

# Using Avizo for better Minecraft constructions

Erwan Plougonven

Lab of Chemical Engineering, Department of Applied Chemistry  
University of Liège, Belgium  
eplougonven@ulg.ac.be

## Abstract

Minecraft is an independent video game released in 2009 that has quickly become a phenomenon, counting over 23 million registered users, and regarded as one of the best games to play at work [1]. It is a multiplayer sandbox game focused on creativity and building. The open world in which the player evolves is a set of cubes of equal size and textured according to what they represent (dirt, stone), defined on a regular 3D grid. The player is free to dig or mine such blocks, storing these in his inventory, and replacing them one by one, where desired. This is the core principle in the construction process.

Even though this open world is defined as cubic blocks on a regular grid, it provides a strong potential for creating elaborate 3D shapes. The difficulty lies in recreating a shape that one wants to model, which is where Avizo has been a valuable tool. In most situations, 3D models are defined as meshes, typically triangulated 2D meshes. If one such model is to be reproduced in Minecraft, either one has to be able to mentally visualize with astounding precision the positions in space where individual blocks should be placed, or possess a tool that firstly converts the model from a set of triangles to a set of cubes on a regular grid, and secondly allows to efficiently locate the positions in 3D space where each block should be placed.

To solve the first point, a module I have developed and integrated in Avizo takes as input a triangulated mesh and a 3D image (a 3D uniform scalar field), the latter providing the 3D grid. For each triangle in the mesh, the module determines what unit cubes, or voxels it intersects in the 3D image, and set the voxel values accordingly. In order for this process to be algorithmically efficient, a small number of voxels should be tested, and the cube/triangle intersection test should be very quick. The first point is handled by beginning with the voxels that contain the triangle vertices and iteratively examining the 26-neighbourhood of the voxels found to intersect the triangle. The second point is handled by decomposing the intersection test using various projections of the voxel centre along the grid diagonals and length measurements using the infinity norm. The center image in figure 2 is the result of such a conversion from the mesh shown in the left image. In order to display a 3D image as a set of cubes, another module has been integrated in Avizo.

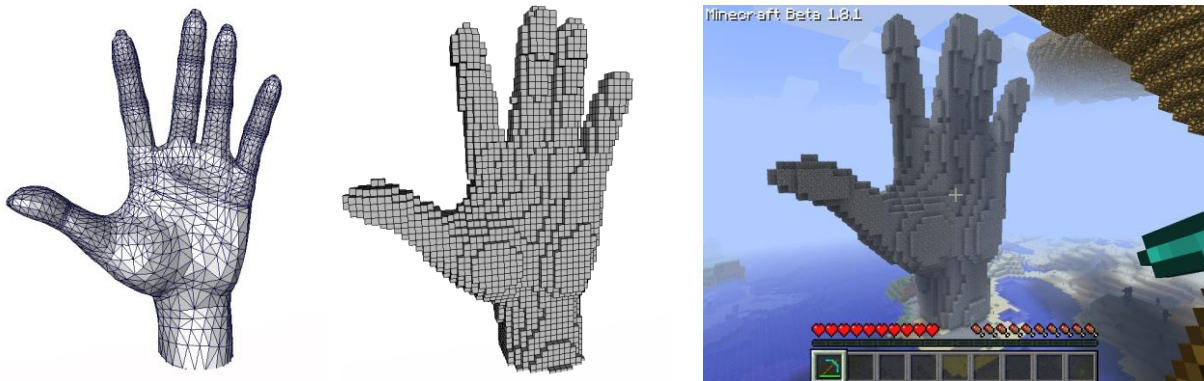
Finally, concerning the second point, by effective usage of the BoundingBox and LocalAxis (for displaying gridlines and axis positions), and OrthoSlice modules, the Minecraft player can easily determine, slice by slice, where to place each block in the game. One such result is the right image of figure 2: the blocks have been placed one by one from the ground up, using the setup described above in Avizo. Although the block placing is tedious, the resulting satisfaction and praise by fellow players is well worth the effort.

## References:

- [1] Jim Rossignol, "50 games to play at work", PC Gamer, July 2010.  
<http://www.pcgamer.com/2010/07/05/50-games-to-play-at-work/5/>



**Figure 1.** A sunset in the cubic world of Minecraft.



**Figure 2.** Illustration of the passage from a surface model (left) that a Minecraft player would like to reproduce to its final construction in the game (right). The center image shows the result of a new compute module that converts the triangle set into a 3D image.