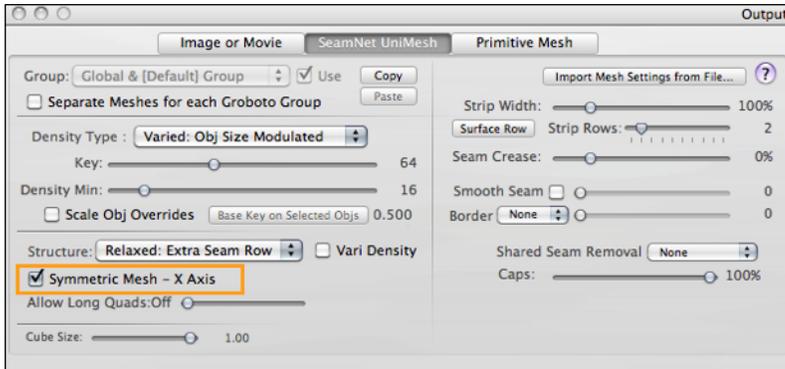
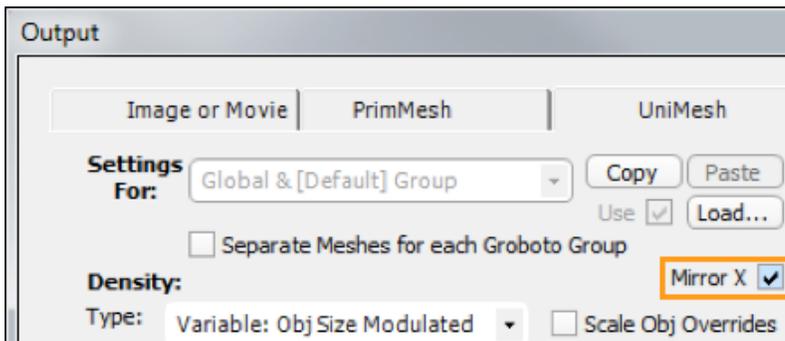


Symmetric Meshes • Groboto 3.3.0

The major new feature in this version is creation of mirror-symmetric mesh from Groboto models. It was always possible to create mirror-symmetric models in Groboto, and when converted to mesh they produced mirror-symmetric shapes within the accuracy determined by mesh density, but there was no exact mesh symmetry, vertex for vertex and face for face. This is what is needed to perform mirror-symmetric editing in any polygonal modeler - for every vertex and polygon on the right there has to exist an exact match on the left.



Mac OS X



Windows

Mirror symmetry has to be explicitly enabled when exporting (or previewing) the mesh, by checking Mirror X in the UniMesh Output dialog (File->Export):

Only X mirror symmetry is available. The Groboto model should itself be mirror-symmetric, but this is not an absolute requirement. Groboto simply creates a mesh from the right-hand side of the model ($X > 0$), then generates the exact mirror image of this mesh on the left and stitches the two meshes together. If the left side of the model is different, it will have no effect - you'll still get a mirror image of what's on the right. We do not recommend creating symmetric meshes from asymmetric models because it can produce weird results (see discussion of UV mapping and mirror symmetry below).

Note that when separate meshes are created for each Groboto Group, Mirror X can be enabled or disabled for each group separately.

Groboto currently has no dedicated mirror symmetry mode where any modification on one side automatically changes the other side, but there are a number of commands that allow you to create an exactly symmetric model.

A Groboto model consists of primitive objects (and boolean clusters, but these are built from primitive objects too). In a mirror-symmetric model there are two kinds of objects. The first kind can have arbitrary position and orientation in space, but each such object on the right has to have a mirror match on the left. The second kind are not paired, but they themselves have to be mirror-symmetric. Their center has to be on the plane $X = 0$ (Y-Z plane) and from the standard orientation they can only be rotated around the X axis.

To manipulate the paired symmetric objects two commands in the right-click menu are useful: Duplicate and Mirror Object creates a mirror image of an object or a selection of objects. If you moved, rotated or reshaped an object on one side and you want its counterpart on the other side to follow as a mirror image, select the first object then right-click on the second object and choose Set Object to Mirror of Selected.

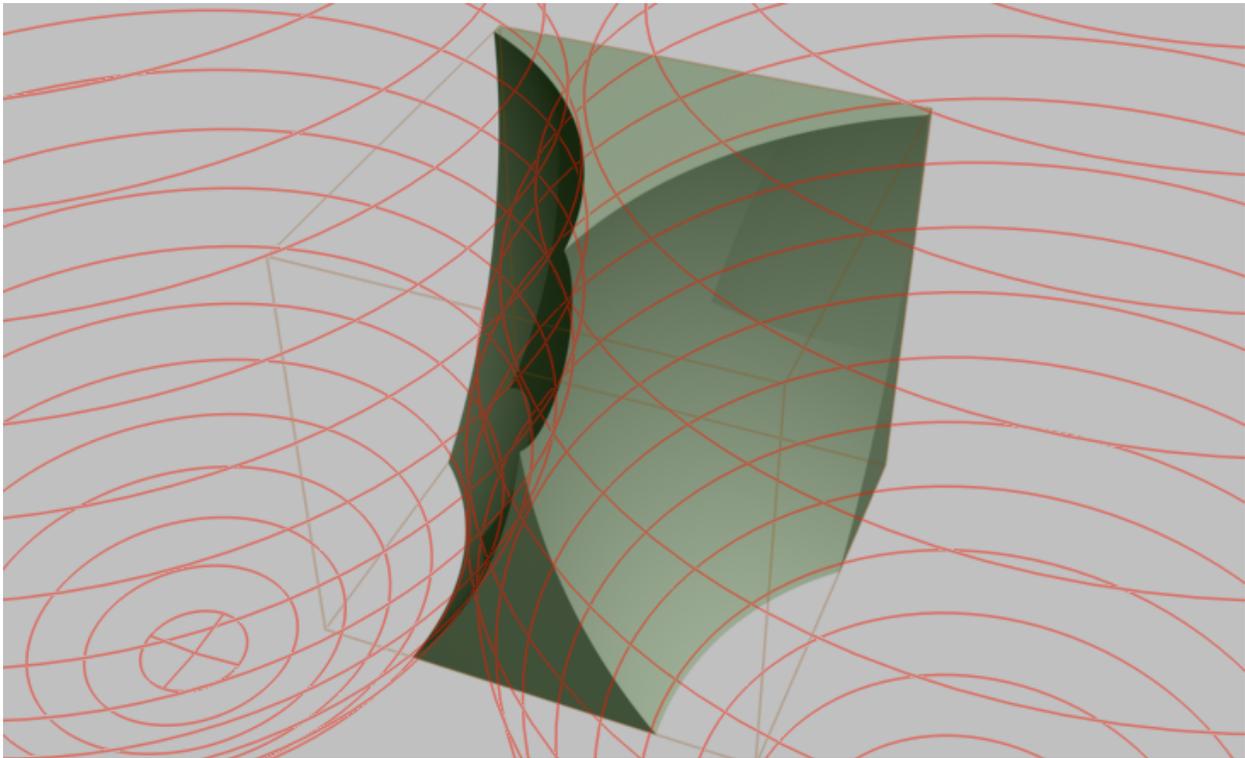
For the second kind, non-paired symmetric objects, you usually start by right-clicking in the Library on the left and choosing Place at World Center. From there you can Move it anywhere in Y and Z and Rotate it around X, and it will remain mirror-symmetric. You can also use Center Object in the right-click menu to center an object only in X, Y or Z. And another command in the right-click menu, Center X and Orient for Symmetry, is explicitly designed to set any object to its X mirror-symmetric state. Its center will move to the Y-Z plane and it will lose any Y or Z rotation, but its X rotation will be preserved. The same commands are available by right-clicking on a primitive in the Boolean Cluster Object List (Window->Modeling Tools, Booleans tab).

Mirror symmetry and UV mapping

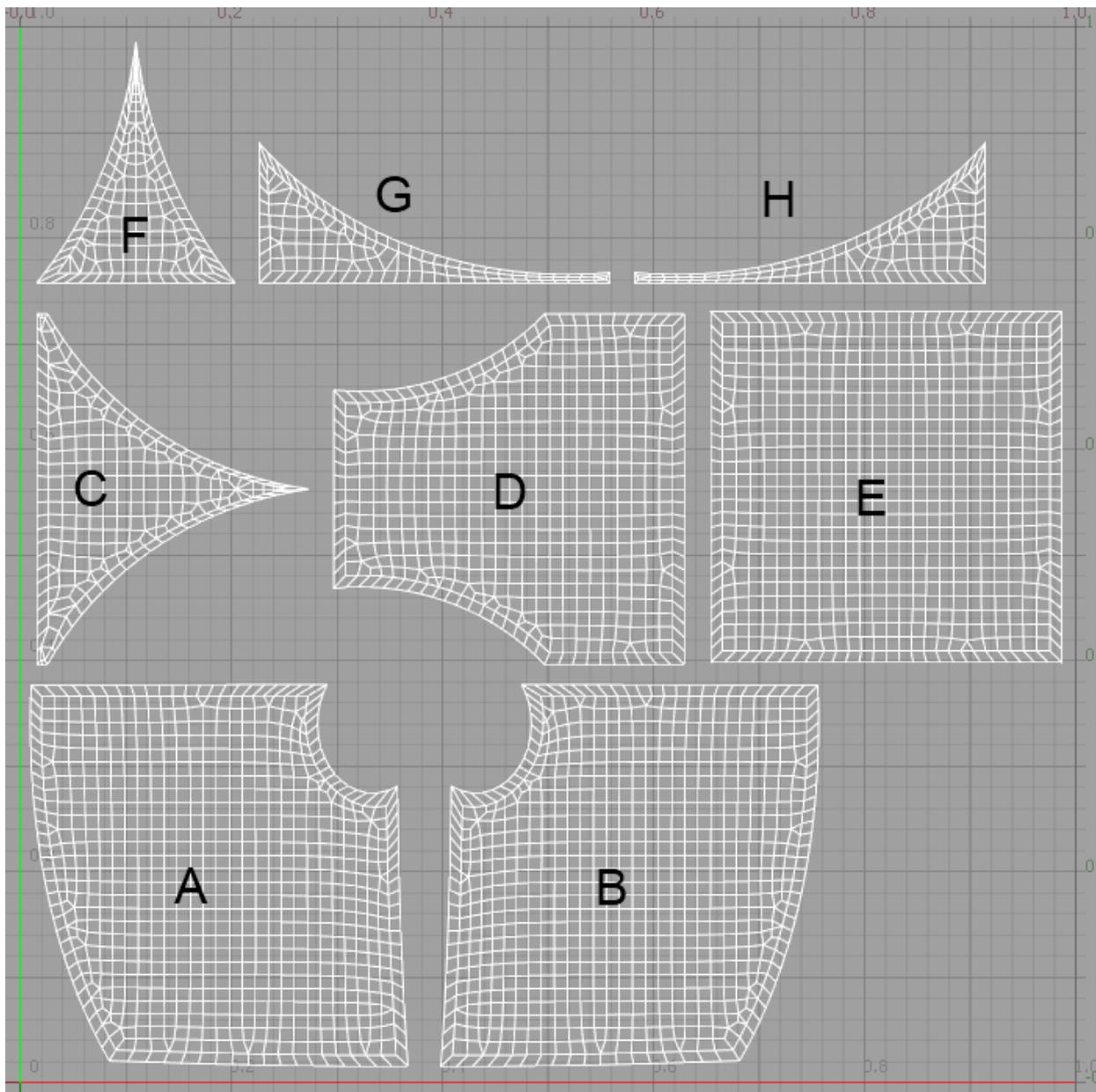
When Mirror X is enabled and Standard UV Map is chosen in the UV Map pop-up of the UniMesh Output dialog, Groboto creates UV mapping in a way that allows you to create an exactly symmetric result when mapping images, displacement, etc. And when you want to generate UV with Mirror X turned on, the model itself has to be exactly mirror-symmetric, or else different UV maps will overlap.

Groboto creates a number of separate UV maps on the surface of the model. On a mirror-symmetric mesh there are two kinds of maps: the first kind can have arbitrary shape, but each such UV map covering a piece of the mesh on the right-hand side has a matching map covering the mirror-image piece on the left. To create mirror-symmetric texture, first of all you need to find in the UV space a mirror match for each UV map. Your polygonal modeling program will help you with that: make a mirror-symmetric polygonal selection on the model in 3D and it will show you the corresponding polygons in the UV window. Groboto tries to place mirror-matched maps close to each other in the UV square.

As you look at the matching maps in UV you'll notice that they are exactly the same except that they are flipped. Some are flipped horizontally, others vertically (this depends on the geometry of the primitive object). To create exact mirror-symmetric image on the surface of your model the image that you use with one of these maps has to be a flipped copy of the image used with the other (horizontally or vertically, like the map itself). Here, for instance, is a vaguely axe-like boolean object created by subtracting two ellipsoids from a box:



After exporting it with Mirror X checked and Standard UV enabled we import the OBJ file, for example, into Modo and open the UV window there:



The two large concave surfaces on the sides of the model are represented in UV by two maps A and B. In this case we have a horizontal flip. To create a symmetric texture you need to slide your two horizontally flipped images into the same positions relative to the two maps. For instance, you would align the bottom left corner of the B map image with the bottom left corner of the B map, and the bottom right corner of the map A image with the bottom right corner of the A map.

If visual alignment is not accurate enough for you, you can use Groboto's precise alignment feature (it's always turned on when Mirror X is checked). The lower left corner in UV of any right-hand map like B (its vertices have $X > 0$ in 3D) is always placed on a grid which is 128x128 in the UV square. For

instance, if you use an image 2048x2048 pixels to cover the UV square, the grid cell is $2048/128 = 16$ pixels. The vertical and horizontal position of the lower left corner is a multiple of 16 pixels. For left-hand maps like A it is the lower right corner that is placed on this grid when the flip is horizontal and the upper left corner when the flip is vertical.

The second kind of UV map that we have on a mirror-symmetric model is not paired, but itself has mirror symmetry in UV. This symmetry can be vertical or horizontal (this again depends on the geometry of the primitive). In the above example, except the paired maps A,B and G,H all other maps are single and symmetric. Maps C and D have vertical mirror symmetry, maps E and F - horizontal. In this case to create symmetric texture you need to use a mirror-symmetric image on these maps, vertical or horizontal, and here the vertical or horizontal middle of the image has to be aligned with the middle of the map.

For these maps the precise alignment feature places on the same 128x128 grid: for maps with horizontal symmetry the lowest point of their (vertical) center line; for maps with vertical symmetry the leftmost point of their (horizontal) center line.