

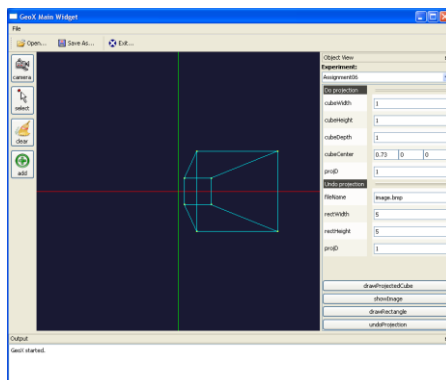


Geometric Modeling

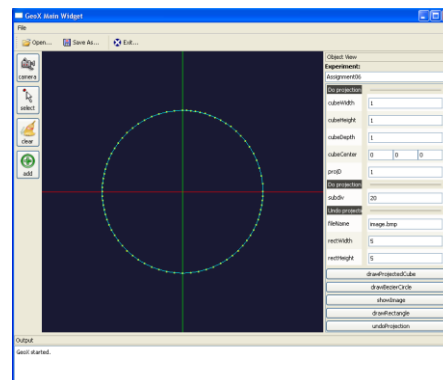
Assignment sheet 6 (Projection & Rational Bezier Curves, due June 15th 2010)

(1) "Do" projection / Rational Bezier Curves [2 + 5 + 3 points]

- Create a new experiment. Draw a 3D cube $[-d,d]^3$ (with $0 < d \leq 1$ specified by user) centered at \mathbf{x} (user should specify \mathbf{x}) and projected to 2D plane by perspective projection. Assume pinhole camera model. Connect cube's corner points with lines.
- Draw 4 quadratic Bezier curves so that their projection corresponds to a circle in 2D. Hint: You can hard code curve parameters/weights in your code.
- Create a new experiment with GLGeometryViewer3D as main viewer. Draw same circle as in b) however, now in 3D space. The 2D circle must be visible from the initial camera position. Move camera around to see how Bezier curves looks alike in 3D space which visualizes the homogenous space.



a.



b.

(2) "Undo" projection [7 + 3 points]

For this exercise you need to update your GeoX package with files provided on the lecture webpage. The updated `GLGeometryViewer` provides you with an interface to access selected points. `ImageLoader.h/cpp` contains a routine to load a bitmap (*.bmp) image and add it as simple 2D point cloud to the viewer. Add a button which loads the image.

In this exercise you are asked to implement a method for removing projective distortion from a perspective image of a plane. This is for example useful for creating orthogonal/parallel projection image out of one taken by pinhole camera. In the following we assume that we are working with a projection of a 3D plane onto 2D image plane only. Everything is applied in 2D space.

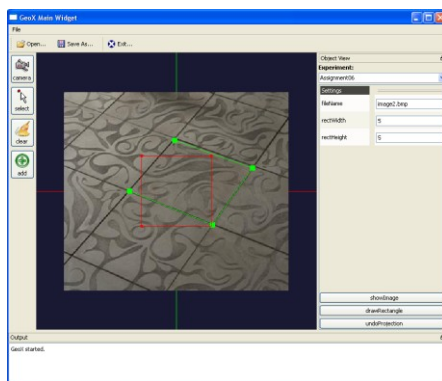
- A pinhole camera model is a perspective projection of a 3D point on 2D image plane. However due to perspective projection the original shape is distorted. In general parallel

lines on a scene plane are not parallel and converge to a finite point in the image plane. There exists a 3×3 matrix \mathbf{H} , which transforms the real projection plane (no distortion) to image plane (perspective distortion), so $\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i$. \mathbf{H} has capability of mapping straight lines to straight lines. Given four corresponding point pairs, i.e. $(\hat{\mathbf{x}}_1, \mathbf{x}_1)$, ..., $(\hat{\mathbf{x}}_4, \mathbf{x}_4)$, where $\hat{\mathbf{x}}_i$ represents \mathbf{x}'_i in homogenous coordinates, compute \mathbf{H} . Assume that $h_{33}=1$. How would you compute \mathbf{H} if h_{33} were unknown?

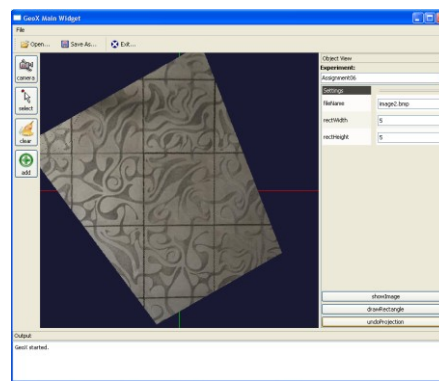
Hint: \mathbf{x} and $\hat{\mathbf{x}}$ are represented in homogenous coordinates, i.e. $(x, y, 1)$.

- b. Add a button which draws a rectangle made of $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$ with user specified width and height centered at origin. Load an image and select four points $(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \hat{\mathbf{x}}_3, \hat{\mathbf{x}}_4)$ corresponding to the drawn rectangle. "Undo" the perspective projection by transforming image points with computed \mathbf{H}^{-1} .

Hint: take care of the order for your points.



(a)



(b)

(a) Loaded image of tiles distorted by perspective projection. Green rectangle was selected by the user. (b) Result of "undo" projection.