# Computer Graphics (CS 543)
# Lecture 11 (Part 1): Sphere Mapping, Normal Maps, Parametrization

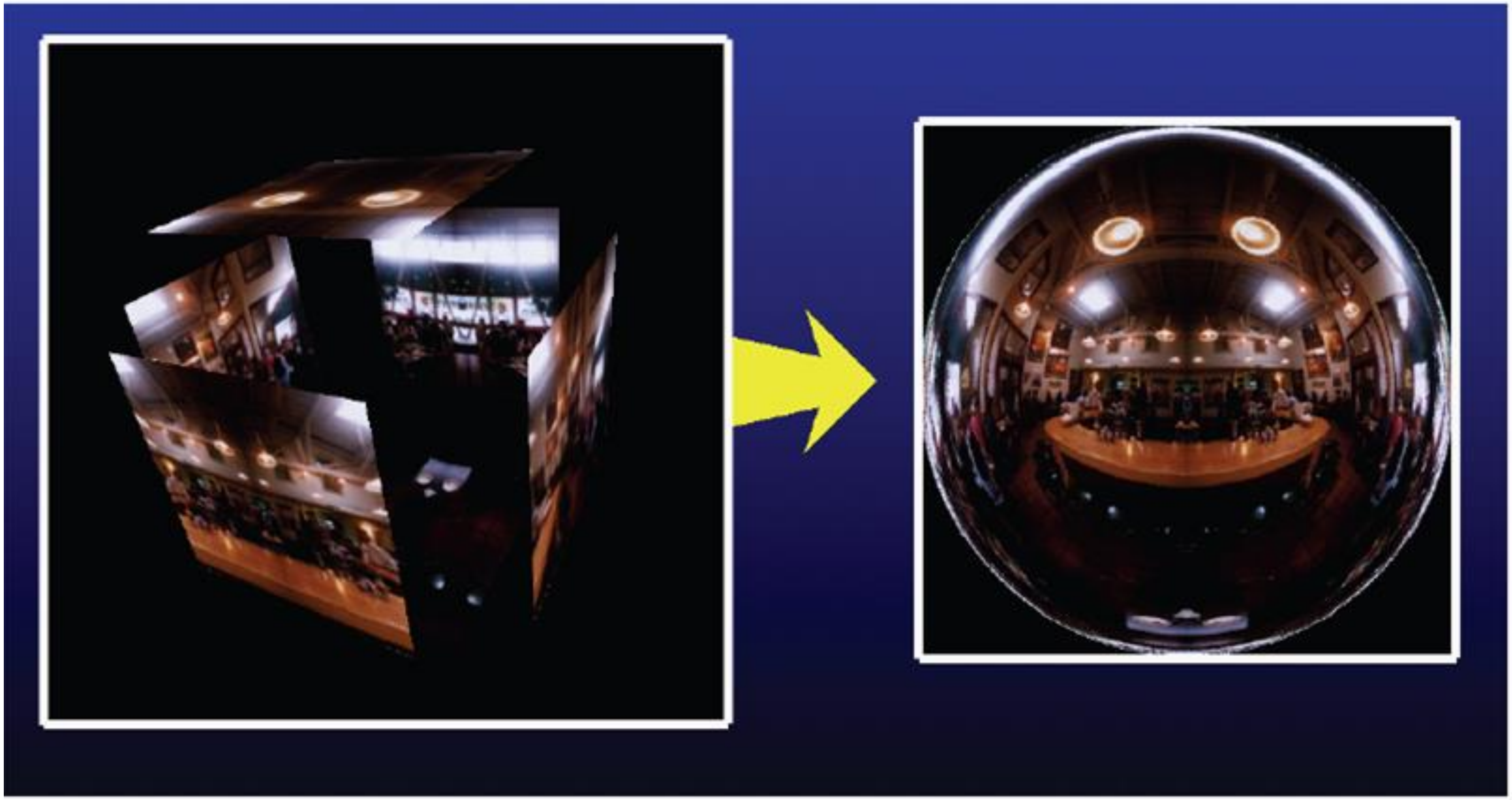
Prof Emmanuel Agu

*Computer Science Dept.*

*Worcester Polytechnic Institute (WPI)*

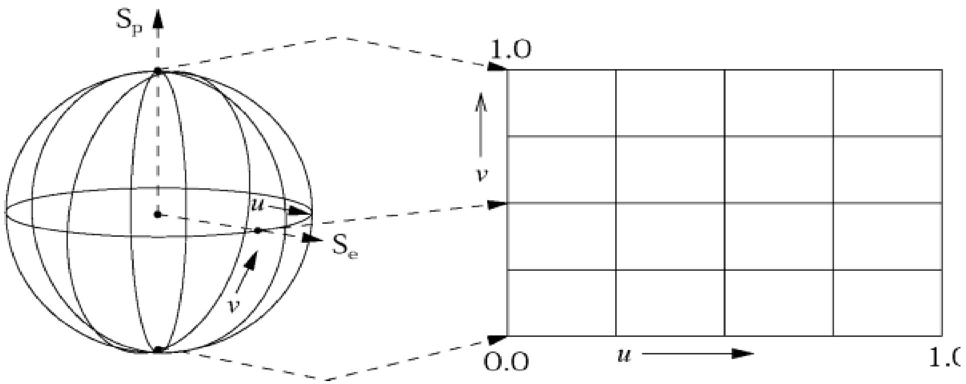
# Sphere Environment Map

- Cube can be replaced by a sphere (sphere map)



# Sphere Mapping

- Original environmental mapping technique
- Proposed by Blinn and Newell
- Uses lines of longitude and latitude to map parametric variables to texture coordinates
- OpenGL supports sphere mapping
- Requires a circular texture map equivalent to an image taken with a fisheye lens



# Sphere Map

- A sphere maps is basically a photograph of a reflective sphere in an environment

Paul DeBevec, www.debevec.org

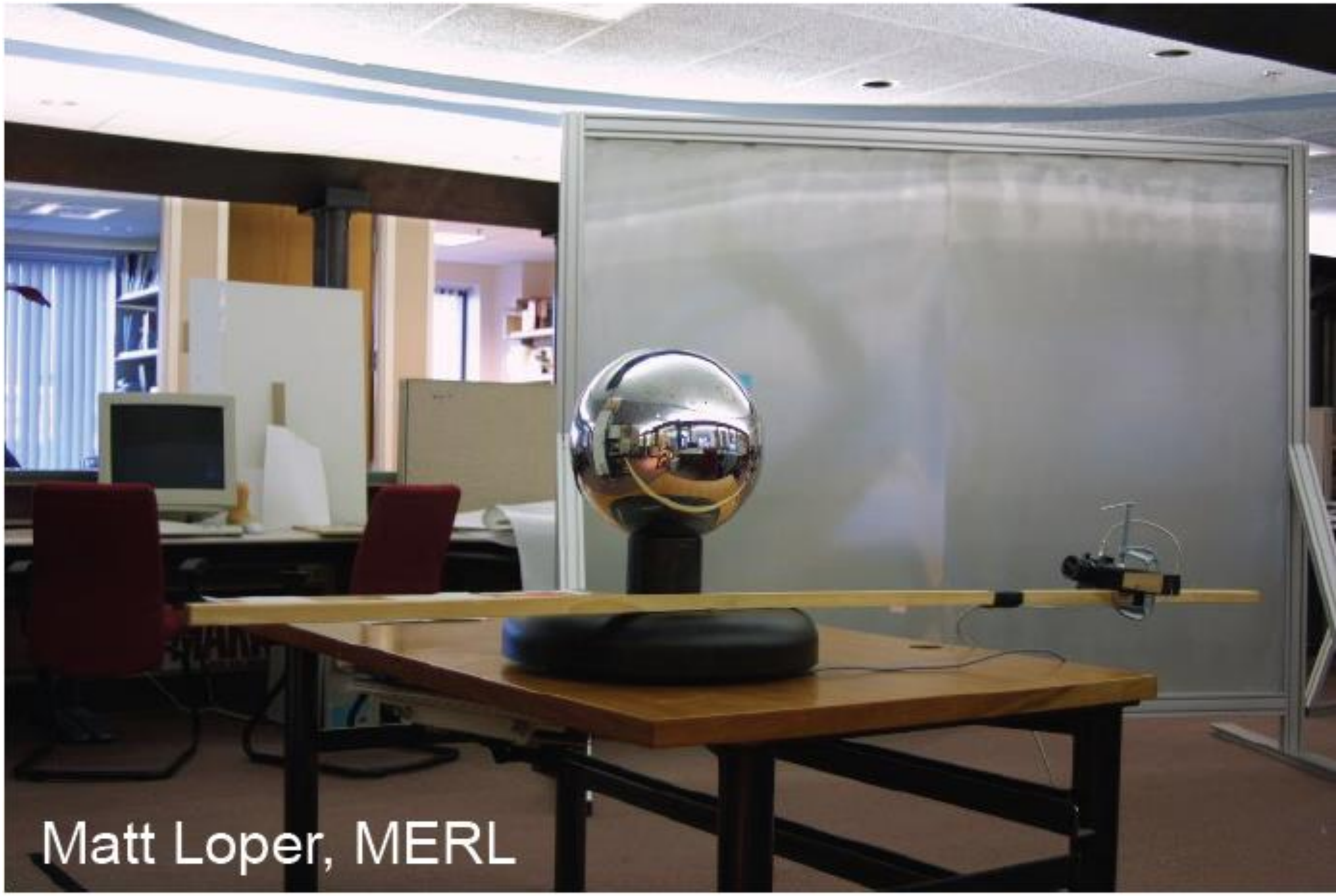
# Sphere map

- example

Sphere map
(texture)

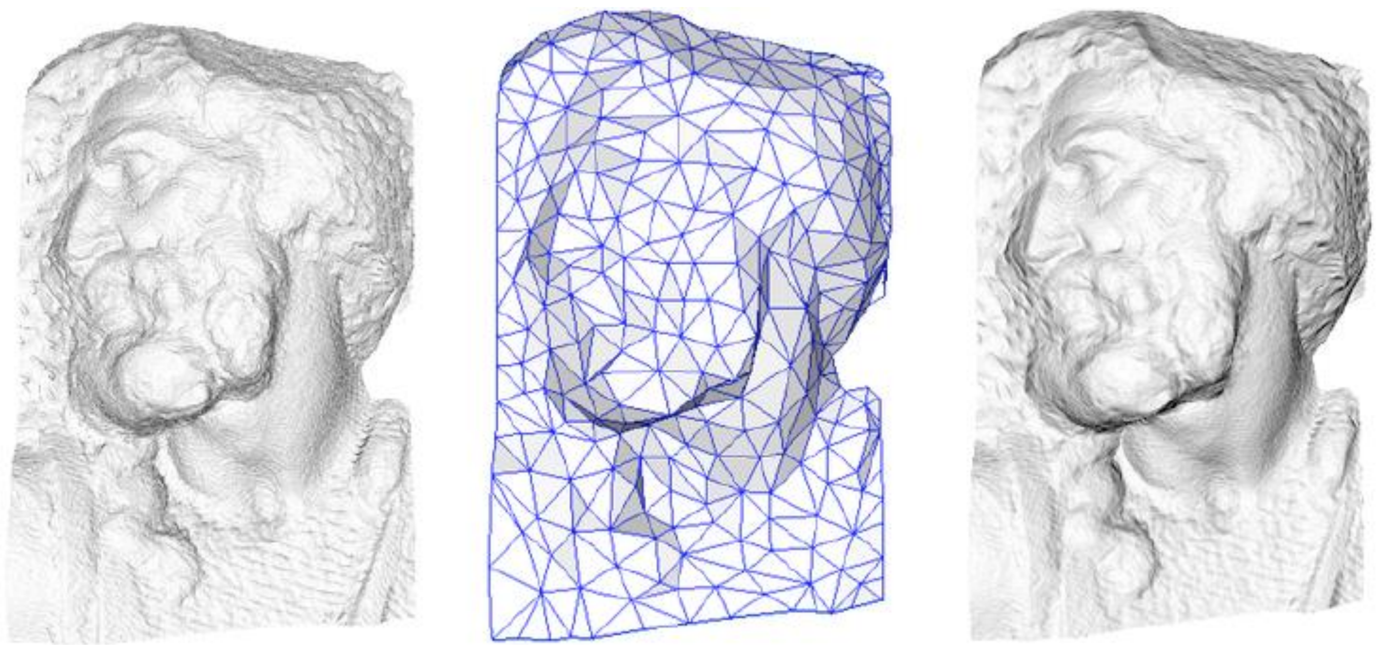
Sphere map
applied on torus

# Capturing a Sphere Map



# Normal Mapping

- Store normals in texture
- Very useful for making low-resolution geometry look like it's much more detailed



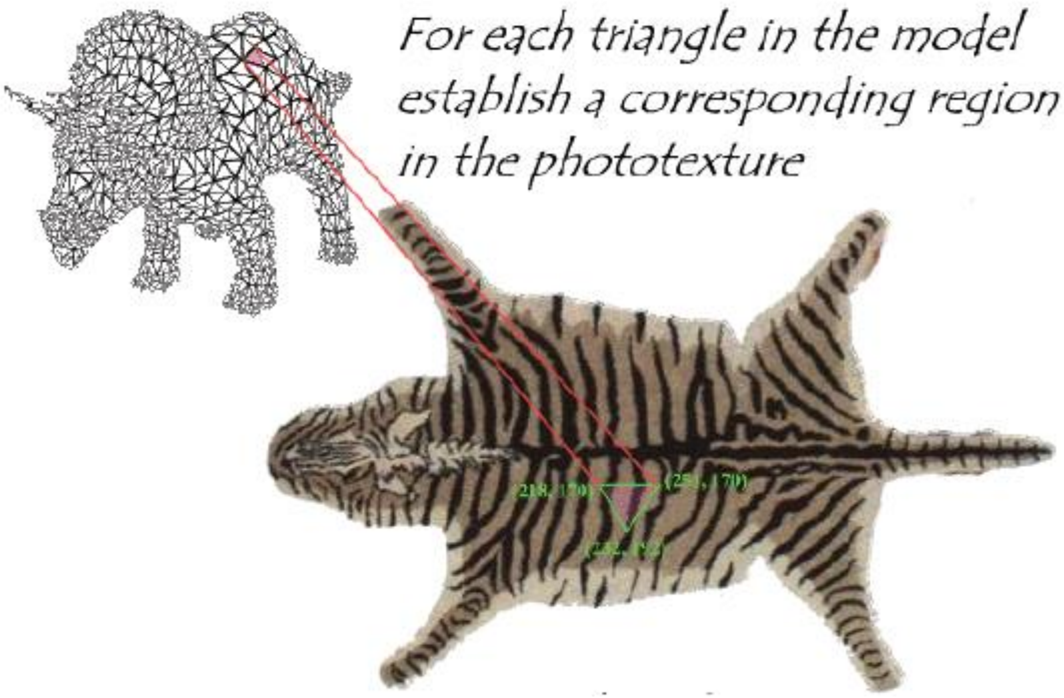
original mesh
4M triangles

simplified mesh
500 triangles

simplified mesh
and normal mapping
500 triangles

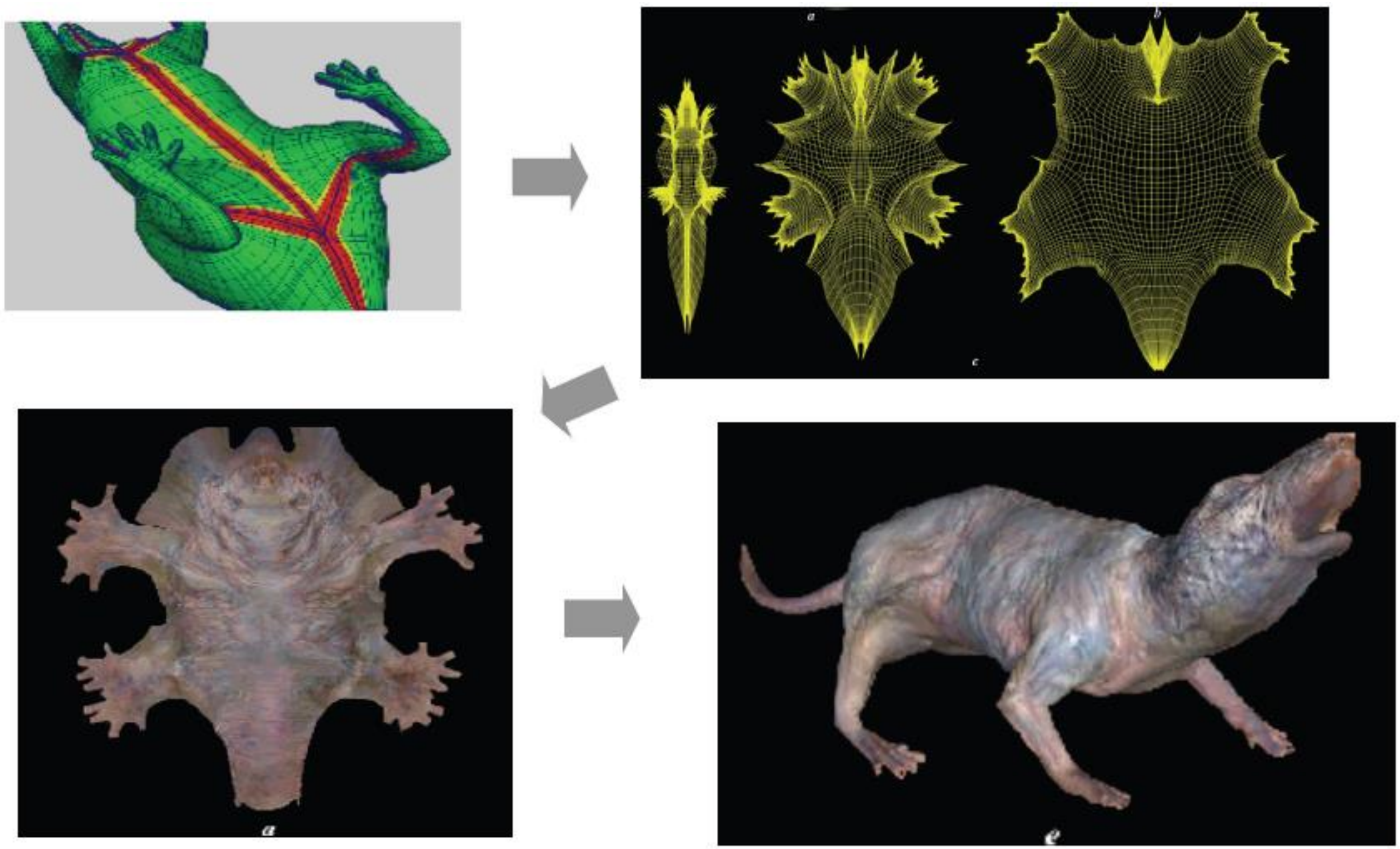
# Hot Research Topic: Parametrization

- The concept is very simple: define a mapping from the surface to the plane



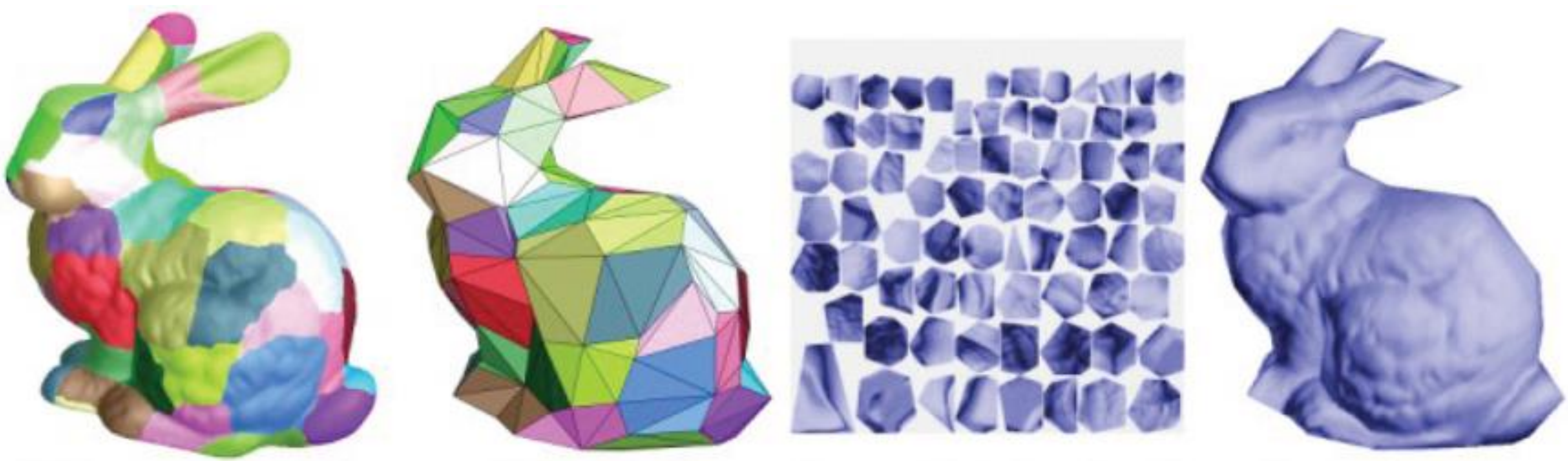
# Parametrization in Practice

- Texture creation and parametrization is an art form
- Option: Unfold the surface



# Parametrization in Practice

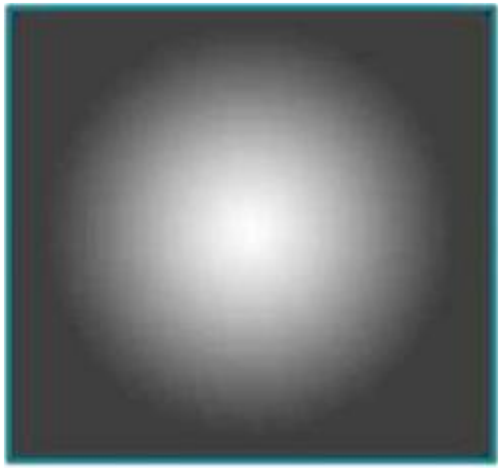
- Option: Create a Texture Atlas
- Break large mesh into smaller pieces



(a) charts on original mesh $M$ (b) base mesh $M'$ (c) texture atlas (before pull-push) (d) textured base mesh

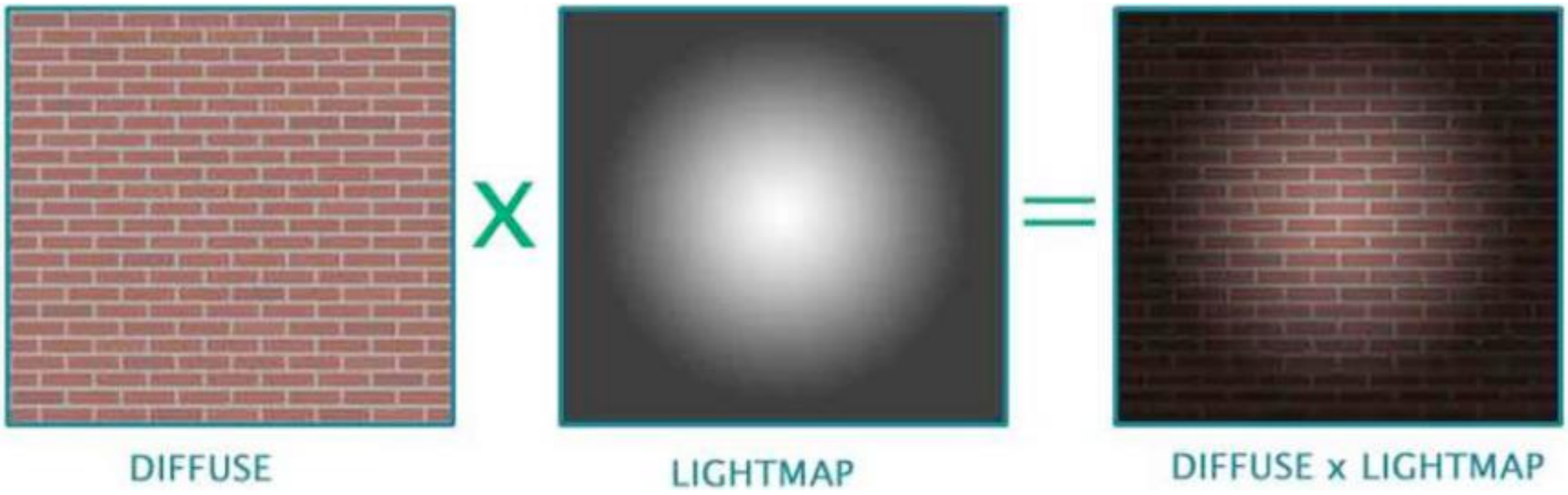
# Light Maps

- Good shadows are complicated and expensive
- If lighting and objects will not change, neither are the shadows
- Can "bake" the shadows into a texture map as a preprocess step (called **lightmap**)
- During shading, lightmap values are multiplied into resulting pixel



LIGHTMAP

# Light Maps



# Specular Mapping

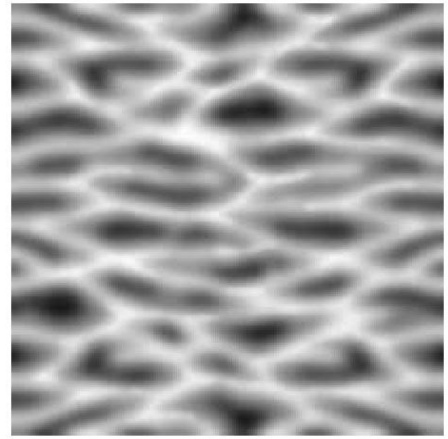
- Use a greyscale texture as a multiplier for the specular component

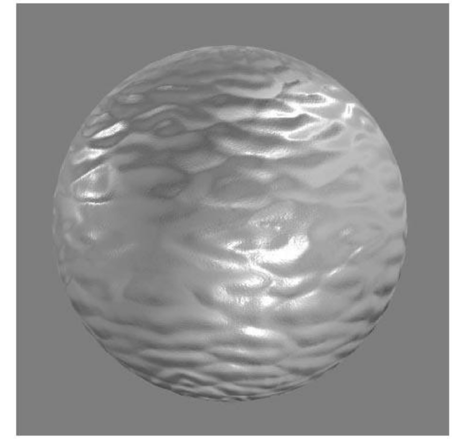
# Alpha Mapping

- Represent the alpha channel with a texture
- Can give complex outlines, used for plants

Render Bush
on 1 polygon

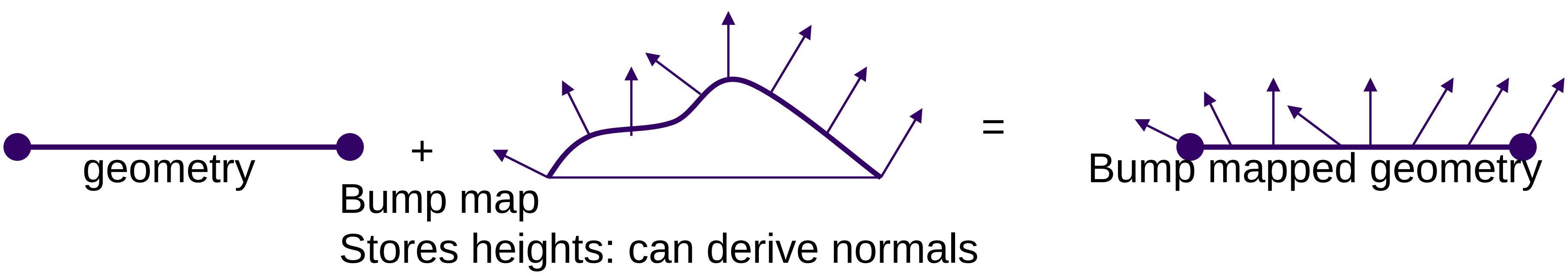
Render Bush
on polygon rotated
90 degrees



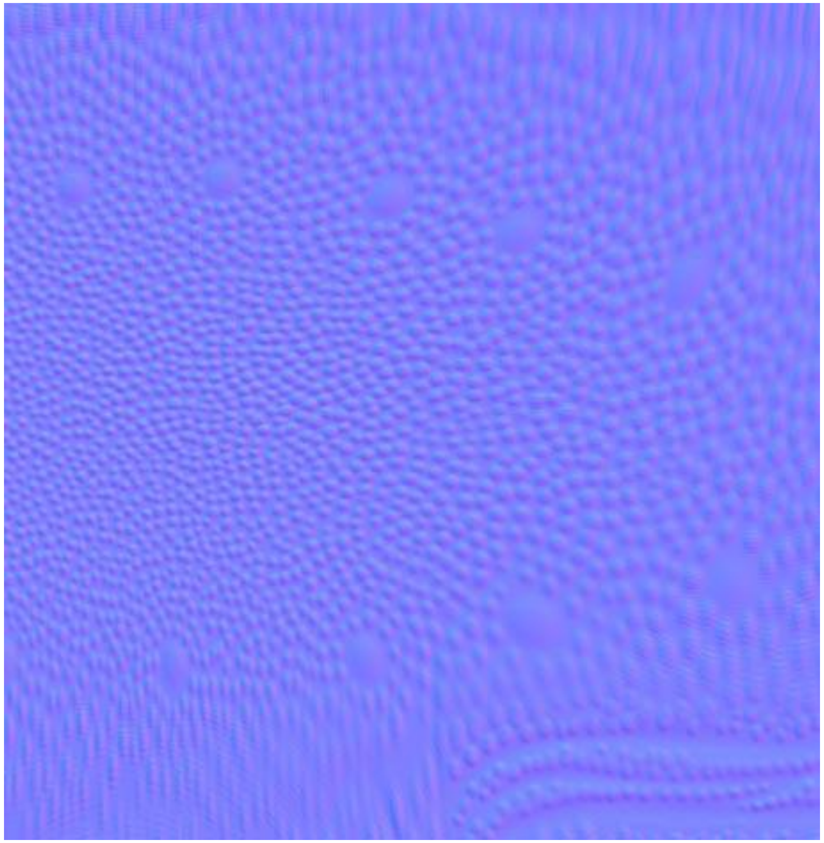


# Bump mapping

- by Blinn in 1978
- Inexpensive way of simulating wrinkles and bumps on geometry
  - Too expensive to model these geometrically
- Instead let a texture modify the normal at each pixel, and then use this normal to compute lighting

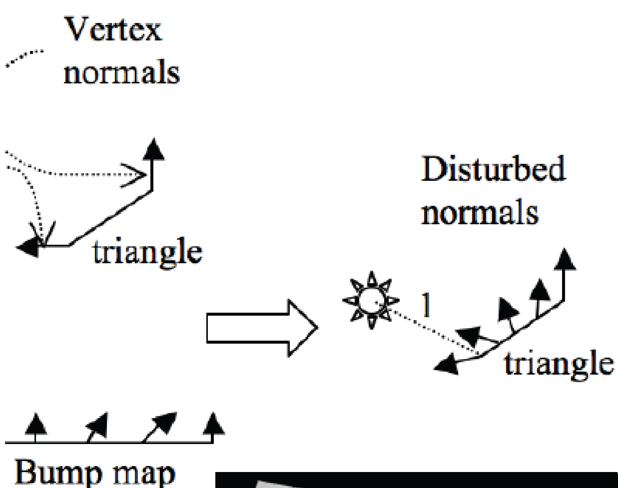


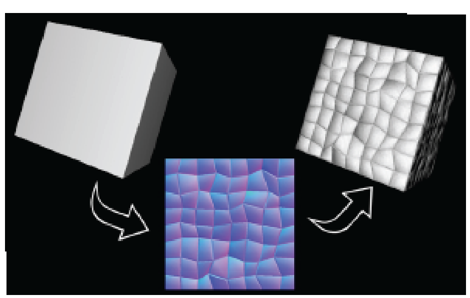
# Bump mapping: examples



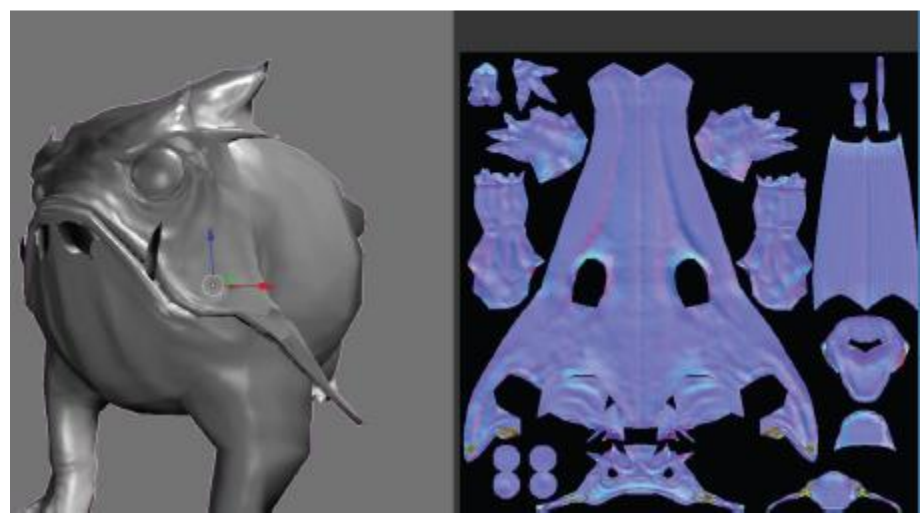
# Bump Mapping Vs Normal Mapping

- **Bump mapping**
- (Normals $\mathbf{n}=(n_x, n_y, n_z)$) stored as *distortion of face orientation*. Same bump map can be tiled/repeated and reused for many faces)



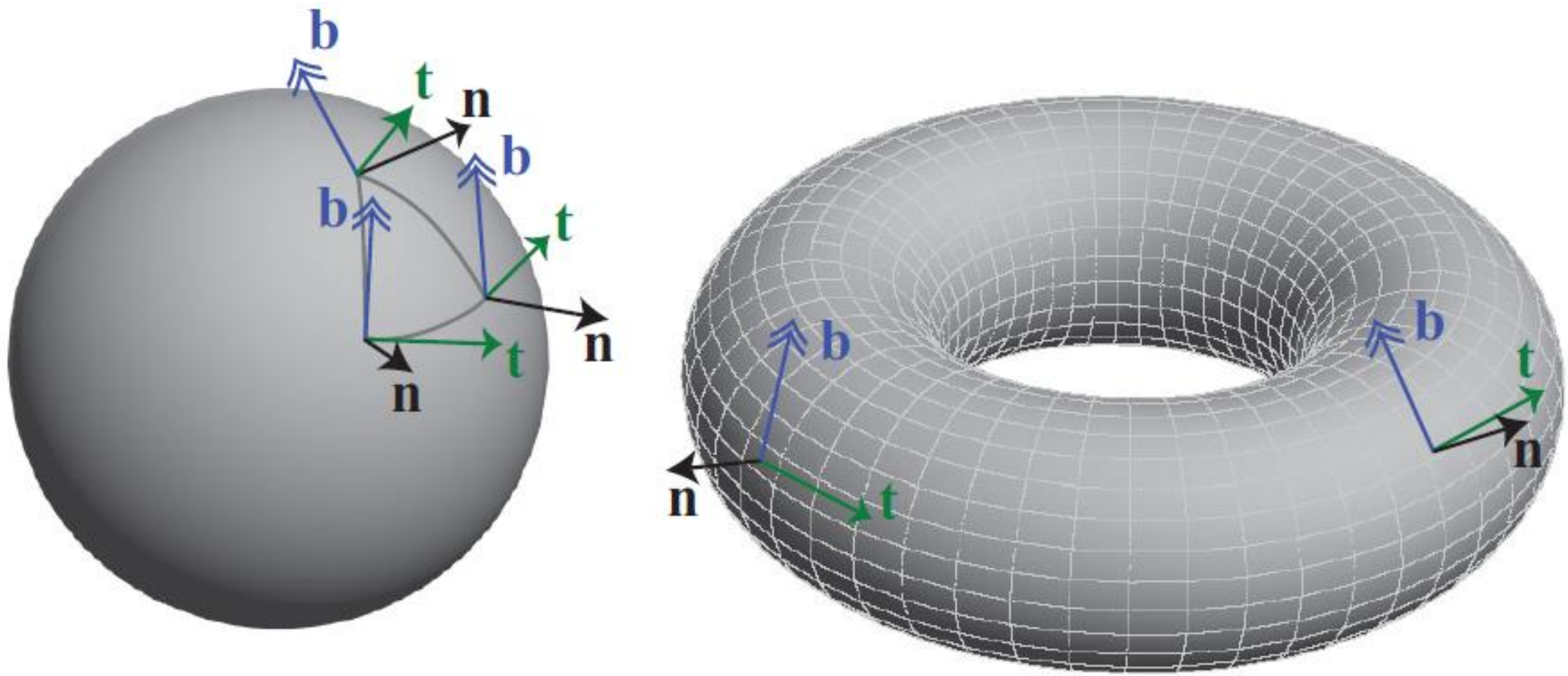



- **Normal mapping**
- Coordinates of normal (relative to tangent space) are encoded in color channels
- Normals stored include face orientation + plus distortion. )



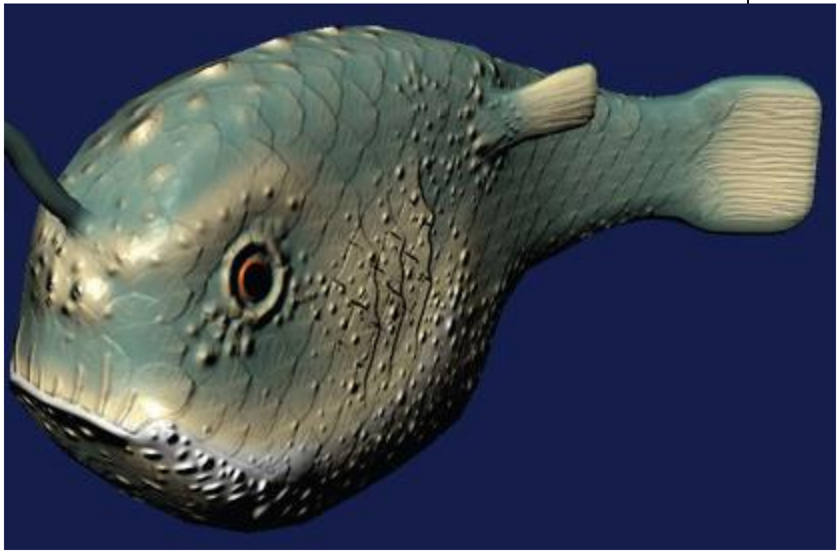
# Tangent Space Vectors

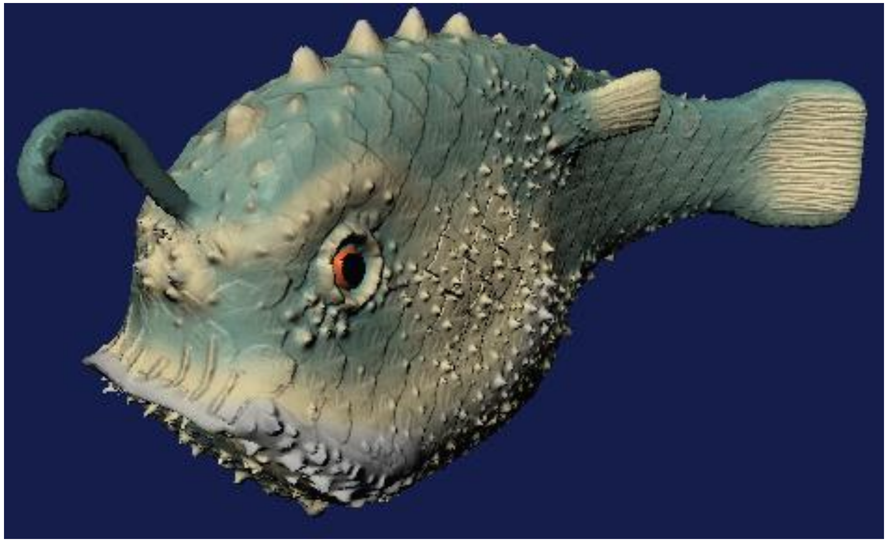
- Normals stored in local coordinate frame
- Need Tangent, normal and bi-tangent vectors



# Displacement Mapping

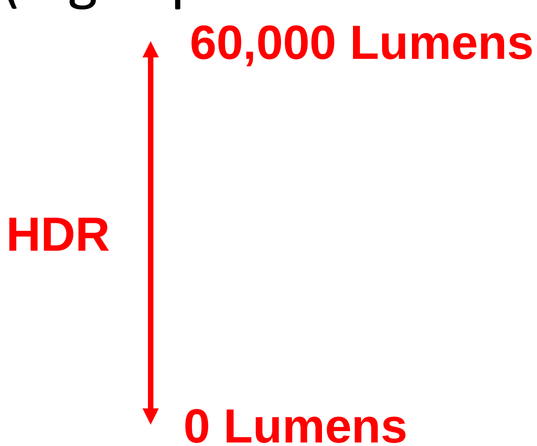
- Uses a map to displace the surface at each position
- Offsets the position per pixel or per vertex
  - Offsetting per vertex is easy in vertex shader
  - Offsetting per pixel is architecturally hard





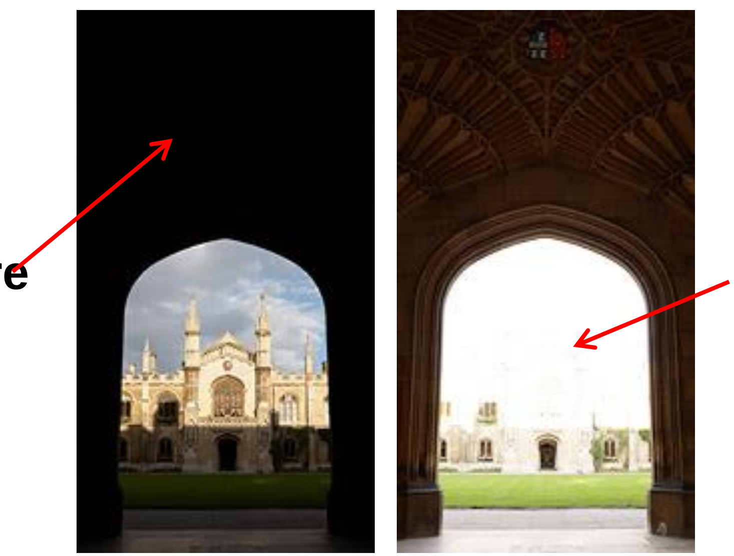
# High Dynamic Range

- Sun’s brightness is about 60,000 lumens
- Dark areas of earth has brightness of 0 lumens
- Basically, world around us has range of 0 – 60,000 lumens **(High Dynamic Range)**
- However, monitor has ranges of colors between 0 – 255 **(Low Dynamic Range)**
- New file formats have been created for HDR images (wider ranges). (E.g. OpenEXR file format)



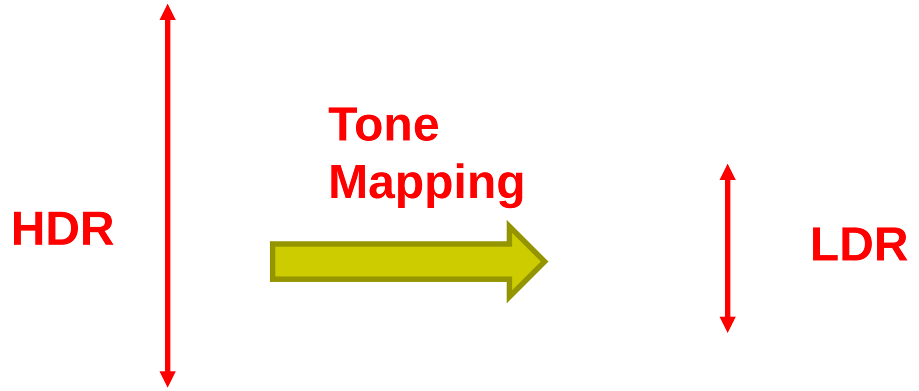
# High Dynamic Range

- Some scenes contain **very bright** + **very dark** areas
- Using uniform scaling factor to map actual intensity to displayed pixel intensity means:
  - Either some areas are unexposed, or
  - Some areas of picture are overexposed



# Tone Mapping

- Technique for scaling intensities in real world images (e.g HDR images) to fit in displayable range
- Try to capture feeling of real scene: **non-trivial**
- **Example:** If coming out of dark tunnel, lights should seem bright
- **General idea:** apply different scaling factors to diffferent parts of the image



# Tone Mapping

***Figure 10**. Scene from Lost Coast at Varying Exposure Levels*

# Types of Tone Mapping Operators

- **Global:** Use same scaling factor for all pixels
- **Local:** Use different scaling factor for different parts of image
- **Time-dependent:** Scaling factor changes over time
- **Time independent:** Scaling factor does NOT change over time
- Real-time rendering usually does **NOT** implement local operators due to their complexity

# Simple (Global) Tone Mapping Methods

Maping to mean value

Division by maximum

Clipping on value 1

Interval mapping (interactive calibration)

Exponential mapping

# Motion Blur

- Motion blur caused by exposing film to moving objects
- Motion blur: Blurring of samples taken over time (temporal)
- Makes fast moving scenes appear less jerky
- 30 fps + motion blur better than 60 fps + no motion blur

# Motion Blur

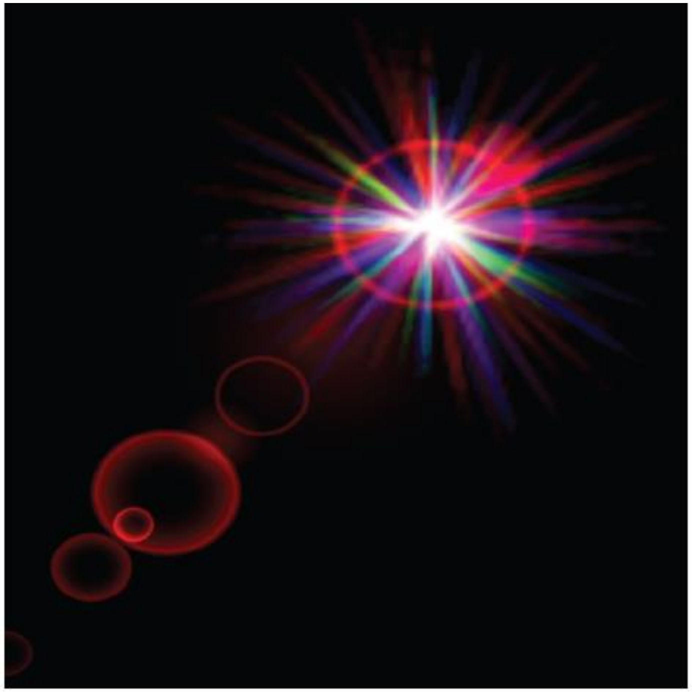
- Basic idea is to average series of images over time
- Move object to set of positions occupied in a frame, blend resulting images together
- Can blur moving average of frames. E.g blur 8 images
- **Velocity buffer:** blur in screen space using velocity of objects

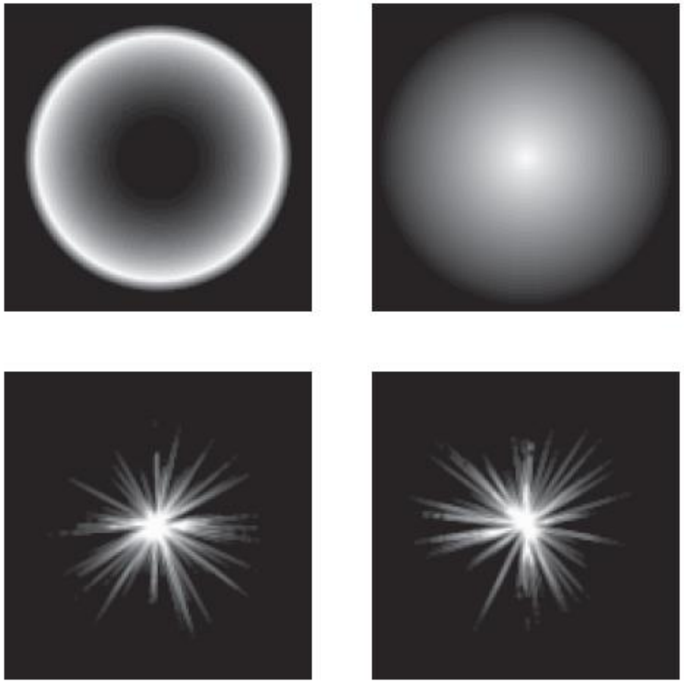
# Depth of Field

- We can simulate a real camera
- In photographs, a range of pixels in focus
- Pixels outside this range are out of focus
- This effect is known as **Depth of field**

# Lens Flare and Bloom

- Caused by lens of eye/camera when directed at light
- Halo – refraction of light by lens
- Ciliary Corona – Density fluctuations of lens
- Bloom – Scattering in lens, glow around light





Halo, Bloom, Ciliary Corona – top to bottom

# Lens Flare and Bloom

- Use set of textures for glare effects
- Each texture is bill boarded
- Alpha map – how much to blend
- Can be given colors for corona
- Overlap all of them !
- Animate – create sparkle

# Reference

- Tomas Akenine-Moller, Eric Haines and Naty Hoffman, Real Time Rendering