Modeling and animation

Summaries of lab 1, lab 2 and the final project

Ebbe Strandell,
ebbst916@student.liu.se

Lab1

Implemented required features:

- Reads in and displays 3D objects in PLY-format. Uses the r-Ply library.
- Splits polygons with n>3 vertices into n-2 triangles.
- Calculates face and vertex normals
- Allows the user to toggle rendering with or without normals.
- Includes a basic scene graph where the user can rotate, pan and zoom the camera.

Extra features:

- Uses Vertex Buffer Object Arrays (VBO's) to display the meshes
- Displays text information in the window (basic UI :p)
- Allows the uses to translate, scale and center the mesh by moving the vertices (different from changing the transformation matrix).

Lab2

Implemented required features:

- Reads in volumetric data from a file.
- Uses the marching cubes algorithm to triangulate a mesh depending on the defined ISO-value.
- Subsampling
- Allows for boolean operations between 2 level sets. Addition, subtraction and difference are defined.

Extra features:

- Marching Virus (triangulation in O(n)). (Find one spot on the surface and only look in its neighbors if they share an edge.)
- Supersampling
- Allows the user to interactively move around the level sets freely in the world before applying boolean operations to them.
- Very basic leap-frog algoritm
- Uses a transformation matrix to give the level set its global transformation. Local position is given through the inverse of that matrix. This also makes translation, rotation and scaling very simple. (No ugly hacks, this is quite nice :p (but requires a lot of coding)).

Project 3D Smoke Solver

The project evolves around creating a smoke solver in three dimensions. The model is an extension to a 2D solver written by Jos Stam in 2003.

The project is manly based on thoughts in the following two papers:

- “Stable Fluids”, Jos Stam, 1999.
- “Visual Simulation of Smoke”, R. Fedkiw, J. Stam, HW. Jensen.2001.

The project is built upon the the labs in the course, and almost everything in the lab-course has been used during this project. The project includes two main parts; volume rendering and smoke generation.

The volume rendering system casts rays from a “near clipping plane” towards the opposite “far clipping plane”. The intersection points between the ray and the bounding box of the level set is evaluated. As the ray traverses between these intersections (inside the level set volume) voxel values are sampled to give color and intensity to the corresponding pixel on the near clipping plane.

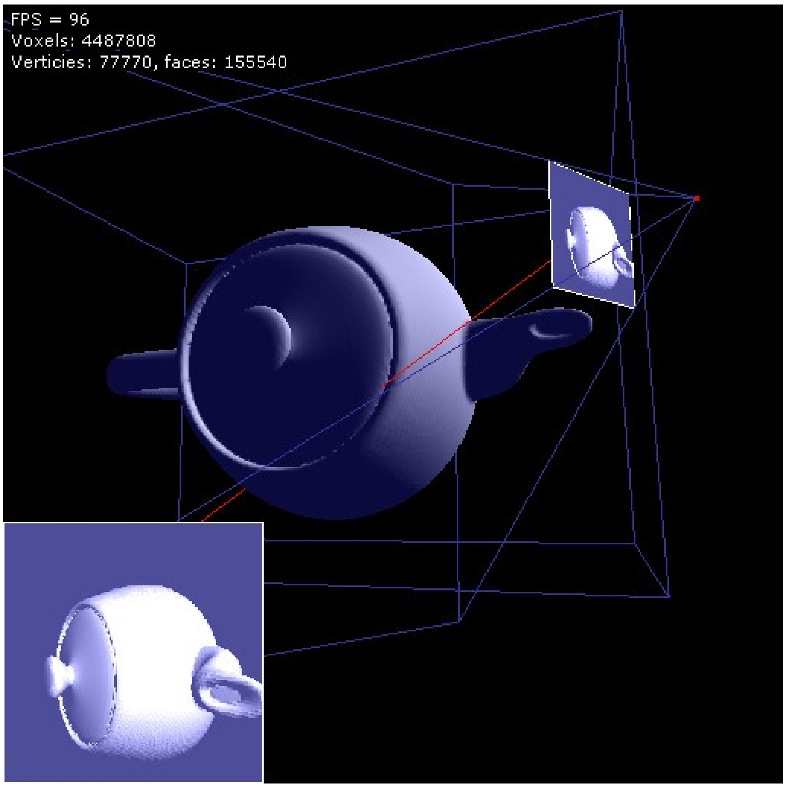
The raycasting system can be used to visualize both ISO-surfaces and volumetric data, such as smoke.

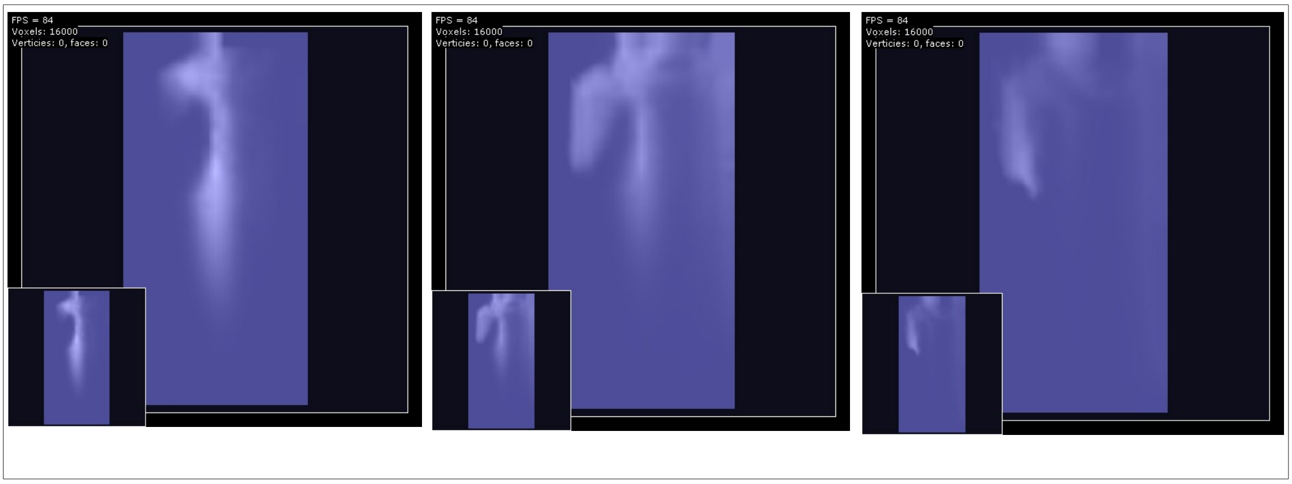
The smoke solver is based on the Navier-Stokes equation for stable fluids. Here forces, densities and temperatures are advected and diffused in space. To ensure that the scalar field remains divergence free and stable a projection step is implemented. The implementation also includes a vorticity refinement step to create realistic movement of the smoke.

The smoke implementation is however not ready yet. It still lacks self-shadowing (which would give it a 3D look) and it needs proper rules at the boarders.

Screenshots from the project:

Volume renderer




FPS = 84
Voxels: 16000
Verticies: 0, faces: 0
FPS = 84
Voxels: 16000
Verticies: 0, faces: 0
FPS = 84
Voxels: 16000
Verticies: 0, faces: 0