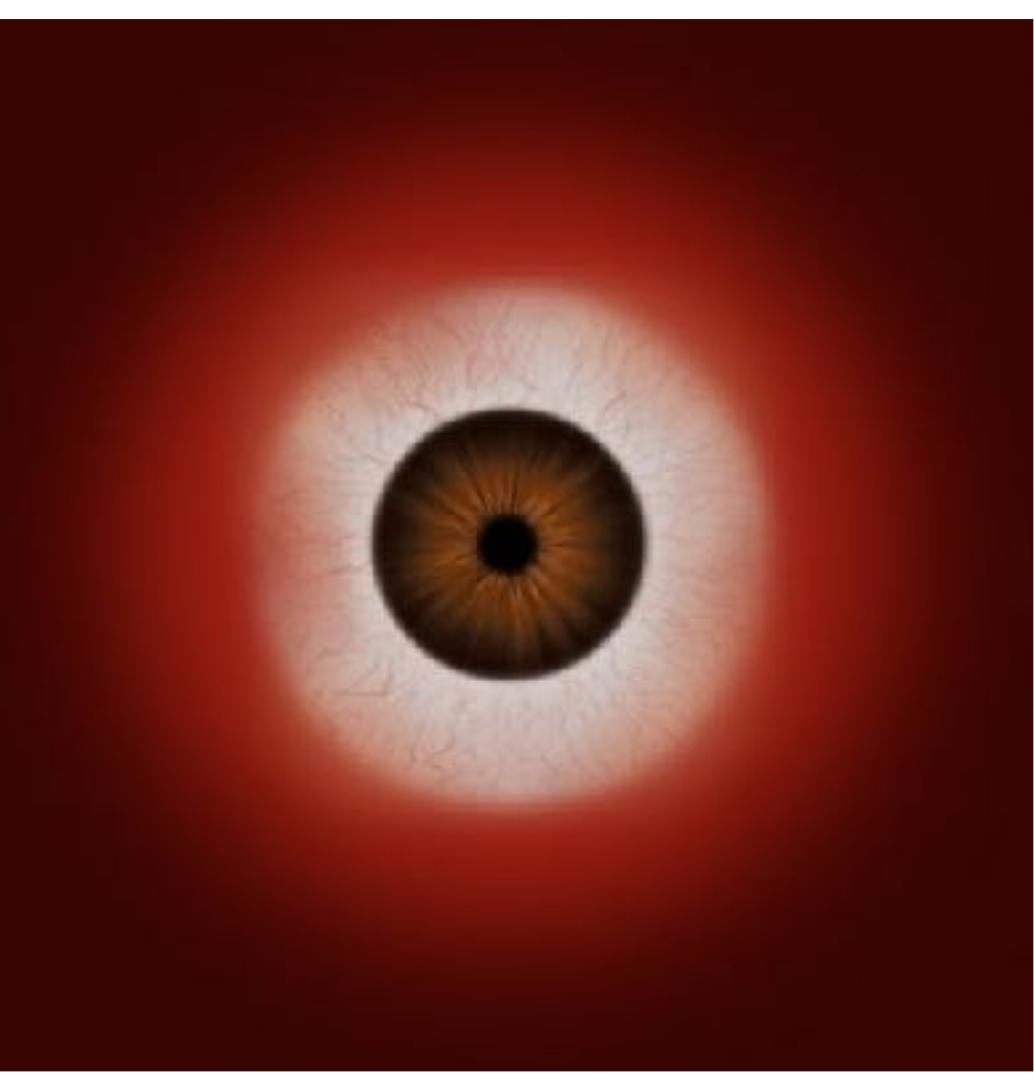
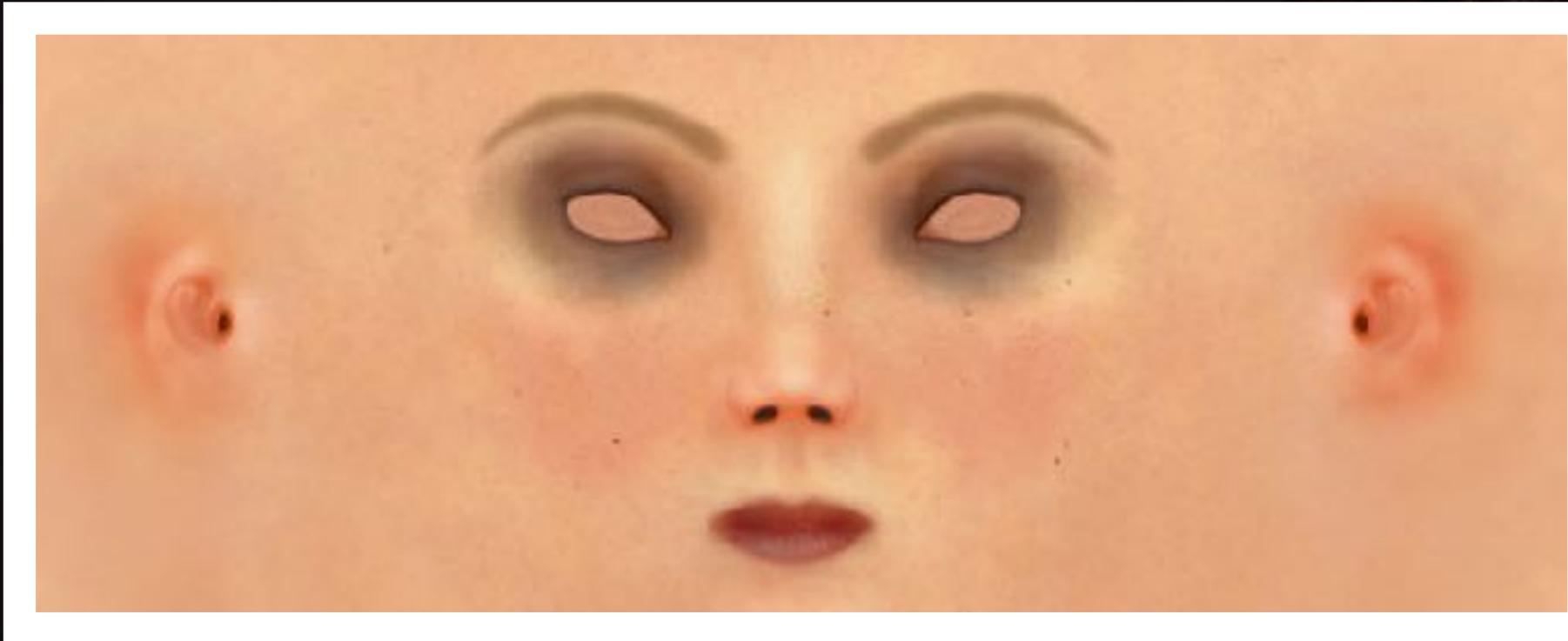


Texture Mapping

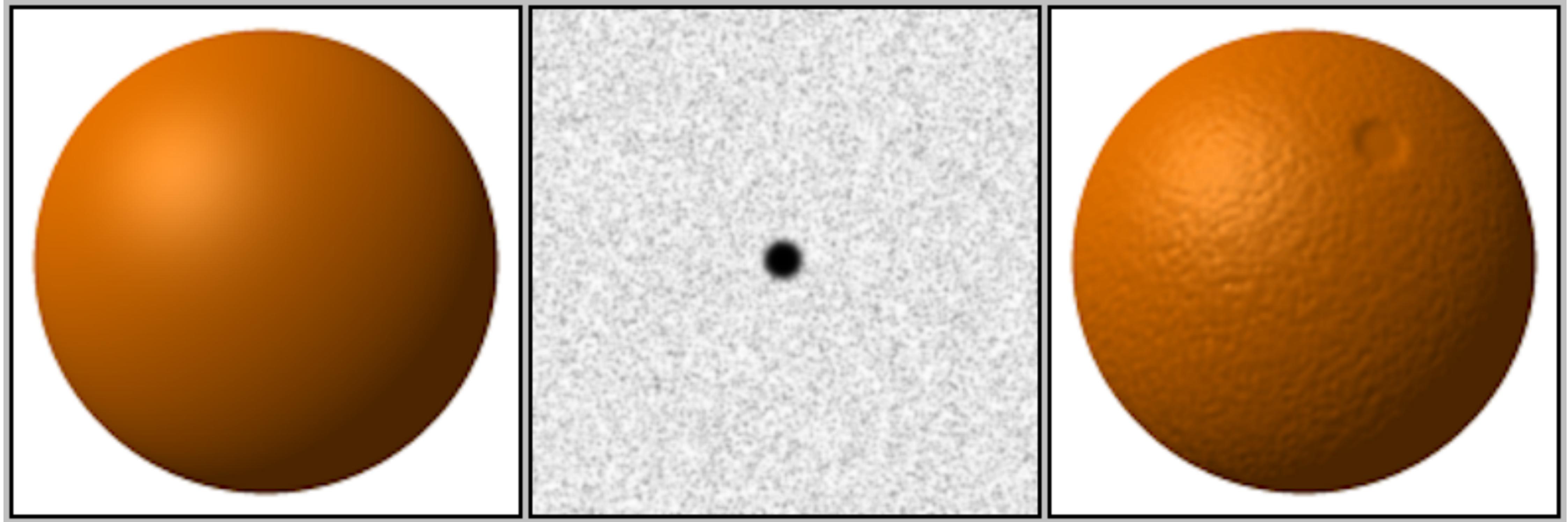




Sintel
Blender Open Movie

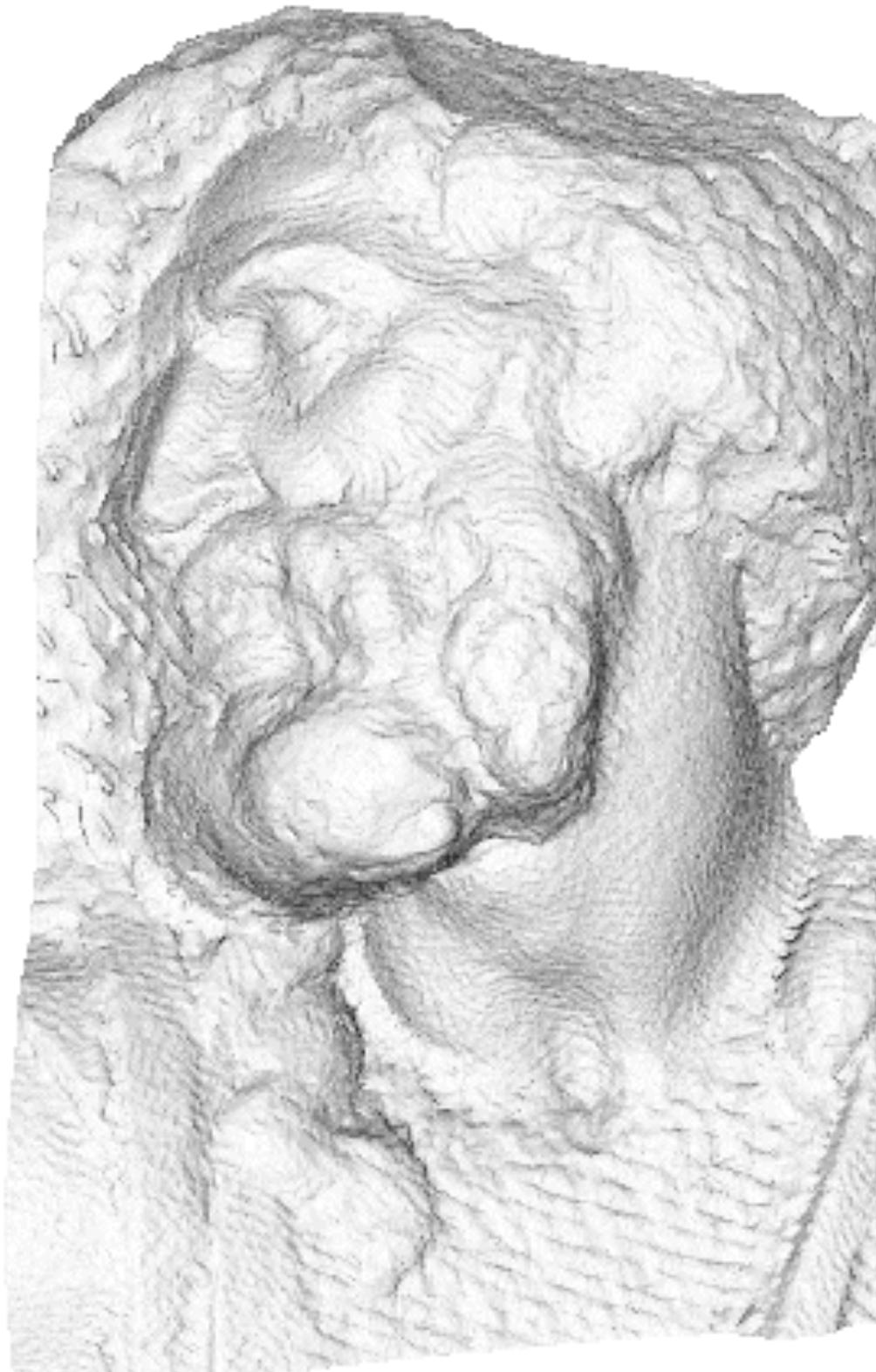
Bump Mapping

Instead of encoding colors in a texture, you encode normals!

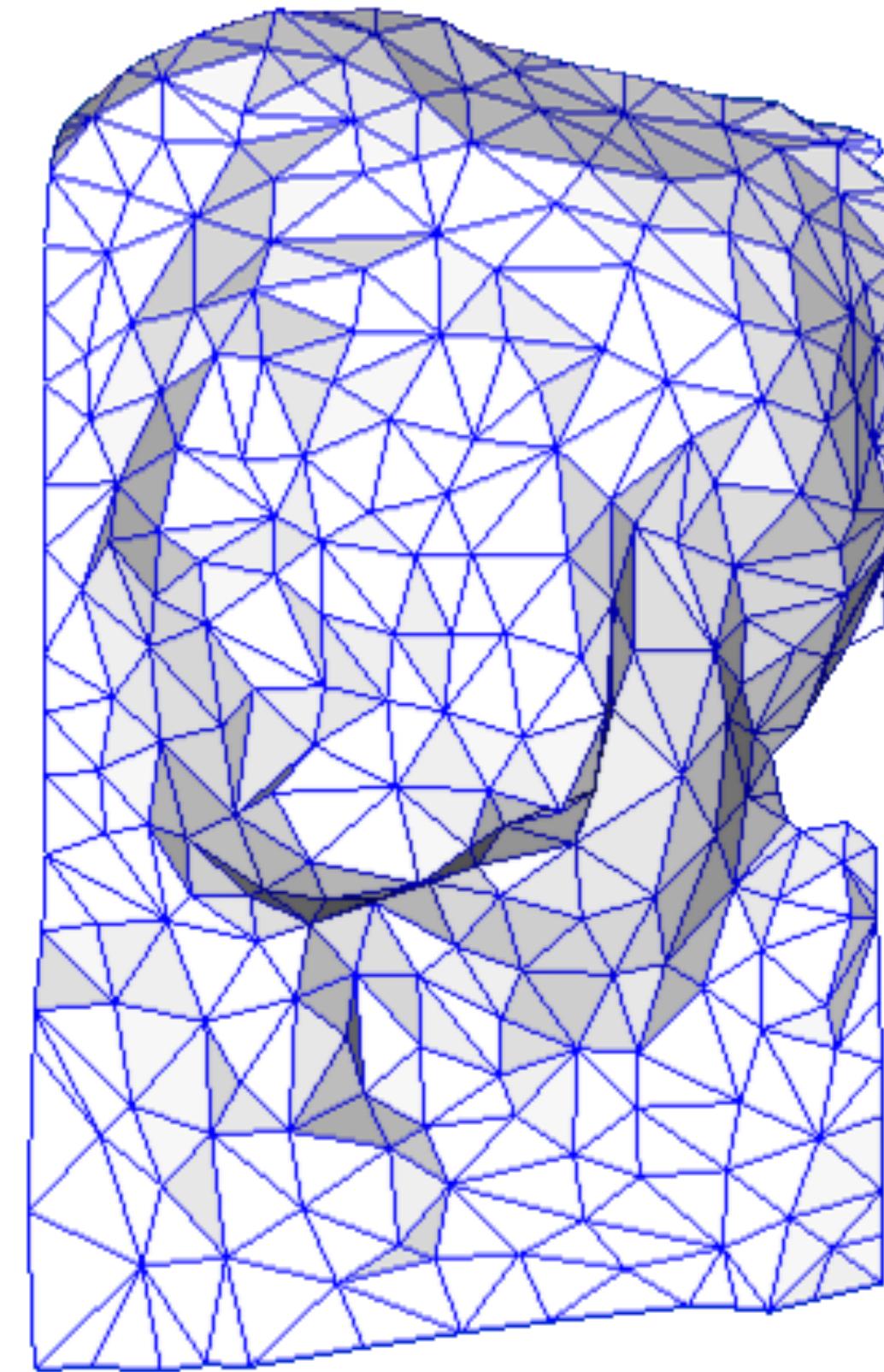


By Bump-map-demo-smooth.png, Orange-bumpmap.png and Bump-map-demo-bumpy.png: Original uploader was Brion VIBBER at en.wikipediaLater version(s) were uploaded by McLoaf at en.wikipedia.derived work: GDallimore (talk) - Bump-map-demo-smooth.png, Orange-bumpmap.png and Bump-map-demo-bumpy.png, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=11747953>

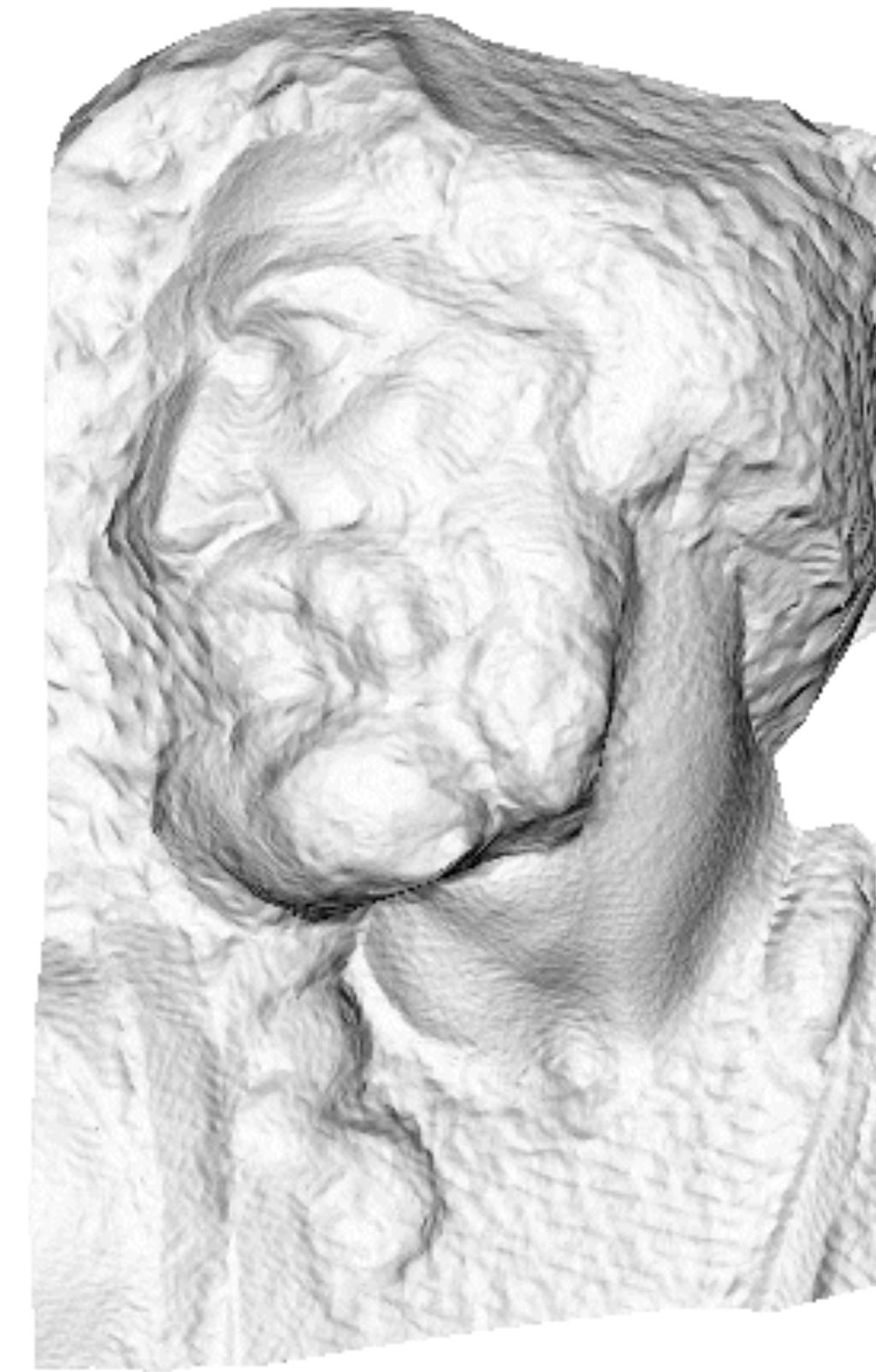
Normal/Bump Mapping



original mesh
4M triangles



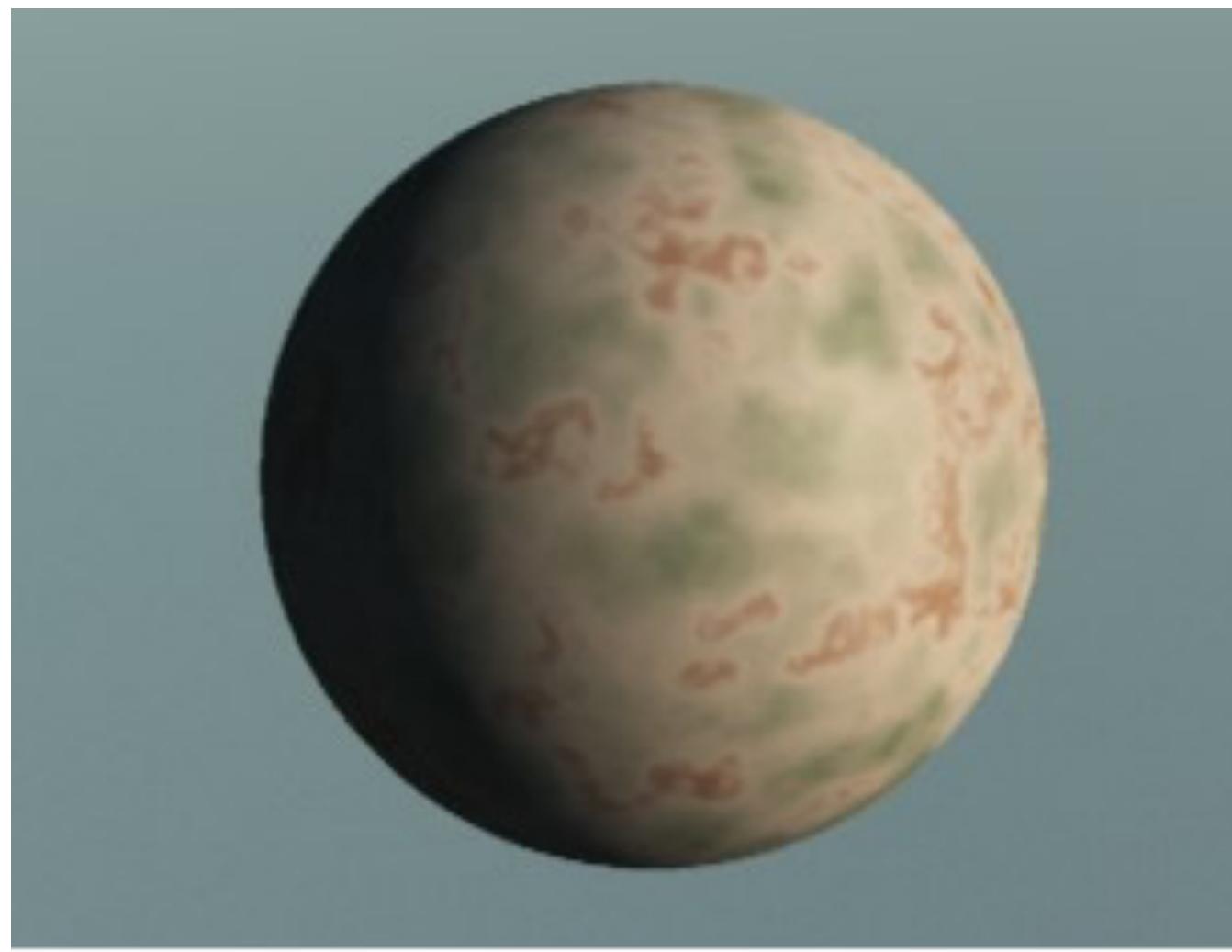
simplified mesh
500 triangles



simplified mesh
and normal mapping
500 triangles

Displacement Mapping

Instead of normals, you encode a displacement.



Original



Bump Mapping



Displacement Mapping

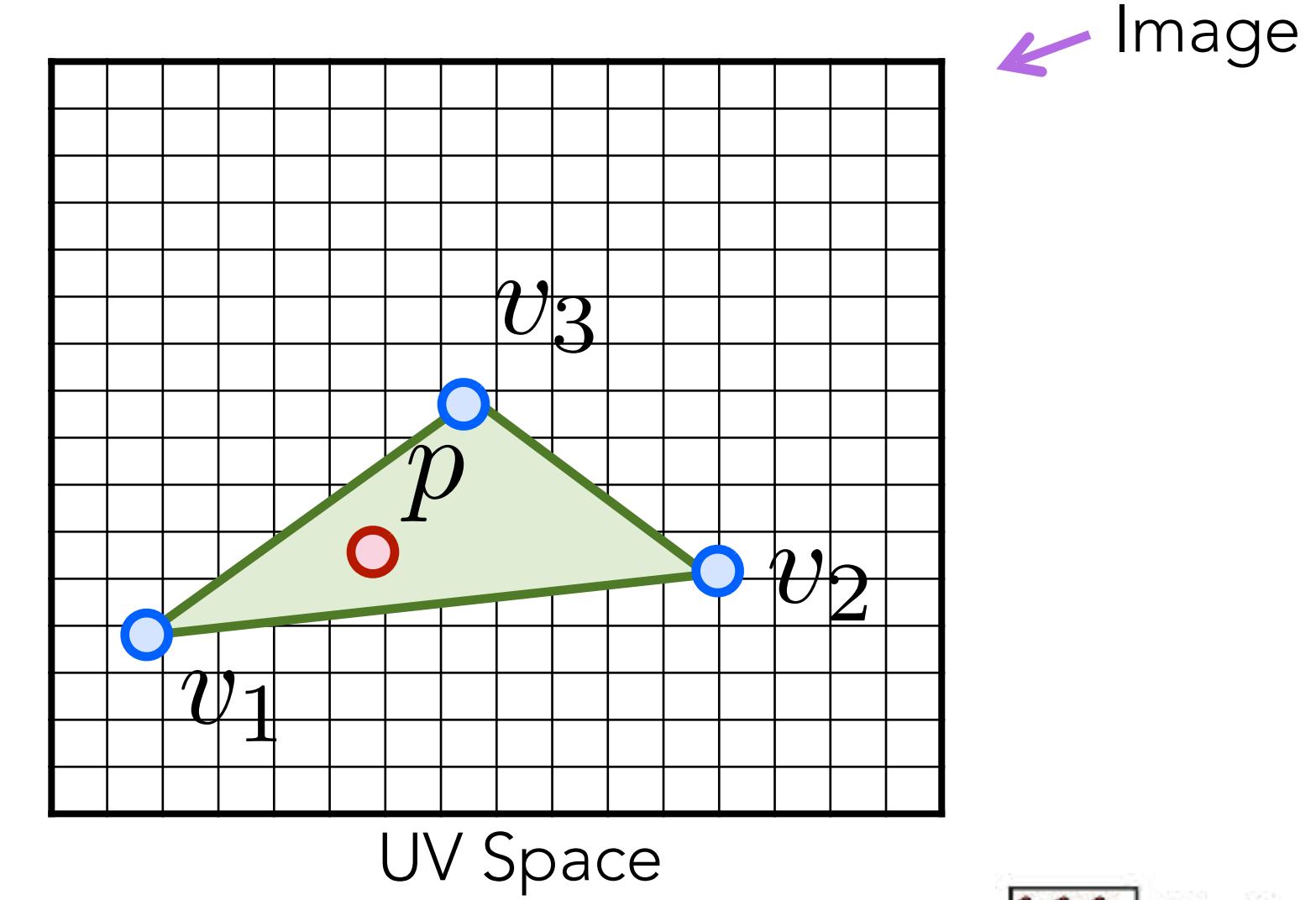
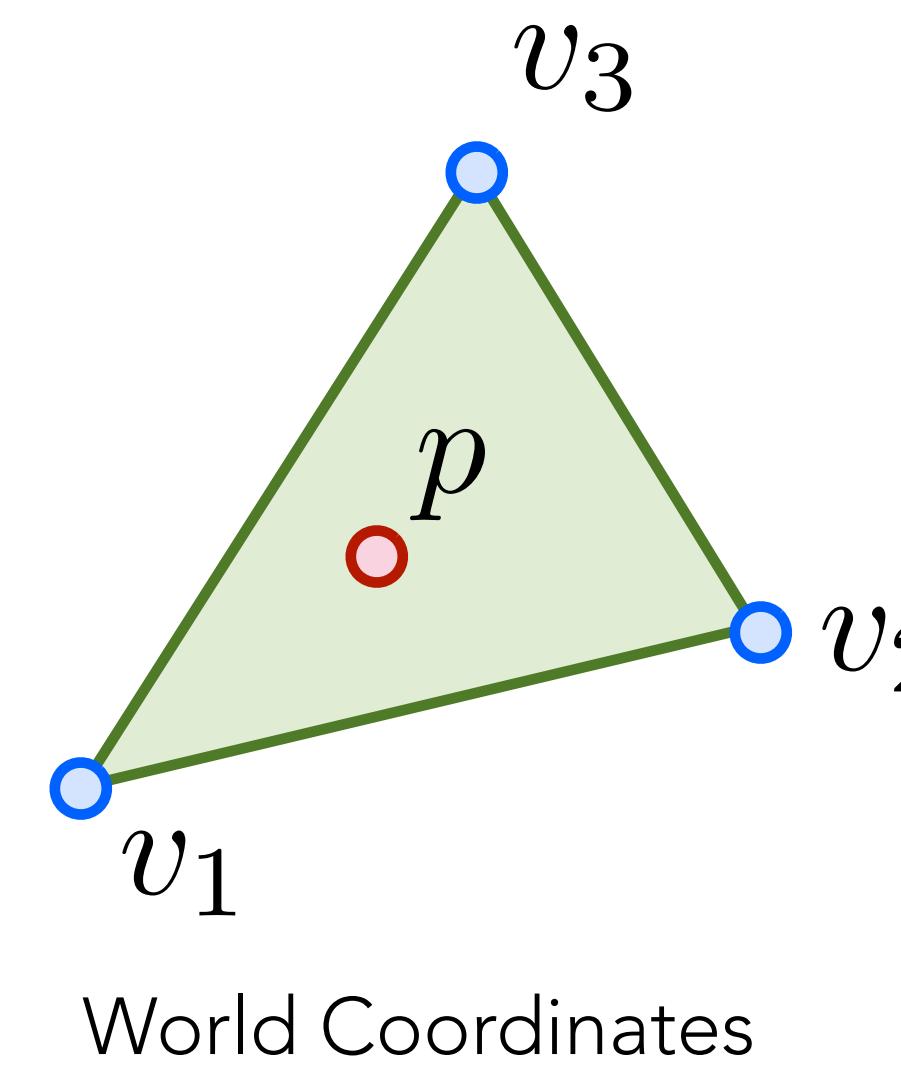
Image courtesy of: <http://www.chromesphere.com/Tutorials/Vue6/Optics-Basic.html>

Texture Mapping

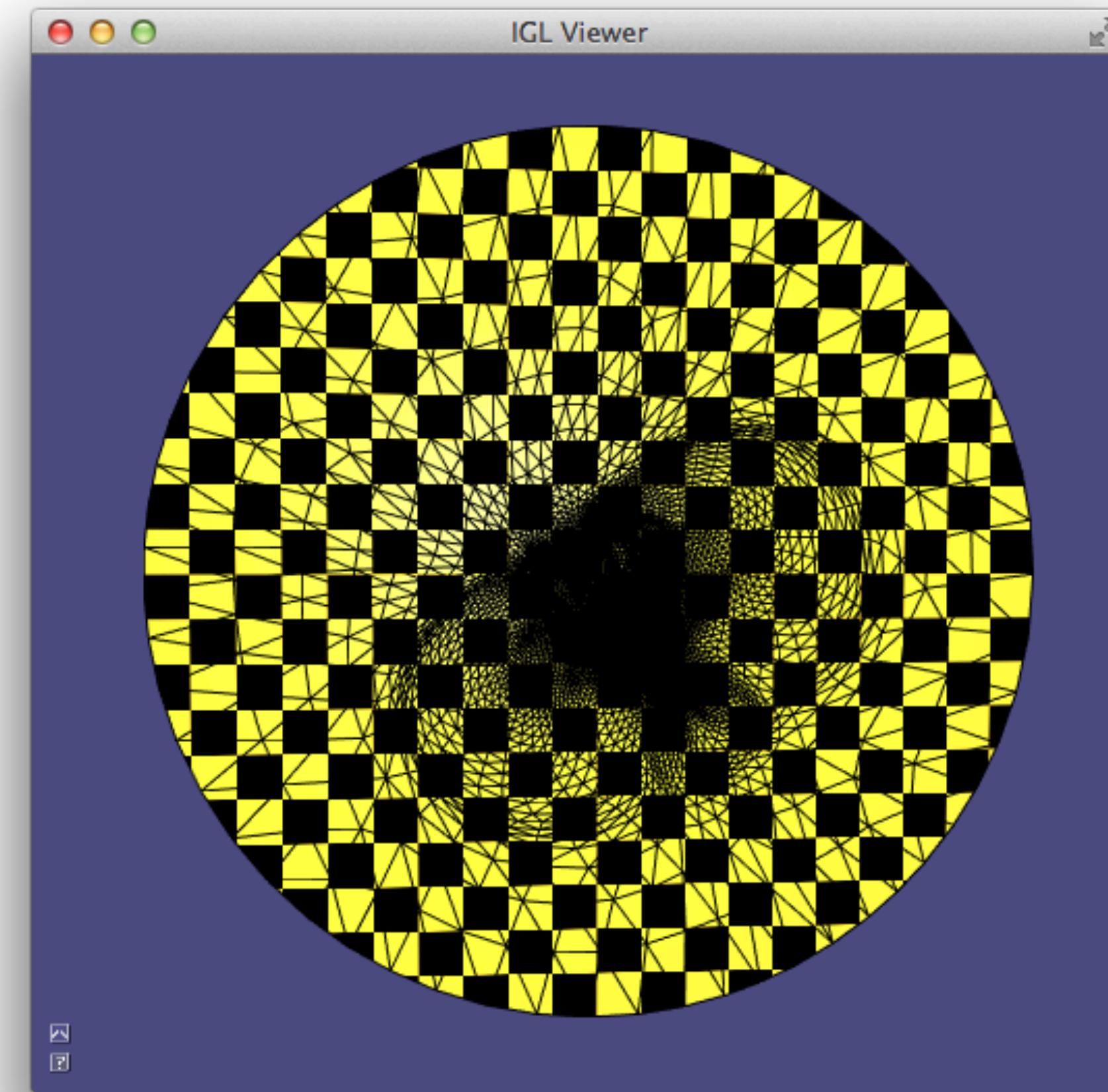
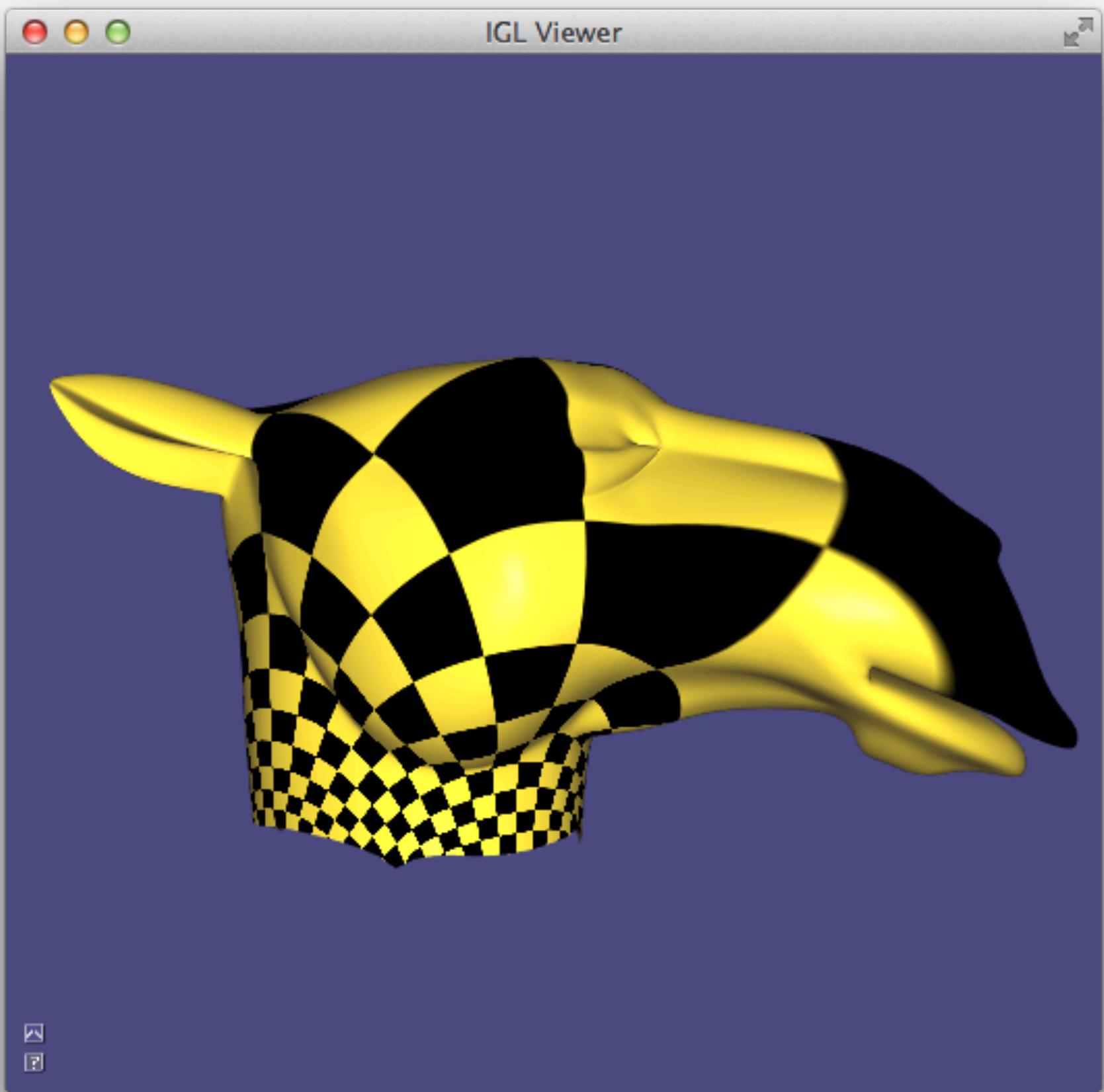
- The idea is the same. Instead of encoding values at vertices of triangles, you encode them in images.
- You gain all the advantages of images (easy to store, compress, interpolate, sample)
- You can encode any property that you want, the most common are:
 - Colors (Texture Mapping)
 - Normals (Bump Mapping)
 - Displacements (Displacement Mapping)

What do you need?

- One additional per-vertex property, the UV coordinates
- An image uploaded to the GPU memory (2D texture)
- The UV coordinates are interpolated inside each triangle, and used to find the corresponding value in the texture
- The texture value is interpolated before it is used in the shader



Checkerboards are great to visualize a UV map



"Seams" are needed for complex objects



Image from Vallet and Levy, techreport INRIA

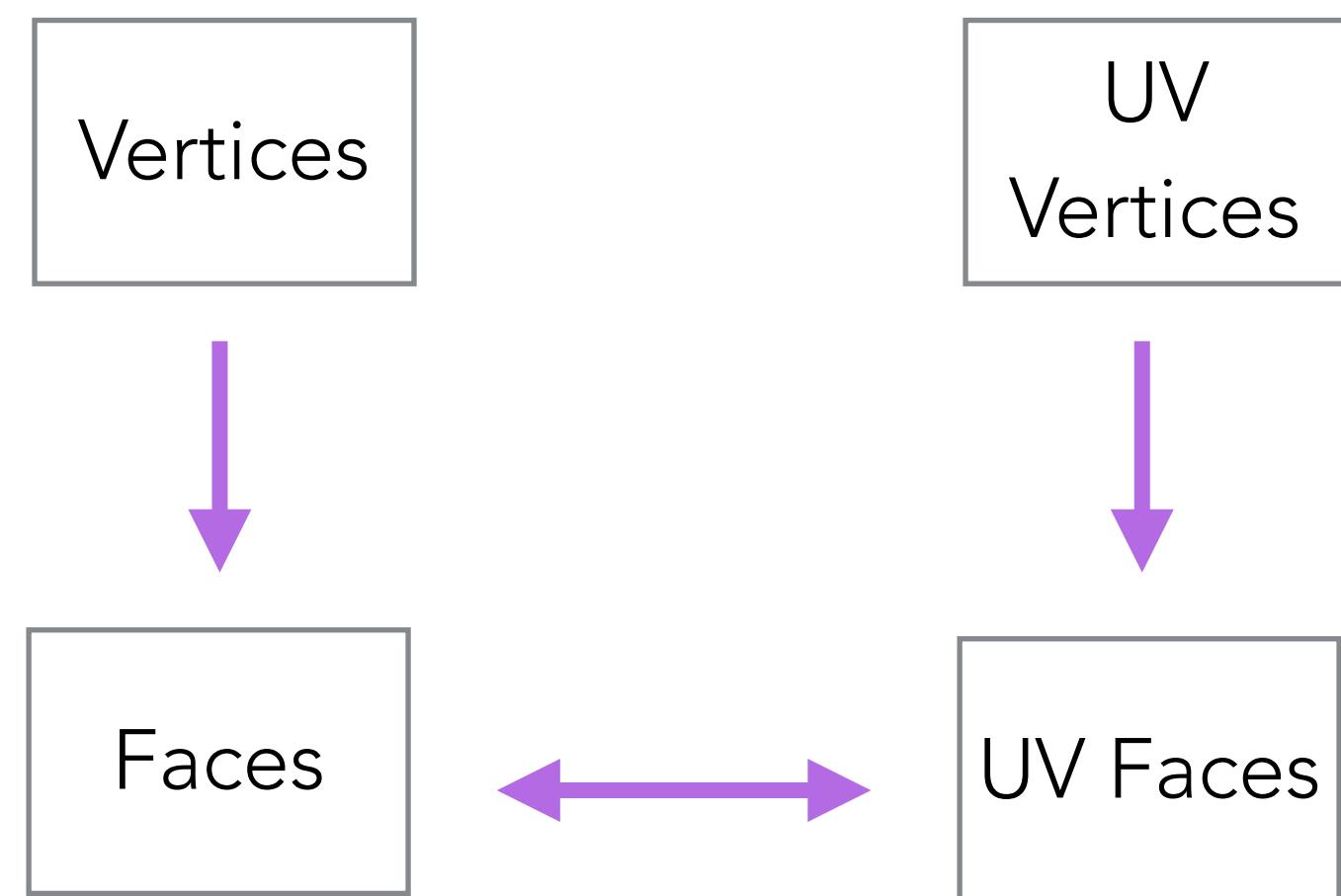


**University
of Victoria**

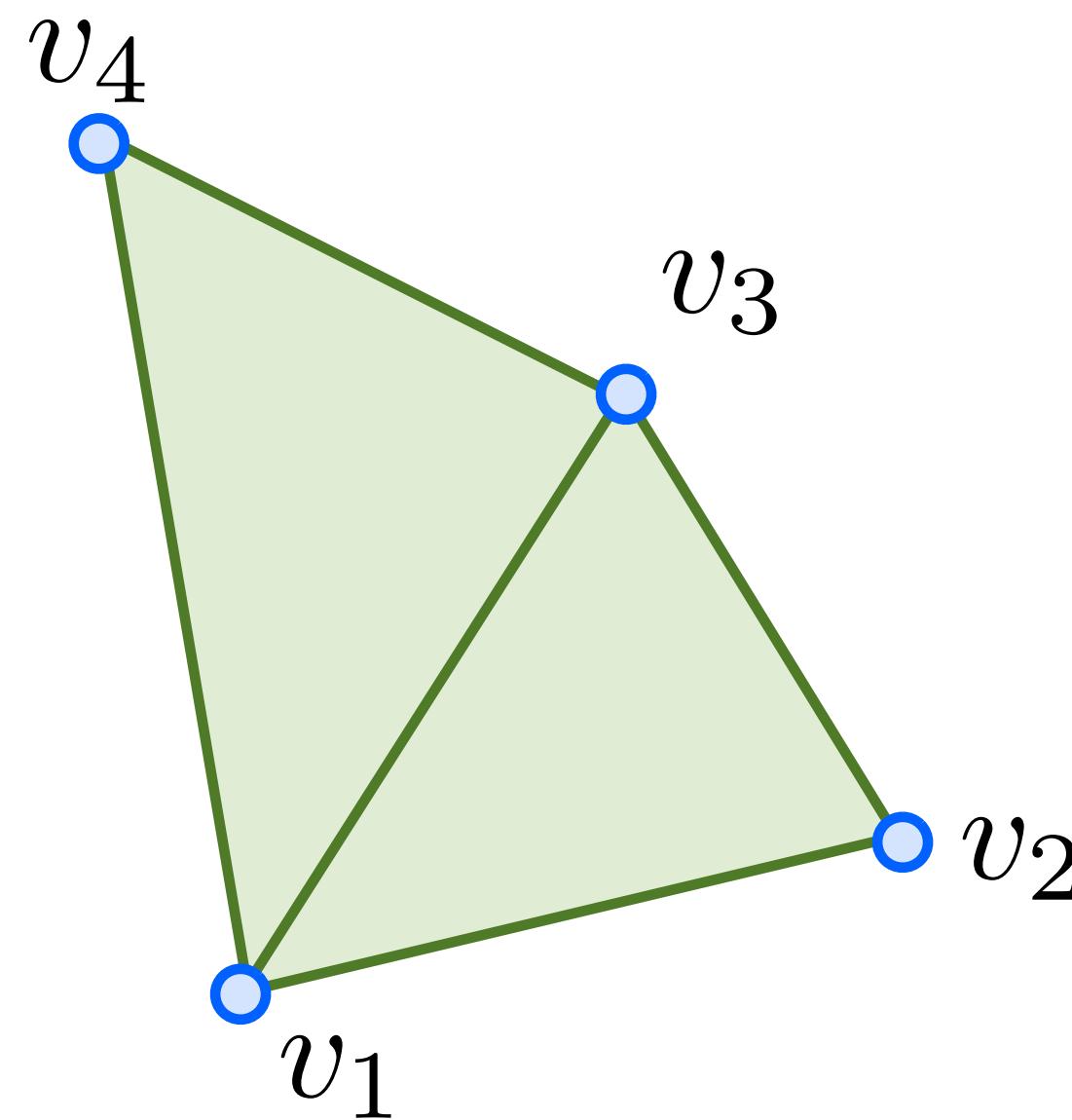
Computer Science

How are UV maps encoded?

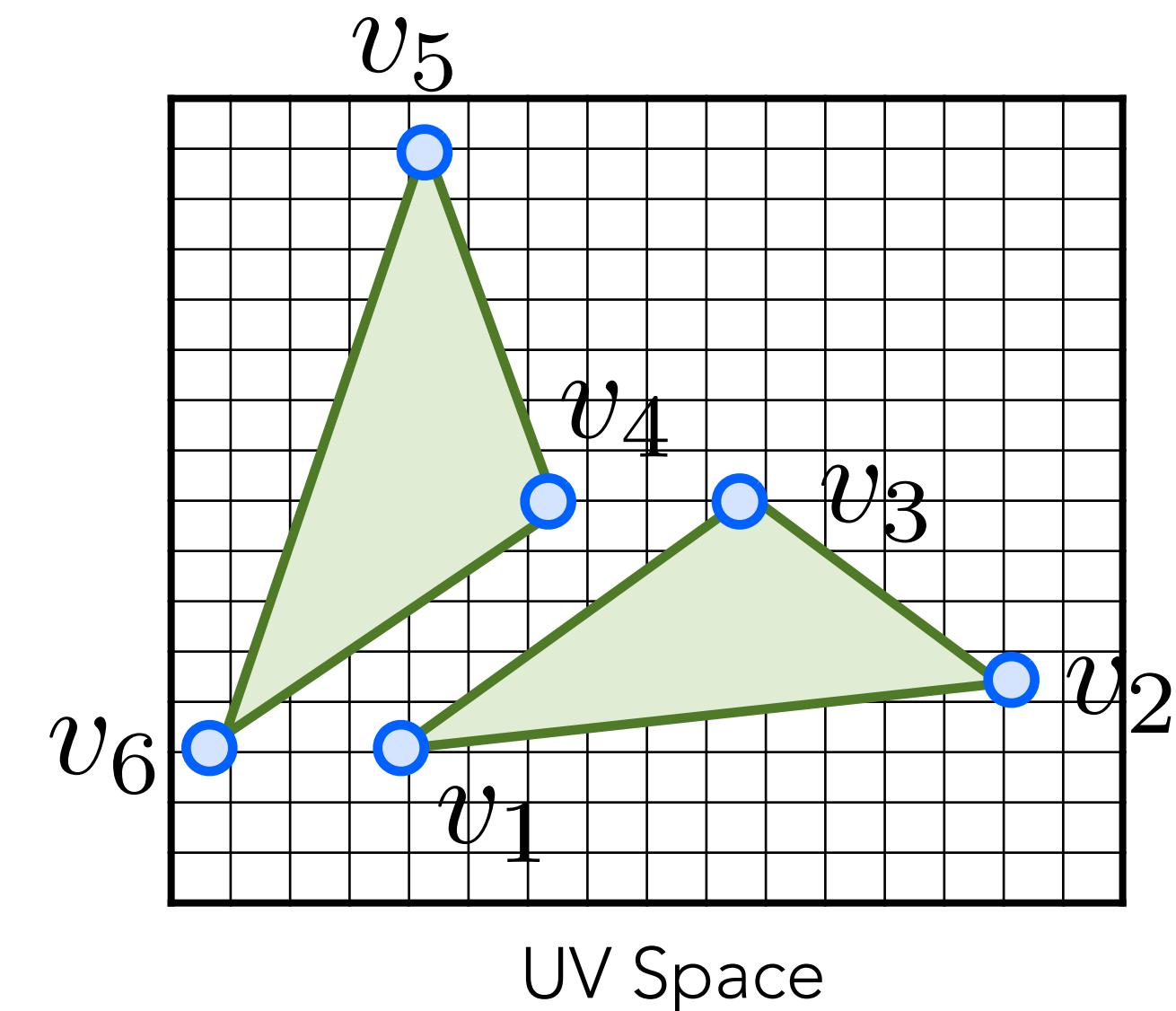
- 2 versions of the mesh are stored, one for the triangles in 3D and one for those in 2D
- The faces of the 2 meshes are in one-to-one correspondence
- OpenGL does not support this you need to duplicate all the vertices on the seams and pass one single mesh. An easy (and inefficient) way to do this is by duplicating all vertices and not using an element buffer.



A minimal example



World Coordinates



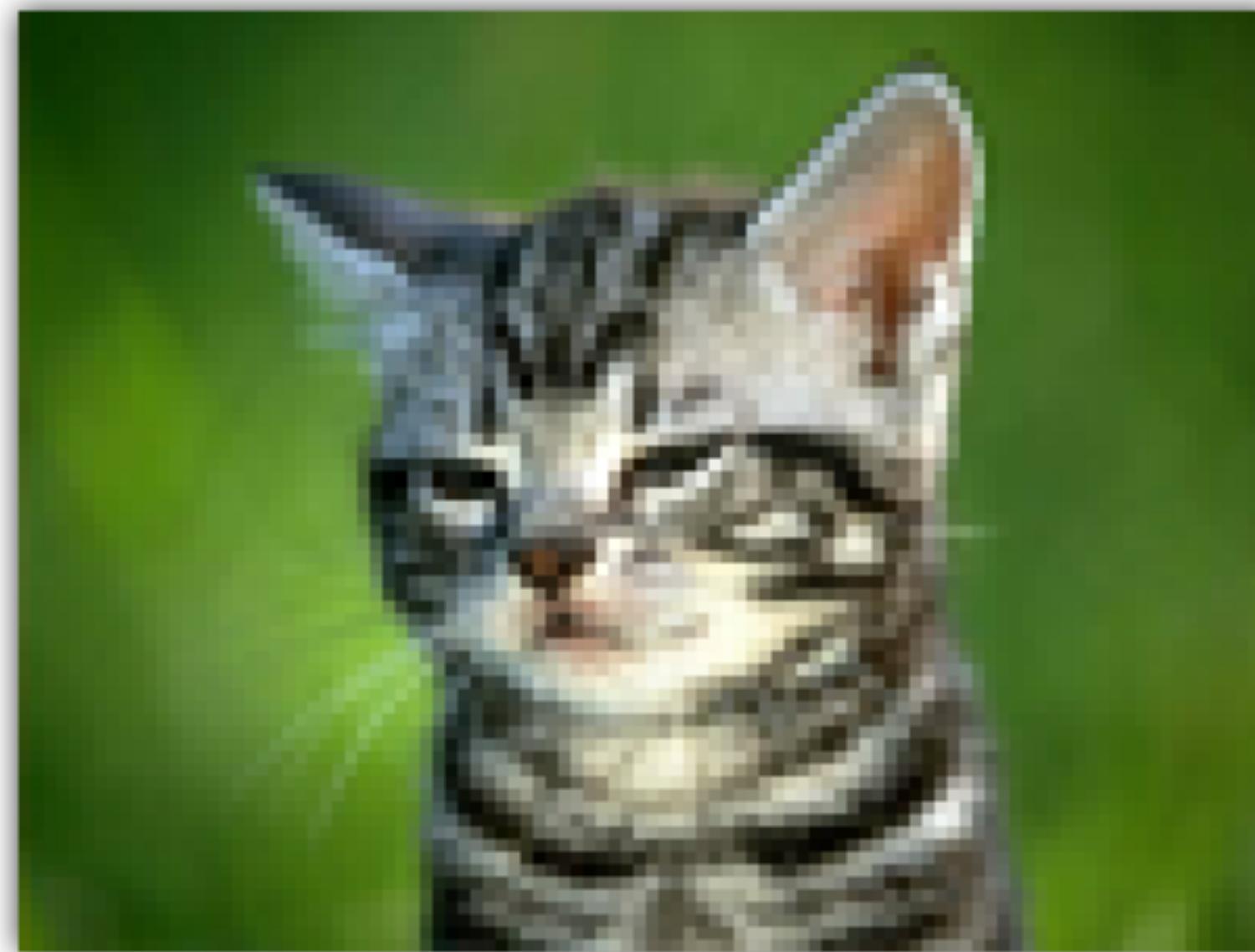
UV Space

Faces
$v_1 \ v_2 \ v_3$
$v_1 \ v_3 \ v_4$



UV Faces
$v_1 \ v_2 \ v_3$
$v_4 \ v_5 \ v_6$

Texture Filtering

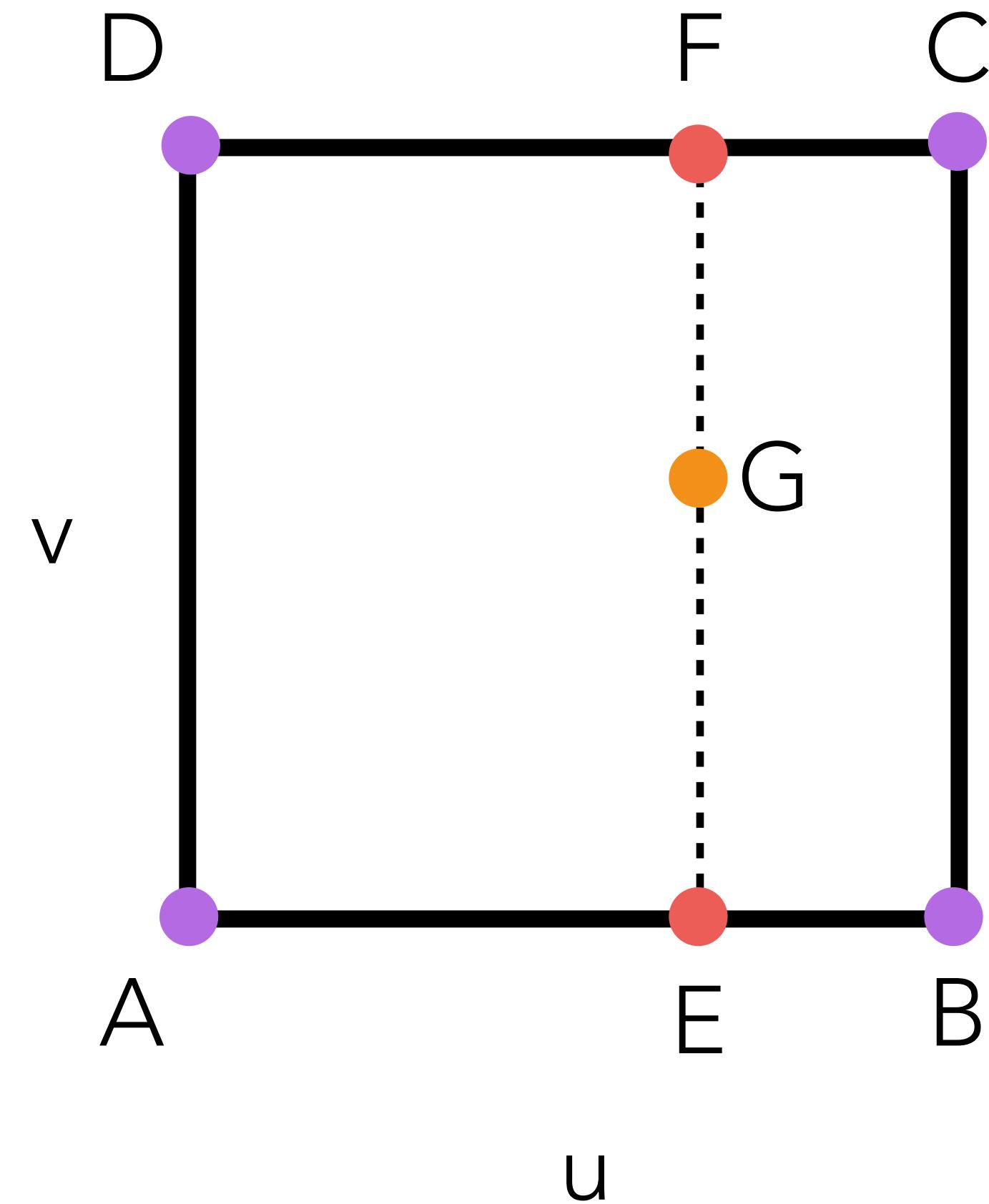


Nearest Point



Bilinear Interpolation

Bilinear Interpolation



$$E = (1-u) A + u B$$

$$F = (1-u) D + u C$$

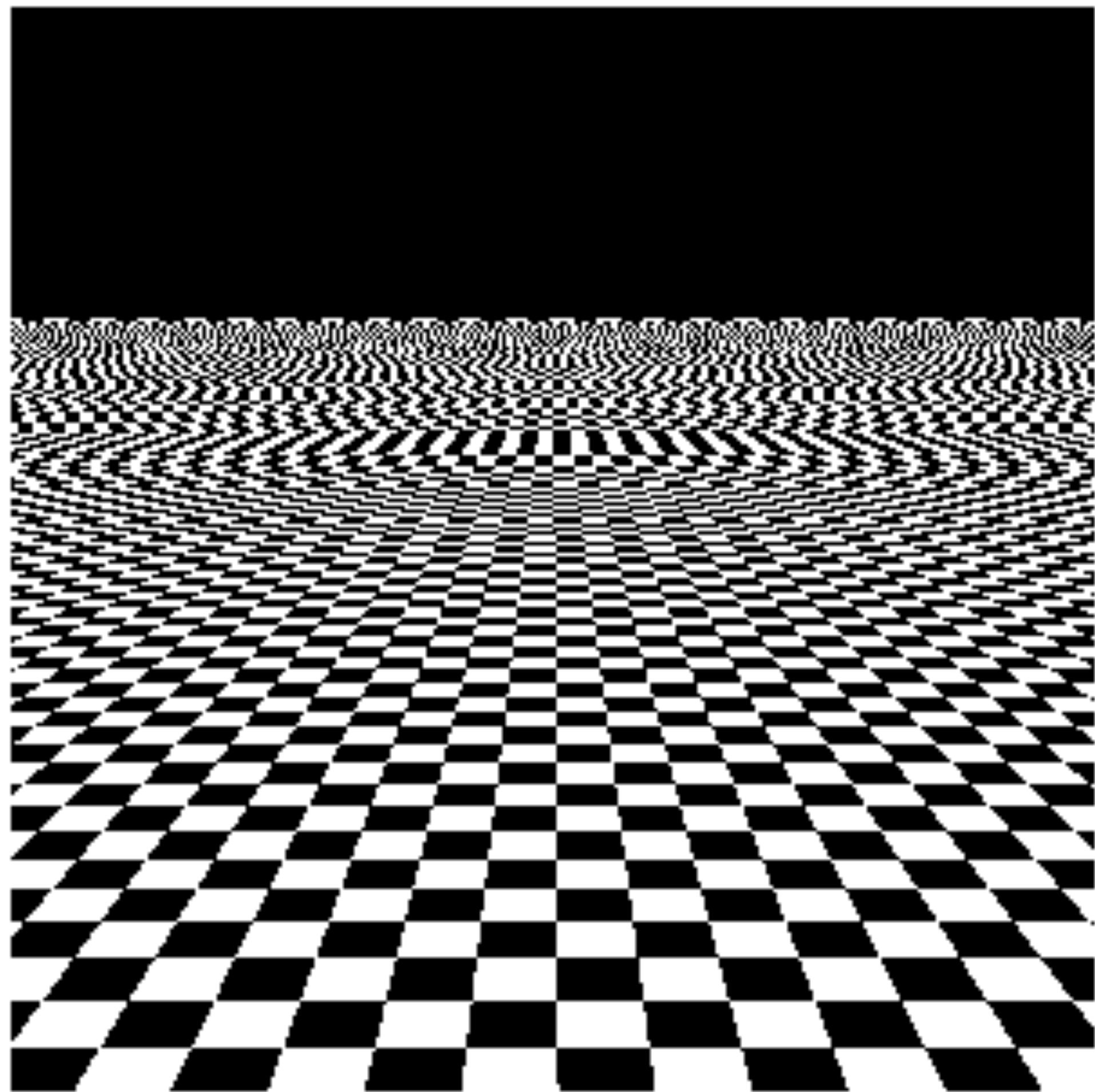
$$G = (1-v) E + v F$$

Moire Pattern



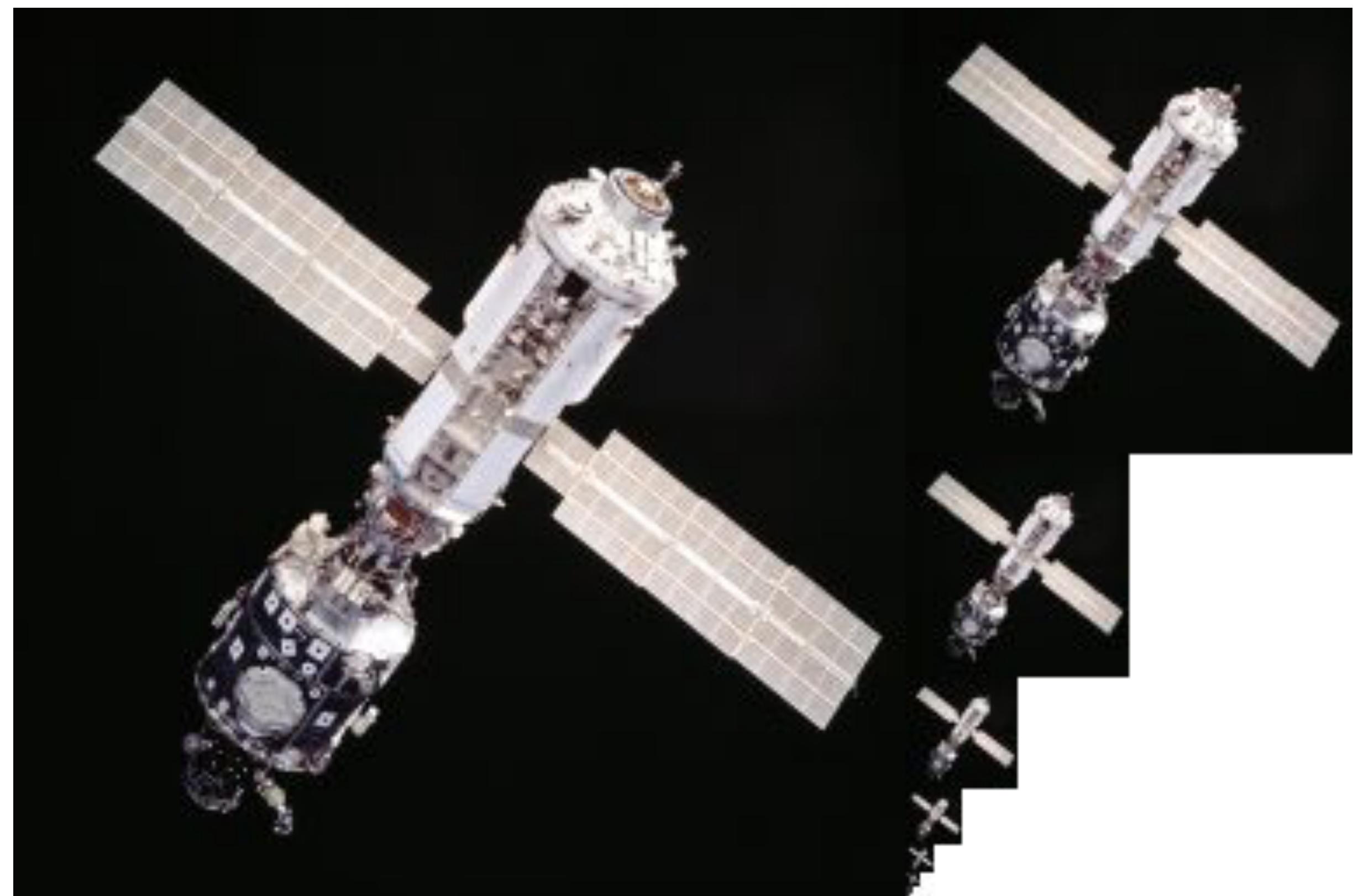
<http://photo.net/digital-darkroom-forum/00W8gC>

Mipmapping



Mipmapping

- Moire pattern can appear if the resolution of the texture is much higher than the sampling rate
- A good solution for this problem is mipmapping



By en:User:Mulad, based on a NASA image - Created by en:User:Mulad based on File:ISS from Atlantis - Sts101-714-016.jpg, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=1140741>