XIAMEN UNIVERSITY

COMPUTER GRAPHICS

OpenGL Texture Mapping

Dr. Zhonggui Chen
School of Informatics, Xiamen University

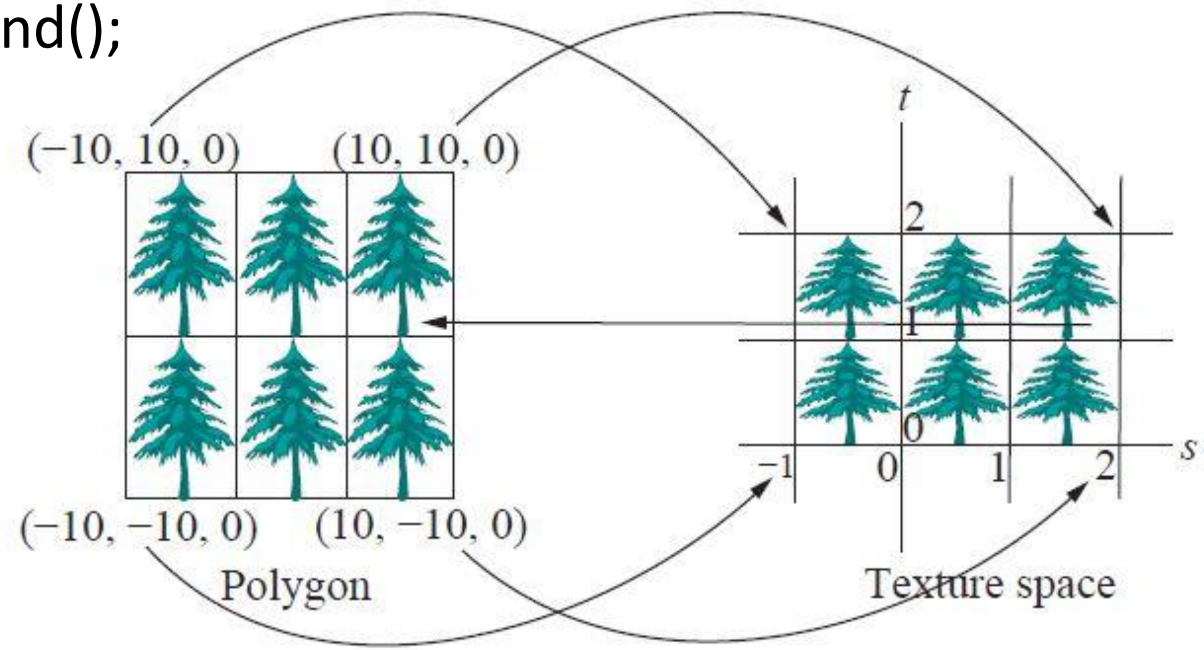
http://graphics.xmu.edu.cn

Texture Parameters

- OpenGL has a variety of parameters that determine how texture is applied
  - Wrapping parameters determine what happens if texture coordinates are outside the (0,1) range
  - Filter modes allow us to use area averaging instead of point samples
  - Mipmapping allows us to use textures at multiple resolutions
  - Environment parameters determine how texture mapping interacts with shading

Wrapping Mode

```
glBegin(GL POLYGON);
glTexCoord2f(-1.0, 0.0); glVertex3f(-10.0, -10.0, 0.0);
glTexCoord2f(2.0, 0.0); glVertex3f(10.0, -10.0, 0.0);
glTexCoord2f(2.0, 2.0); glVertex3f(10.0, 10.0, 0.0);
glTexCoord2f(-1.0, 2.0); glVertex3f(-10.0, 10.0, 0.0);
glEnd();
```

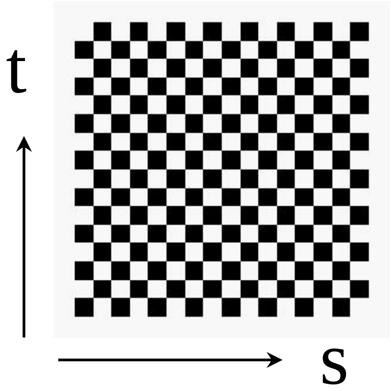


Wrapping Mode

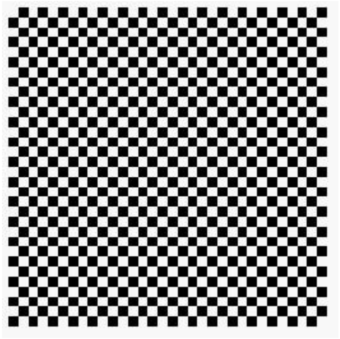
Clamping: if s,t > 1 use 1, if s,t <0 use 0

Wrapping: use s,t modulo 1

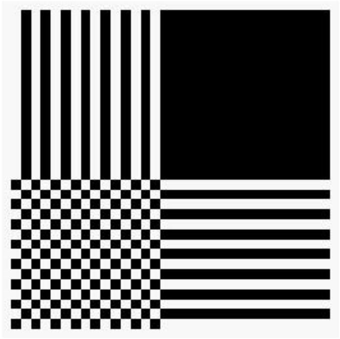
```
glTexParameteri( GL_TEXTURE_2D,
     GL_TEXTURE_WRAP_T, GL_REPEAT )
glTexParameteri( GL_TEXTURE_2D,
     GL_TEXTURE_WRAP_S, GL_CLAMP )
```



texture



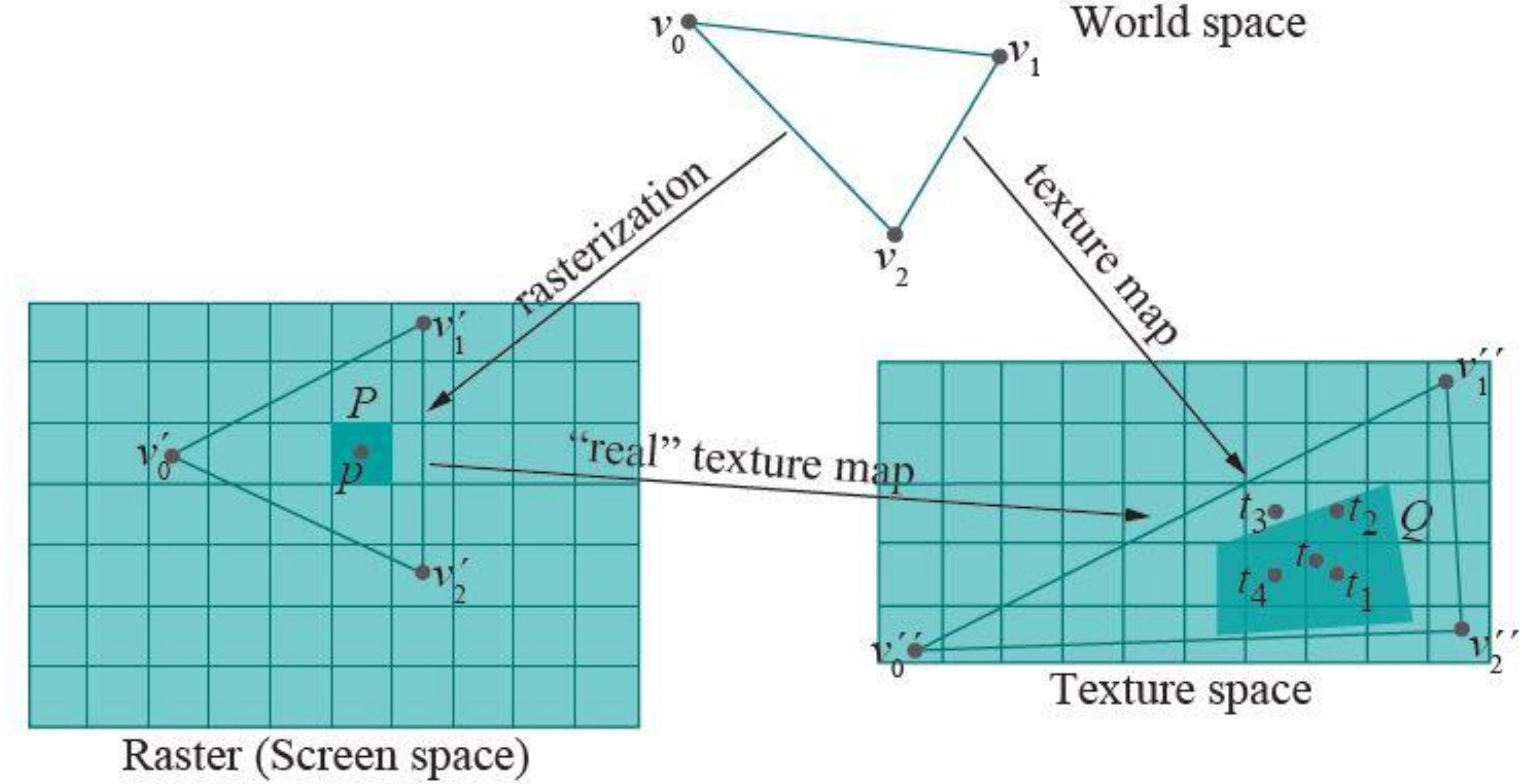
GL_REPEAT wrapping



GL_CLAMP wrapping

Aliasing

- A single pixel P is mapped to a quadrilateral Q covering many texels (minification).



Filtering

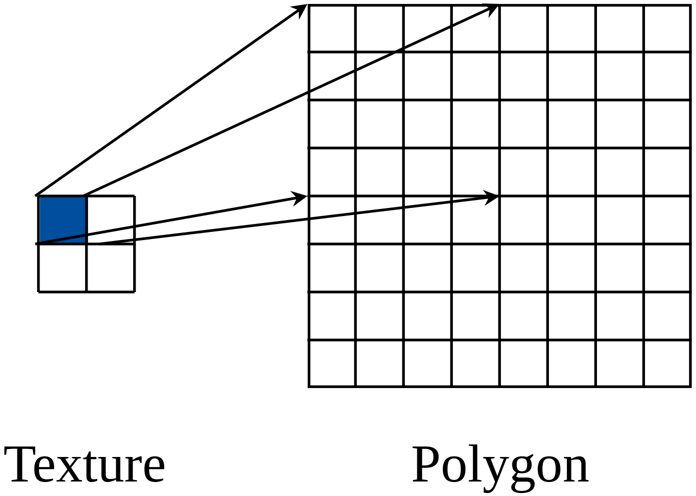
- The process of sampling color values for pixels based on the texture map is called filtering

```
// Set texture parameters for filtering.
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
```

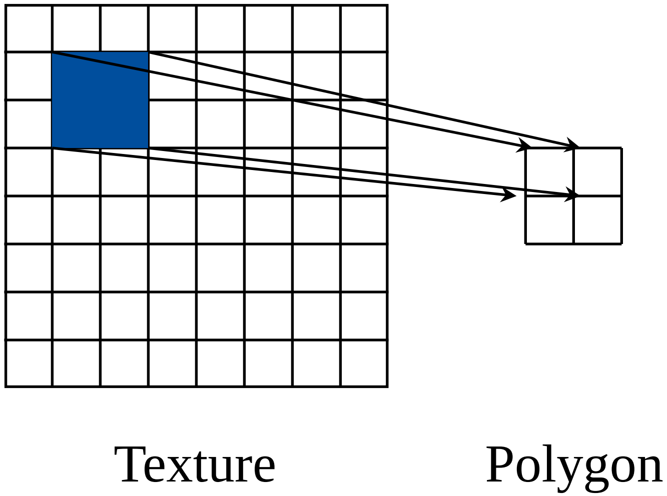
Magnification and Minification

More than one texel can cover a pixel (*minification*) or more than one pixel can cover a texel (*magnification*)

Can use point sampling (nearest texel) or linear filtering (2 x 2 filter) to obtain texture values



Magnification



Minification

Filter Modes

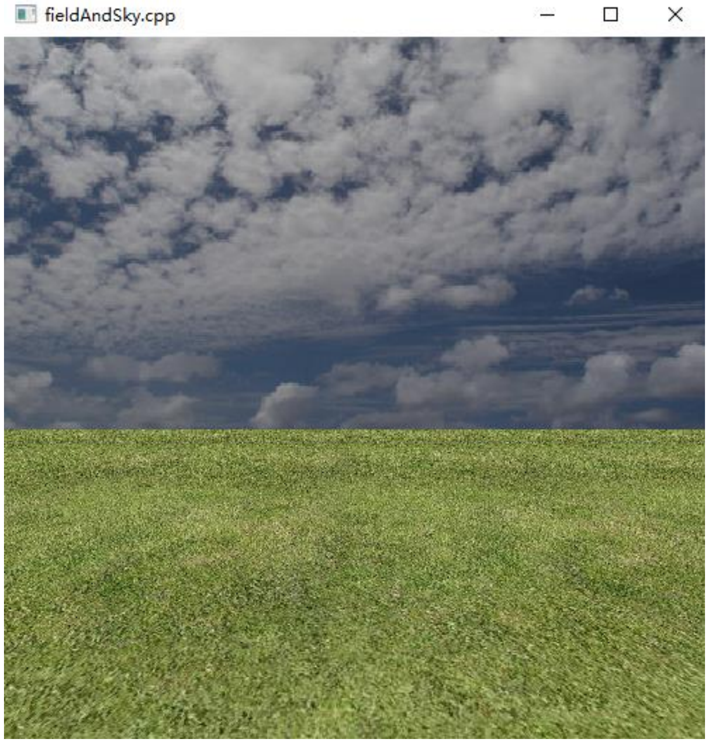
Modes determined by

- `glTexParameteri( target, type, mode )`

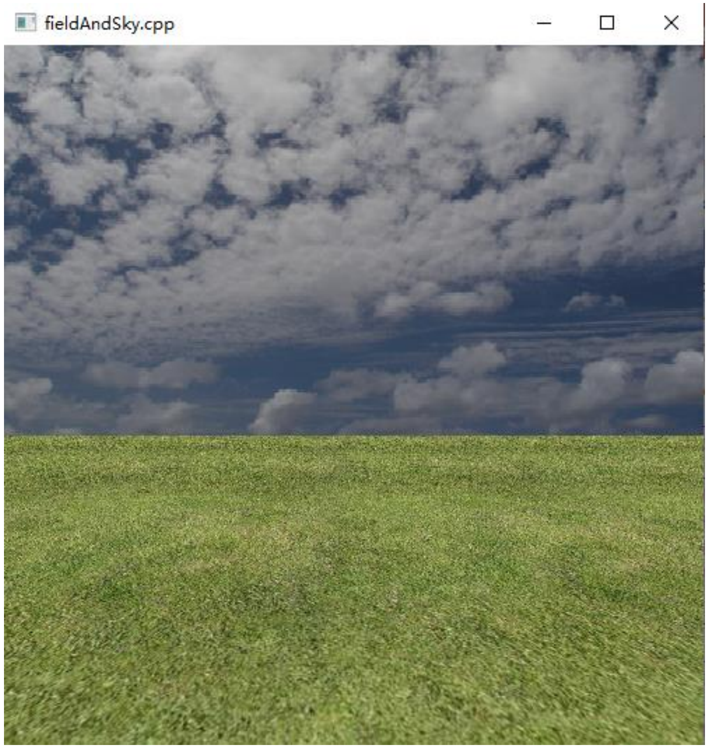
```
glTexParameteri(GL_TEXTURE_2D, GL_TEXURE_MAG_FILTER,
             GL_NEAREST);

glTexParameteri(GL_TEXTURE_2D, GL_TEXURE_MIN_FILTER,
             GL_LINEAR);
```

Filter Modes

Chapter12\FieldAndSky\FieldAndSky.cpp



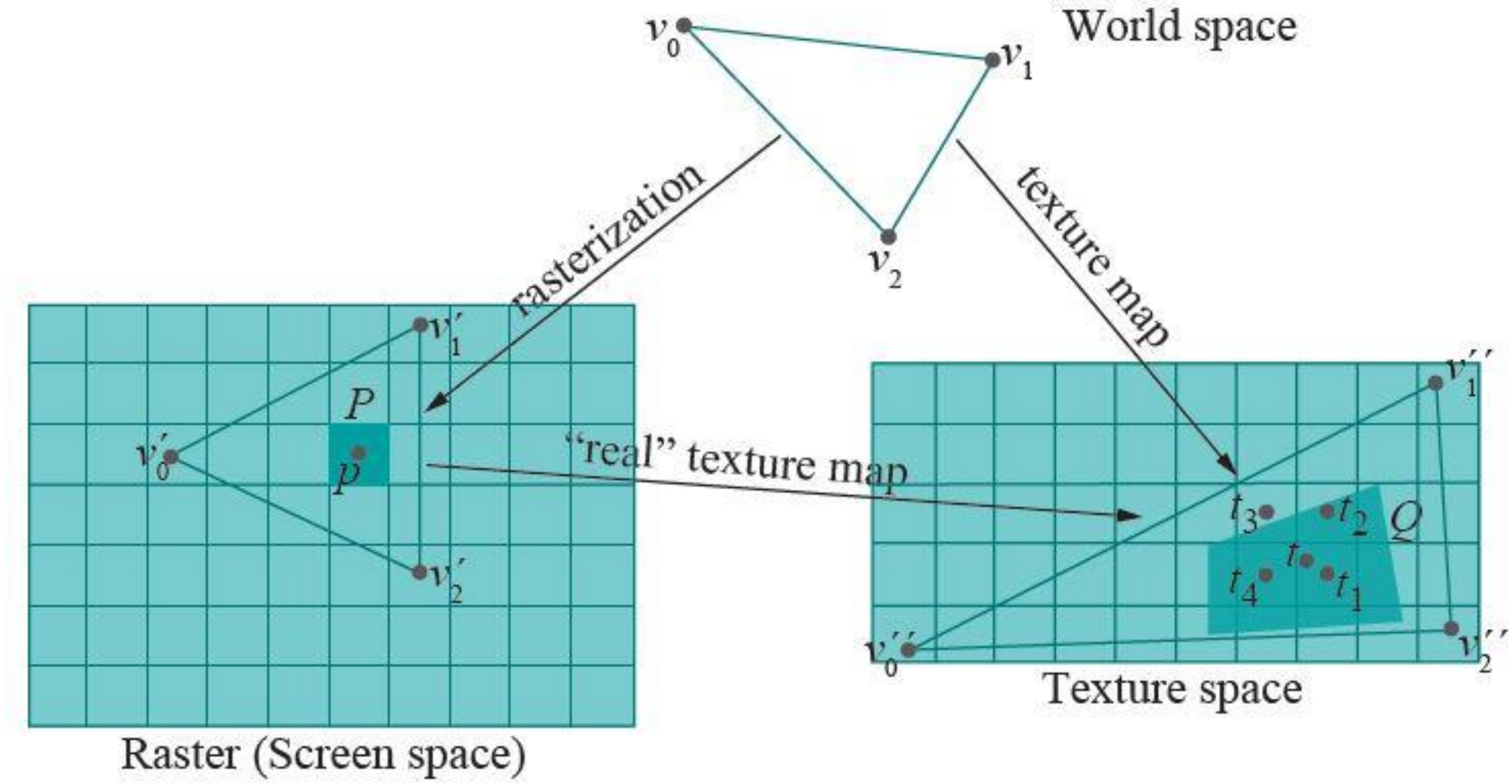
```
glTexParameteri(
GL_TEXTURE_2D,
GL_TEXURE_MAG_FILTER,
GL_NEAREST);
```



```
glTexParameteri(
GL_TEXTURE_2D,
GL_TEXURE_MIN_FILTER,
GL_LINEAR);
```

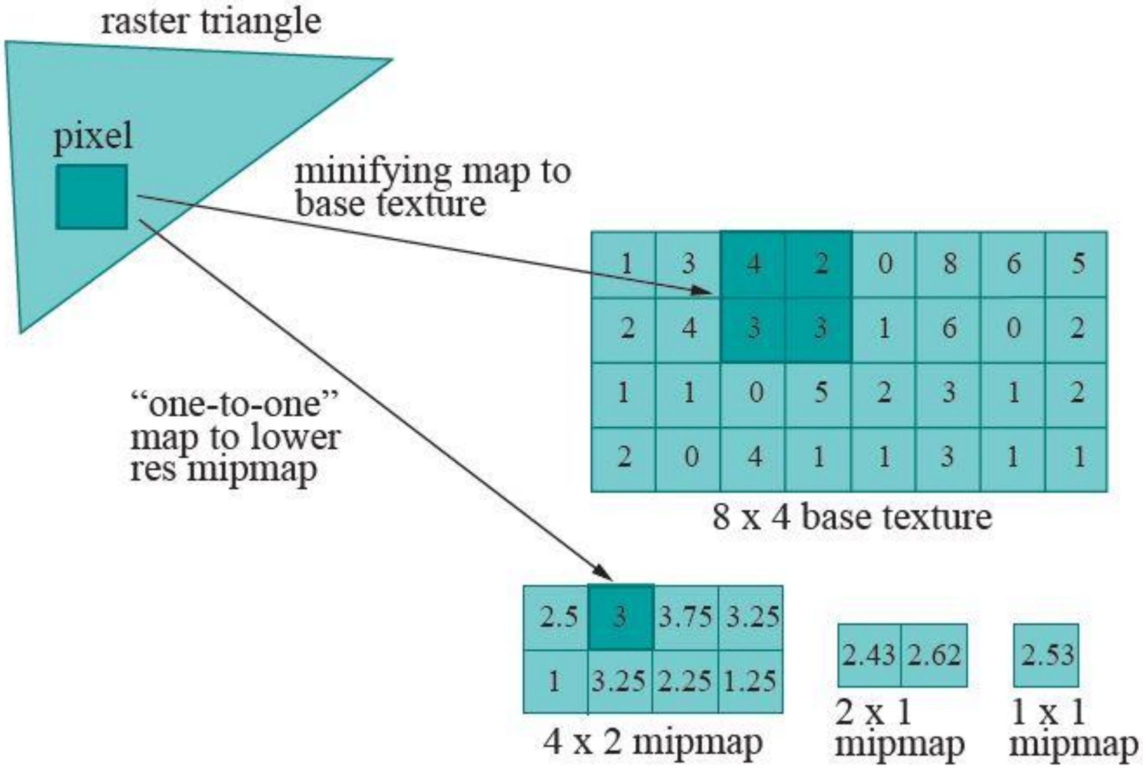
Aliasing

- Reason for the shimmer observed in fieldAndSky.cpp



Mipmapping

- Starting with the original texture, a set of textures of progressively lower resolution, called mipmaps, is prepared.
- During run-time, OpenGL maps a geometric primitive to the mipmap which affords a nearly *one-to-one* correspondence between pixels and texels



1	3	4	2	0	8	6	5
2	4	3	3	1	6	0	2
1	1	0	5	2	3	1	2
2	0	4	1	1	3	1	1

2.5	3	3.75	3.25
1	3.25	2.25	1.25

2.43	2.62

2.53

Mipmapping Modes

Modes determined by

- `glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, mode )`

□ GL_NEAREST_MIPMAP_NEAREST : Applies the mipmap that’s a closest fit resolution-wise to the rasterized primitive and then uses the GL_NEAREST filtering option within that mipmap.

□ GL LINEAR MIPMAP NEAREST : Applies the mipmap that’s a closest fit resolution-wise to the rasterized primitive and then the GL_LINEAR filtering option within that mipmap.

Mipmapping Modes

Modes determined by

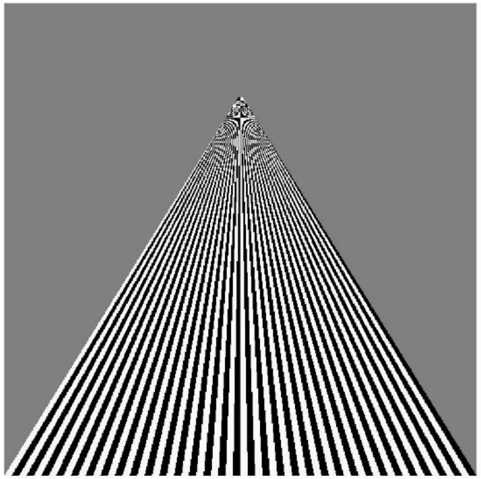
- **glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, mode)**

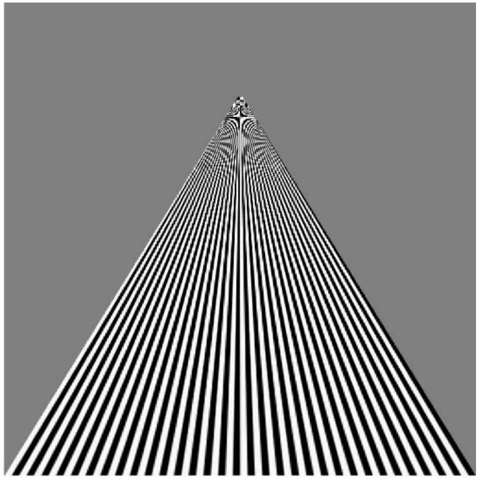
□ GL_NEAREST_MIPMAP_LINEAR : Finds the two mipmaps that are closest resolution-wise to the rasterized primitive, then uses the GL_NEAREST filtering option within either mipmap to produce two sets of color values and, finally, takes a weighted average of the two sets.

□ GL_LINEAR_MIPMAP_LINEAR : Finds the two mipmaps that are closest resolution-wise to the rasterized primitive, then uses the GL_LINEAR filtering option within either mipmap to produce two sets of color values and, finally, takes a weighted average of the two sets.

Example

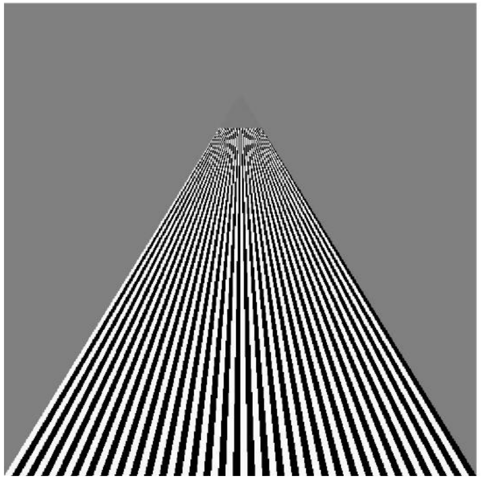
nearest
filtering

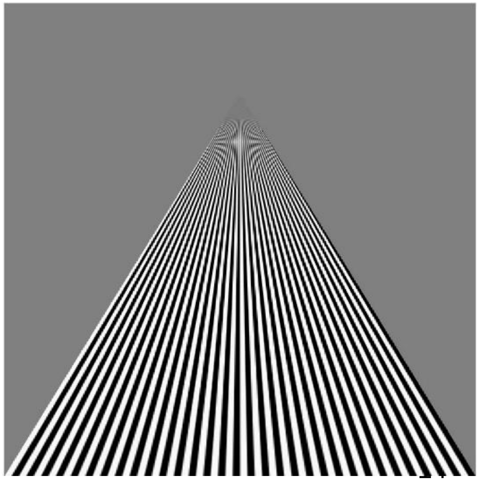




linear
filtering

mipmapped
nearest
filtering

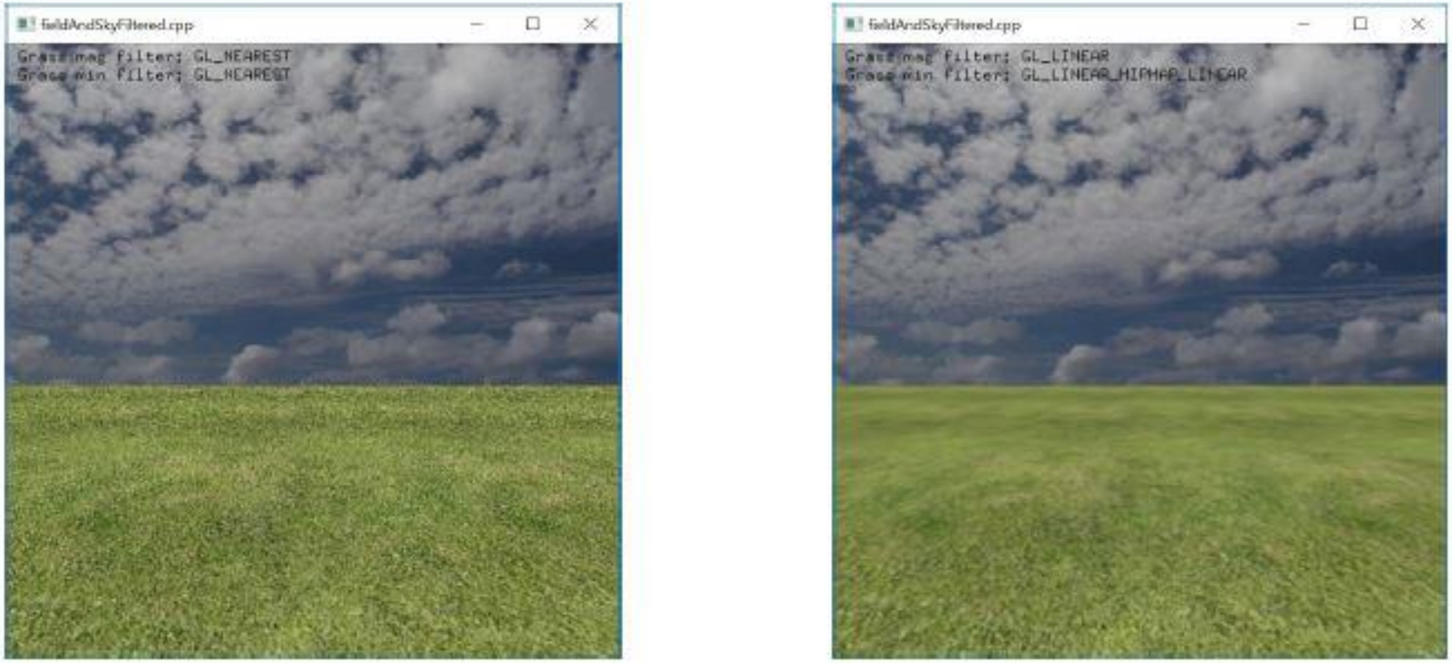




mipmapped
linear
filtering

Example

- □ Chapter12\FieldAndSkyFiltered\fieldAndSkyFiltered.cpp



```
Grass mag filter: GL_NEAREST
Grass min filter: GL_NEAREST
```

(a)

```
Grass mag filter: GL_LINEAR
Grass min filter: GL_LINEAR_MIPMAP_LINEAR
```

(b)

Figure 12.28: Screenshots of `fieldAndSkyFiltered.cpp`: (a) Weakest filter (b) Strongest filter.

LOD (level-of-detail)

- Mipmapping is one of a class of LOD (level-of-detail) methods
- Representing objects by polygonal meshes of varying levels of refinement is another practically important LOD application

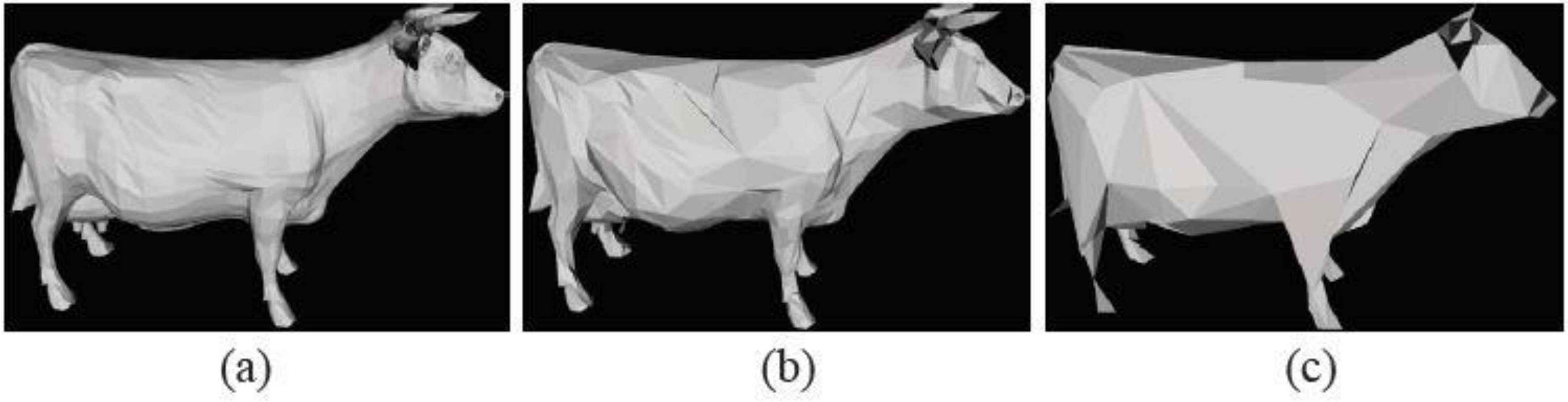


Figure 12.31: Cow at 3 different resolutions: (a) 5804 (b) 1772 (c) 328 triangles.