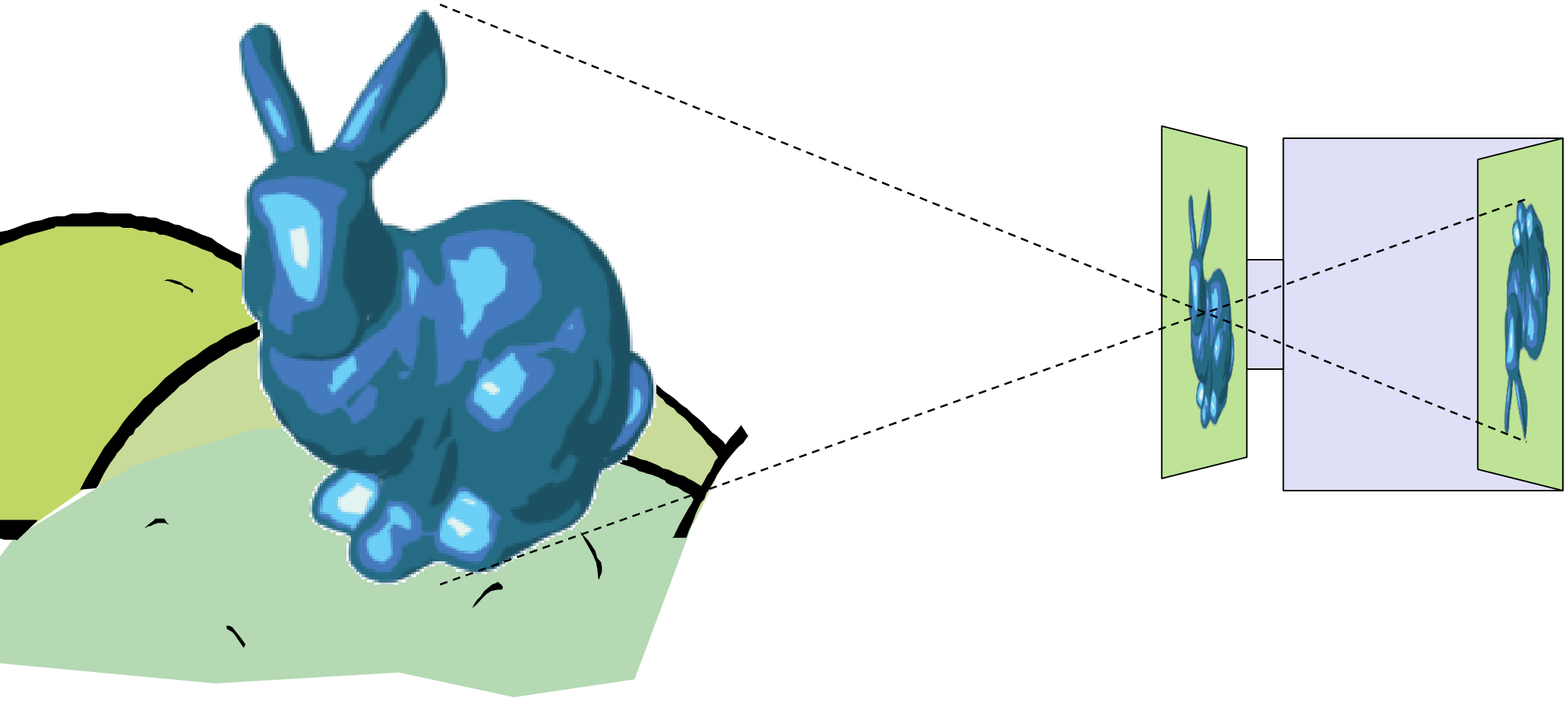
Image Formation

CS418 Computer Graphics

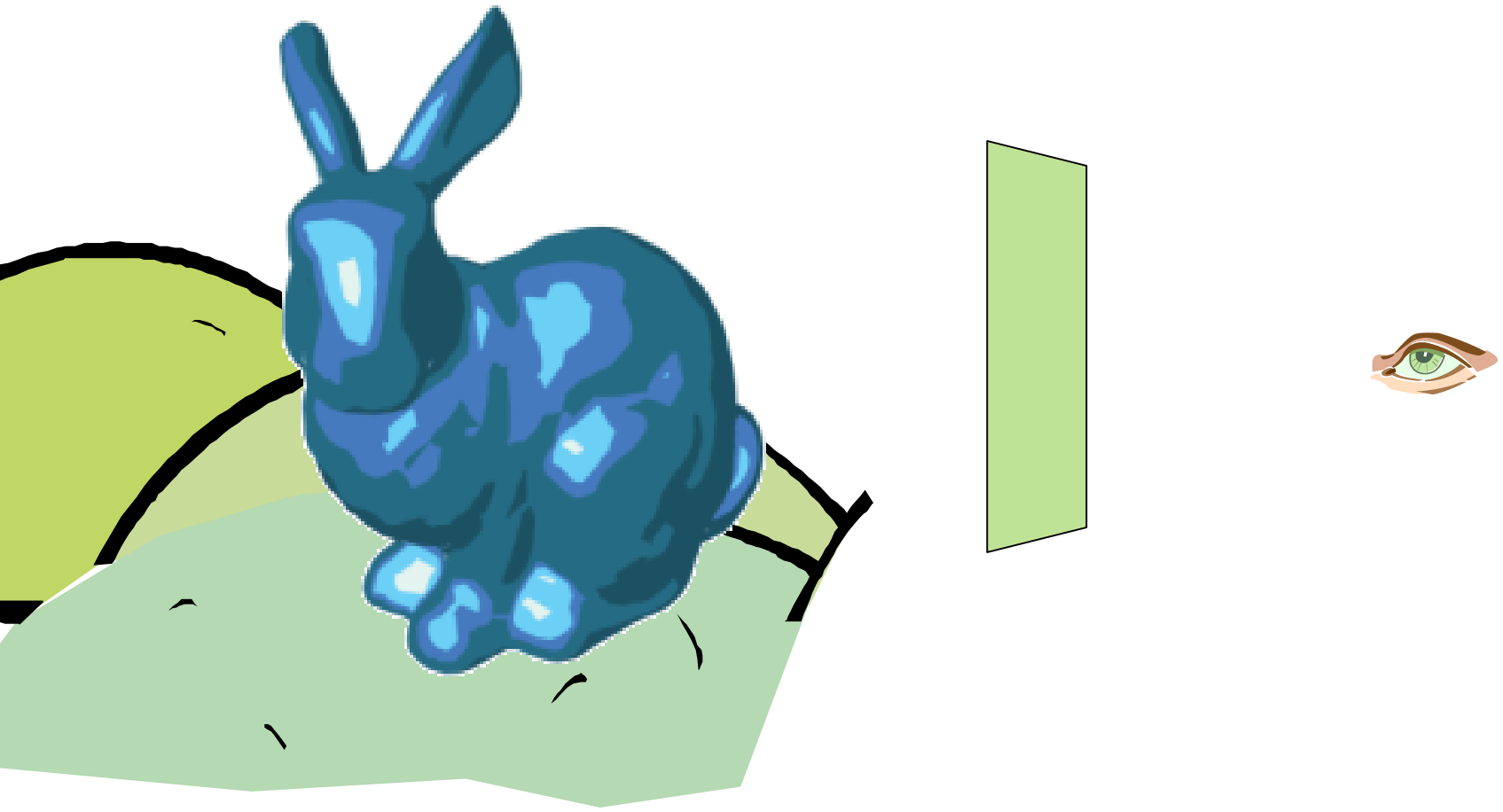
John C. Hart

The Camera



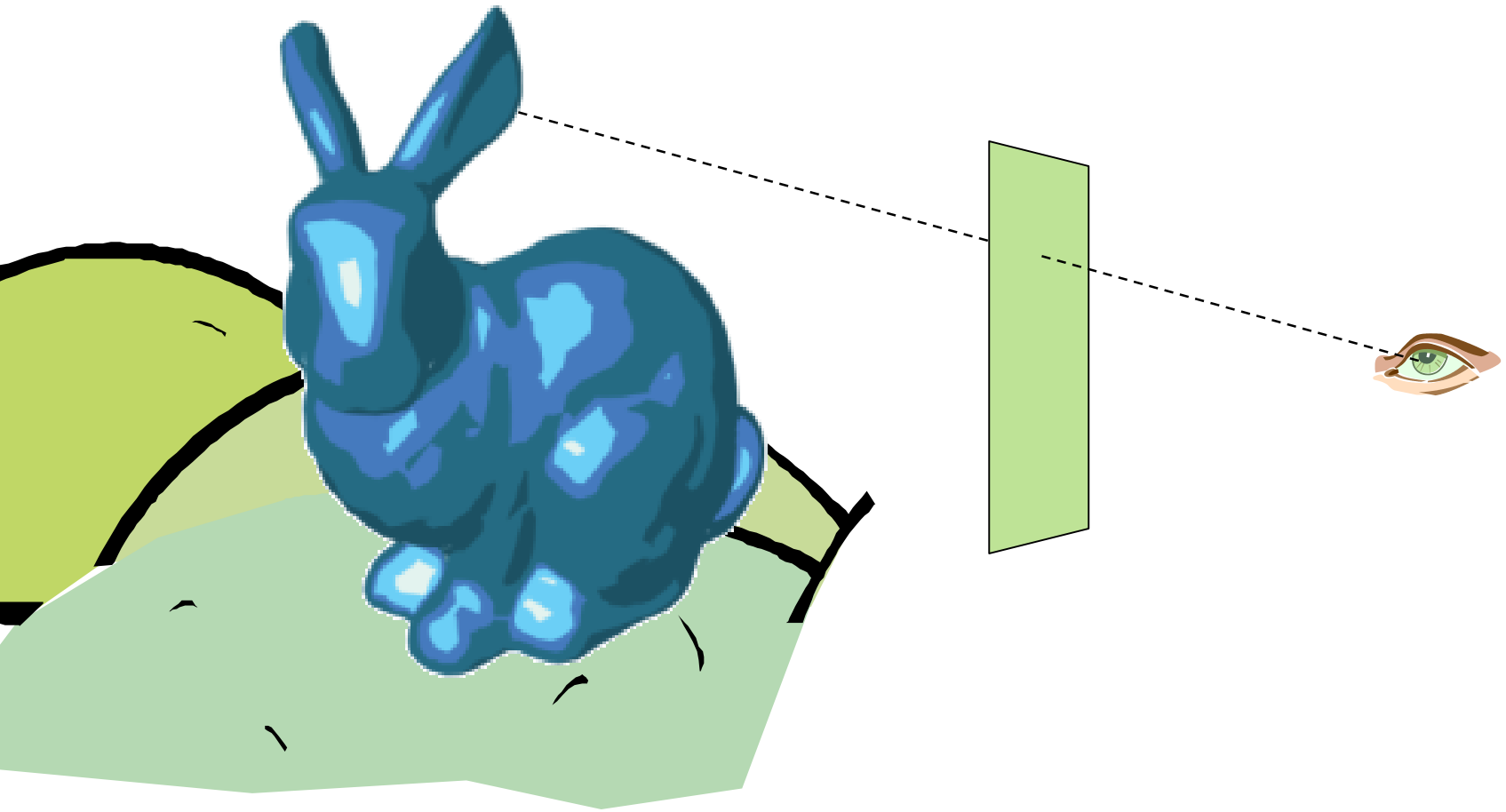
The Image Plane

1. Place an image plane (window) in front of the eye.



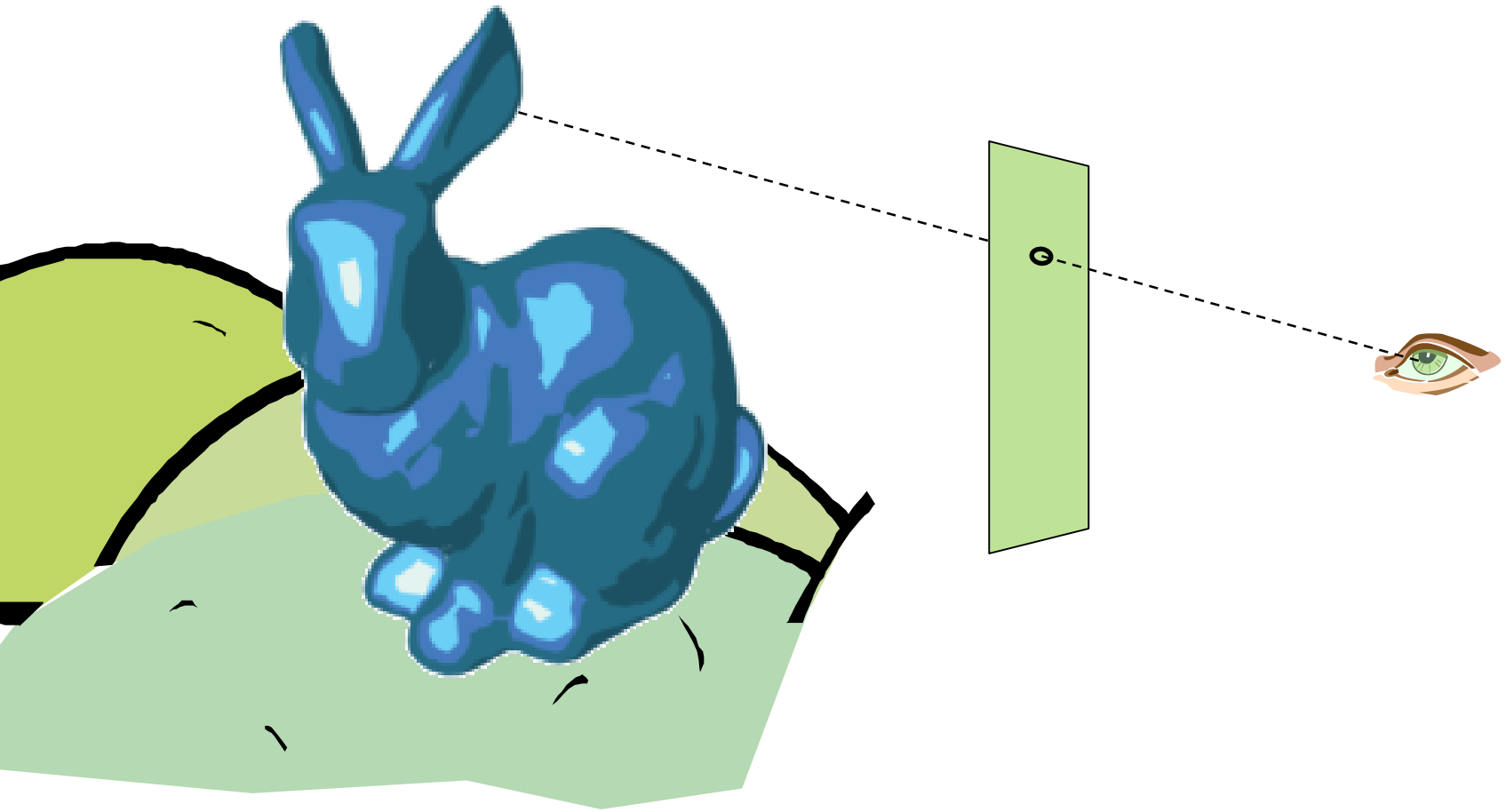
The Image Plane

1. Place an image plane (window) in front of the eye.
2. Draw a line from each point in the scene to the eye (pupil).



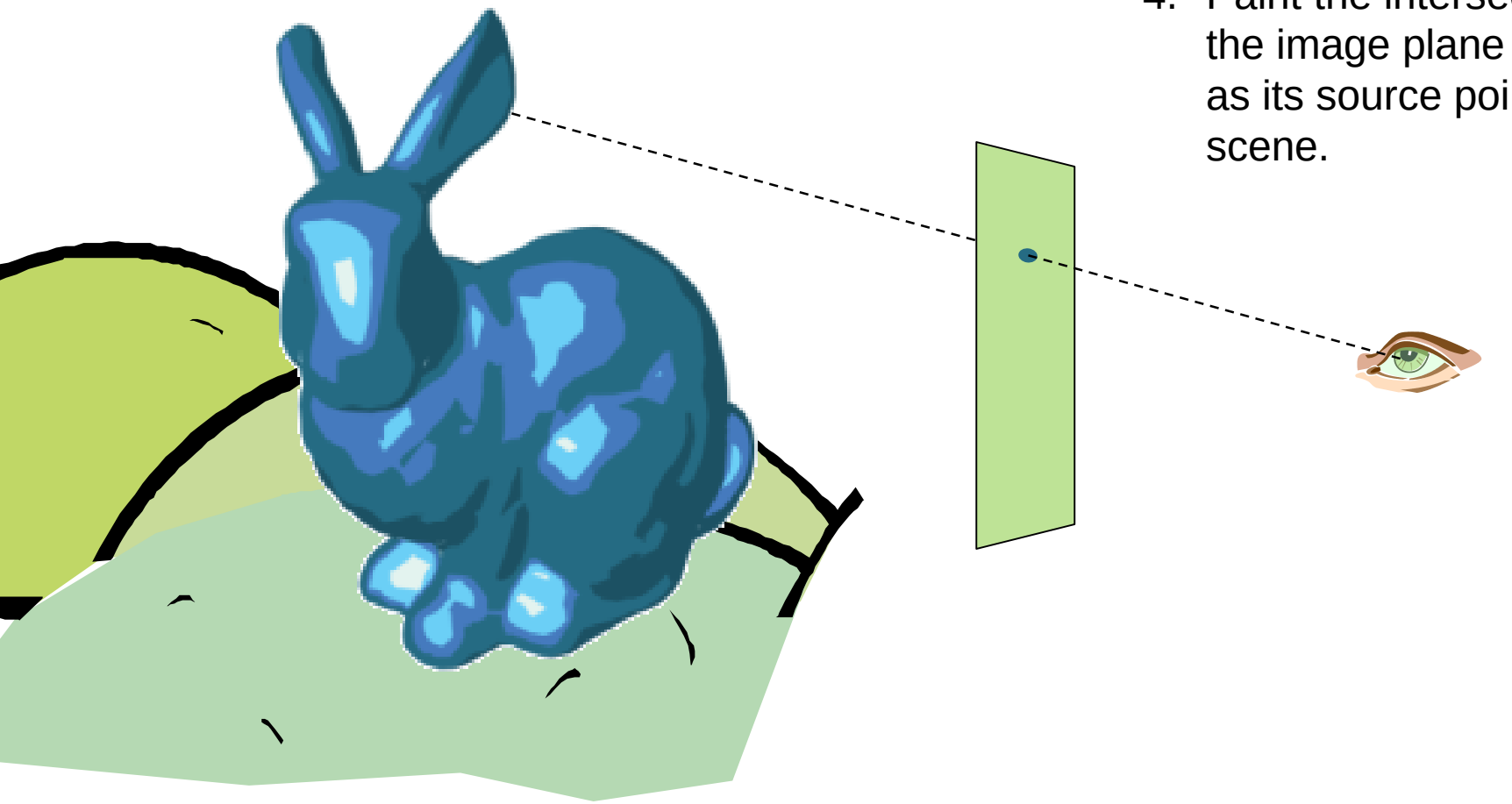
The Image Plane

1. Place an image plane (window) in front of the eye.
2. Draw a line from each point in the scene to the eye (pupil).
3. Find the intersection of this line with the image plane.



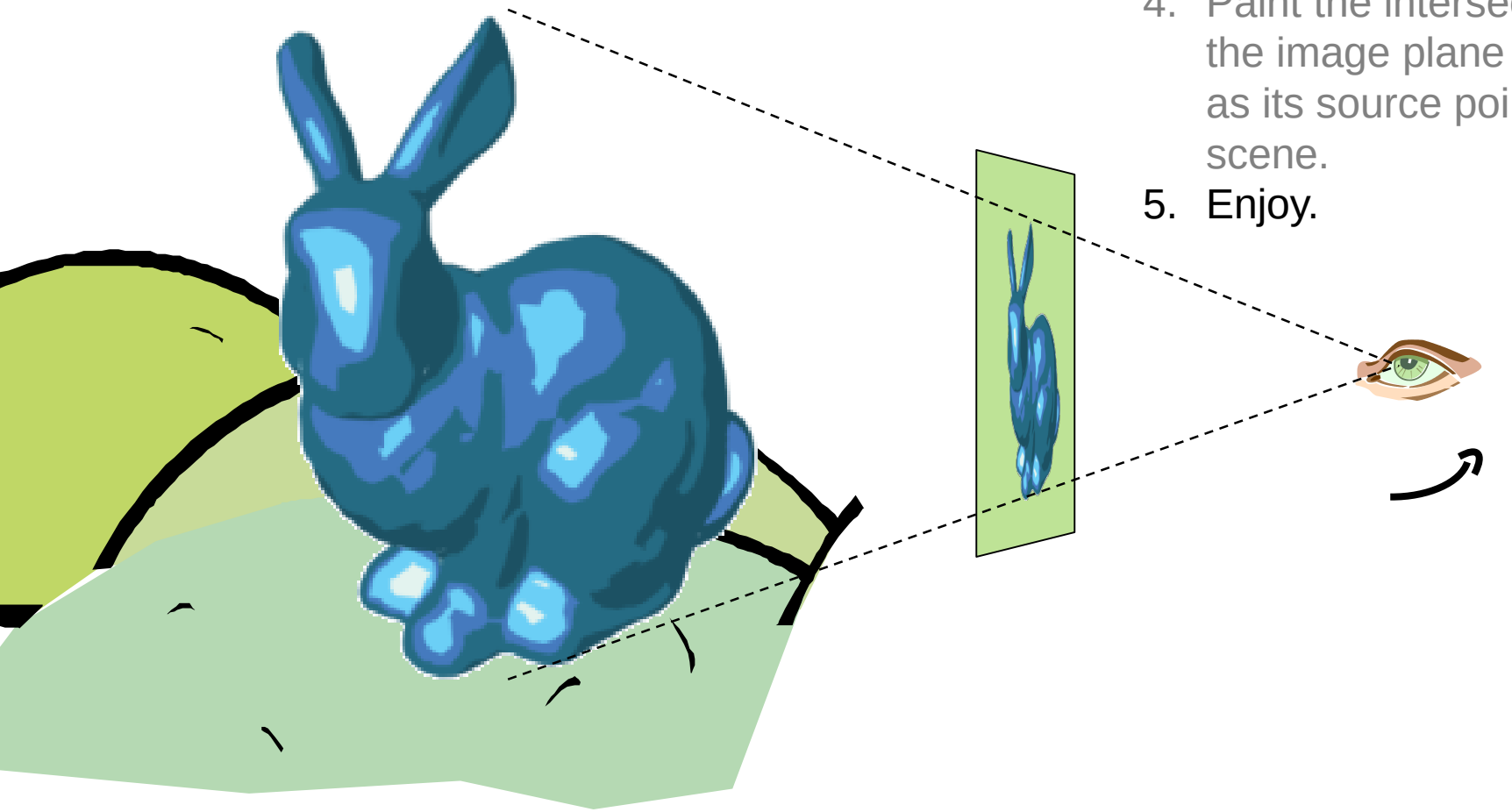
The Image Plane

1. Place an image plane (window) in front of the eye.
2. Draw a line from each point in the scene to the eye (pupil).
3. Find the intersection of this line with the image plane.
4. Paint the intersection point on the image plane the same color as its source point in the scene.



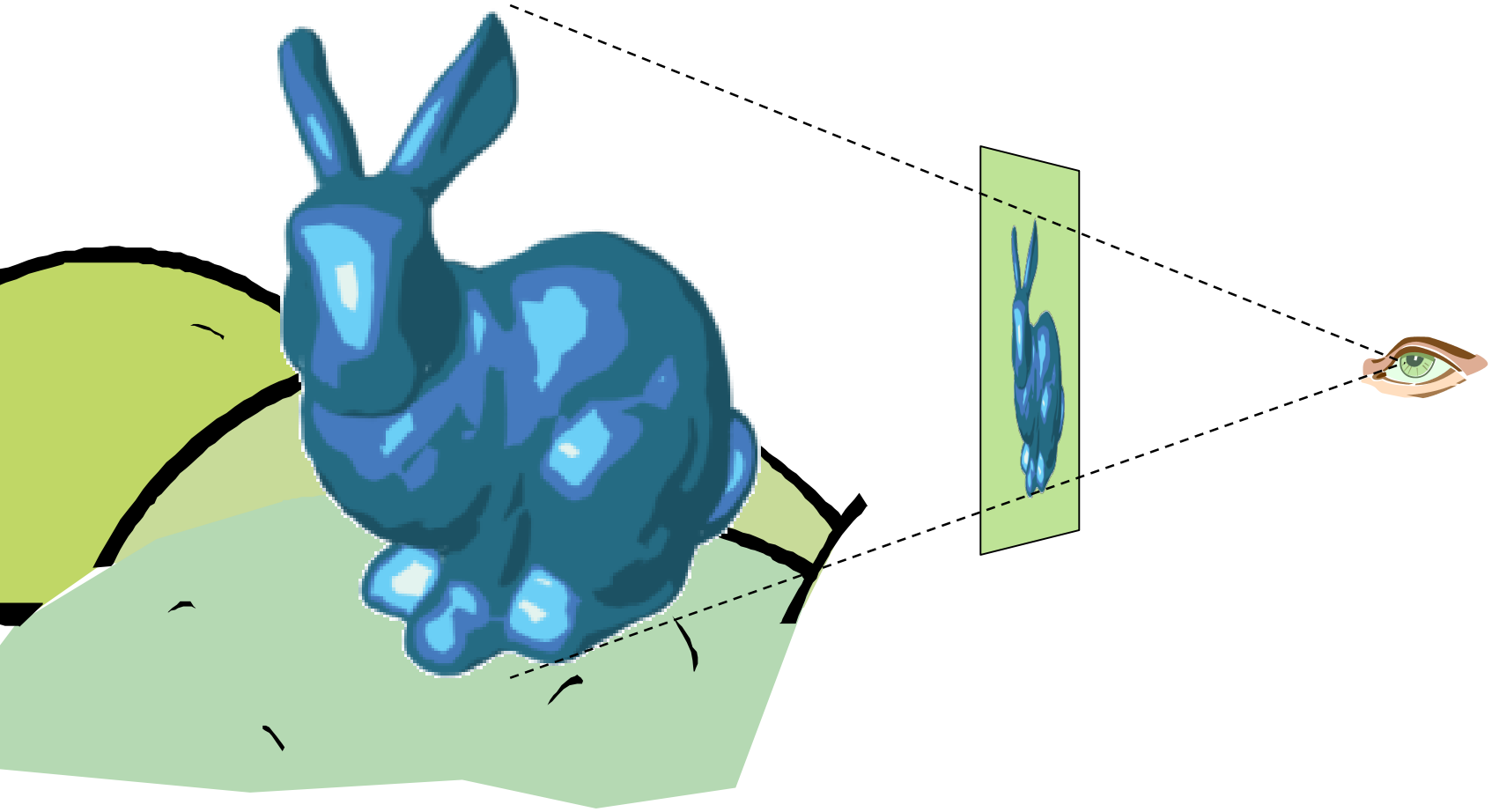
The Image Plane

1. Place an image plane (window) in front of the eye.
2. Draw a line from each point in the scene to the eye (pupil).
3. Find the intersection of this line with the image plane.
4. Paint the intersection point on the image plane the same color as its source point in the scene.
5. Enjoy.



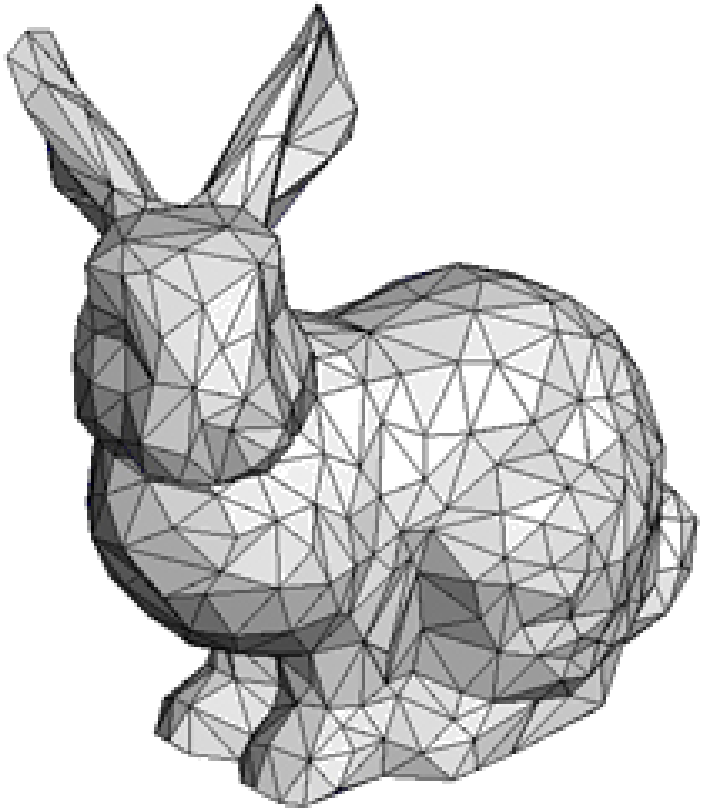
Polygonal Models

- Need a data structure to represent the scene in computer memory



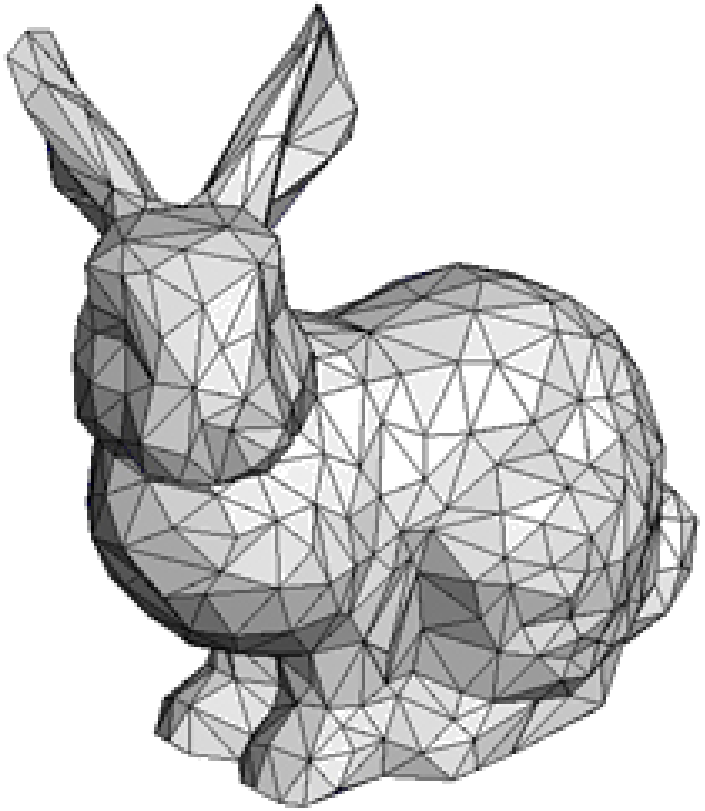
Polygonal Models

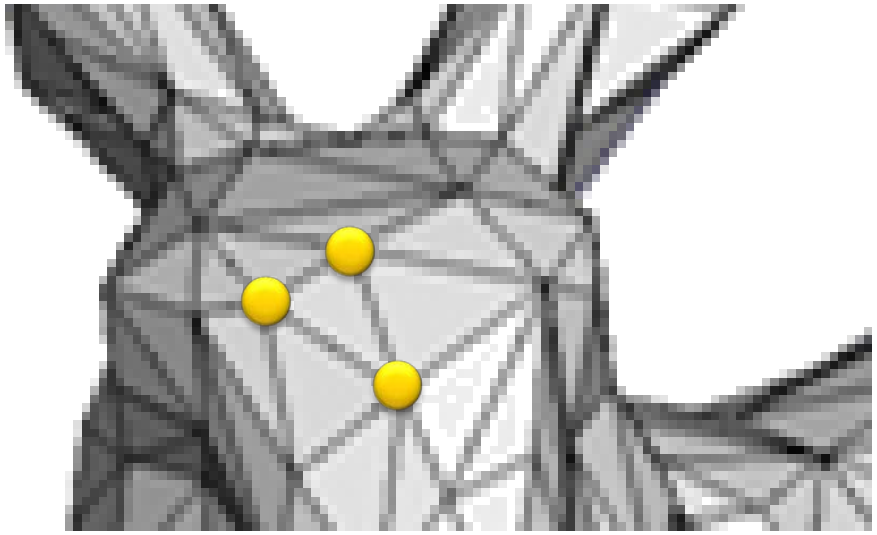
- Need a data structure to represent the scene in computer memory
- Most common data structure is a surface mesh of triangles



Polygonal Models

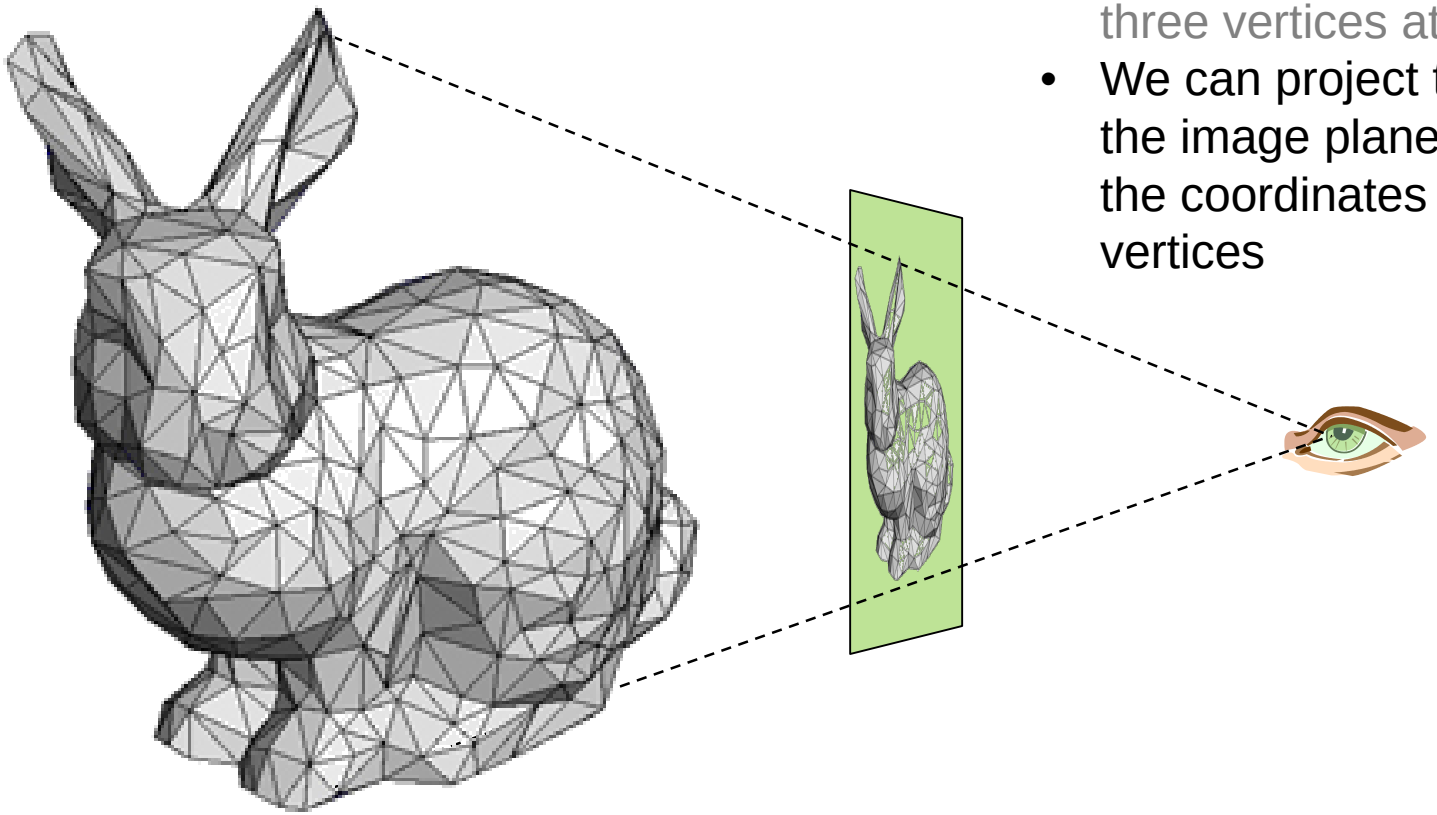
- Need a data structure to represent the scene in computer memory
- Most common data structure is a surface mesh of triangles
- Each triangle is represented by three vertices at its corners





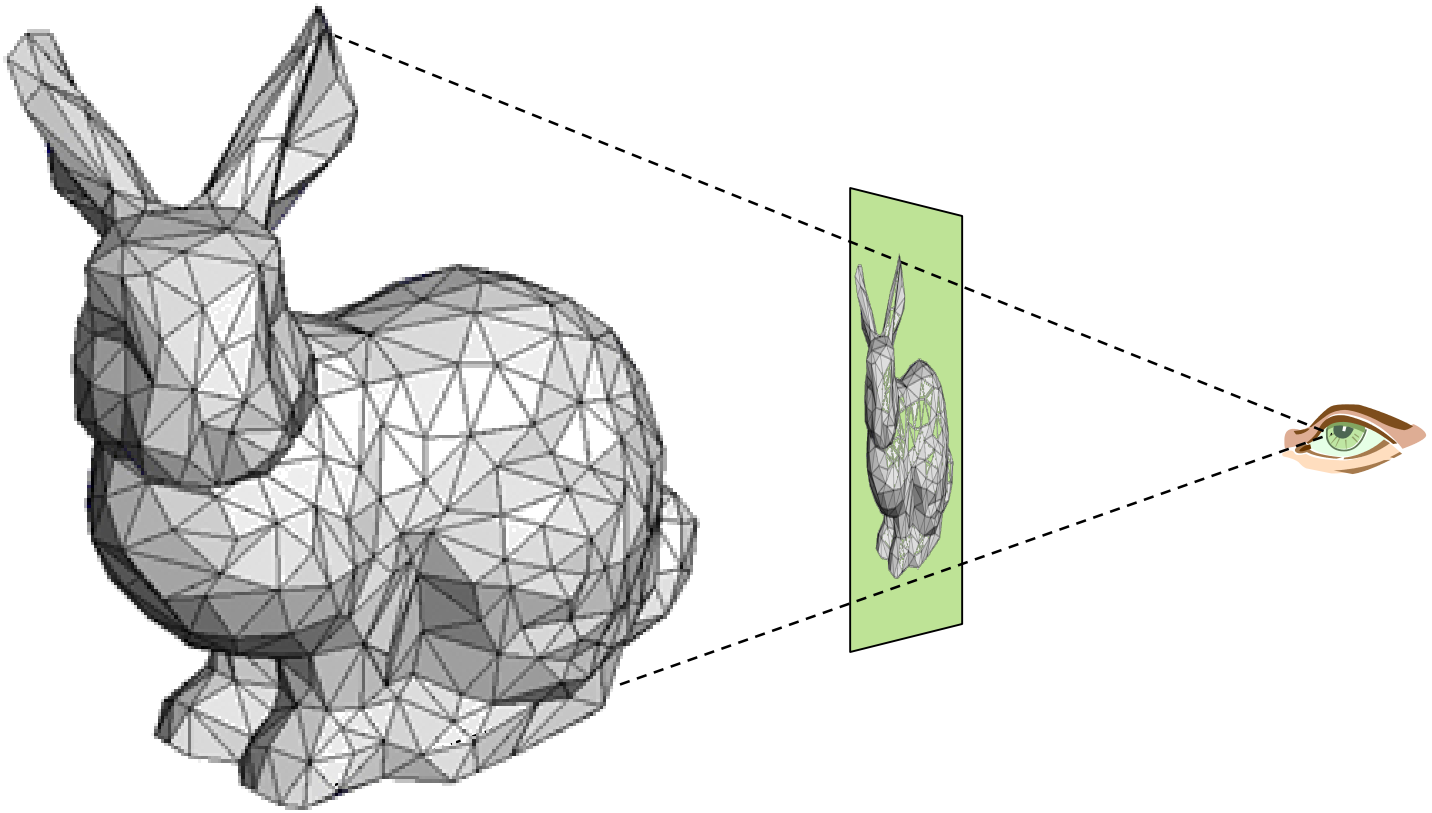
Polygonal Models

- Need a data structure to represent the scene in computer memory
- Most common data structure is a surface mesh of triangles
- Each triangle is represented by three vertices at its corners
- We can project the mesh onto the image plane by changing the coordinates of these vertices



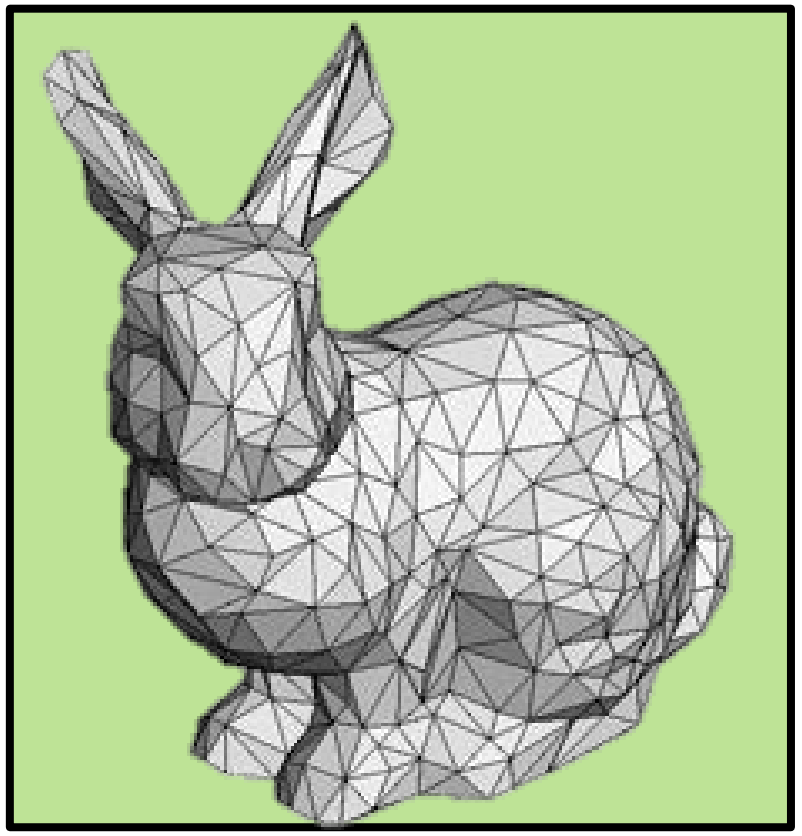
Pixel Discretization

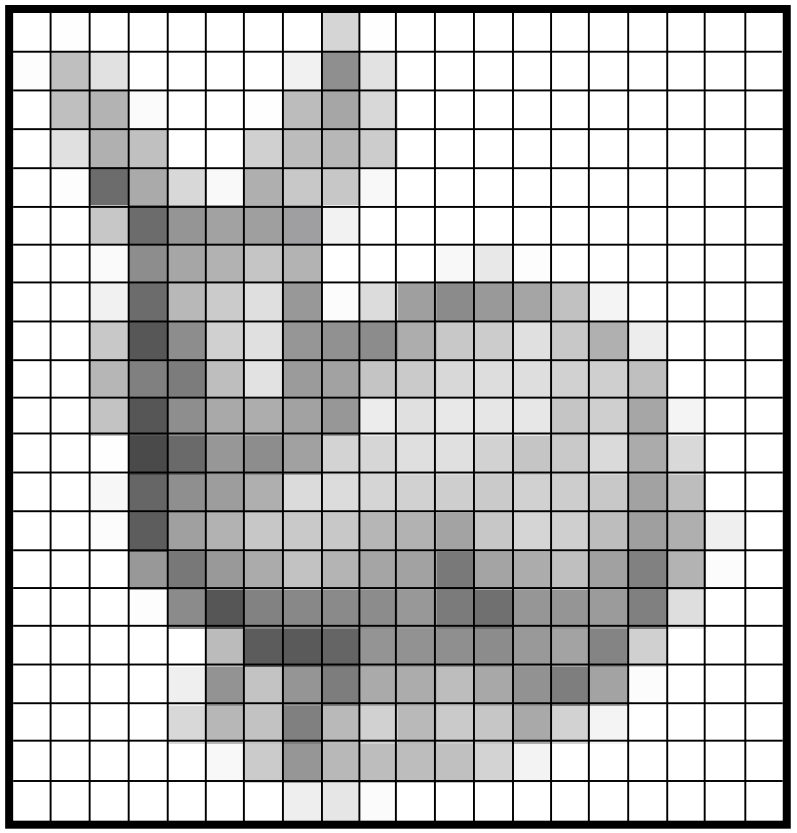
- Need a data structure to represent the image in computer memory



Pixel Discretization

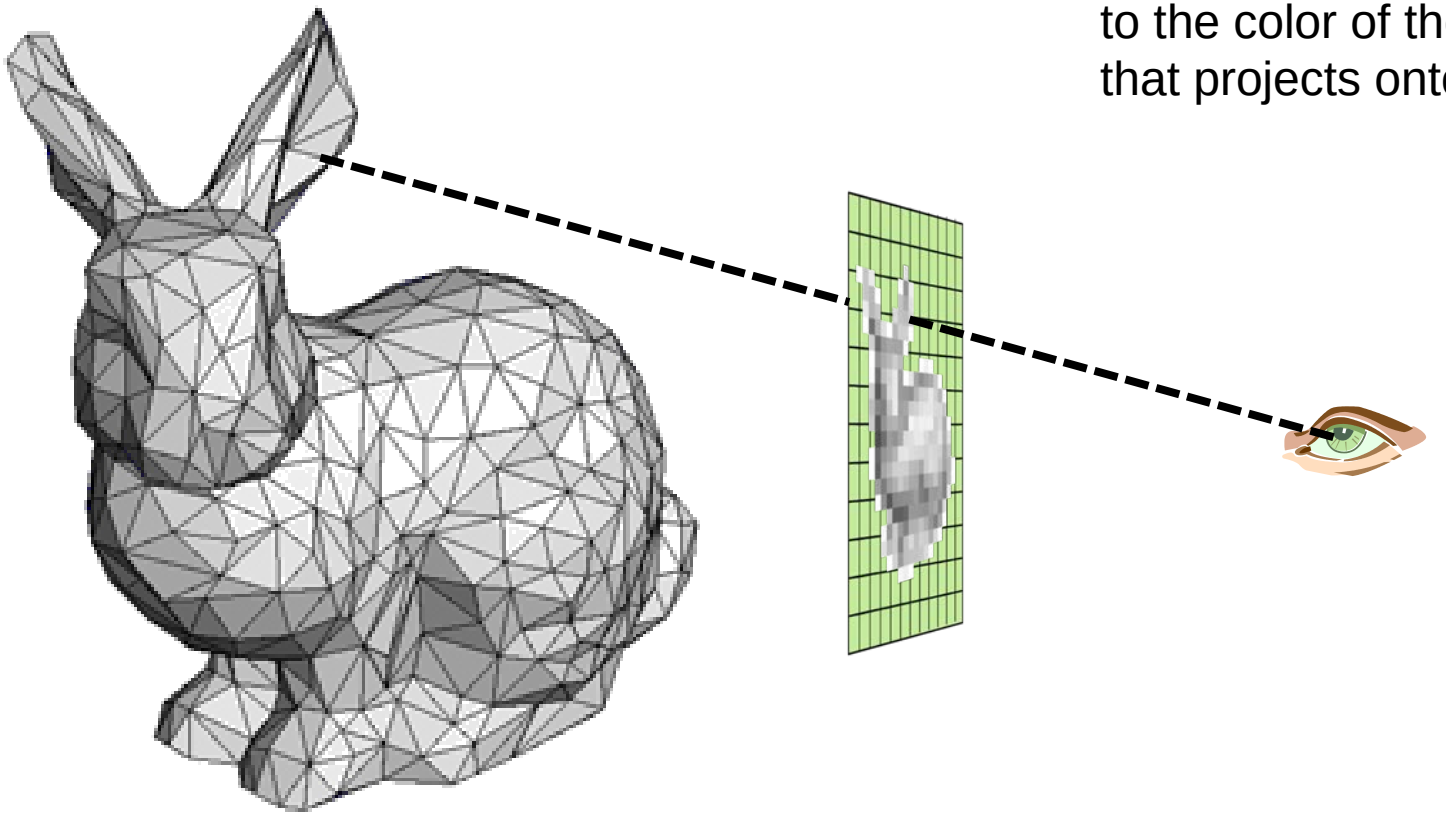
- Need a data structure to represent the image in computer memory
- Most common representation is as an array of pixels





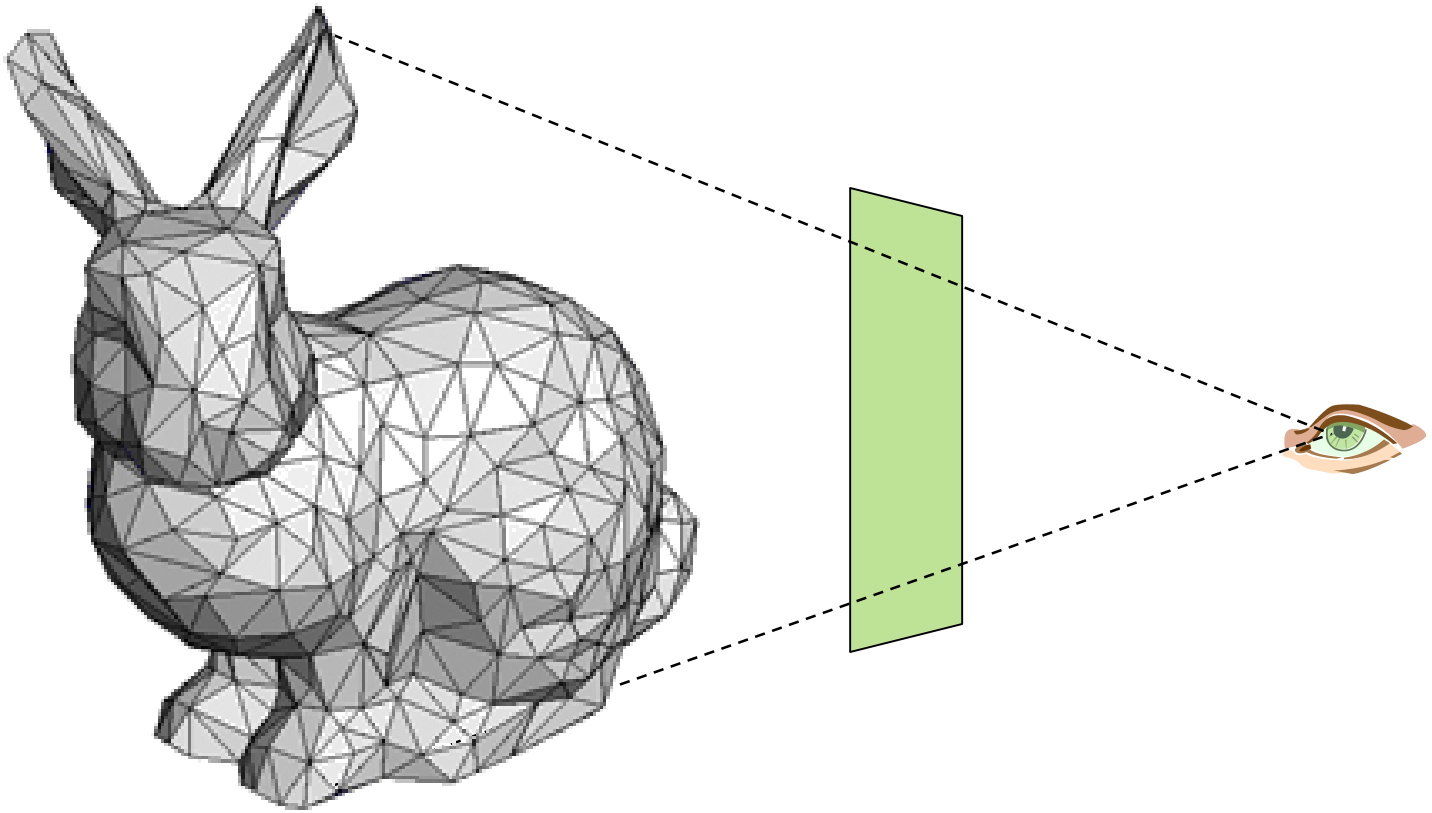
Pixel Discretization

- Need a data structure to represent the image in computer memory
- Most common representation is as an array of pixels
- Each pixel's color set according to the color of the mesh triangle that projects onto the pixel



Raster Rendering

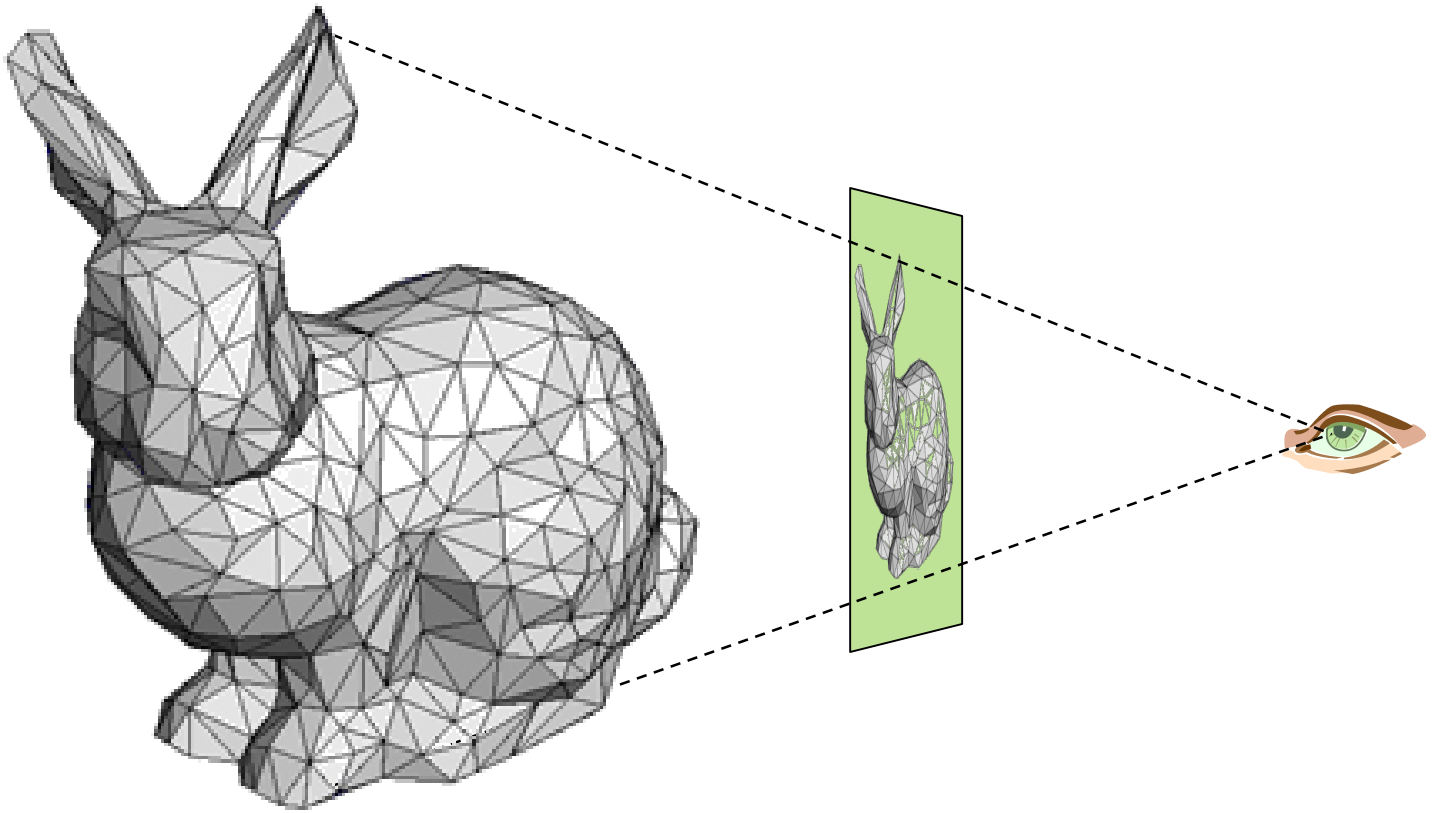
For each triangle in the scene
mesh representation:



Raster Rendering

For each triangle in the scene mesh representation:

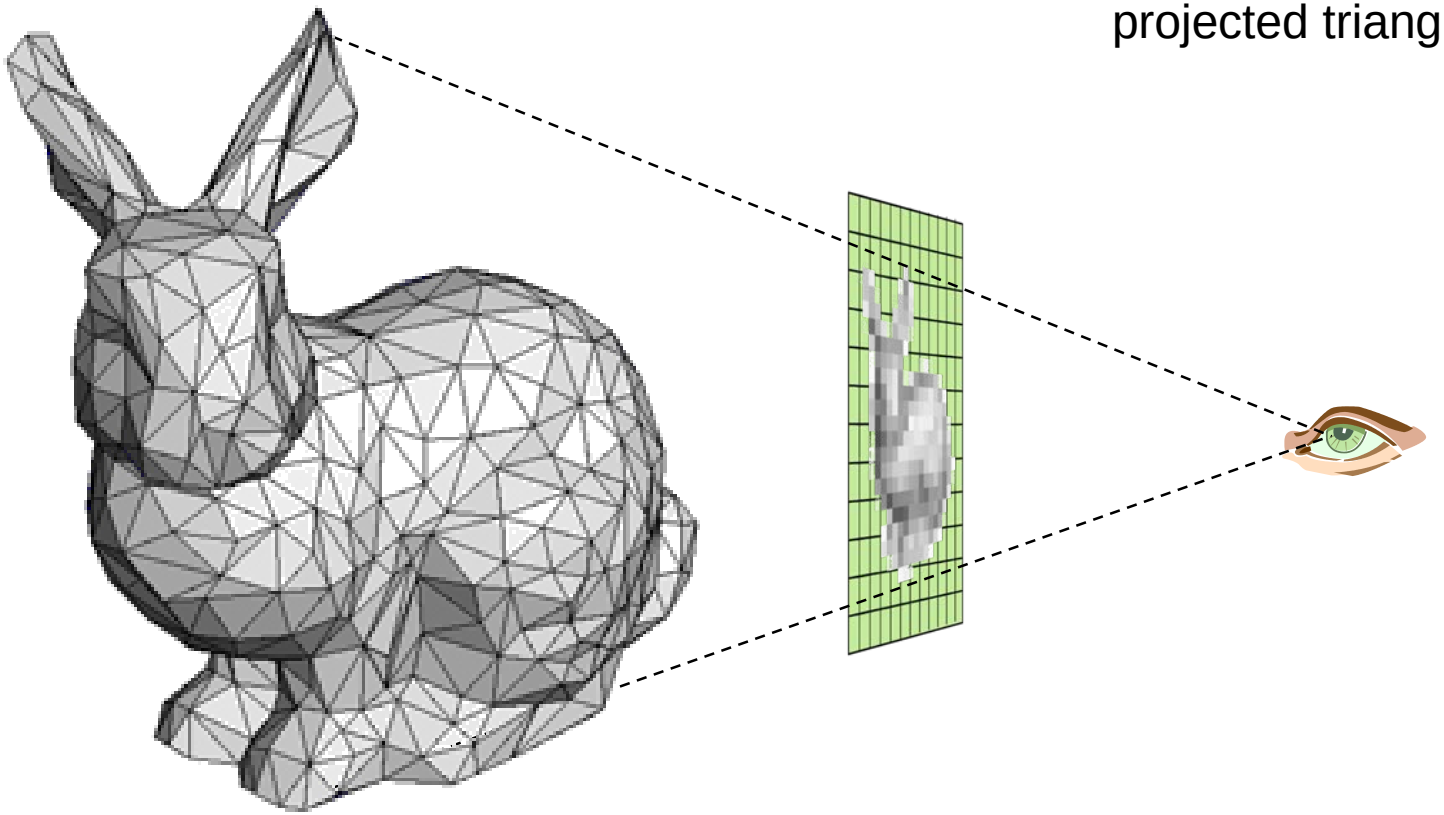
1. Project the triangle onto the image plane (by computing the projection of its vertices)



Raster Rendering

For each triangle in the scene mesh representation:

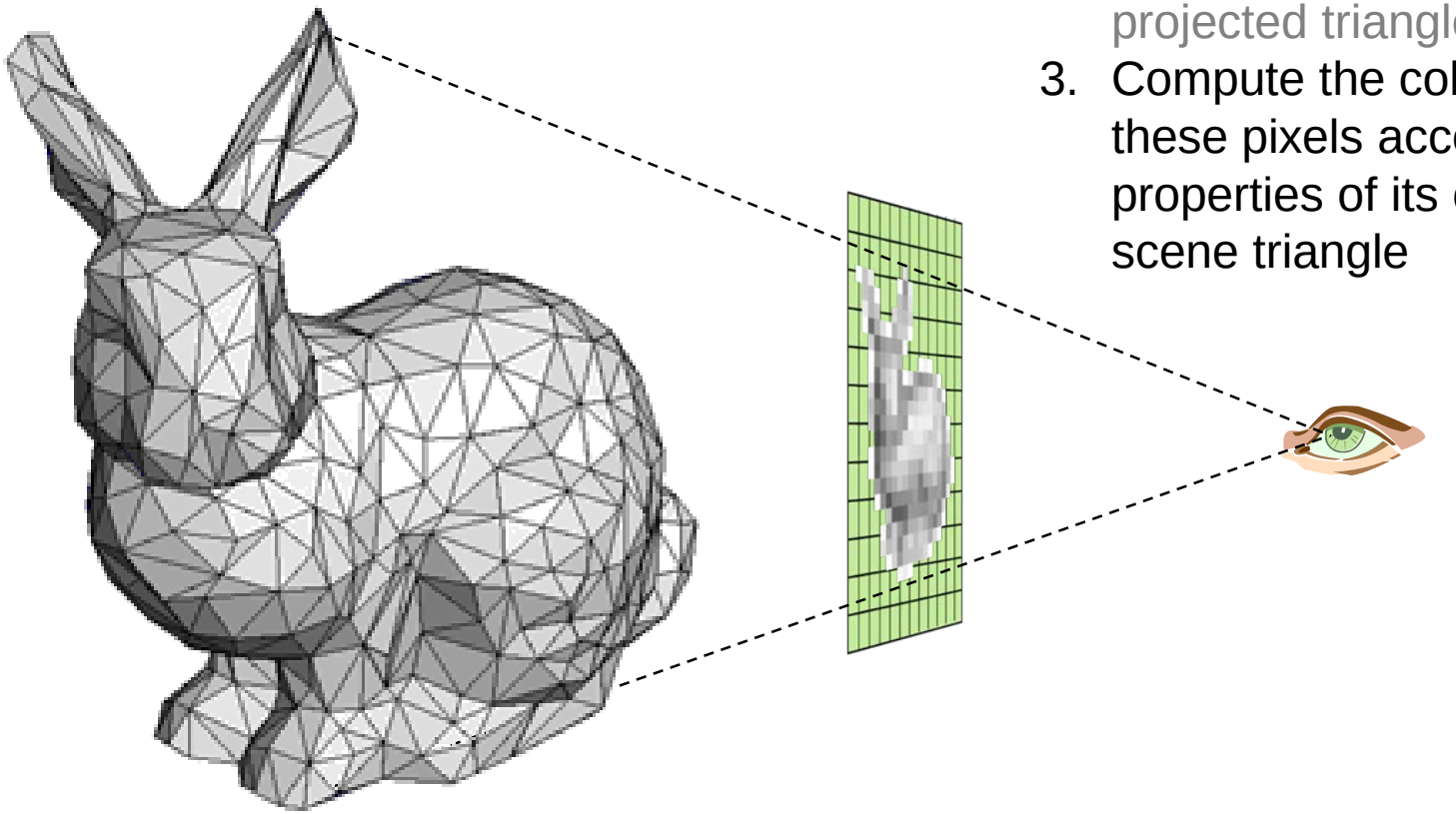
1. Project the triangle onto the image plane (by computing the projection of its vertices)
2. Find the pixels covered by the projected triangle



Raster Rendering

For each triangle in the scene mesh representation:

1. Project the triangle onto the image plane (by computing the projection of its vertices)
2. Find the pixels covered by the projected triangle
3. Compute the color each of these pixels according to the properties of its corresponding scene triangle



Raster Rendering

For each triangle in the scene mesh representation:

Raster Rendering

For each triangle in the scene mesh representation:

1. Project the triangle onto the image plane (by computing the projection of its vertices)

Raster Rendering

For each triangle in the scene mesh representation:

1. Project the triangle onto the image plane (by computing the projection of its vertices)

2. Find the pixels covered by the projected triangle

Raster Rendering

For each triangle in the scene mesh representation:

1. Project the triangle onto the image plane (by computing the projection of its vertices)

2. Find the pixels covered by the projected triangle

3. Compute the color each of these pixels according to the properties of its corresponding scene triangle

Raster Rendering

For each triangle in the scene mesh representation:

1. Project the triangle onto the image plane (by computing the projection of its vertices)

2. Find the pixels covered by the projected triangle

3. Compute the color each of these pixels according to the properties of its corresponding scene triangle

Vertex Processing

Raster Rendering

For each triangle in the scene mesh representation:

1. Project the triangle onto the image plane (by computing the projection of its vertices)

2. Find the pixels covered by the projected triangle

3. Compute the color each of these pixels according to the properties of its corresponding scene triangle

Vertex Processing

Scan Conversion

Raster Rendering

For each triangle in the scene mesh representation:

1. Project the triangle onto the image plane (by computing the projection of its vertices)

Vertex Processing

2. Find the pixels covered by the projected triangle

Scan Conversion

3. Compute the color each of these pixels according to the properties of its corresponding scene triangle

Pixel Processing

Raster Rendering

For each triangle in the scene mesh representation:

1. Project the triangle onto the image plane (by computing the projection of its vertices)

2. Find the pixels covered by the projected triangle

3. Compute the color each of these pixels according to the properties of its corresponding scene triangle

First third of class
Exam 1

Raster Rendering

For each triangle in the scene mesh representation:

1. Project the triangle onto the image plane (by computing the projection of its vertices)

First third of class
Exam 1

2. Find the pixels covered by the projected triangle
3. Compute the color each of these pixels according to the properties of its corresponding scene triangle

Middle third of class
Exam 2

Raster Rendering

For each triangle in the **scene mesh** representation:

1. Project the triangle onto the image plane (by computing the projection of its vertices)
2. Find the pixels covered by the projected triangle
3. Compute the color each of these pixels according to the properties of its corresponding scene triangle

Final third of class
Exam 3

First third of class
Exam 1

Middle third of class
Exam 2