

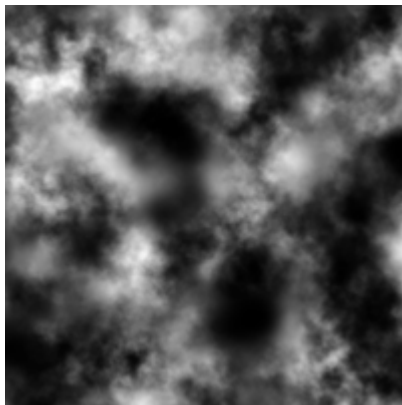
HeightMap Exporter for SketchUp v6 and higher V 1.0

D. Bur, December 2009

This script is intended to generate images of your models that can be used as heightmaps.

It has been tested with SketchUp 6 (Pro and free) and SketchUp 7 (Pro and free).

What is a heightmap ?



In computer graphics, a heightmap or heightfield is a raster graphics image or bitmap representing a rectangular grid of pixels, or points of color, used to store values, such as surface elevation.

Generally, a height map is a black and white 256 grayscale image, exported in RAW format. Whiter areas are more elevated than darker areas.

Heightmaps are used for instance in game engines or in fractal terrain generators, or as bump mapping textures. In 3D softwares, it is often the 3d mesh that is generated by the heightmap. So what this script is for ? It takes the 3D meshes or faces of your model and creates a grayscale or color bitmap that can be used to be re-mapped on the model itself, or to create more efficient terrains in other softwares.

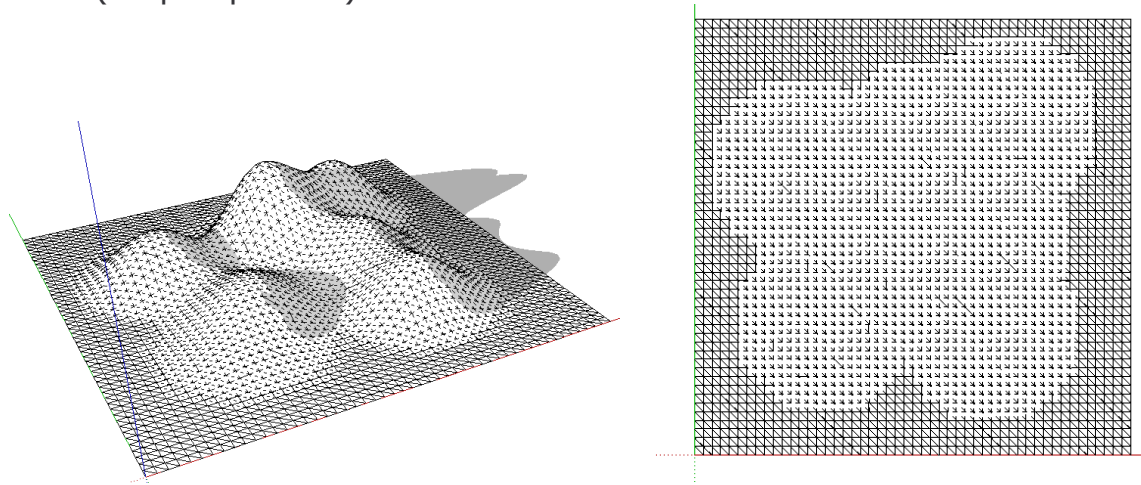
See this page for more details: <http://en.wikipedia.org/wiki/Heightmap>

1. Installation:

Put the file " mesh2HeightMap.rbs" in your Plugins folder. Restart SketchUp. You should find a "Model to HeightMap" item in your "Plugins" menu.

2. Exporting a heightmap:

Activate the top view in SketchUp and set your camera to orthographic view (no perspective).



Example mesh in perspective view and top view

Select "Model to HeightMap" from the "Plugins" menu.

Select the options in the dialog box:



- **Heighmap:**

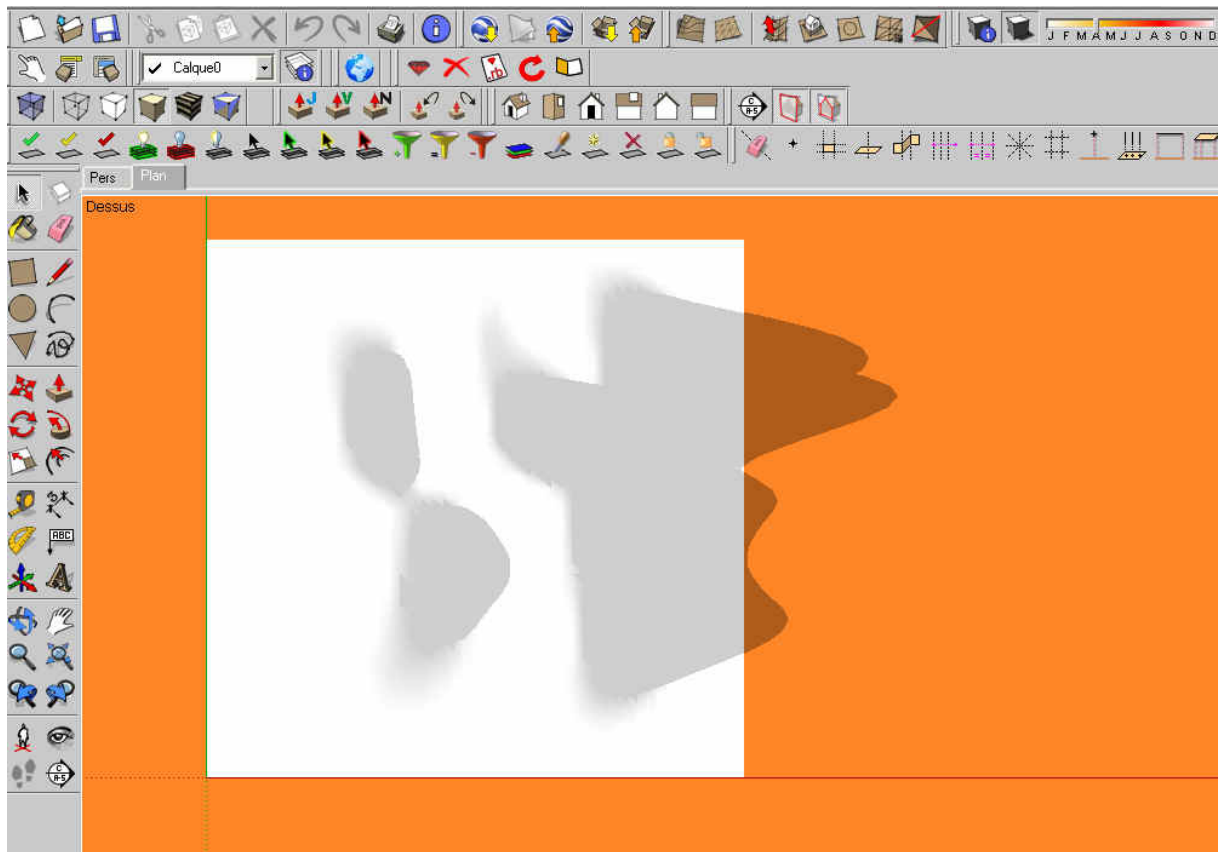
- Greyscale 8 bits BMP: creates an image with pixels colored on a 256 grey values scale, from white to black at the BMP format.
- Color 8 bits BMP: creates an image with pixels colored on a 256 red, green, blue values scale, from purple to red (conventional altitudes color chart scale) at the BMP format.
- Greyscale 8 bits RAW: creates an image with pixels colored on a 256 grey values scale, from white to black at the RAW format.
- Greyscale 16 bits RAW: creates an image with pixels colored on a 65536 grey values scale, from white to black at the RAW format.

- Color 16 bits RAW: creates an image with pixels colored on a 65536 red, green, and blue values scale, from purple to red, at the RAW format.



- **Mask:**

The image that is created covers the entire viewport (the graphic window of SketchUp). The model never matches exactly the viewport, so a mask is created to easily depict the model outline, as show below (orange color):



Select a color for the mask. Note that you selected the "greyscale 8 bits raw" option this has no effect: the mask will always be black.

- **Blur:**

The resulting heightmap will have better gradients if a Gaussian blur is applied to the image, especially if it is a 8 bits one (256 values).

Hit the OK button, then enter a file name (with or without extension) for the resulting heightmap.

- **Crop:**

If "Yes", the resulting heightmap will be cropped to the smallest size encompassing the model projection in top view:

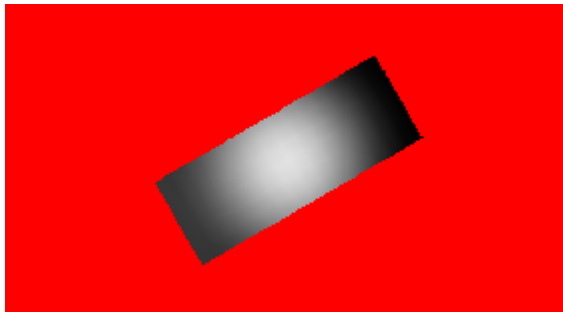


Image not cropped

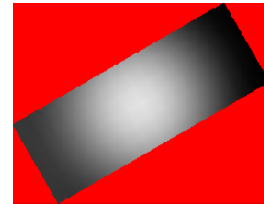
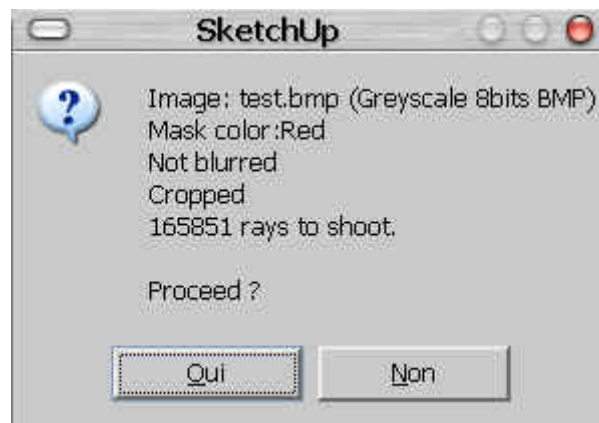
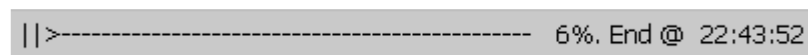


Image cropped

A message will be displayed before launching the script:



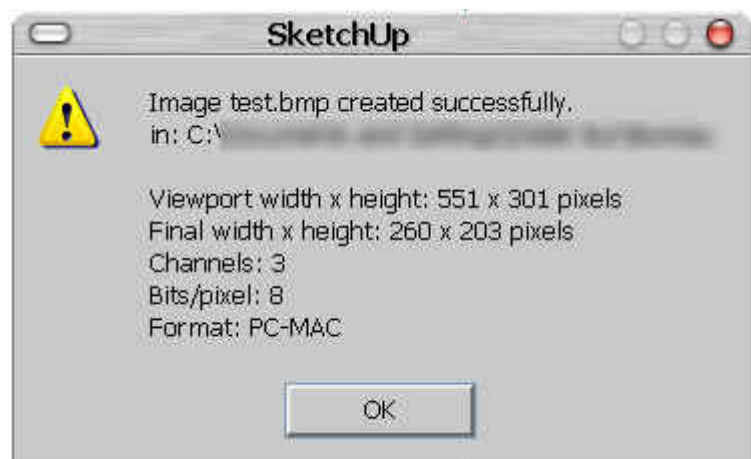
Watch the status bar as the script computes the image. The process can take some time, especially if the model has a lot of faces.



Status bar when exporting the heightmap

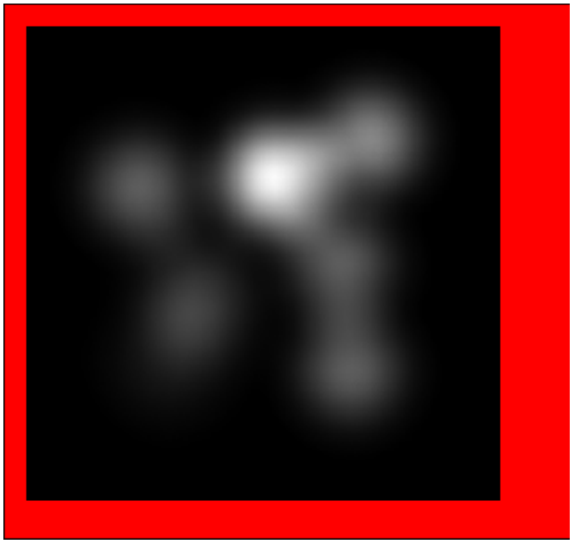
When the heightmap is ready, some information is displayed:

Note: This information will be necessary if you want to open a RAW image in Photoshop for instance.

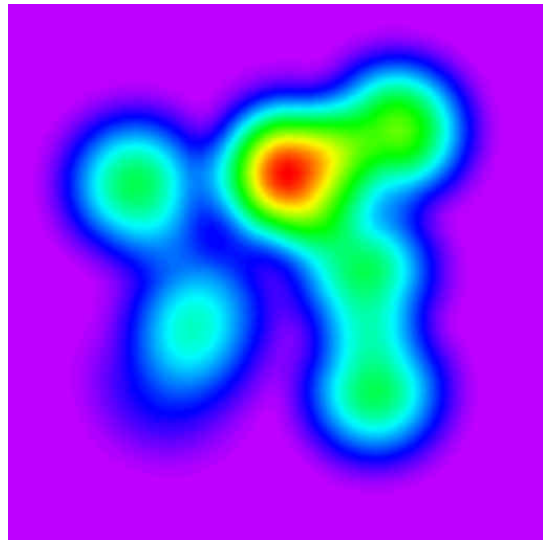


Important:

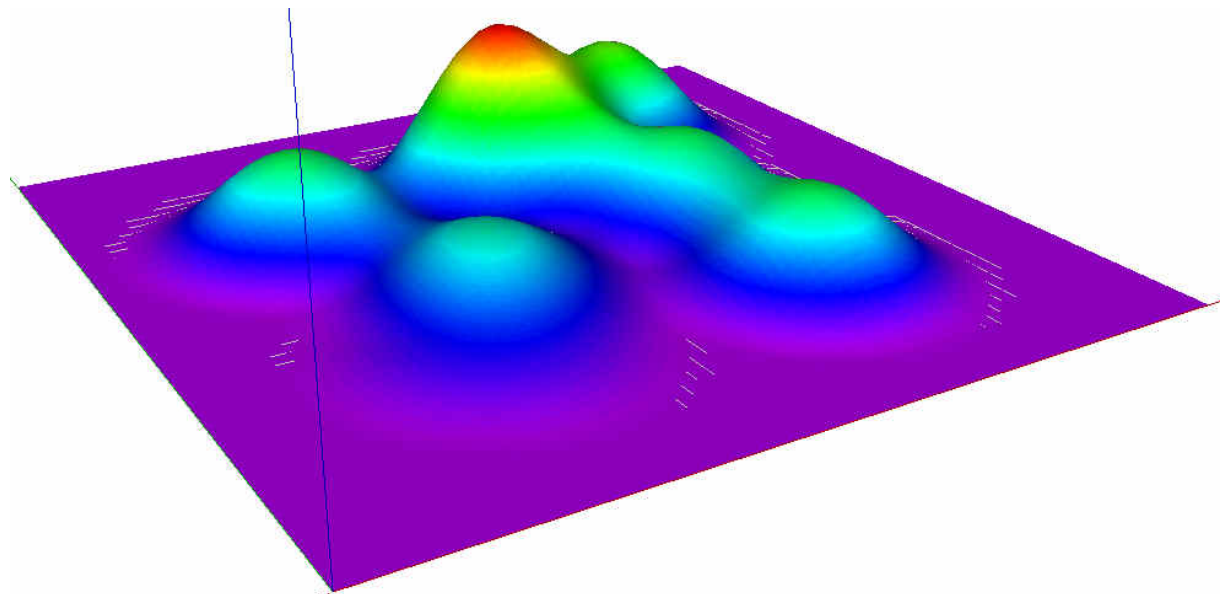
Because the script uses a ray-casting method to create the heightmap, it shoots rays through each pixel of the viewport, from top to bottom, along the camera direction. That's why you should have any obstacle between the camera and the model, even of frozen layers. No hidden objects should interfere the ray casting process as well.

Examples:

Greyscale 8bits BMP, red mask, not cropped



Color 8bits BMP, cropped



Color heightmap re-mapped onto the original terrain mesh