

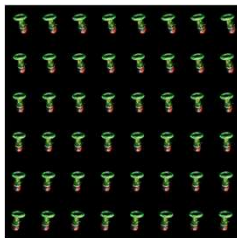
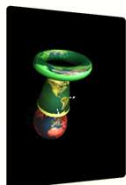
Looking Glass Quilts for a Web Page- Embedded 3D Display, Using both OpenGL and Blender



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License



Oregon State University
Mike Bailey
mjb@cs.oregonstate.edu



Here is How to Generate a Quilt from OpenGL

Here are some pre-defined constants you will need:

```
const int QUILTWIDTH = 3360;
const int QUILTHEIGHT = 3360;
const int QUILTNUMWIDTH = 8;
const int QUILTNUMHEIGHT = 6;
const int QUILTTOTALBLOCKS = QUILTNUMWIDTH * QUILTNUMHEIGHT;
const int BLOCKWIDTH = QUILTWIDTH / QUILTNUMWIDTH; // 420
const int BLOCKHEIGHT = QUILTHEIGHT / QUILTNUMHEIGHT; // 560
const float QUILTZOP = 5.f;
const float QUILTDELTAEPY = 0.10f;
const float QUILT_ASPECT_Y_OVER_X = (float)BLOCKHEIGHT / (float)BLOCKWIDTH; // 4:3 = 1.3333

const char * QUILTFILENAME = "Quilt_qe8x6a0.75.bmp"; // must be in the format "_qe8x6a0.75.*"

unsigned char QuiltArray[QUILTHEIGHT][QUILTWIDTH][3]; // holds the entire quilt
unsigned char BlockArray[BLOCKHEIGHT][BLOCKWIDTH][3]; // holds one block
```

Triggering the File Output

```
void
Keyboard( unsigned char c, int x, int y )
{
    switch( c )
    {
        case 'w':
        case 'W':
            CreateAndWriteQuilt( );
            break;

        . . .

        case 'q':
        case 'Q':
        case ESCAPE:
            DomainMenu( QUIT ); // will not return here
            break; // happy compiler

        default:
            fprintf( stderr, "Don't know what to do with keyboard hit: '%c' (0x%x)\n", c, c );
    }

    // force a call to Display( ):
    glutSetWindow( MainWindow );
    glutPostRedisplay( );
}
```

Hit the 'w' key to trigger the 48 renderings and file-writing

Generating the 48 Individual Images for the Quilt, I

```
void
CreateAndWriteQuilt( )
{
    glutSetWindow( MainWindow );

    float eyex = - QUILTDELTAEPY * (float)(QUILTTOTALBLOCKS) / 2.f;
    for( int y = 0; y < QUILTNUMHEIGHT; y++ )
    {
        for( int x = 0; x < QUILTNUMWIDTH; x++ )
        {
            glDrawBuffer( GL_FRONT );
            glEnable( GL_DEPTH_TEST );
            glShadeModel( GL_FLAT );
            int vx = glutGet( GLUT_WINDOW_WIDTH );
            int vy = glutGet( GLUT_WINDOW_HEIGHT );
            int xl = (vx - BLOCKWIDTH) / 2;
            int yb = (vy - BLOCKHEIGHT) / 2;
            glViewport( xl, yb, BLOCKWIDTH, BLOCKHEIGHT );
            glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );

            glMatrixMode( GL_PROJECTION );
            glLoadIdentity( );
            OffAxisPersp( 70.f, QUILT_ASPECT_Y_OVER_X, 0.1f, 1000.f, QUILTZOP, eyex );

            glMatrixMode( GL_MODELVIEW );
            glLoadIdentity( );
            gluLookAt( 0.f, 0.f, 0.f, 0.f, 0.f, 0.f, 1.f, 0.f );

            DisplayScene( );

            glFlush( );
            glFinish( );

            . . .
        }
    }
}
```

Generating the 48 Individual Images for the Quilt, II

5

```

    . . .

    // done rendering -- upload the pixels:
    glReadBuffer(GL_FRONT);
    glPixelStorei(GL_PACK_ALIGNMENT, 1);
    glReadPixels(xl, yb, BLOCKWIDTH, BLOCKHEIGHT, GL_RGB, GL_UNSIGNED_BYTE, BlockArray);

    // location for this block in the large QuiltArray:
    for( int yb = 0; yb < BLOCKHEIGHT; yb++)
    {
        for( int xb = 0; xb < BLOCKWIDTH; xb++)
        {
            QuiltArray[y*BLOCKHEIGHT+yb][x*BLOCKWIDTH+xb][0] = BlockArray[yb][xb][0];
            QuiltArray[y*BLOCKHEIGHT+yb][x*BLOCKWIDTH+xb][1] = BlockArray[yb][xb][1];
            QuiltArray[y*BLOCKHEIGHT+yb][x*BLOCKWIDTH+xb][2] = BlockArray[yb][xb][2];
        }
    }

    Sleep(250);

    eyex += QUILTDELTAEXE;
    // y
} // x

WriteArray( QUILTFILENAME );
}

```

DisplayScene() – just do the scene drawing, I

6

```

void
DisplayScene()
{
    // rotate the scene:
    glRotatef((GLfloat)Yrot, 0.f, 1.f, 0.f);
    glRotatef((GLfloat)Xrot, 1.f, 0.f, 0.f);

    // uniformly scale the scene:
    if (Scale < MINSCALE)
        Scale = MINSCALE;
    glScalef((GLfloat)Scale, (GLfloat)Scale, (GLfloat)Scale);

    // possibly draw the axes:
    if (AxesOn != 0)
    {
        glColor3fv(4Colors[WhichColor][0]);
        glCallList(AxesList);
    }

    // since we are using glScalef( ), be sure the normals get unitized:
    glEnable(GL_NORMALIZE);
    . . .
}

```

DisplayScene() – just do the scene drawing, II

7

```

    . . .

    glShadeModel(GL_FLAT);
    if (SmoothOn)
    {
        glShadeModel(GL_SMOOTH);
    }

    glDisable(GL_LIGHTING);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE);
    if (LightingOn)
    {
        glEnable(GL_LIGHTING);
        glEnable(GL_LIGHT0);
        glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
    }

    glBindTexture(GL_TEXTURE_2D, Tex);
    glDisable(GL_TEXTURE_2D);
    if (TextureOn)
    {
        glEnable(GL_TEXTURE_2D);
    }

    . . .

```

DisplayScene() – just do the scene drawing, III

8

```

    . . .

    // draw the objects by calling up their display lists:
    glTranslatef(0.f, -2.25f, 0.);
    glColor3f(0.8f, 0.2f, 0.2f);
    SetMaterial(0.8f, 0.2f, 0.2f, 10.f);
    glCallList(SphereDL);

    glTranslatef(0.f, 1.75f, 0.);
    glColor3f(0.8f, 0.8f, 0.2f);
    SetMaterial(0.8f, 0.8f, 0.2f, 8.f);
    glCallList(ConeDL);

    glTranslatef(0.f, 2.5f, 0.);
    glColor3f(0.2f, 0.8f, 0.2f);
    SetMaterial(0.2f, 0.8f, 0.2f, 6.f);
    glCallList(TorusDL);

    glDisable(GL_LIGHTING);
    glDisable(GL_TEXTURE_2D);
    glShadeModel(GL_FLAT);
}

```

Doing the Off-Axis Projection

9

```
void
FrustumZ( float left, float right, float bottom, float top, float znear, float zfar, float sproj )
{
    if( sproj != 0.0 )
    {
        left  *= ( znear/sproj );
        right *= ( znear/sproj );
        bottom *= ( znear/sproj );
        top    *= ( znear/sproj );
    }

    glFrustum( left, right, bottom, top, znear, zfar );
}

void
OffAxisPersp( float fovdeg, float aspect_y_over_x, float znear, float zfar, float z0p, float eyex )
{
    float tanfov = tanf( (float)(M_PI/180.) * fovdeg / 2.f );
    float right = z0p * tanfov;
    float left  = -right;

    float bottom = aspect_y_over_x * left;
    float top    = aspect_y_over_x * right;

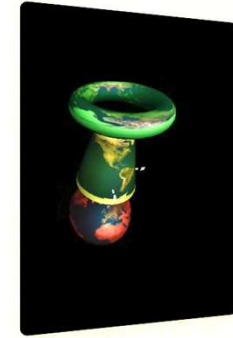
    left  -= eyex;
    right -= eyex;

    FrustumZ( left, right, bottom, top, znear, zfar, z0p );
    glTranslatef( -eyex, 0.0, 0.0 );
}

}
```

Quilt_qs8x6a0.75.bmp

10



How the usual Display() function calls DisplayScene()

11

```
void
Display()
{
    glutSetWindow( MainWindow );
    glDrawBuffer( GL_BACK );
    glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );
    glEnable( GL_DEPTH_TEST );
    glShadeModel( GL_FLAT );

    // set the viewport to a square centered in the window:
    GLint vx = glutGet( GLUT_WINDOW_WIDTH );
    GLint vy = glutGet( GLUT_WINDOW_HEIGHT );
    GLint v = vx < vy ? vx : vy; // minimum dimension
    GLint xl = ( vx - v ) / 2;
    GLint yb = ( vy - v ) / 2;
    glViewport( xl, yb, v, v );

    // set the viewing volume:
    glMatrixMode( GL_PROJECTION );
    glLoadIdentity();
    if( WhichProjection == ORTHO )
        glOrtho( -2.f, 2.f, -2.f, 2.f, 0.1f, 1000.f );
    else
        gluPerspective( 70.f, 1.f, 0.1f, 1000.f );

    // place the objects into the scene:
    glMatrixMode( GL_MODELVIEW );
    glLoadIdentity();
    gluLookAt( 0.f, 0.f, 7.f, 0.f, 0.f, 0.f, 0.f, 1.f, 0.f );

    // draw just the scene:
    DisplayScene();

    glutSwapBuffers();
    glFlush();
}
```

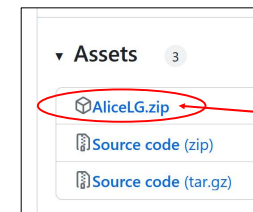
Here is How to Generate a Quilt from Blender

12

You can also generate these quilts automatically from Blender.

The first thing you need to do is retrieve a Blender Add-on.

The Add-on's name is **AliceLG.zip** and it can be found at <https://github.com/regcs/AliceLG/releases>

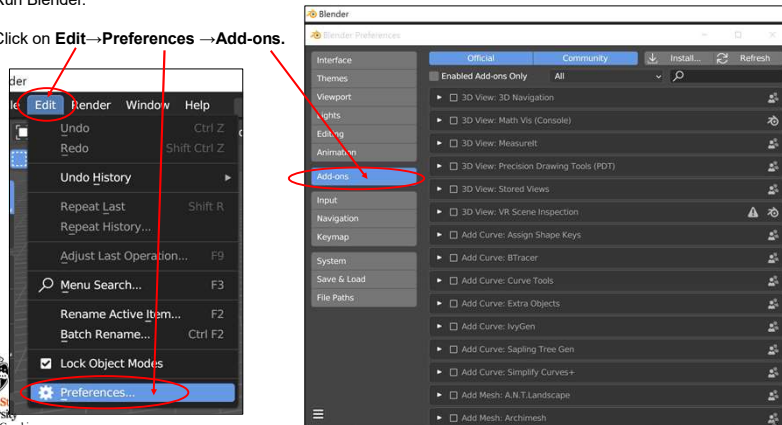


Click here and save this file anywhere.
Do not un-zip it!

Run Blender.

Getting the AliceLG.zip Blender Add-on, I

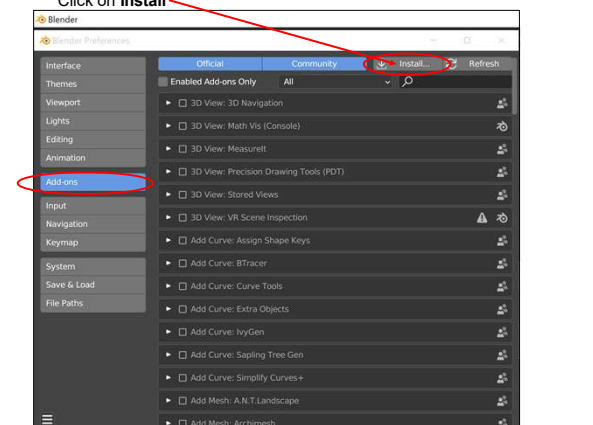
Click on **Edit** → **Preferences** → **Add-ons**.



mpb - September 18, 2023

Getting the AliceLG.zip Blender Add-on, II

Click on **Install**.



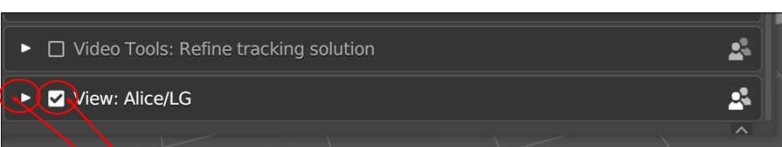
Then navigate to where you installed **AliceLG.zip** and select it.

Don't un-zip it!

mpb - September 18, 2023

Getting the AliceLG.zip Blender Add-on, III

Scroll down the list of Add-ons until you see **View: Alice/LG**



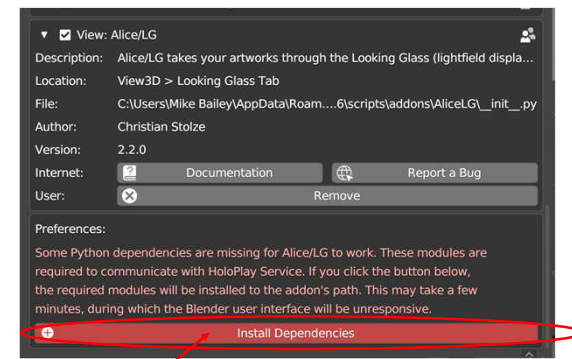
Click the **checkbox** to enable it.

Then click the **arrow** and scroll down so you can see all the information.

mpb - September 18, 2023

Getting the AliceLG.zip Blender Add-on, IV

Scroll down so you can see all the information.



Then click on **Install Dependencies**. This might take a while. Be patient.

mpb - September 18, 2023

Getting the AliceLG.zip Blender Add-on, V

17

When it's done, you will see this:

Preferences:

All required Python modules were installed.

⚠ Please restart Blender to activate the changes!

Now exit Blender and start it up again.



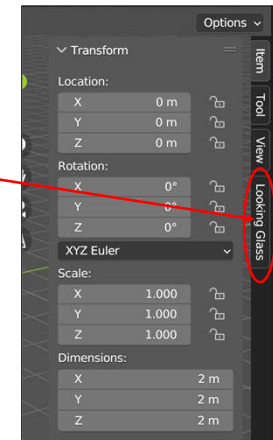
mjb - September 18, 2023

Finding the Looking Glass Tab in Blender

18

Hit the 'n' key and you will see the usual Transform panel, but something new has been added.

Click on the **Looking Glass** tab.



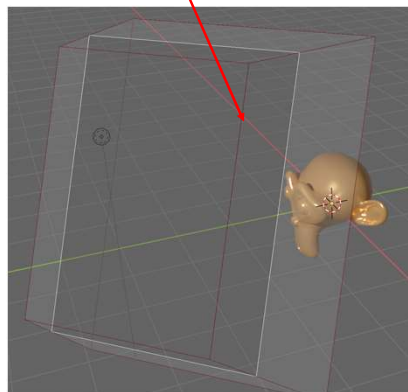
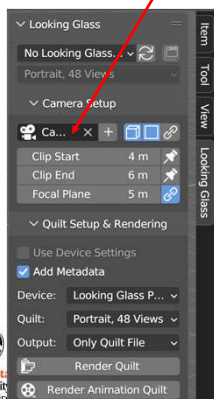
mjb - September 18, 2023

Using the Looking Glass Tab, I

19

Click here and select **Camera**

This will bring up a viewing volume, like this:



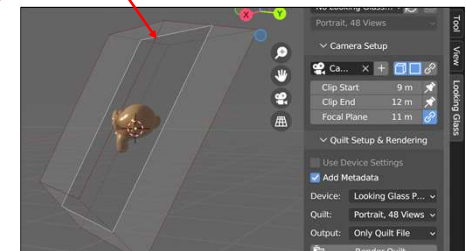
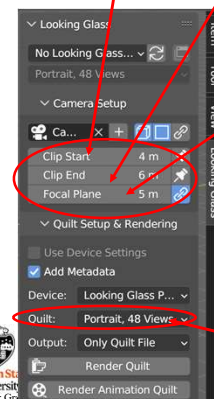
mjb - September 18, 2023

Using the Looking Glass Tab, II

20

Now adjust the **Clip Start** and **Clip End** to completely surround your scene.

Then adjust the **Focal Plane** to be roughly down the middle of the scene, like this:



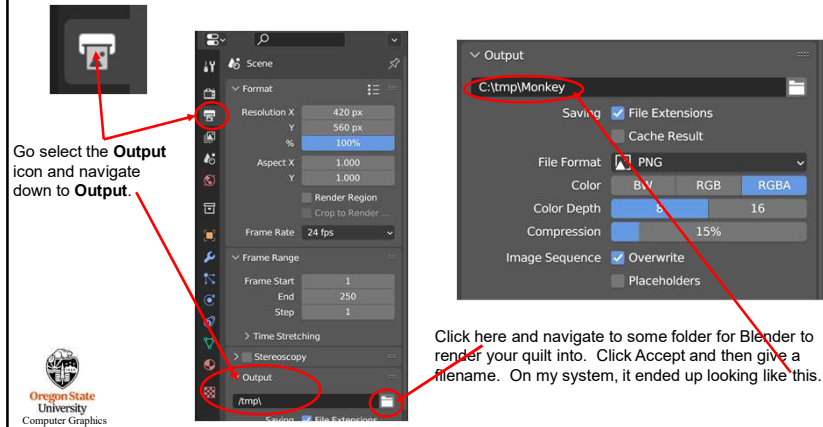
Select **Portrait, 48 Views**



mjb - September 18, 2023

Using the Looking Glass Tab, III

21



Go select the **Output** icon and navigate down to **Output**.

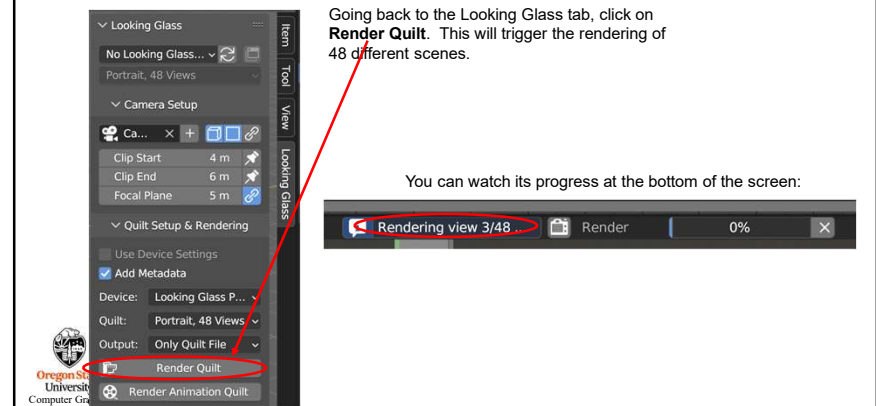
Click here and navigate to some folder for Blender to render your quilt into. Click Accept and then give a filename. On my system, it ended up looking like this.

Oregon State University
Computer Graphics

mpb - September 18, 2023

Using the Looking Glass Tab, IV

22



Going back to the Looking Glass tab, click on **Render Quilt**. This will trigger the rendering of 48 different scenes.

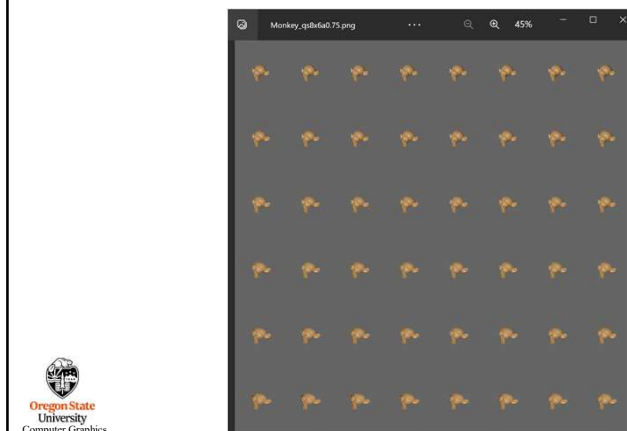
You can watch its progress at the bottom of the screen:

Oregon State University
Computer Graphics

mpb - September 18, 2023

Monkey_qs8x6a0.75.png

23



How to Go From a Quilt to a Live "Hologram", I

24

Navigate to <https://blocks.glass>

If you don't have an account, **sign up** to get one. If you are under 18 years old, be sure to have your parents help you do this!

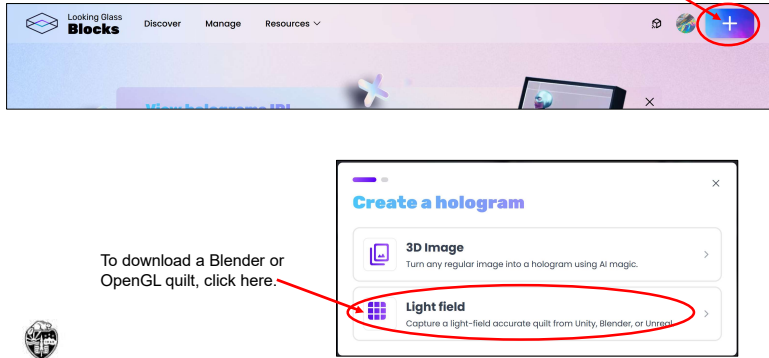
If you do have an account, **Log in** to it.



How to Go From a Quilt to a Live "Hologram", II

25

Click on the big Plus Sign



To download a Blender or OpenGL quilt, click here.

Looking Glass Blocks Discover Manage Resources

Create a hologram

- 3D Image**
Turn any regular image into a hologram using AI magic.
- Light field**
Capture a light-field accurate quilt from Unity, Blender, or Unreal

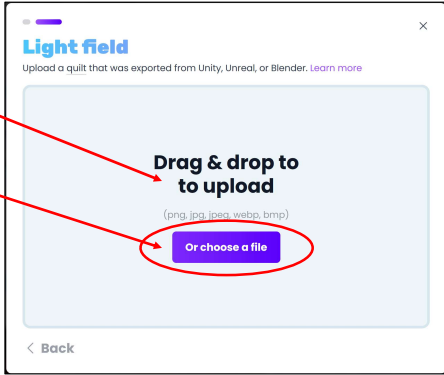
Oregon State University Computer Graphics

mjb - September 18, 2023

How to Go From a Quilt to a Live "Hologram", III

26

Drag-and-Drop your quilt file here, or click here and navigate



Light field
Upload a quilt that was exported from Unity, Unreal, or Blender. [Learn more](#)

Drag & drop to upload
(png, jpg, jpeg, webp, bmp)

Or choose a file

< Back

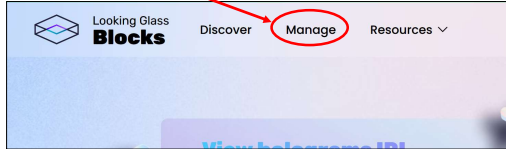
Oregon State University Computer Graphics

mjb - September 18, 2023

How to Go From a Quilt to a Live "Hologram", IV

27

Click on **Manage**, then click on the name of the file you just uploaded



Wiggle the mouse left and right to see your hologram move in 3D

Looking Glass Blocks Discover Manage Resources

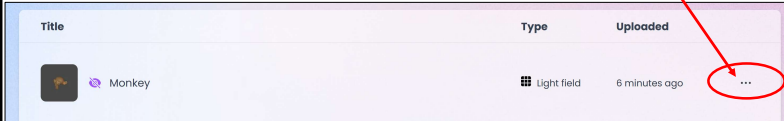
Oregon State University Computer Graphics

mjb - September 18, 2023

How to Go From a Quilt to a Live "Hologram", V

28

Click on **Manage** again then click here



Title	Type	Uploaded
Monkey	Light field	6 minutes ago

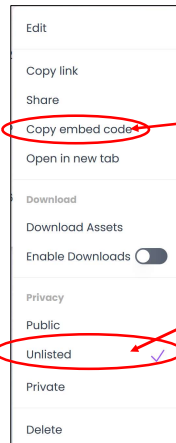
Oregon State University Computer Graphics

mjb - September 18, 2023

How to Go From a Quilt to a Live "Hologram", VI

29

That will bring up a bunch of things you can do with your new "hologram"

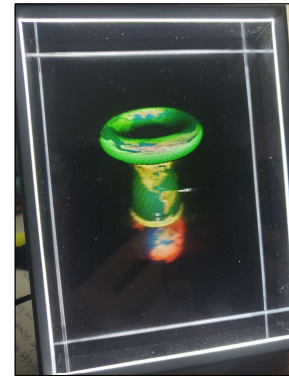


Clicking here will copy the embedded HTML code, to get at this hologram from the web, to the clipboard. Paste that into one of your HTML web pages!

Be sure this permission is set to **Unlisted**.

"Casting" a Quilt to a 3D Image on a Looking Glass Portrait Display

30



<https://lookingglassfactory.com/looking-glass-portrait>