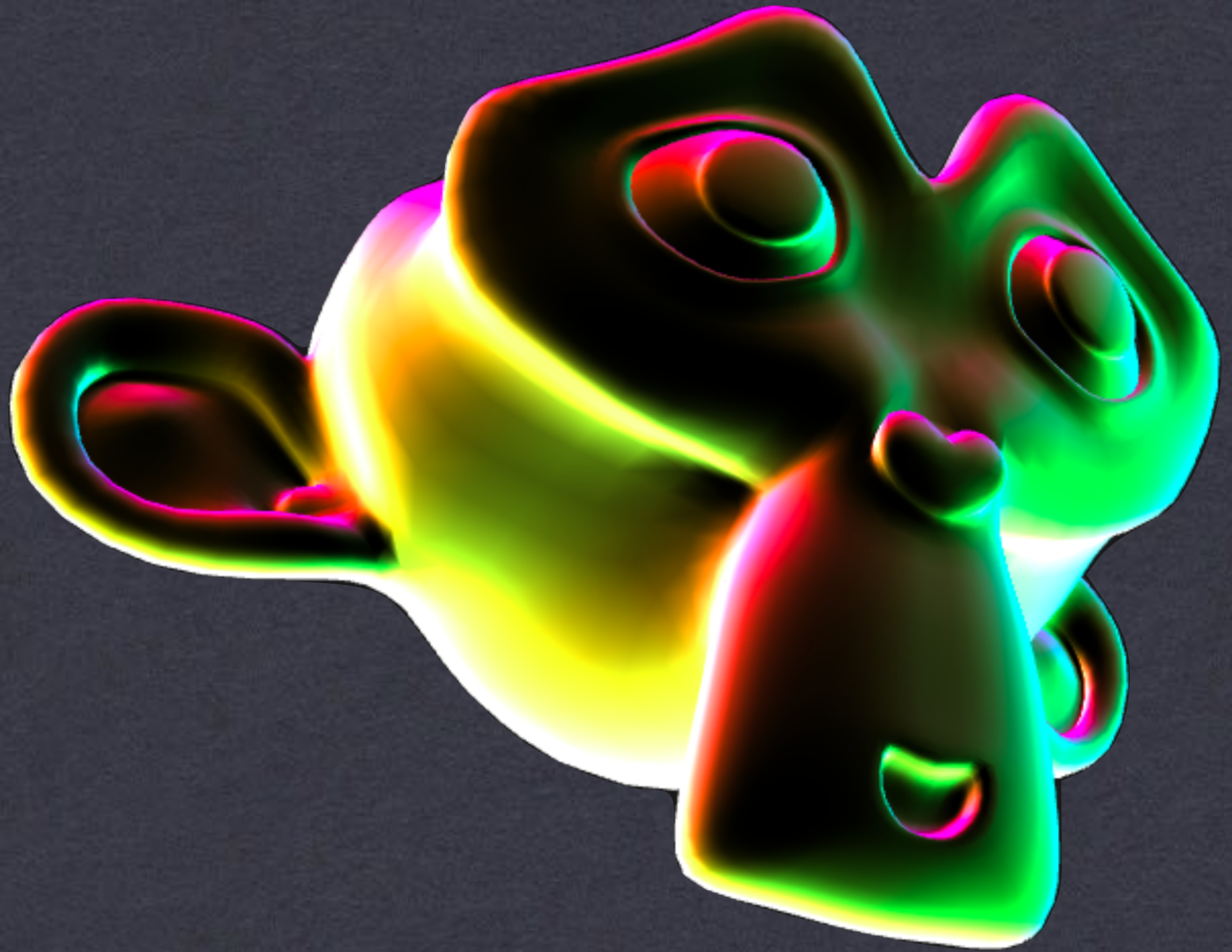
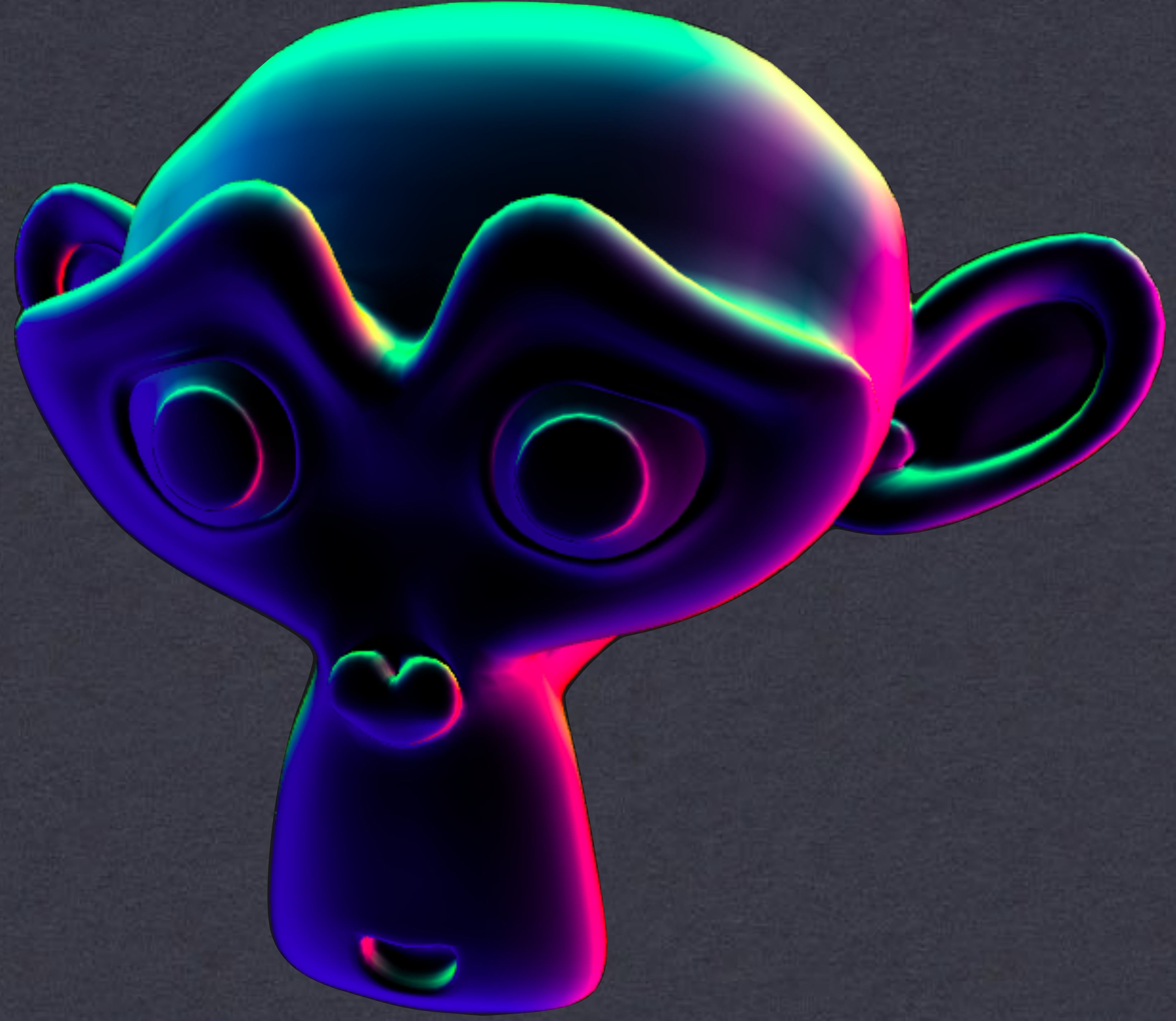




TEXTURE MAPPING

MUSL2361 - VR ADVENTURE

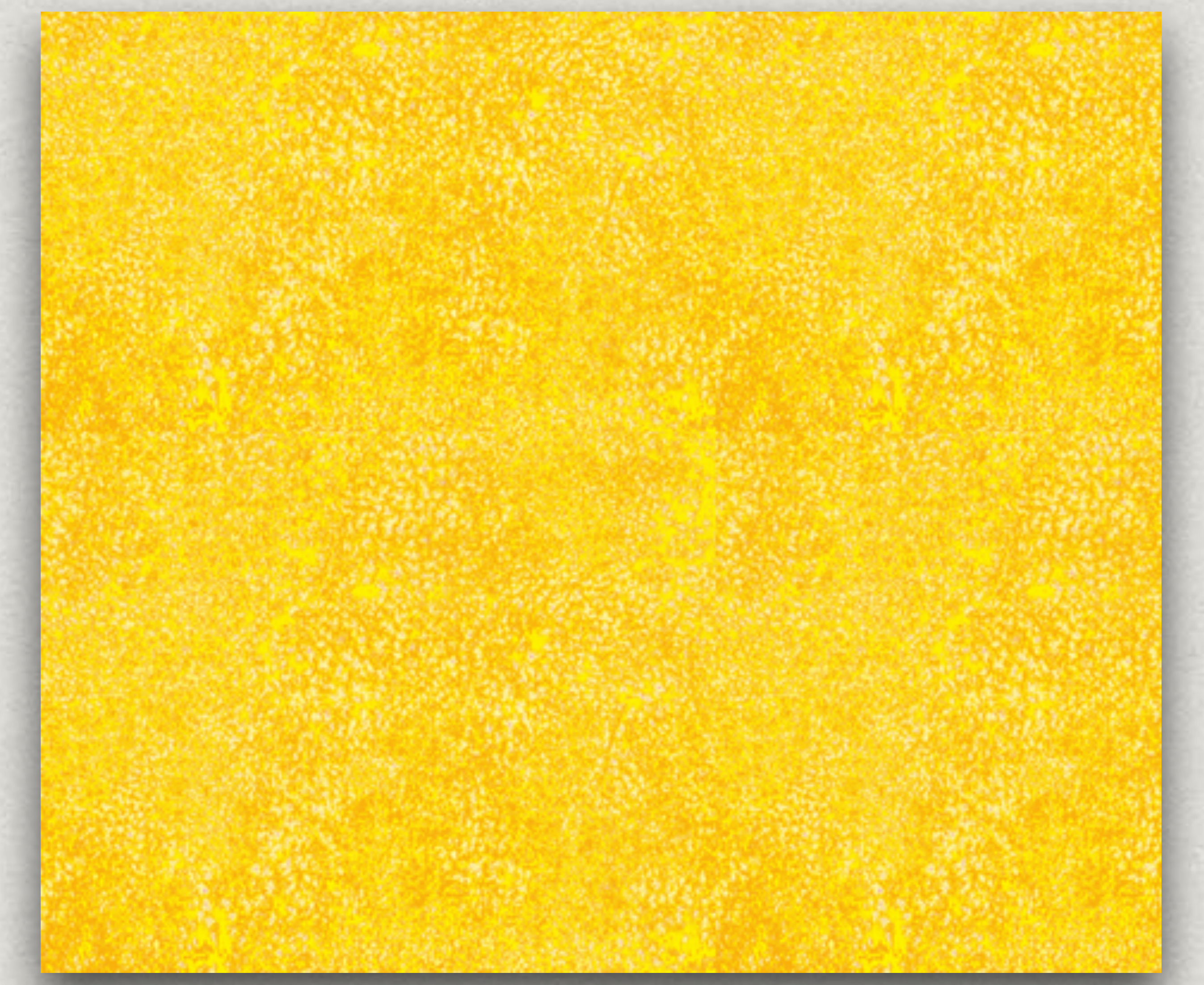
香 港 資 優 教 育 學 苑
The Hong Kong Academy for Gifted Education

Modeling an Orange

- Consider the problem of modeling an orange (the fruit)
- Start with an **orange-colored sphere**
  - Too simple
- Replace sphere with a more complex shape
  - Takes **too many polygons** to model all the small dimples

Modeling an Orange (2)

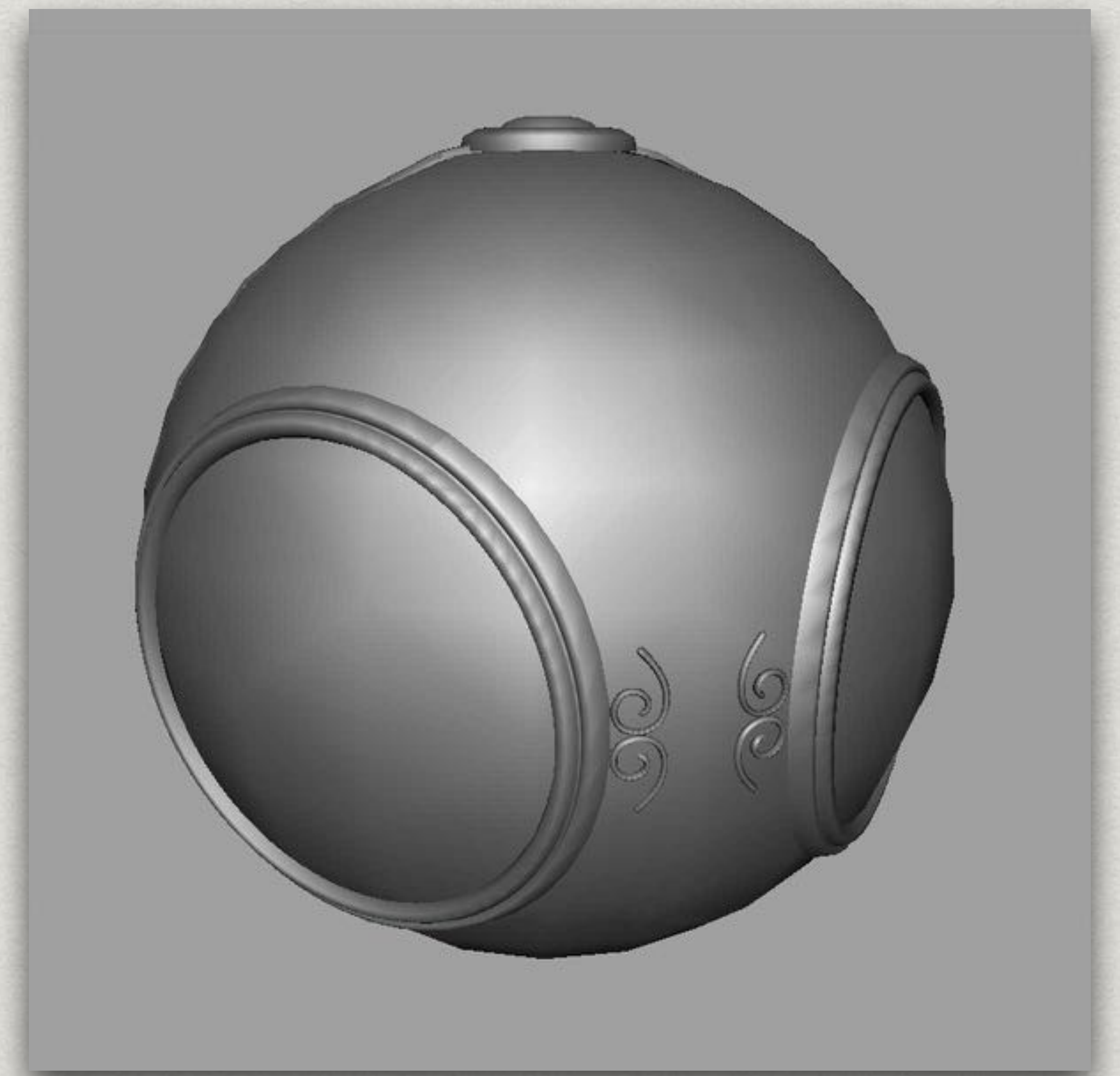
- Take a picture of a real orange, scan it, and "paste" onto simple geometric model
  - This process is known as **texture mapping**
- Still might not be sufficient because **resulting surface will be smooth**
  - Need to change local shape
  - Normal mapping



Three Types of Mapping

- Texture Mapping
  - Uses **images** to fill inside of polygons
- Environment (reflection mapping)
  - Uses a **picture of the environment** for texture maps, which allow simulation of **highly specular surfaces**
- Normal mapping
  - Emulates **altering normal vectors** during the rendering process

Texture Mapping

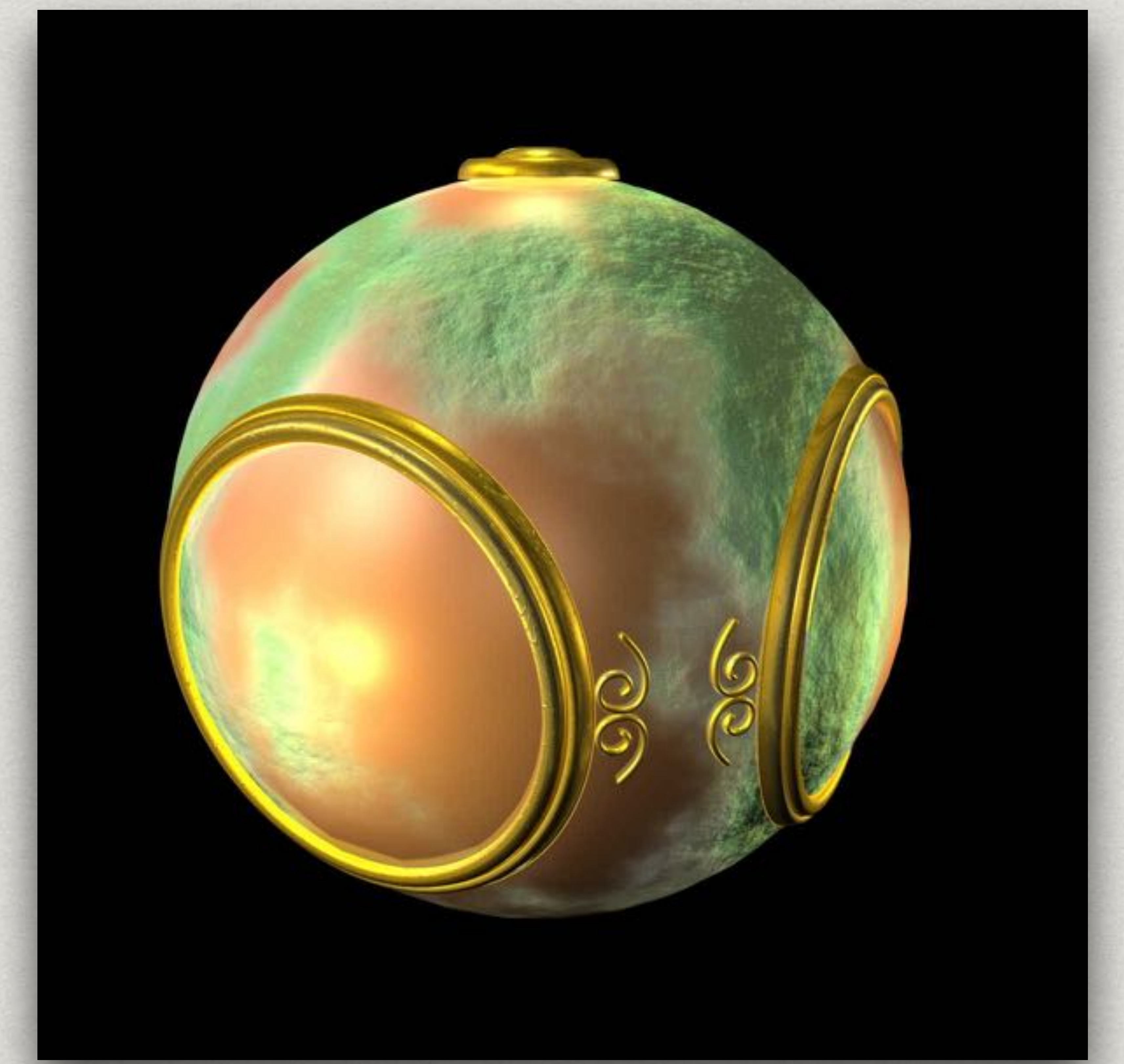


geometric model

texture mapped

Mappings

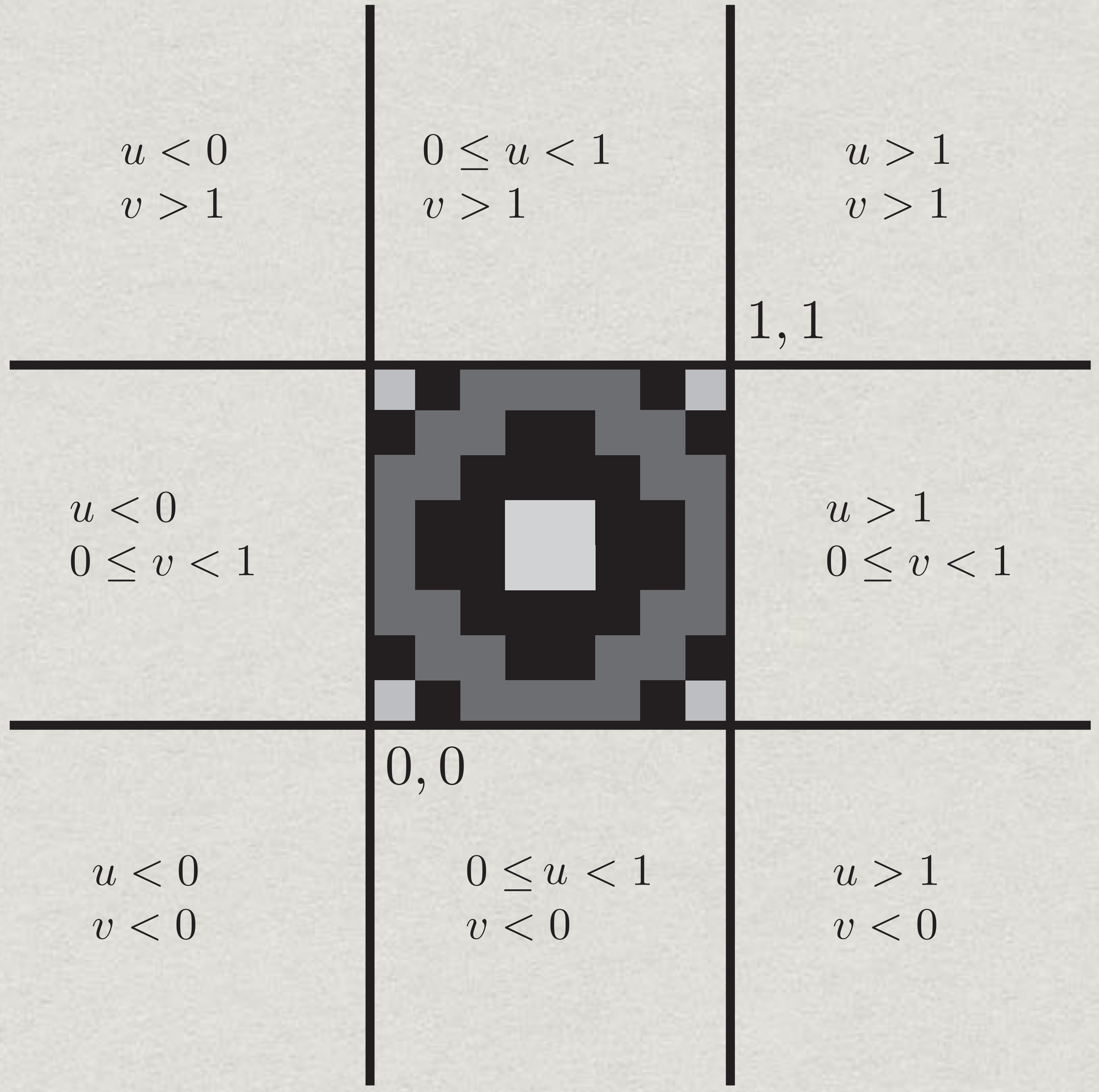
Environment mapped



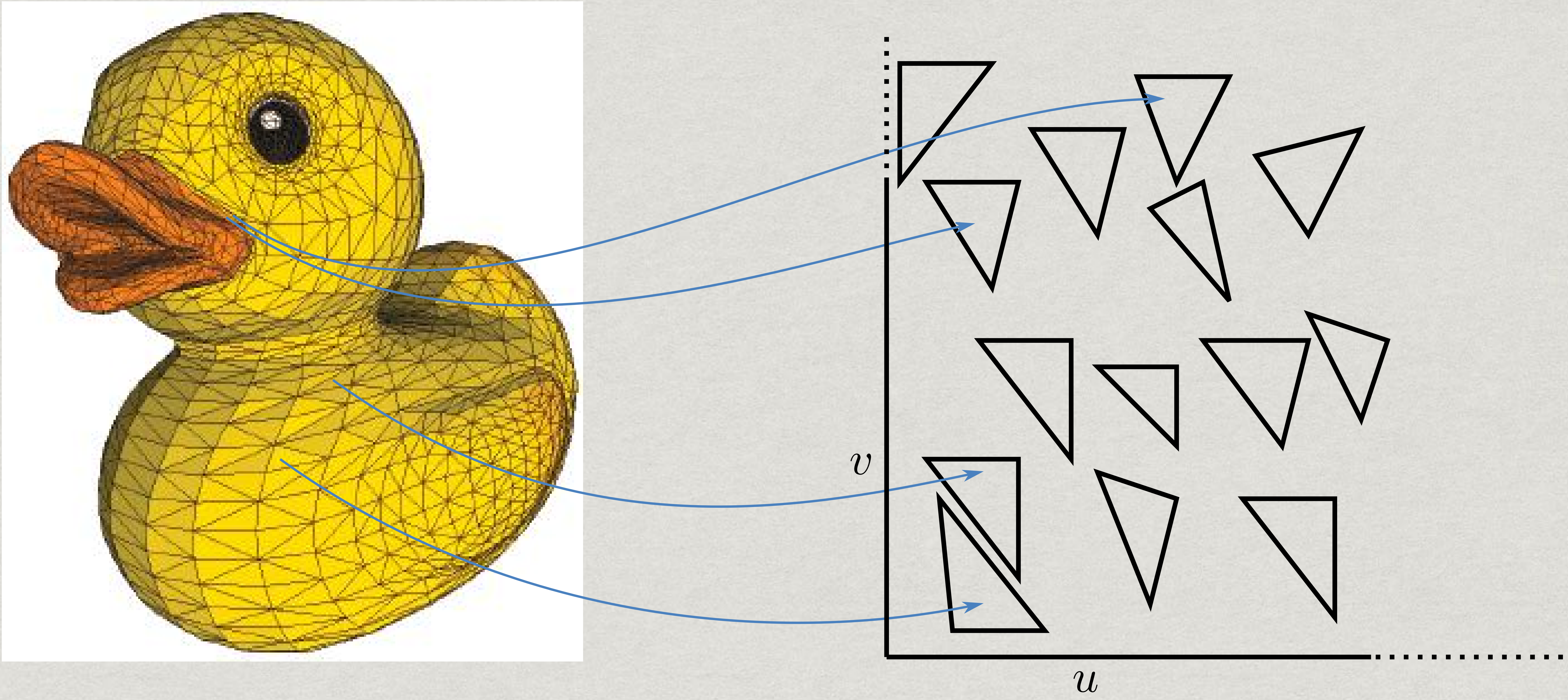
Normal mapped

Basic Concepts

- A **texture** is a raster image, and its pixels are called **texels** (**tex**ture **el**ements).
- A position in a texture is conventionally referred to as **texture coordinates**, or **UV-coordinates**.
- The texture corresponds to the rectangular space between **(0, 0)** and **(1, 1)** is called **texture space**.



Assign each vertex with a (u,v)



Not every texel is used in the 3D model!!!

Environment Mapping

- Texture mapping used to show the **reflection of the surrounding environment** on the 3D scene to be represented.
- Cube Mapping
    - We can form a **cube map texture** by defining **six 2D texture maps** that correspond to the sides of a box

Cube Mapping

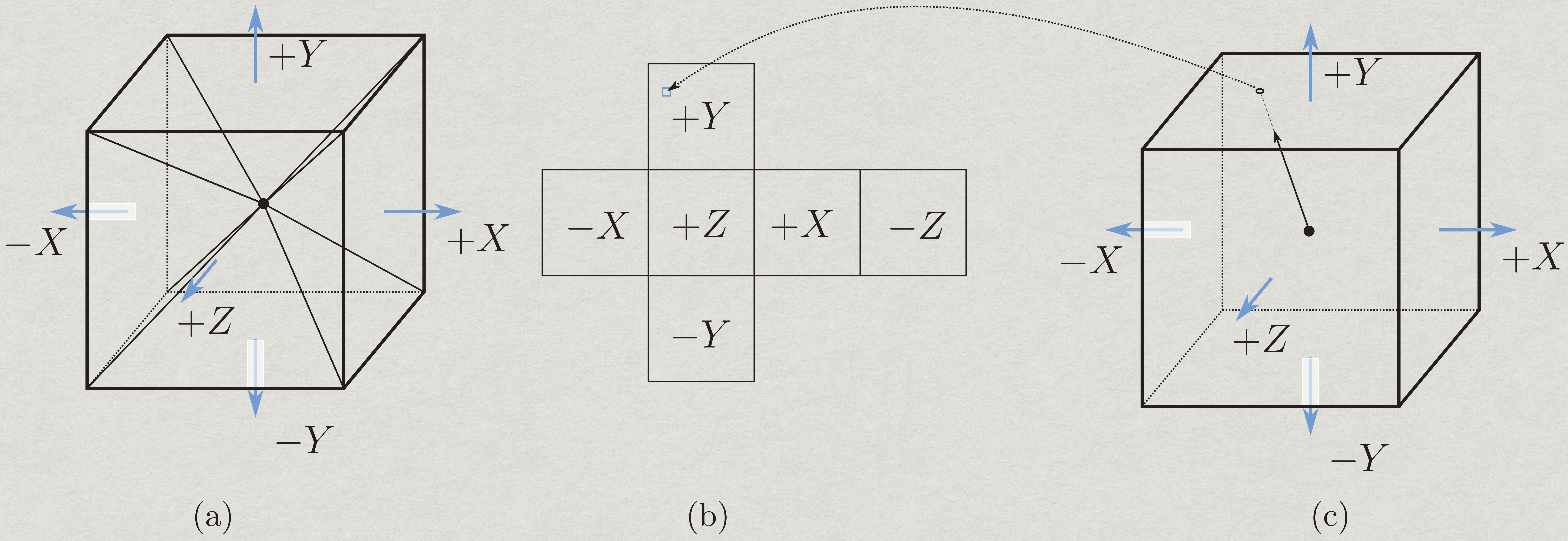
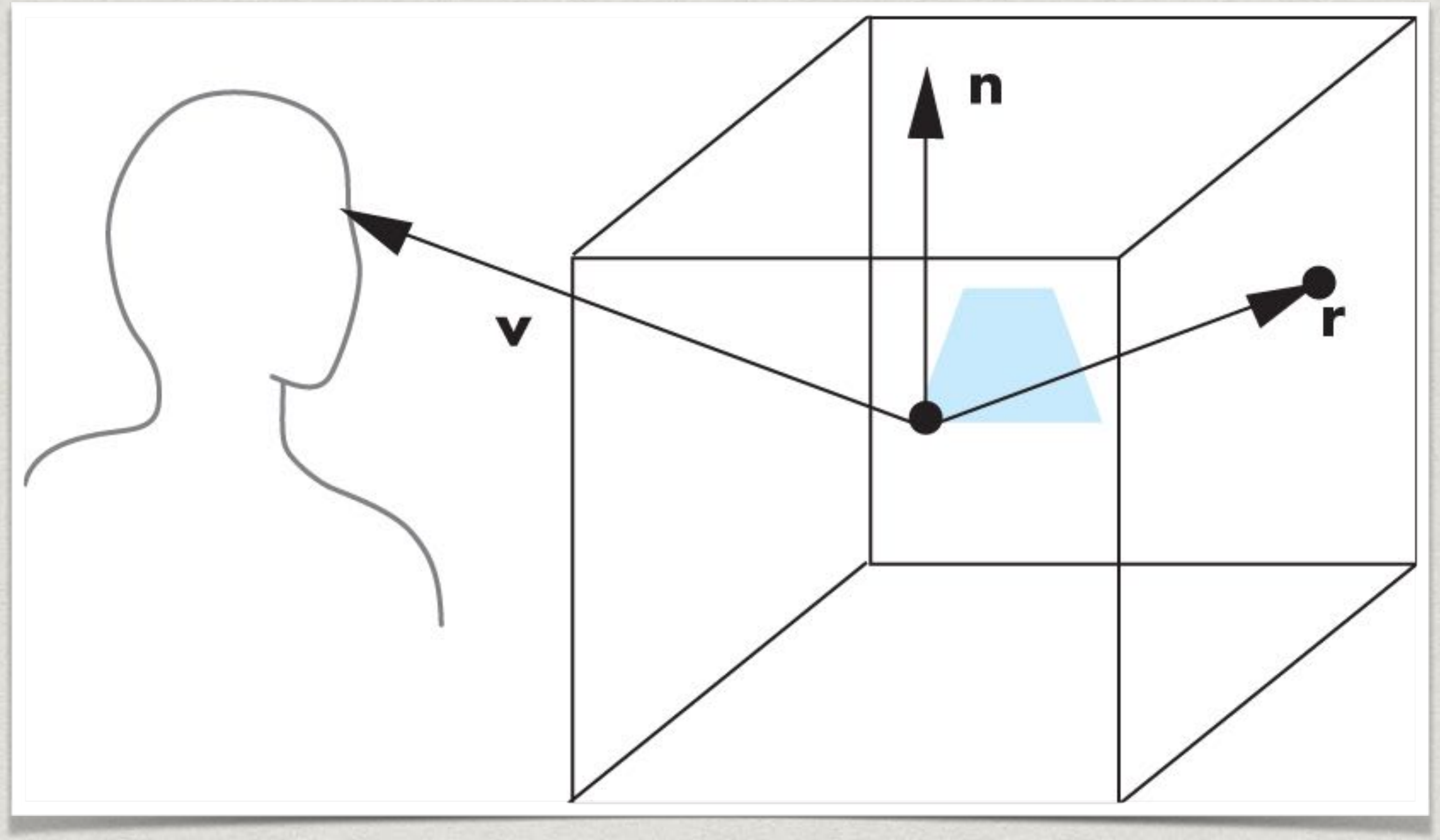


FIGURE 7.18: (a) Six images are taken from the center of the cube. (b) The cube map: the cube is unfolded as six square images on the plane. (c) Mapping from a direction to texture coordinates.

Finding the Texture Coordinates

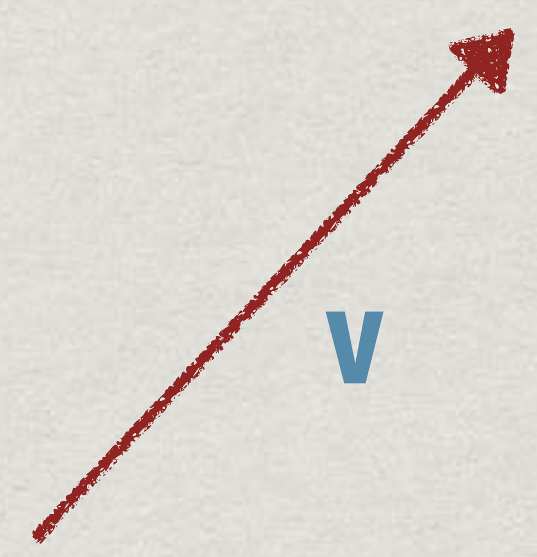
- Find the **intersection** between the **extension of the reflection direction** and the cube.



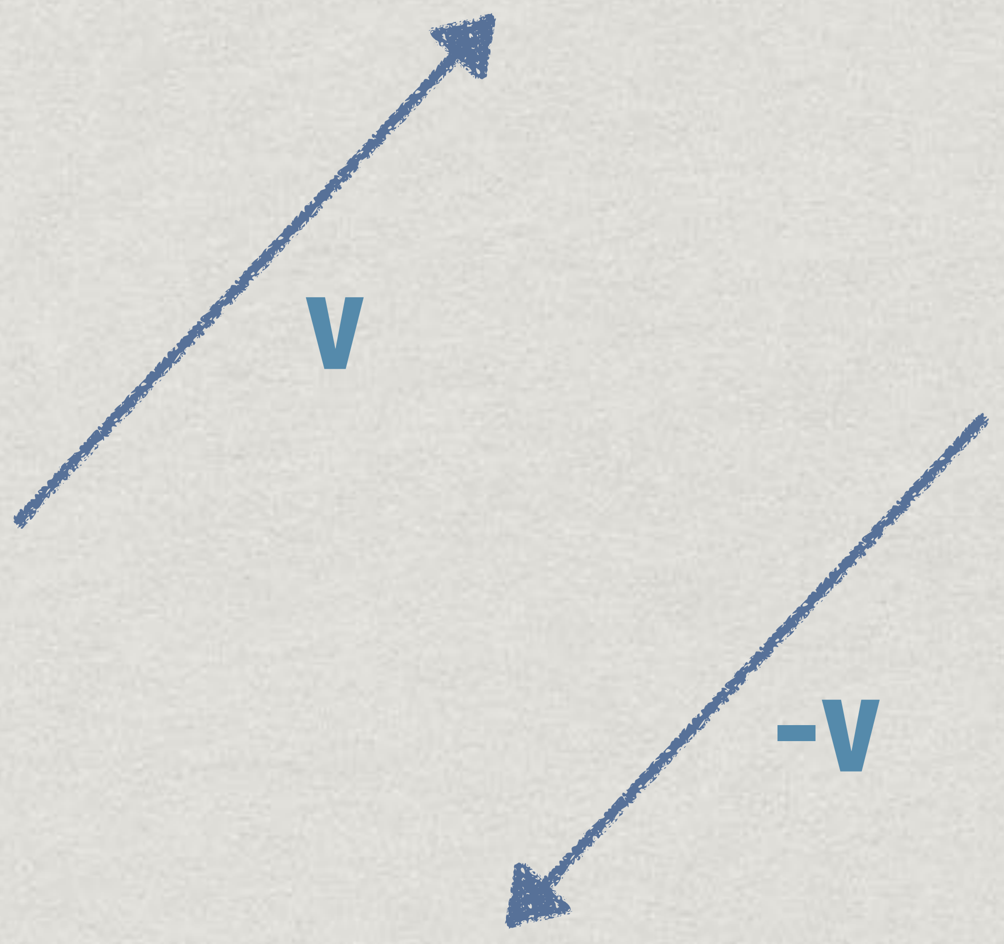
VECTORS

Vectors

- Physical definition: **a vector is a quantity with two attributes**
  - **Direction** and **Magnitude**
- Examples include
  - Force, Velocity
  - Directed line segments

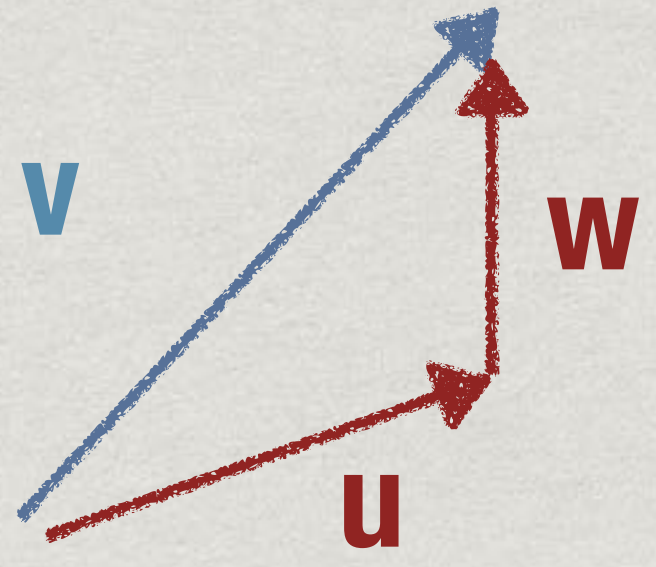


Vector Operations



- **Every vector has an inverse**
  - Same magnitude but **points in opposite direction**

- Every vector can be **multiplied by a scalar**
- **The sum of any two vectors is a vector**
  - Use head-to-tail axiom



Mathematical Expression

* Vector is usually expressed with **column notation**.

$$\vec{v} = \begin{pmatrix} 2 \\ 1 \\ -2 \end{pmatrix}$$

* Vector can be **multiplied by a scalar.**

$$3\vec{v} = 3 \times \begin{pmatrix} 2 \\ 1 \\ -2 \end{pmatrix} = \begin{pmatrix} 6 \\ 3 \\ -6 \end{pmatrix}$$

Mathematical Expression

- **The sum of any two vectors is a vector**

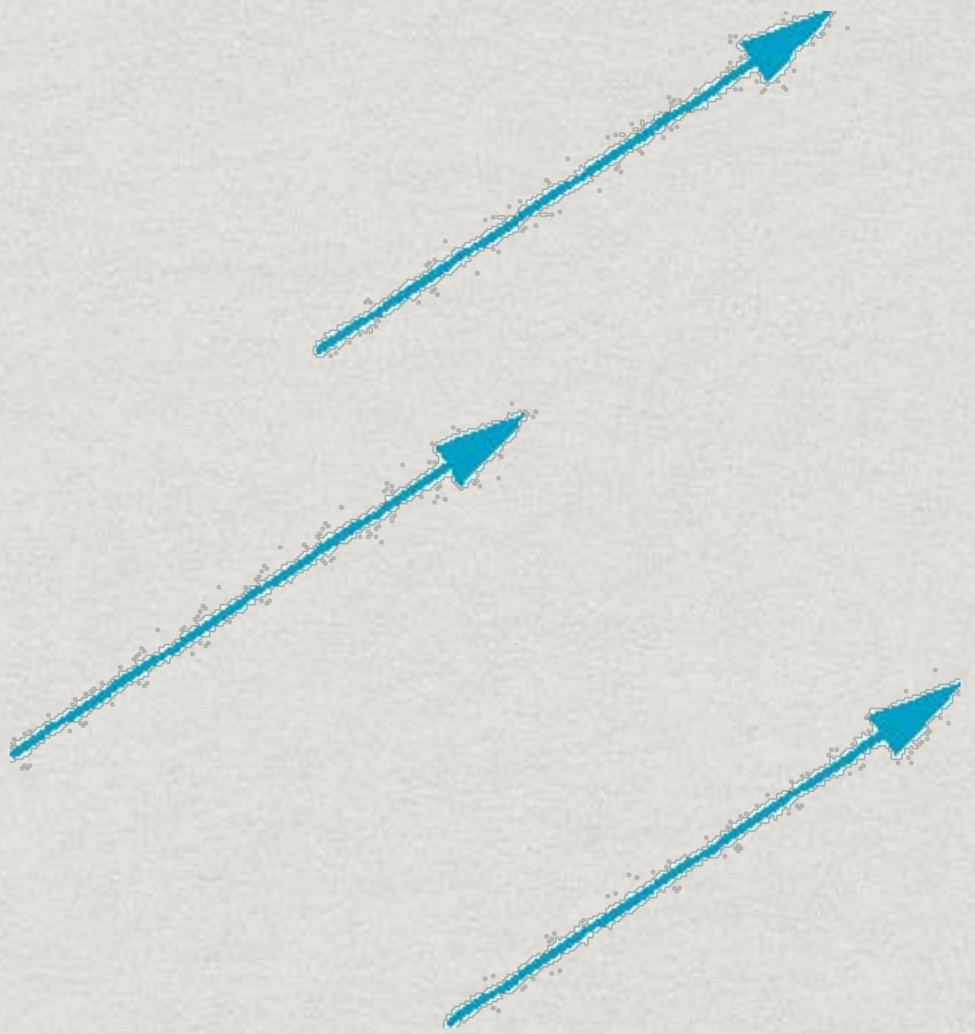
$$\vec{v} = \vec{u} + \vec{w} = \begin{pmatrix} 5 \\ -2 \\ 3 \end{pmatrix} + \begin{pmatrix} 0 \\ 8 \\ -7 \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \\ -4 \end{pmatrix}$$

- To find the **magnitude** of a vector?
  - **Pythagorean Theorem**.

$$|\vec{v}| = \sqrt{2^2 + 1^2 + (-2)^2} = 3$$

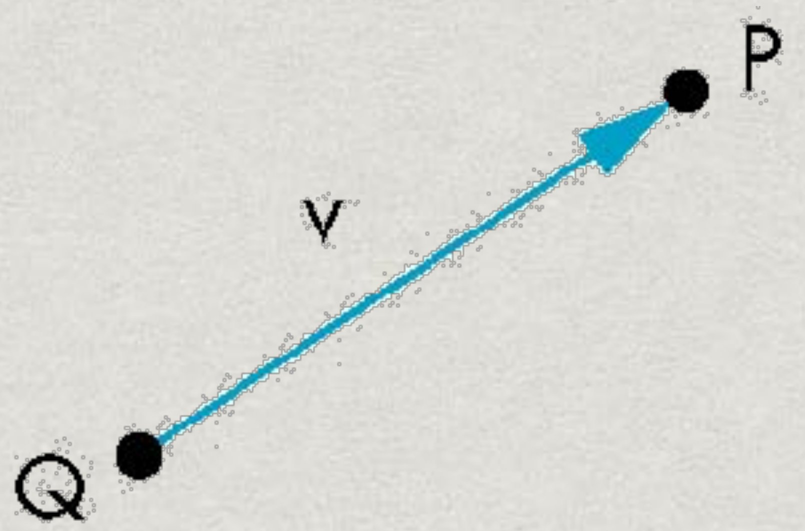
Vectors Lack Position

- These vectors are identical
  - Same length and magnitude
- Vectors spaces insufficient for geometry
  - Need points



Points

- Location in space

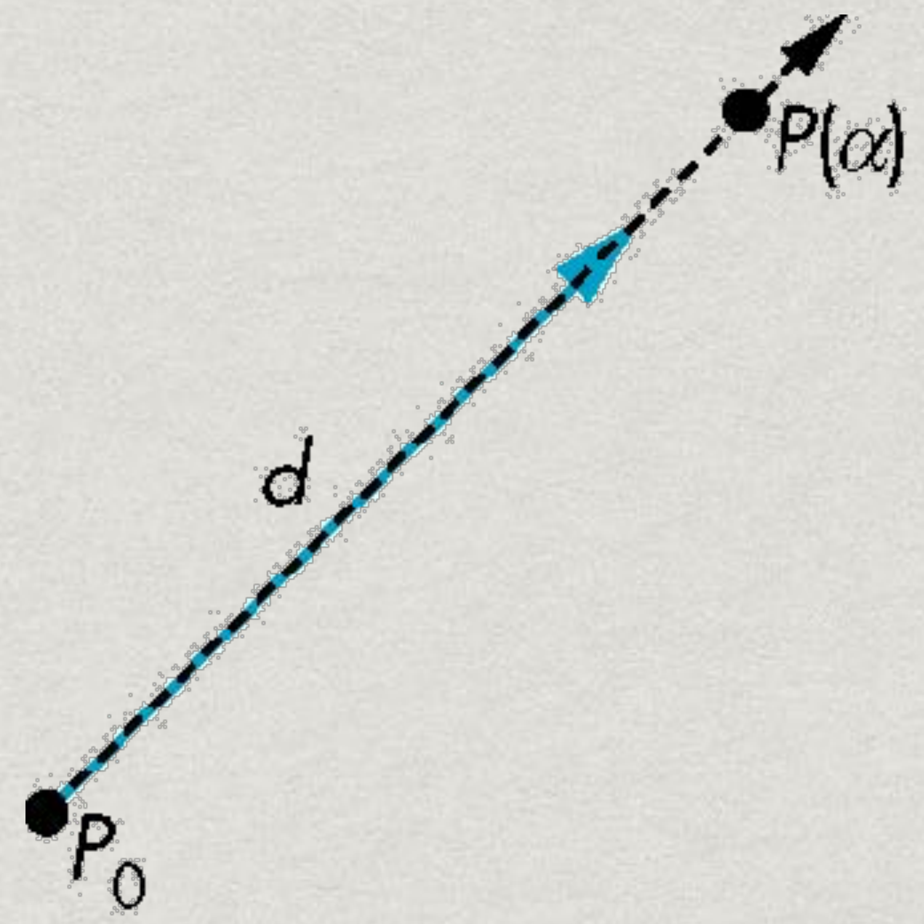


- Operations allowed between points and vectors
  - Point-point subtraction yields a vector $v = P - Q$
  - Equivalent to point-vector addition $P = v + Q$

Lines

- Consider all points of the form

$$P(\alpha) = P_0 + \alpha\vec{d}$$

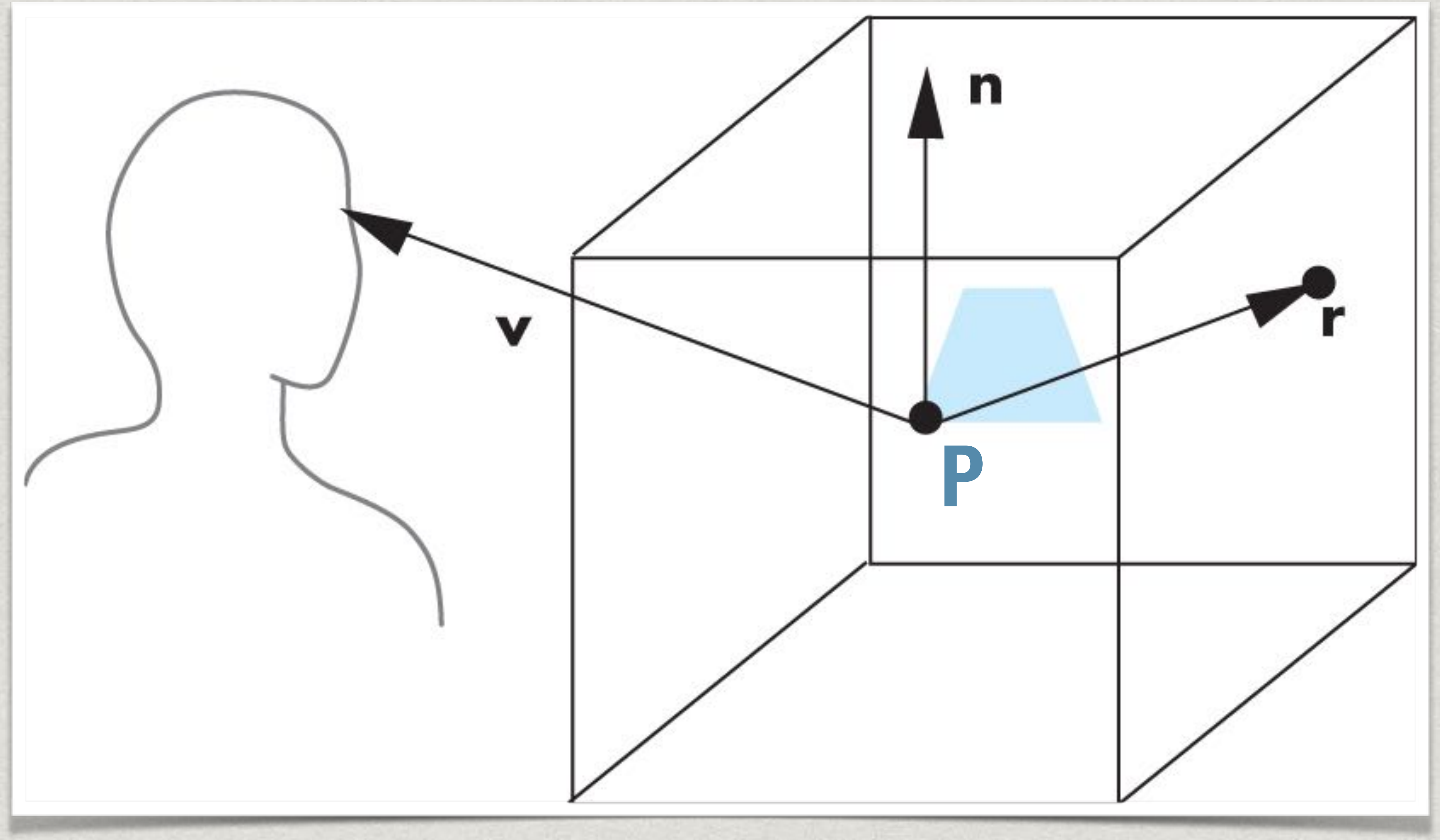


set of all points that pass through P_0 in the direction of the vector d

Finding the Texture Coordinates

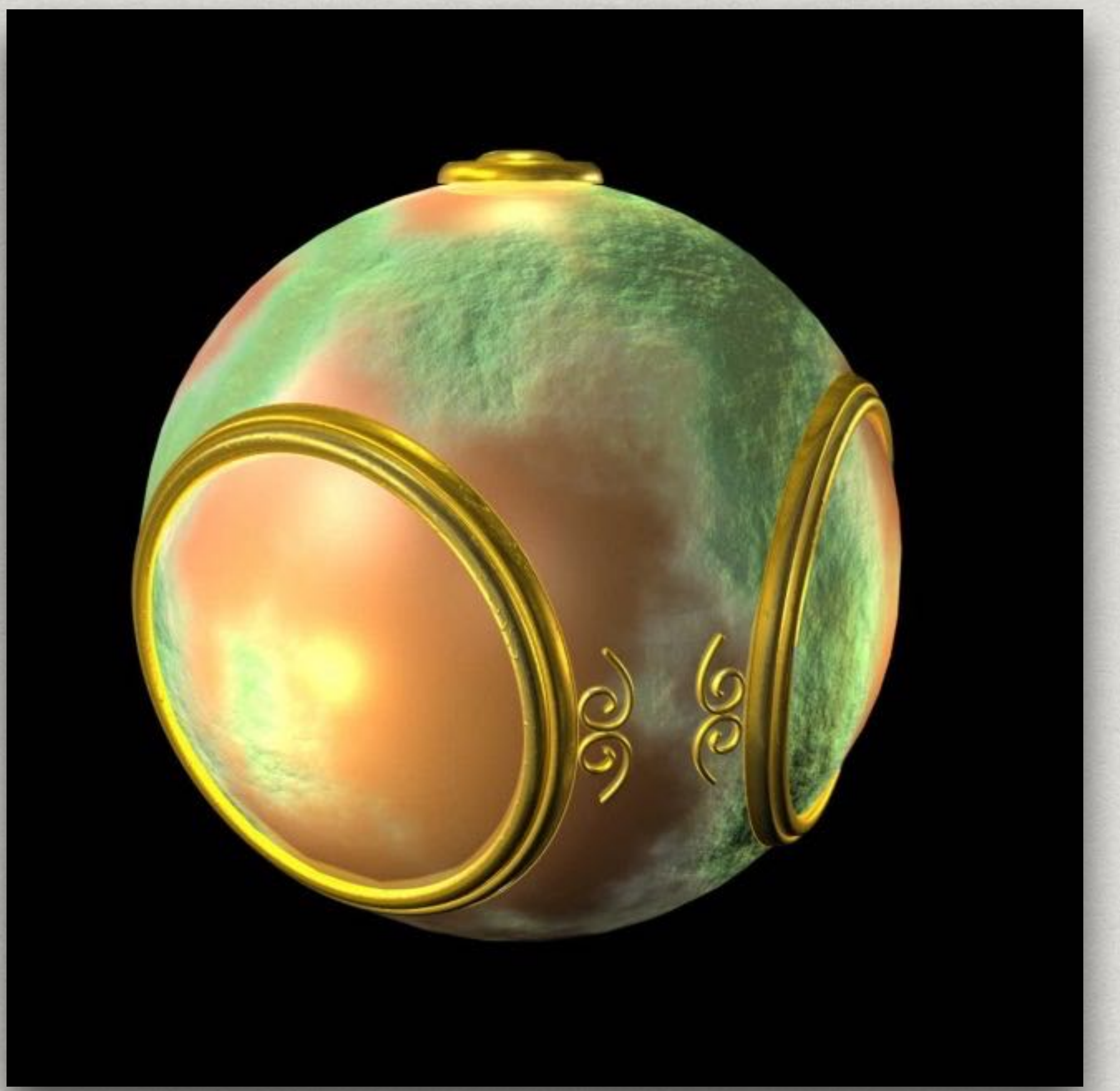
- Suppose the 6 environment maps are all placed at **planes 10 units away from the origin**.
- What is the intersect point if

$$P = \begin{pmatrix} 4 \\ 5 \\ -3 \end{pmatrix} \qquad \vec{r} = \begin{pmatrix} 2 \\ 1 \\ -2 \end{pmatrix}$$



Normal Mapping

- With **normal mapping,** we use textures to encode the **normals** and use them during rasterization to compute **per-fragment lighting**.
- We refer to these textures with the name of **normal maps**.

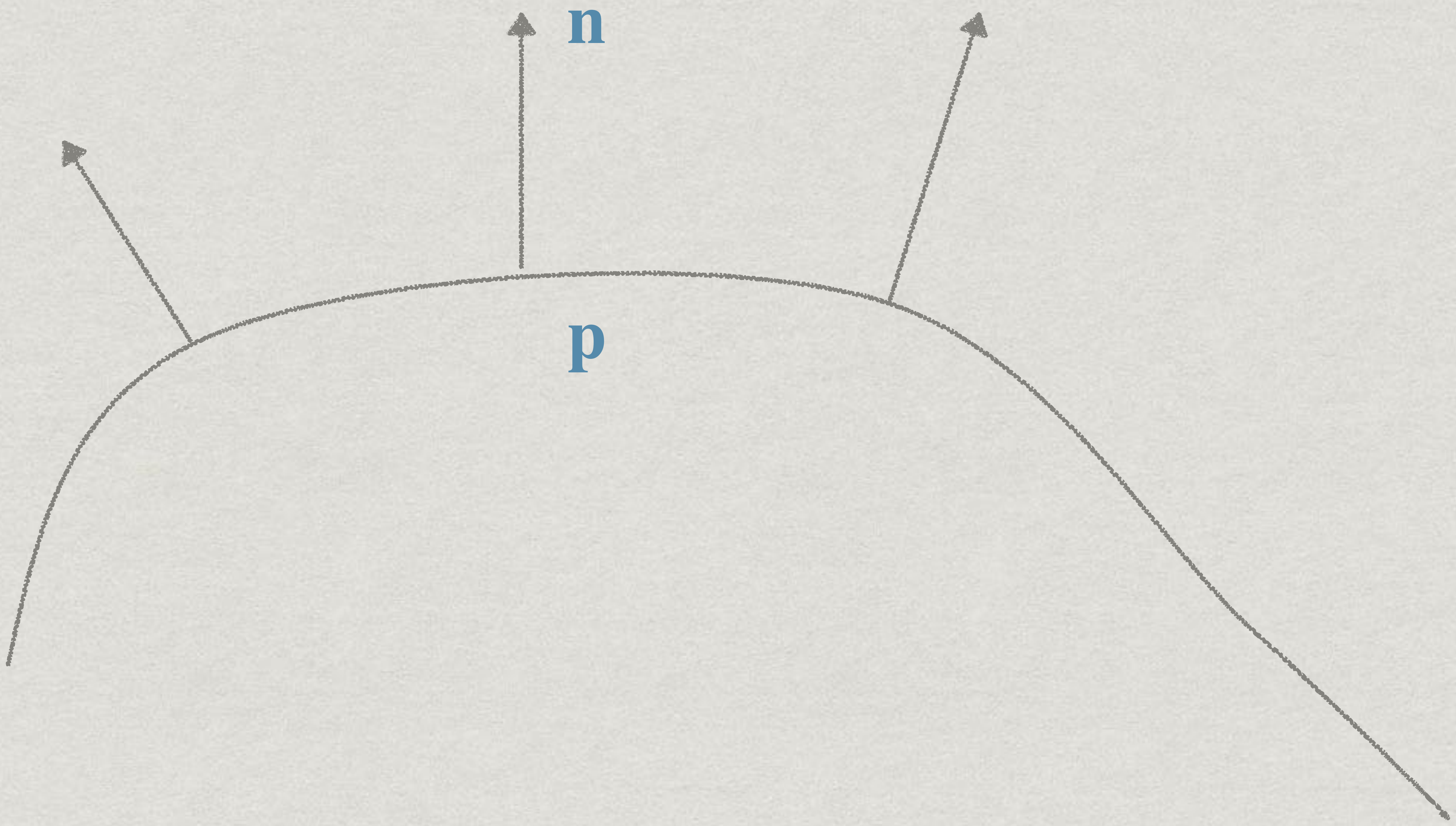


Modeling an Orange

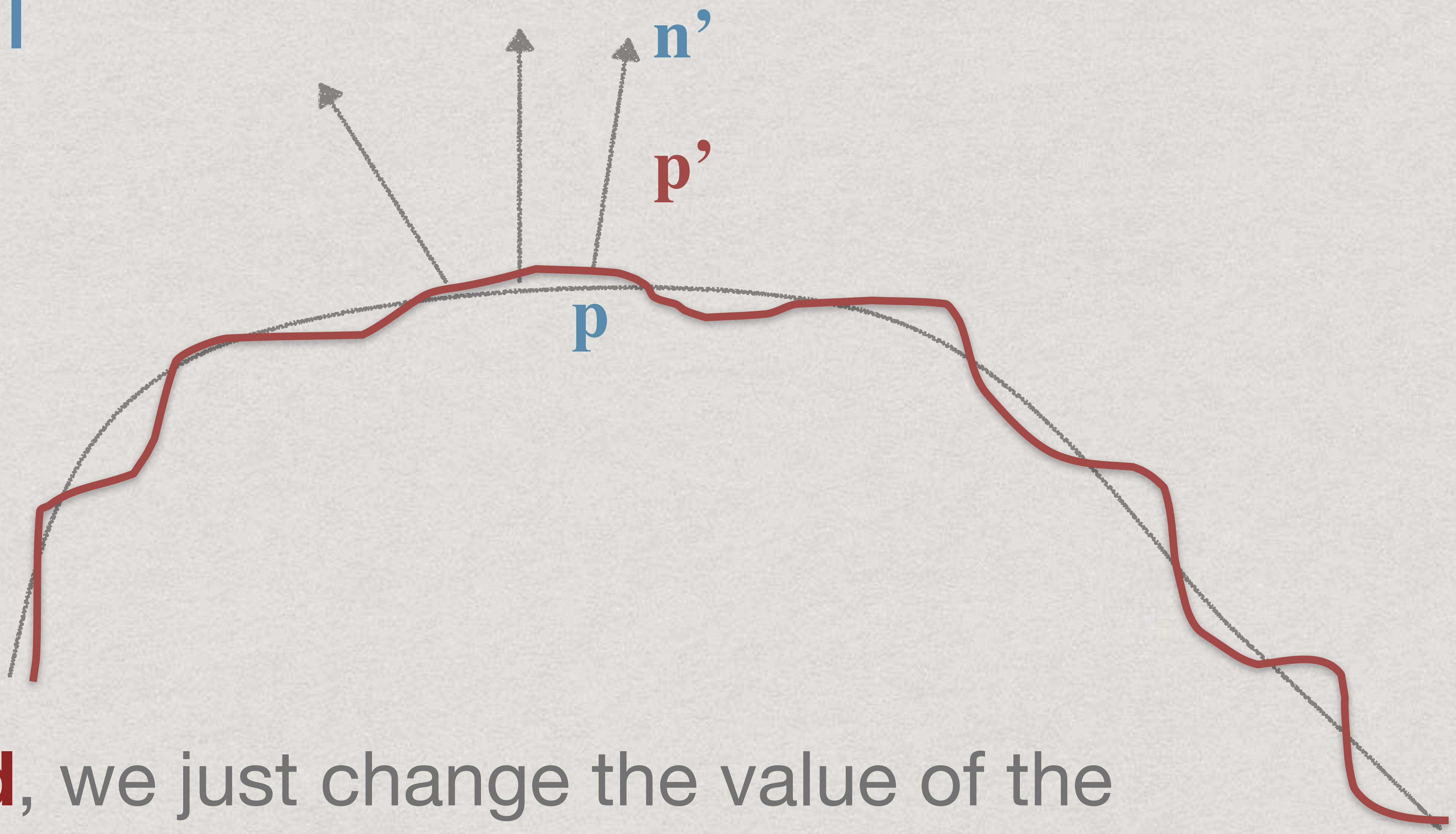
- Consider **modeling an orange**
- Texture map a photo of an orange onto a surface
  - Captures dimples
  - Will **not be correct if we move viewer or light.** We have shades of dimples rather than their correct orientation
- Ideally we need to **adjust normal** across surface of object and **compute a new color at each interior point**

Normal Mapping

- Consider a smooth surface

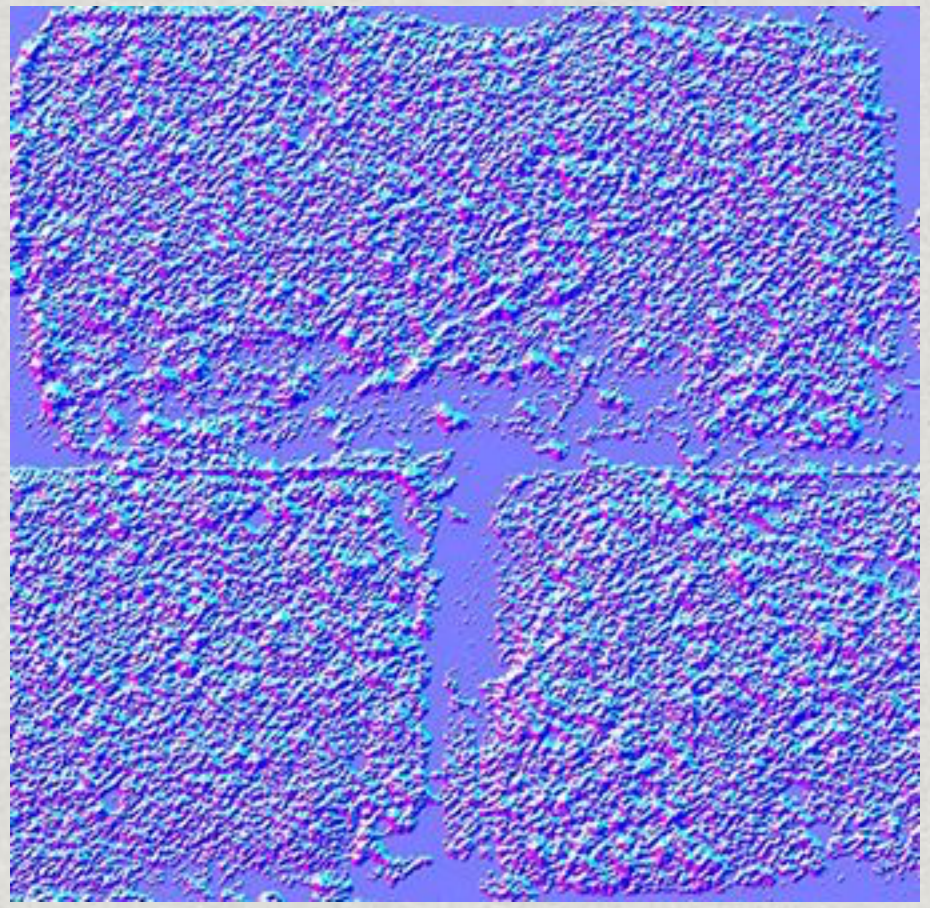


Rougher Version



- It seems point p has been moved to p’ as the **normal of that point** changed to n’.
- **The geometry is not altered**, we just change the value of the normal to be the one that we would have if the geometry was actually displaced.

Normal Maps



- Also called **Normal Textures**
- In each RGB texel is encoded **a XYZ vector** :
  each color component is between 0 and 1,
  and each vector component is between -1 and 1,
  so this simple mapping goes from the texel to the normal :

$$normal = (2 \times color) - 1$$

// on each component

Tangent Space Normal Map

- Tangent space normal map has a **blue tone**
- It stores normal **relative to the surface orientation**.
- The texture is blue as normal is **pointing outwards**.