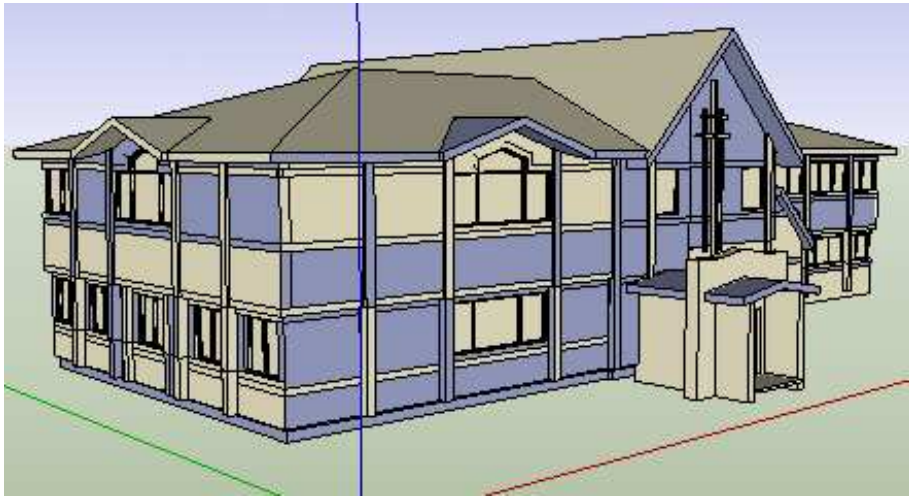


## View-Based Face Reverser v1.0. for SketchUp v5 and v6 (Free and Pro)

© D. Bur, 01.21.2008

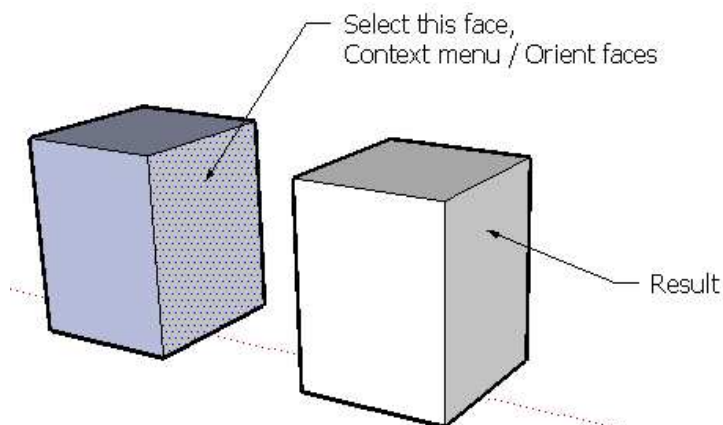
This script contains two tools to quickly orient/invert faces if they are not "pointing" in the correct direction. You all know the "blue syndrom", especially when importing model from other CAD apps:



*Model by Boofredlay, imported from Revit*

Sketchup has two commands to manually reverse wrong faces: "Context menu / Invert faces", and "Context menu / Orient faces"

With the first, it is up to you to select faces to reverse, with the second you can select an object or a set of contiguous faces, Sketchup will orient faces, assuming the face you have selected is oriented correctly.

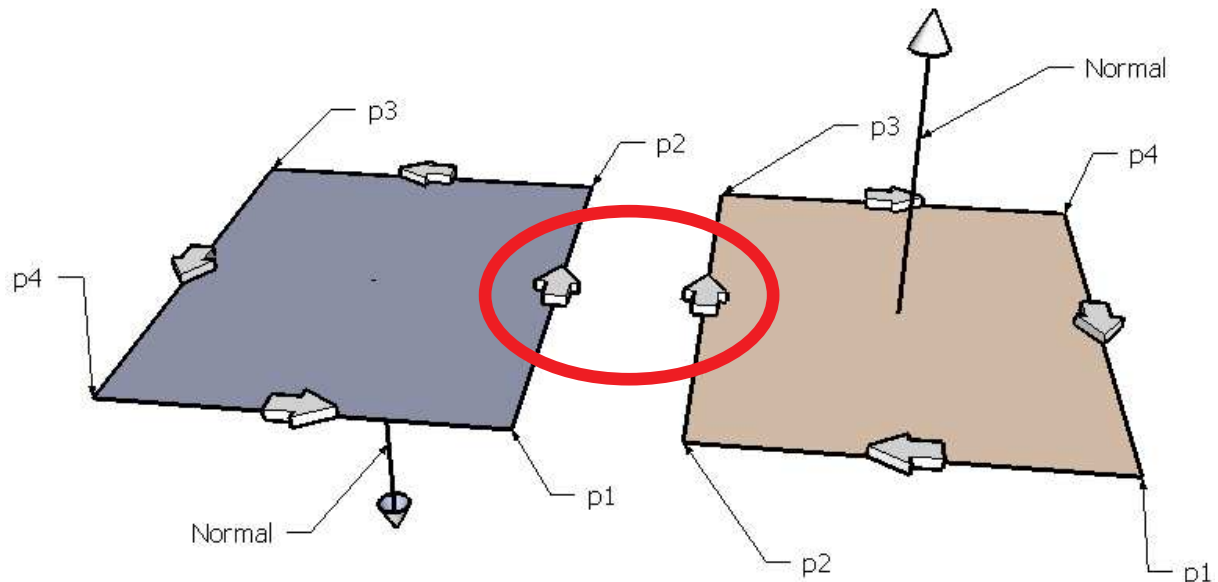


*Selecting a single "good" face of an homogeneous object makes the entire object look good.*

### A bit of theory:

SketchUp is the only modeler (AFAIK) with two-sided faces. Let's see how it defines whether you see the front or back side:

In BREP (boundary representation models), a face is stored as a sorted set of edges, each edge pointing to two points. The normal of a face is the cross product between two edges of a face (assumed all edges of the face are coplanar), and assumed that the points are stored in a clockwise order.



When you select "Orient faces" in the context menu, SketchUp parses adjacent faces, and if it finds that a shared edge is pointing to vertices which are stored in the same order (they have the same vector shown in the red circle above), it reverses the adjacent face (the left one in this example). This is, say, the "data-base based" way of reversing wrong faces, the observer's viewpoint (camera) not being involved.

In image synthesis and 3D model display, a face is considered visible when its normal vector is pointing to the camera and its points are visible in a clockwise order. This is what I call the "view-based" way.

This made me think of a script to scan the SketchUp viewport from a perspective camera, checking whether each visible face's normal vector is pointing to the camera or not. Adding virtual cameras around the model and iterate through these viewpoints can scan almost the entire model at a given eye's height. This can take some time on huge models, but it is better for you to launch the script and go drink a coffee, than manually reverse each wrong face, IMHO ;)

## Installation:

Unpack the archive in the SketchUp Plugins folder. Restart SketchUp.

You should have:

"vbfr.rb" in your Plugins folder (vbfr stands for view-based face reverser)  
" VBFR User Guide.pdf" in your Plugins folder. You can move this file where you want.

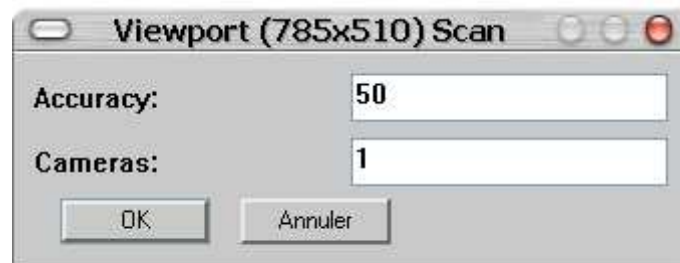
## Menu:

There is no option or sub-menu in the standard menus of SketchUp.  
The tool options will only appear in the context menu (right-click).



## 1. Scanning the model:

Select anything in the model,  
Right-click and choose "View-based Face Reverser / Scan model".  
The below dialog is displayed:

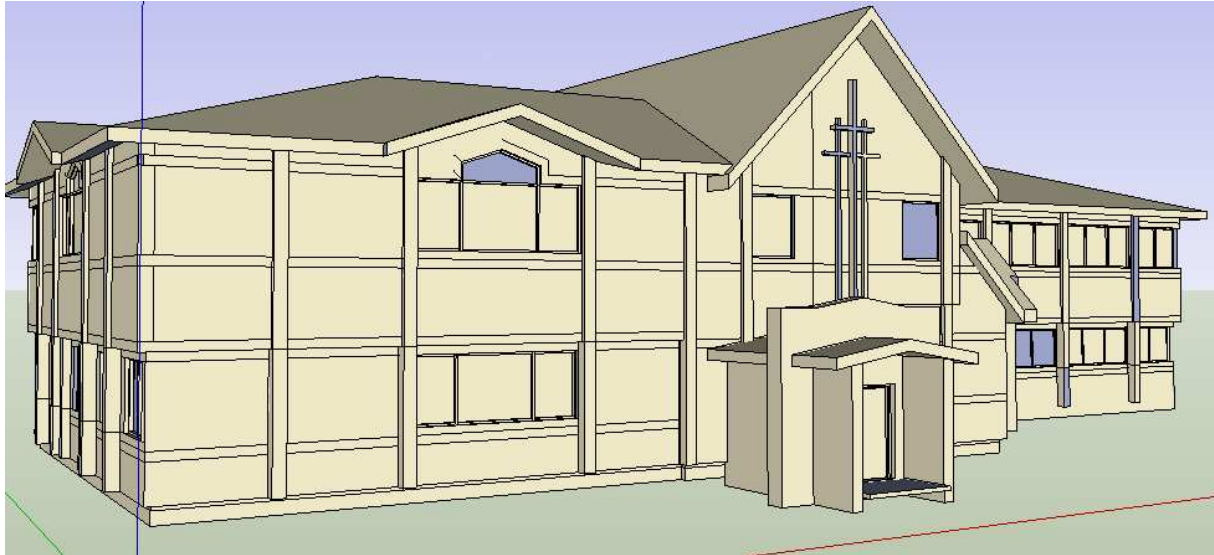


**Accuracy:** this is basically the grid size of the viewport (see "How it works" paragraph below). The higher the value, the longer the processing time.

**Cameras:** this is the number of virtual cameras around the model, which will scan for wrong faces. Default is 1, using only the current viewpoint. Using 4 to 6 cameras takes obviously 4 to 6 more time but will give good results in most cases.

*Note: Have a look at the status bar when processing to see the progress.  
It is not mandatory to have all the model within the viewport (no need to do a "zoom extents"), but only faces visible on screen will be tested.*

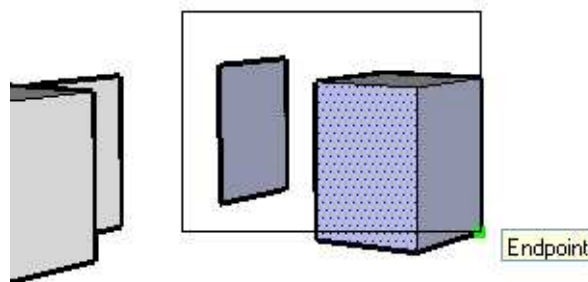
The script can reverse faces inside groups and components, and inside nested groups or nested components, at any level.



*Model shown at page 1, after 1 scan, accuracy 70, viewport size 981x670, grid step 14 pixels, 40 seconds.*

### 1. Scanning a region of the screen:

Select anything in the model, right-click and choose "View-based Face Reverser / Scan screen region". Click and drag a rectangle or click two points (the diagonal of the rectangular area you want to examine):



*Note: Have a look at the status bar when processing to see the progress. Scanning large areas can be slow.*

The script can reverse faces inside groups and components, and inside nested groups or nested components, at any level.

### 3. Scanning the selection:

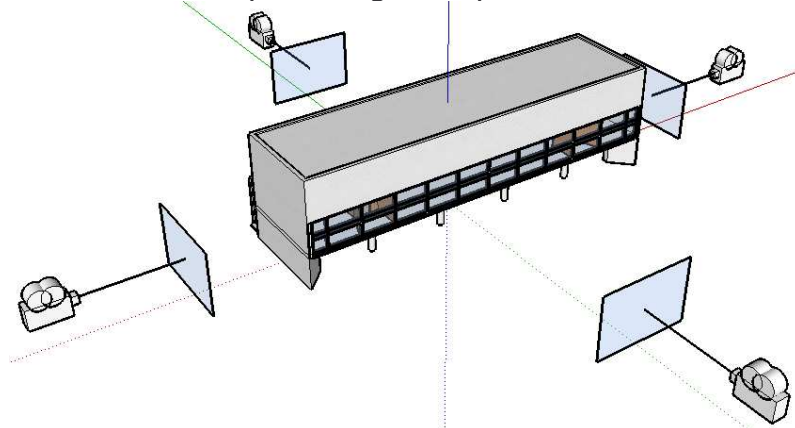
Select anything in the model, right-click and choose "View-based Face Reverser / Scan selection". There is no dialog and the faces of the selection are reversed immediately if they are visible from the camera viewpoint and wrongly oriented.

*Note: scanning the selection can be slow, because when you select faces with the select tool, you also select faces that are hidden, or behind the faces visible on the screen.*

## How it works:

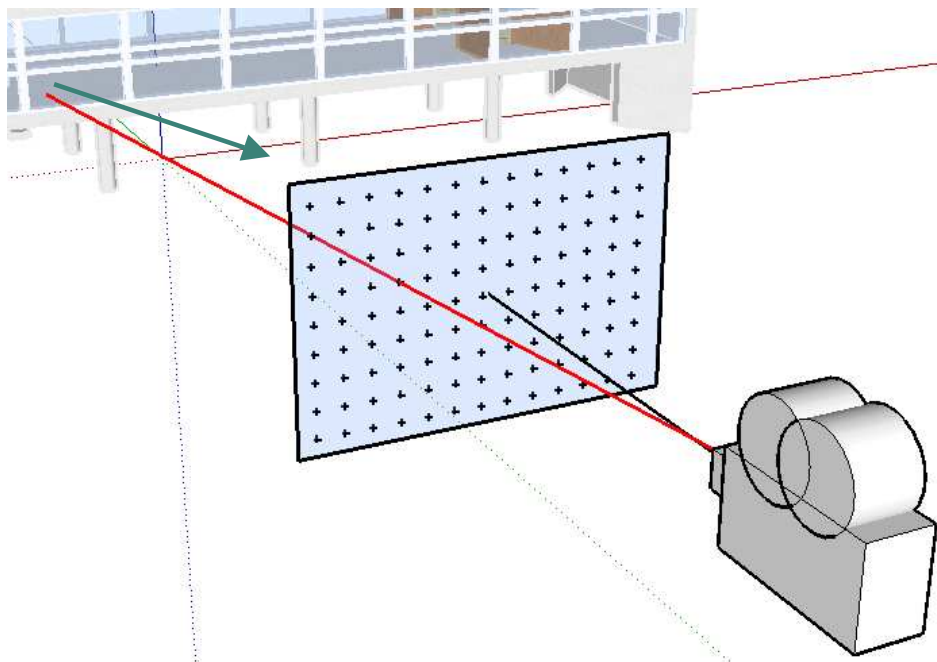
Quite simply in fact:

When seen from a perspective camera, the model is projected (conic projection) on the viewport (the SketchUp window). This viewport is divided by a grid, using the accuracy value to define the step of the grid step.



*Model seen by 4 virtual cameras on 4 viewports (blue rectangles).*

For an accuracy of 50 for instance, the viewport grid will have 2500 points (50 along the width and 50 along the height of the SketchUp window).



A ray is traced through each grid point, from the camera to the model (the red line). When it hits a face, the normal of the face (the green line) is compared to the camera direction vector (the black line). If the resulting angle is less than  $\pi/2$ , the face has to be reversed.

## Known issues:

Inner spaces cannot be scanned by exterior cameras.

You cannot access to interior spaces with section planes, since the raytest Ruby method doesn't care of hidden layers, hidden entities or section planes.

When a face is visually small on the screen, it may not be hit by a ray (at least with a reasonable accuracy value). Zoom to the model when this occurs.  
Some faces won't be reversed when they belong to curved faces.

---

