project them on the MLS surface. As a set of base functions use:

$$\boldsymbol{b}(\boldsymbol{x}) = \{1, x, y\}$$

Experiment with different values for σ and k. What can you observe?

Hint: Don't forget to apply correct coordinate frame transformations.
Hint: In order for MLS to work properly correctly oriented normals are required. Normals computed in (b), however, are not oriented properly, hence MLS reconstruction could be weird. In the case when you plot the points as shown in the screenshot below (yellow points), normals should be just fine.

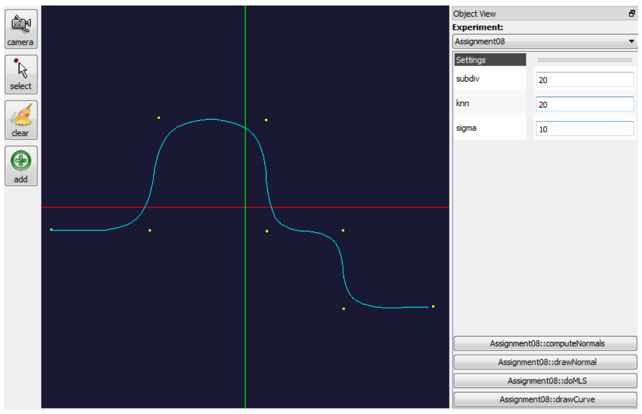


Experiment: Assignment08

Settings

subdiv 20
knn 10
sigma 10

MLS reconstruction using 10 nearest neighbors



Experiment: Assignment08

Settings

subdiv 20
knn 20
sigma 10

MLS reconstruction using 20 nearest neighbors