TDT4195: Visual Computing Fundamentals

Kristoffer Kjørlaug Larsen
Ragnhild Cecilie Neset

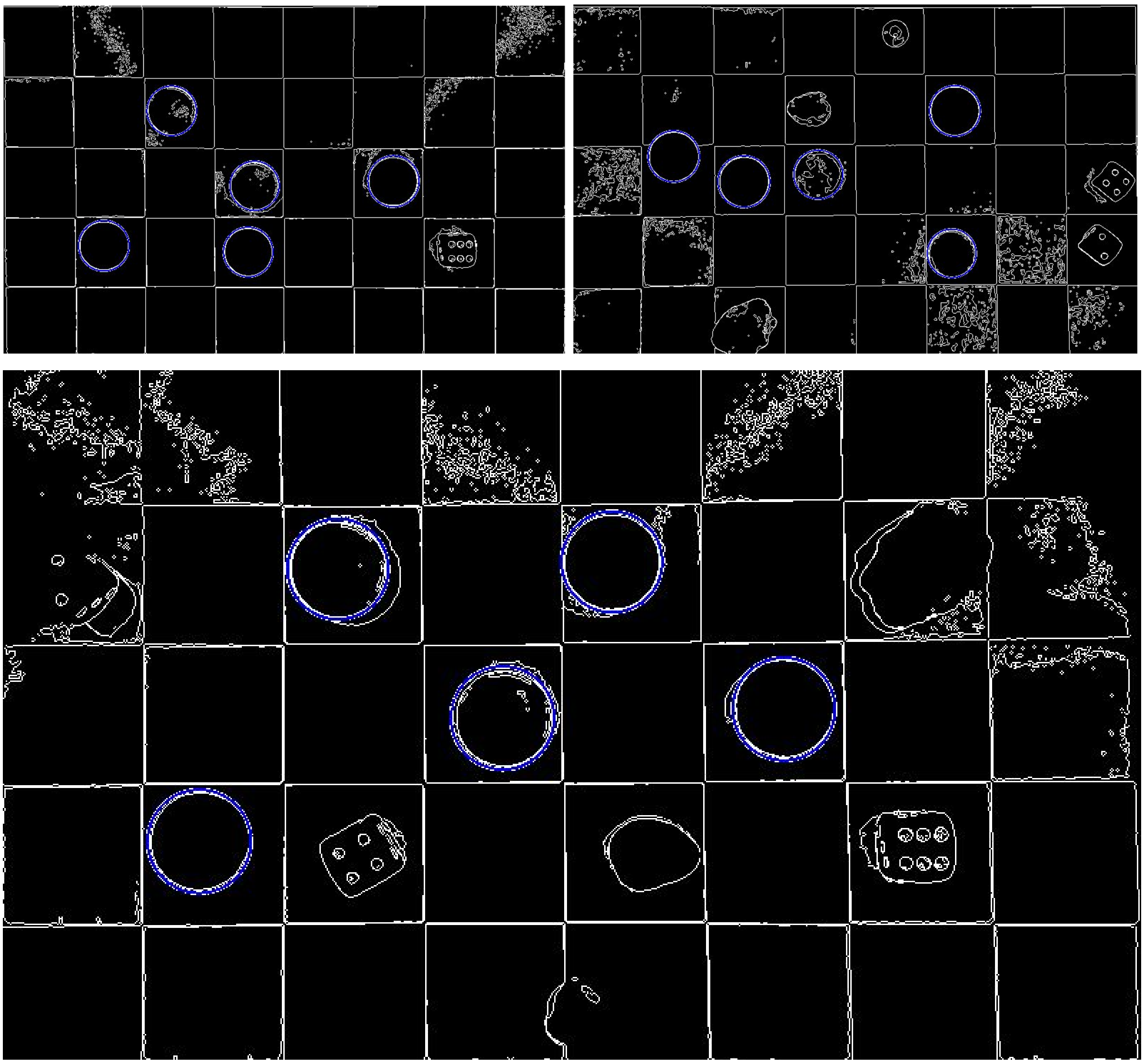
Part 1: Image Processing

The images was processed using Matlab which has a lot of built in functions for image processing, in what is called the Image Processing Toolbox in Matlab.

After the image is imported we split it into three separate images, each image containing one color channel. This is for making it easier to detect pieces that have the same color as the tile. The separate color channels is converted into grayscale images and run through a sobel filter, this is for highlighting the edges in the image.

The three separate images created by the sobel operator are combined, something that really makes all the edges stand out. This again is turned into a grayscale representation, making it easy for matlabs builtin function “imfindcircles” to detect all the circles. “Imfindcircles” finds both the coordinates and the center of all the circles within the given parameters.

The corresponding output is processed and the color in the center of the pieces are extracted to be used by the 3d rendering. The first and second tile colors are also extracted by the program, which makes sure that the 3d rendering is almost identical as the input image.

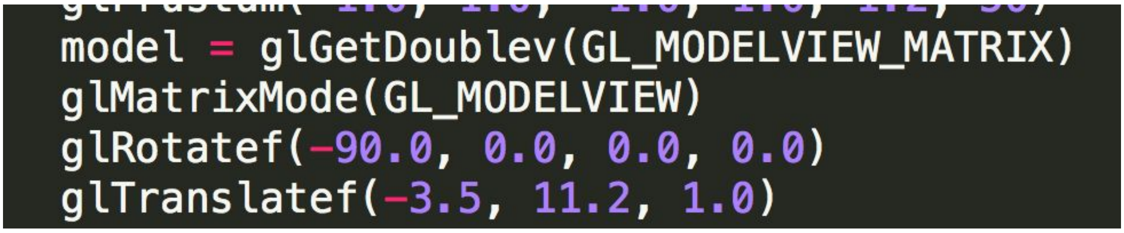


Part 2: Graphics

We used PyOpenGL with Python for this part.

Board: The board was created by 40 scaled cubes, with vertices (for squares) defined in the Draw() - function of the “Tile” class. The board lays flat on the X-Y-plane.
The board is stored as a 2D-array containing Tile-objects. The Tile-objects store the x, y and z values, which are generated using a double loop, starting from 0,0. These values are used to translate each cube in the appropriate directions when the board is rendered.

View: The “camera” - angle has been modified by rotating and translating the Model_View - matrix.



```
model = glGetDoublev(GL_MODELVIEW_MATRIX)
glMatrixMode(GL_MODELVIEW)
glRotatef(-90.0, 0.0, 0.0, 0.0)
glTranslatef(-3.5, 11.2, 1.0)
```

Here the glRotatef- and glTranslatef-functions multiply the current matrix with the x,y,z vector given, and the degrees given as first argument of the glRotate-method.

Pieces: The pieces consists of a scaled gluCylinder with a gluDisc on top. The pieces are placed by modifying the coordinates found in MatLab. As the board, the pieces are placed flat in the X-Y-plane, but with a smaller Z-value, to locate them on top of the board.

The pieces are stored as gamePiece-objects, containing the data for position(x,y,z) and color. These are used when the pieces are rendered.

Moving Pieces: We chose not to implement a mouse-listener due to the added complexity of the code. The most logical choice was then to use keyboard selection. We used Tab to change between pieces, space to deselect, and WASD to move the pieces around the board.

With each user input the scene is rerendered. If for instance the user holds down the the A-key the x-value of the selected piece is subtracted by 0.05 before each rendering. This creates a somewhat smooth animation.

The results: