# 4. Testing Design

// Integration, unit ve system testing e bak bu aşağıdakiler gereksiz.

Considering our project is predominantly research and development based, scientific papers and algorithm methods research takes a lot of time. Because of these reasons testing and design changes deferred to the later stages of our project.

In this section, we have mentioned about various improvements on the POF system. The pros and cons of the algorithms are discussed in this section. The main criteria are performance and efficiency. Nevertheless, visualization quality is important along with memory and CPU usage is important. Different methods have been tried to get better visuals and to get performance improvements. This section aims to satisfy the discussion of the advantages and disadvantages of the methods and reasons why specific methods have selected will be explained.

***4.1 Marching Cubes***

Marching Cubes is an algorithm for rendering iso-surfaces in volumetric data. The basic notion is that we can define a voxel (cube) by the pixel values at the eight corners of the cube. If one or more pixels of a cube have values less than the user-specified iso-value, and one or more have values greater than this value, we know the voxel must contribute some component of the iso-surface.

Advantages:

• Automatically handles topology changes

Disadvantages:

• Diffusion causes loss of volume & surface detail

• Requires periodic redistancing

***4.1.1 Linear Interpolation***

Linear interpolation is basically an estimation between two points, it is useful for creating new data points from discrete two data points.

The position that it cuts the edge will be linearly interpolated, the ratio of the length between the two vertices will be the same as the ratio of the iso-surface value to the values at the vertices of the grid cell.

***4.1.2 Calculating Normal***

While the triangulation part continuing, additionally we can compute the surface normal of every triangle. A surface normal is basically for each vertex, you can compute an average surface normal at that point by averaging the normal of all triangles sharing that vertex. At each vertex, you can use that normal to compute the lighting at that point. You can interpolate for every point between to determine how bright the surface is at that point. The result is the appearance of edges looks blurry without changing the actual geometry of the object.

A picture containing tree

Description automatically generated

***4.1.3 Mesh Limit***

The triangulation section creates many triangles and holds it as a single mesh. However, a single mesh can only consist of 65,000 triangles. In our project, we can easily exceed this limit. We have added a small bug fix part of the mesh part of the code. If mesh triangles exceed the limit, we create another mesh and fills it until all triangles finish.

***4.1.4 Grid Resolution***

One very desirable control when polygonising a field where the values are known or can be interpolated anywhere in space is the resolution of the sampling grid. This allows course or fine approximation to the iso-surface to be generated depending on the smoothness required and the processing power available to display the surface.

***4.1.5 Reducing errors in concave regions***

Solenthaler [7] is a surface reconstruction technique based on considering the movement of the center of mass to reduce rendering errors in concave regions to achieve a smooth surface from particles. Similar method to Zhu et al. [5] but with reduced reconstruction artifacts even for inhomogeneously.

***4.2 Calculating Weight function***

In general, the scalar field computation affects the surface quality and the

computation time. While [5] is faster compared to [SSP07] . It

suffers from artifacts, i.e.„ spurious blobs, in concave regions.

***4.3 Kernel Function***

***/////// Pros and cons of the kernel functions etc.***

***4.3 Hash***

We have mentioned the advantages of the Hash system in final report. However, hash system had a small error when it comes to finding surface particles. After out test phases, we have figured out that error is the result of the cell that particle belongs is exceeding the limits of the AABB. We have added small check with if statements. If the particle is on the surface. It means that cell exceeds the range and we fix the bug by equalize the exceeding boundaries to AABB boundaries.

The comparison tables with details will be added in final report.