fig2pov documentation

# Additional Povray options

In addition to a straightforward conversion from the MATLAB figure to a ray-traced image, it is possible to specify additional properties and options for the graphics objects. This will enable rendering options that are not natively supported in MATLAB itself, such as adding texture to patch objects, adding shadowing, etc ...

Each graphics object has a property called ‘UserData’, which allows any type of data to be attached to the object. The appearance a graphics object can be changed by adding a structure named 'povray' to the UserData of that object.

For instance, adding the following to a graphics object with handle h:

h.UserData.povray.Texture = ‘T\_Stone1’;

will indicate to fig2pov that the (Povray) texture T\_Stone1 has to be applied to that object.

## Additional Povray options for object of type ‘axes’

Plane

PlaneColor

PlaneTexture

Define

**Zoom**

|  |  |  |
| --- | --- | --- |
| cube = patch('Vertices',[0 0 0; 0 0 1; 0 1 0; 0 1 1; 1 0 0; 1 0 1; 1 1 0; 1 1 1], ...  'Faces',[1 2 4 3; 5 6 8 7; 1 2 6 5; 3 4 8 7; 1 3 7 5; 2 4 8 6], ...  'FaceColor', [1 0 0]);  view(3)  axis equal  h\_axes = gca;  h\_axes.UserData.povray.Zoom = 0.5; | | |
|  |  |  |
| Zoom = 0.5 | Zoom = 1 | Zoom = 1.5 |

Default Value: No Zoom is applied (i.e. Zoom = 1)

## Additional Povray options for object of type ‘patch’

**Texture, InteriorTexture**

|  |  |
| --- | --- |
| cube = patch('Vertices',[0 0 0; 0 0 1; 0 1 0; 0 1 1; 1 0 0; 1 0 1; 1 1 0; 1 1 1], ...  'Faces',[ 3 4 2 1; 5 6 8 7; 7 8 4 3; 5 7 3 1], ...  'FaceColor', [1 0 0]);  view(3)  axis equal  h\_axes = gca;  cube.UserData.povray.Texture = 'T\_Stone1';  cube.UserData.povray. InteriorTexture = 'T\_Stone18'; | |
|  |  |
| No InteriorTexture | InteriorTexture = 'T\_Stone18' |

Note that this depends on the correct orientation of the vertices in each face.

**TextureScale**

|  |  |  |
| --- | --- | --- |
| cube.UserData.povray.TextureScale = 1; | | |
|  |  |  |
| TextureScale = 1 | TextureScale = 5 | TextureScale = 0.2 |

Default Value: TextureScale = 1.

**ShadowLess**

**drawAsSphere, drawAsCylinder**

In MATLAB, patch and surface objects are really collections of polygons, even if the intention is to draw a sphere or a cylinder. The appearance of "roundness" is obtained by using large numbers of faces.

Povray natively supports commands that draw smooth spheres or cylinders (or any object with a rotation axis). If we know that an object is a sphere or has an symmetry axis, we can include this information in UserData.povray. fig2pov will then use the appropriate povray commands, rather than treating the object as a collection of faces.

FaceColor

FaceAlpha

drawEdges

EdgeColor

EdgeTexture

MarkerFaceColor

MarkerTexture

## Additional Povray options for object of type ‘surface’

**Texture**

See ‘Additional Povray options for object of type ‘patch’’

**InteriorTexture**

See ‘Additional Povray options for object of type ‘patch’’

**TextureScale**

See ‘Additional Povray options for object of type ‘patch’’

ShadowLess

**drawAsSphere**

Example:

|  |  |
| --- | --- |
| [x,y,z]=sphere;  h\_sphere = surf(x, y, z, 'FaceColor', 'r')  view(3)  axis equal  h\_sphere.UserData.povray.drawAsSphere = true; | |
|  |  |
| drawAsSphere = false | drawAsSphere = true |

Default Value: drawAsSphere = false

**drawAsCylinder**

|  |  |
| --- | --- |
| t = 0:pi/10:2\*pi;  [x, y, z] = cylinder(0.5+0.1\*cos(t));  h\_cyl = surf(x, y, z, 'FaceColor', 'r')  axis equal  h\_cyl.UserData.povray.drawAsCylinder = true; | |
|  |  |
| drawAsCylinder = false | drawAsCylinder = true |

FaceColor

FaceAlpha

**MeshOn**

|  |  |
| --- | --- |
| [X,Y,Z] = peaks(25);  h\_surf = surf(X,Y,Z/3);  view(3)  axis equal  h\_surf.UserData.povray.MeshOn=true; | |
|  |  |
| MeshOn= false | MeshOn=true |

NOTE: in its current form, ‘MeshOn’ will only work or surfaces defined on an (X, Y)-grid. Not for e,g, spheres or cylinders.

## Additional Povray options for object of type ‘line’

Povray is a ray-tracing program. As such, it has no concept of "lines". Instead, lines are converted into thin cylinders.

Texture

TextureScale

SmoothingOn