Lauren Schmidt
CSC313
Private Functions Research Page

Voxel Traversal Algorithm for Ray Tracing:
The traversal algorithm consists of two phases: initialization and incremental traversal. The initialization phase starts by identifying the voxel where the ray originates, → u, is found. If the ray origin is outside the grid, we find the point in which the ray enters the grid and take the adjacent voxel. The integer variables X and Y are initialized to the starting voxel coordinates. The variables stepX and stepY are initialized to either 1 or -1 indicating whether X and Y are incremented or decremented as the ray crosses voxel boundaries (this is determined by the sign of the x and y components of → v). Next, we determine the value of t at which the ray crosses the first vertical voxel boundary and store it in variable tMaxX. We perform a similar computation in y and store the result in tMaxY. The minimum of these two values will indicate how much we can travel along the ray and still remain in the current voxel. Finally, we compute tDeltaX and tDeltaY. TDeltaX indicates how far along the ray we must move (in units of t) for the horizontal component of such a movement to equal the width of a voxel. Similarly, we store in tDeltaY the amount of movement along the ray which has a vertical component equal to the height of a voxel. The incremental phase of the traversal algorithm is very simple. The basic loop is outlined below:

```
loop {
	If (tMaxX < tMaxY)  {
		tMaxX = tMaxX + tDeltaX;
		X = X + stepX;
	}  else       {
		tMaxY = tMaxY + tDeltaY;
		Y = Y + stepY;
	}
	NextVoxel(X, Y);
}
```

This is looped until there is a voxel found with a non-empty object list or the object falls out of the end of the grid.
Resource:
https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.42.3443&rep=rep1&type=pdf