Course: Introduction to Geometric Modeling (ECS 178), SQ 2016
Professor: Dr. Bernd Hamann

Project 4: An Interactive NURBS Surface Editor
Date due: Wednesday, May 25, 2016

The fourth project deals with the development of an interactive three-dimensional (3D) **editor for NURBS** (non-uniform rational B-spline) **surfaces**. Your NURBS surface editor must be developed on a workstation (or PC or laptop) using the OpenGL (or a similar) graphics library for drawing wire-frame and Gouraud-shaded surfaces (= shaded surface triangulations) obtained by evaluating a NURBS surface using the **four-dimensional (4D)**, **homogeneous version** of the **de Boor algorithm**. **You must not use any of OpenGL's NURBS routines for surface evaluation!** A user must be able to interactively define and alter the **number of control points**, the **control points**, the **weights**, the **orders**, the **knot vectors**, and the **rendering resolution**.

A **NURBS surface** $\mathbf{s}(u,v) = \big(x(u,v), y(u,v), z(u,v)\big)^T$ is a piecewise rational surface given as

$$\begin{aligned}\mathbf{s}(u,v) &= \frac{\mathbf{x}(u,v)}{\omega(u,v)} \\ &= \frac{\sum_{j=0}^{n}\sum_{i=0}^{m} \omega_{i,j}\ \mathbf{d}_{i,j}\ N_i^k(u)N_j^l(v)}{\sum_{j=0}^{n}\sum_{i=0}^{m} \omega_{i,j}\ N_i^k(u)N_j^l(v)}, \qquad u \in [u_{k-1}, u_{m+1}], \quad v \in [v_{l-1}, v_{n+1}],\end{aligned}$$

defined by

- two orders k and l,
- control points $\mathbf{d}_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j})^T$, $i = 0, ..., m$, $j = 0, ..., n$,
- real weights $\omega_{i,j}$, $i = 0, ..., m$, $j = 0, ..., n$,
- a set of real $u-$knots, $\{u_0, ..., u_{m+k} \mid u_i \le u_{i+1},\ i = 0, ..., (m+k-1)\}$,
- a set of real $v-$knots, $\{v_0, ..., v_{n+l} \mid v_j \le v_{j+1},\ j = 0, ..., (n+l-1)\}$,
- B-spline basis functions $N_i^k(u)$, $u \in [u_i, u_{i+k}]$, $i = 0, ..., m$,
- B-spline basis functions $N_j^l(v)$, $v \in [v_j, v_{j+l}]$, $j = 0, ..., n$, and
- surface segments $\mathbf{s}_{i,j}(u,v)$, $u \in [u_i, u_{i+1}]$, $i = (k-1), ..., m$,
  $v \in [v_j, v_{j+1}]$, $j = (l-1), ..., n$.

As discussed in class, it is advantageous (for evaluation purposes) to represent the above rational expression in its corresponding homogeneous 4D form. Thus, the de Boor algorithm can be applied to 4D points, and a resulting surface point in 4D space must be projected into 3D space for rendering.

A user must be able to change all parameters by providing window areas used for parameter display and manipulation. Regarding the manipulation of control points $\mathbf{d}_{i,j}$, a user must be able to **pick** them, **change their positions in space**, and **insert/delete entire rows or columns of control points**. If two or more points are within the same "pick" region in a window, your program must select the point closest to the viewing plane (to the eye). To move a point in 3D space use the left, middle, and right mouse buttons to alter its x-, y-, or z-coordinates.

Besides having to demonstrate your program, prepare a short, about one-page "user's manual" explaining how to use your program.

HAVE FUN!!!