

Visual Computing I:

Interactive Computer Graphics and Vision



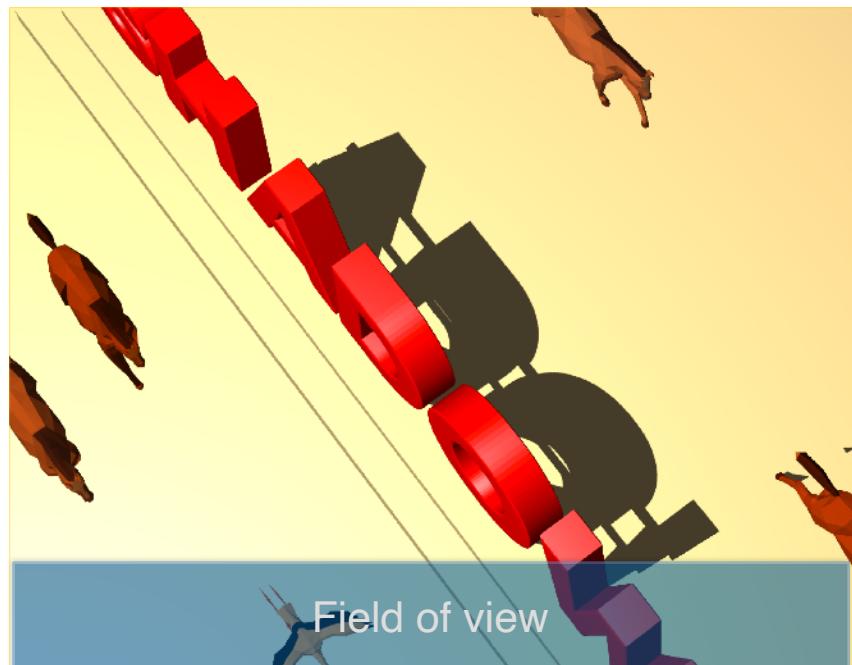
Shadow Mapping Optimisations

Stefanie Zollmann and Tobias Langlotz

Shadow Mapping Challenges

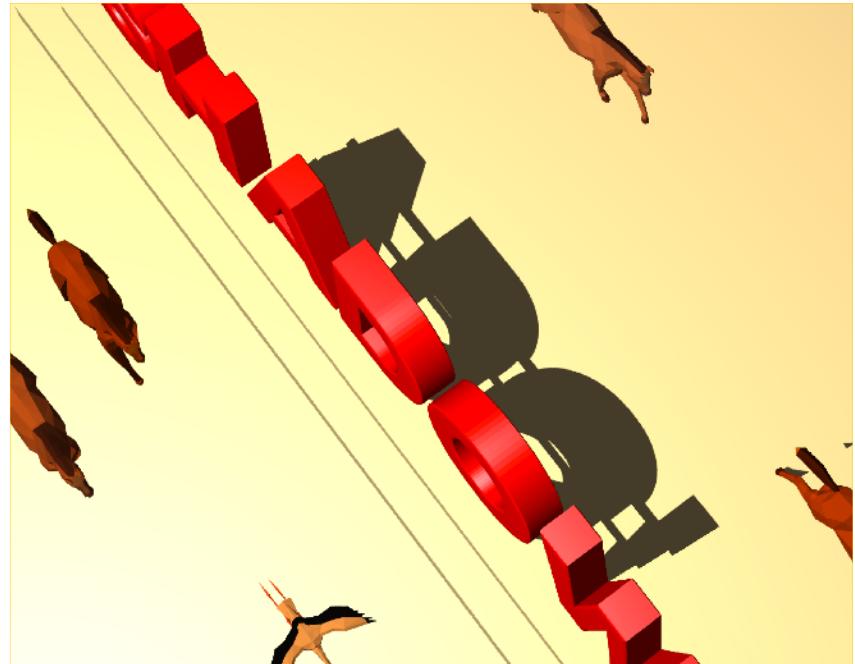
Limitations

- Field of View
- Surface Acne
- Aliasing



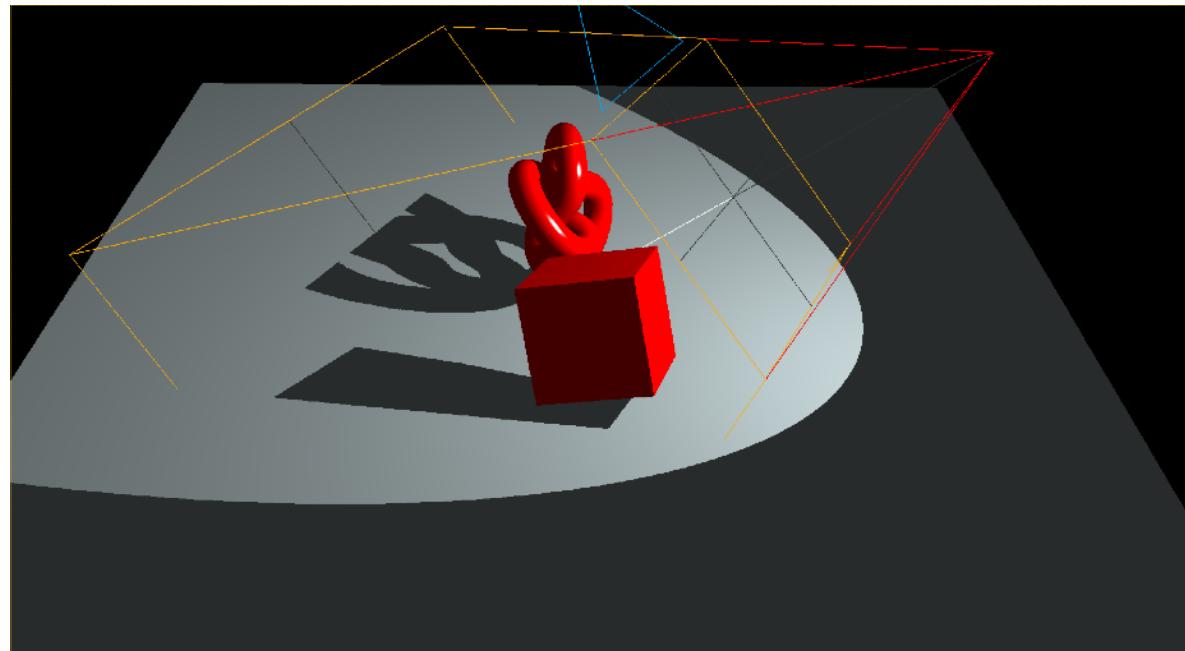
Problem 1: Field of View

- What if object is outside field of view of shadow map?
- No shadows or partial shadows
- For spot lights, this can be changed by tweaking its range
- Problem in particular for larger scenes



Problem 1: Field of View

- What happens?
 - Sampling depth values outside the light's frustum
 - Behaviour depends on texture wrapping method, e.g. for `GL_CLAMP` depends on the last texel



Automatic Light Frustum for Shadow Mapping

- Poorly chosen frustum can waste resolution (too large) or cause missing shadows (too small)
- Automating frustum computation ensures optimal coverage of the visible scene
 - Transform camera frustum corners into light space
 - Find the axis-aligned bounding box (AABB) that encloses those points
 - Use that AABB to build the light's orthographic projection matrix

```
// 1. Get 8 corners of camera view frustum in world space
std::vector<glm::vec3> frustumCorners = getFrustumCornersWS(camera);

// 2. Transform corners into light space
for (auto& v : frustumCorners)
    v = glm::vec3(lightViewMatrix * glm::vec4(v, 1.0));

// 3. Compute light-space AABB
glm::vec3 min = vmin(frustumCorners);
glm::vec3 max = vmax(frustumCorners);

// 4. Build orthographic projection
glm::mat4 lightProj = glm::ortho(min.x, max.x, min.y, max.y, -max.z, -min.z);

// Final light matrix
glm::mat4 lightSpaceMatrix = lightProj * lightViewMatrix;
```

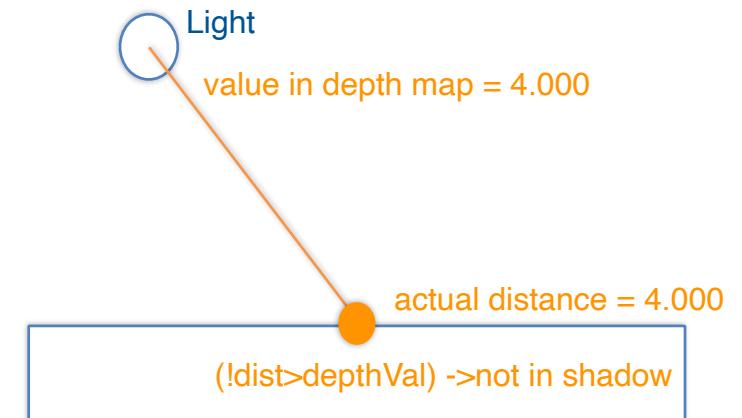
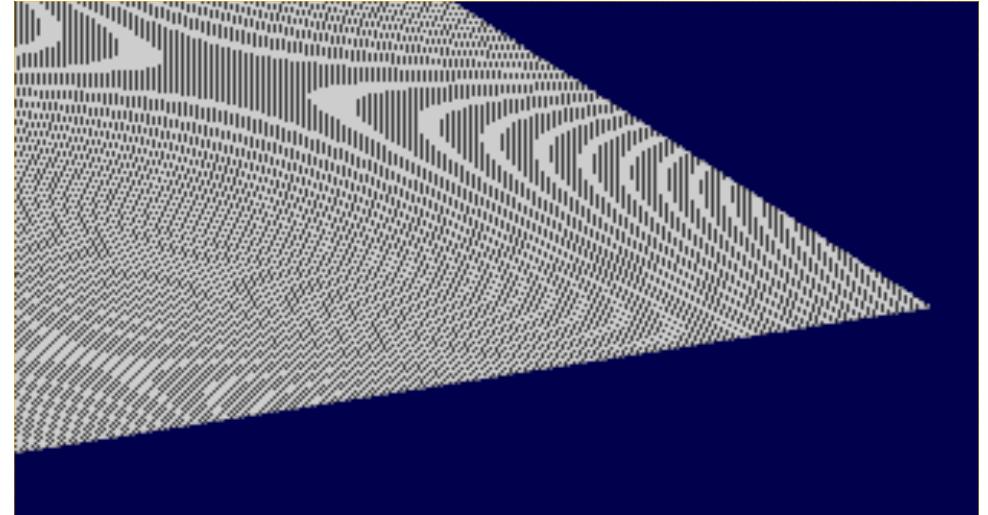
Problem 2: Surface Acne

- Self-shadowing problem due to precision and depth map resolution
 - Remember: we compare depth stored in the **shadow map** (from the light's view) and the distance of the **current fragment** to the light source
 - If they are almost the same, small rounding errors (from limited depth precision) can make the fragment shader think the surface is slightly behind itself, so it gets incorrectly marked as “in shadow”
 - Looks like the lit surface has “shadow pimples” -> surface acne



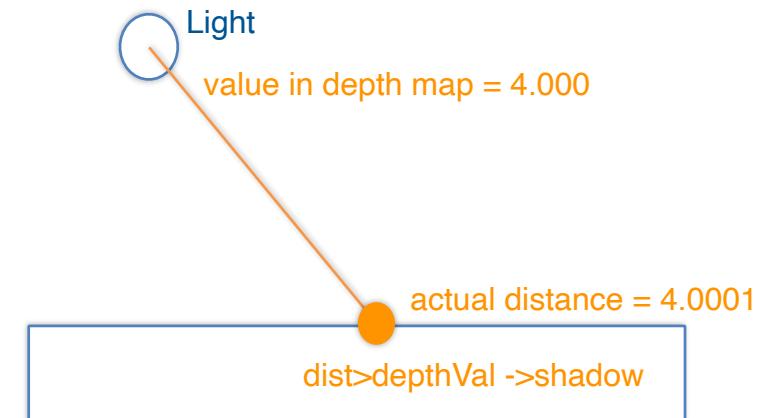
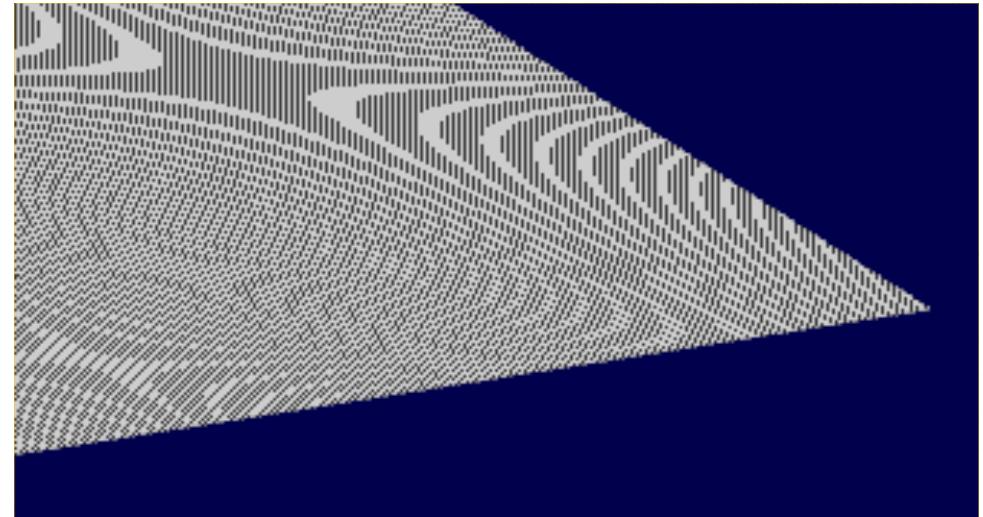
Problem 2: Surface Acne

- Self-shadowing problem due to precision and depth map resolution
 - Depth value in map can differ from actual distance between object and light source
 - Sampling problem: neighbouring vertices map to the same depth map pixel



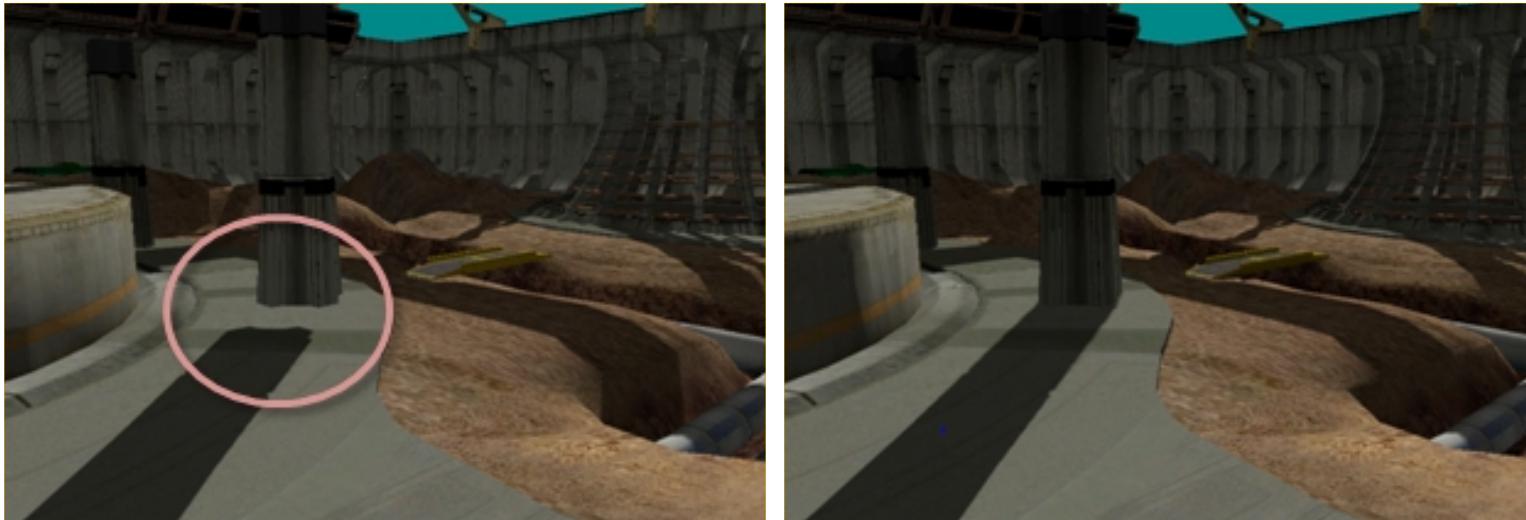
Problem 2: Surface Acne

- Self-shadowing problem due to precision and depth map resolution
 - Depth value in map can differ from actual distance between object and light source
 - Sampling problem: neighbouring vertices map to the same depth map pixel



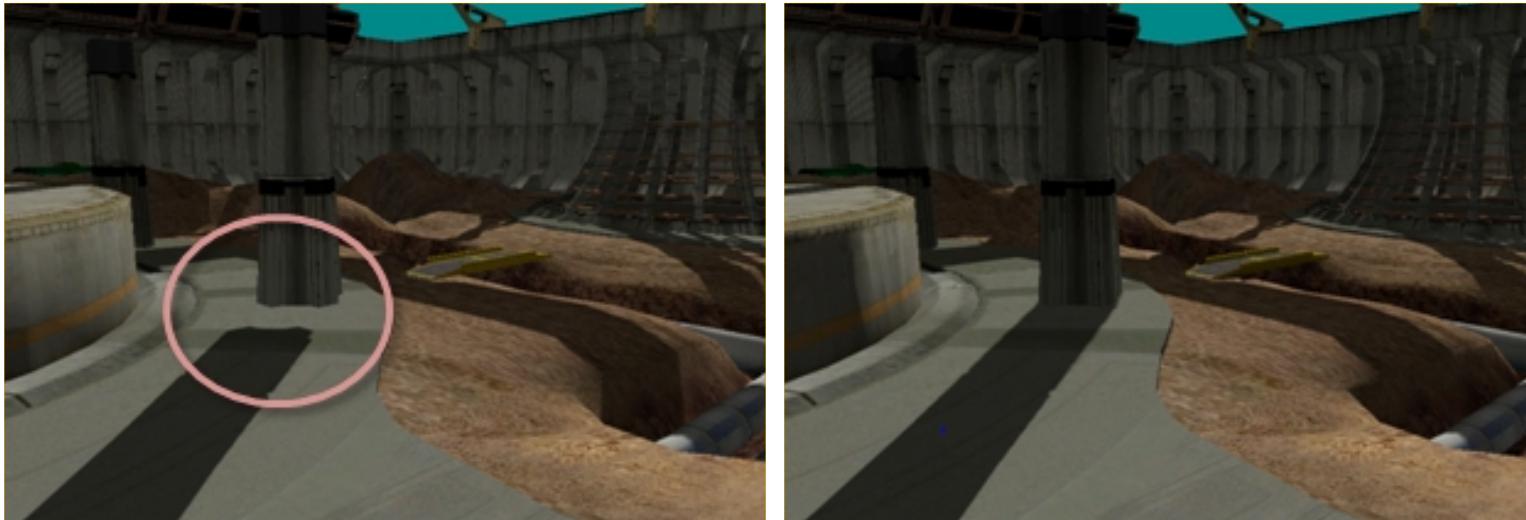
Shadow Map Bias

- Solution for surface acne:
 - Shadow map bias for shadow test: $\text{ShadowMap}(x,y) + \text{bias} < \text{dist}$
 - Choosing a good bias value can be tricky - otherwise we get shadow detachment -> “Peter Panning”



Shadow Map Bias

- Solution for surface acne:
 - Shadow map bias for shadow test: $\text{ShadowMap}(x,y) + \text{bias} < \text{dist}$
 - Choosing a good bias value can be tricky - otherwise we get shadow detachment -> “Peter Panning”

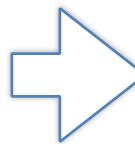


Problem 3: Shadow Map Aliasing

- Quality depends on shadow map resolution
- Higher resolution:
 - Higher quality
 - More memory required



1024x1024



2048x2048

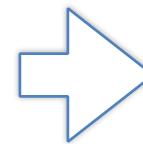


Problem 3: Shadow Map Aliasing

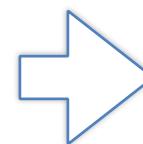
- Quality depends on shadow map resolution
- Higher resolution:
 - Higher quality
 - More memory required



1024x1024

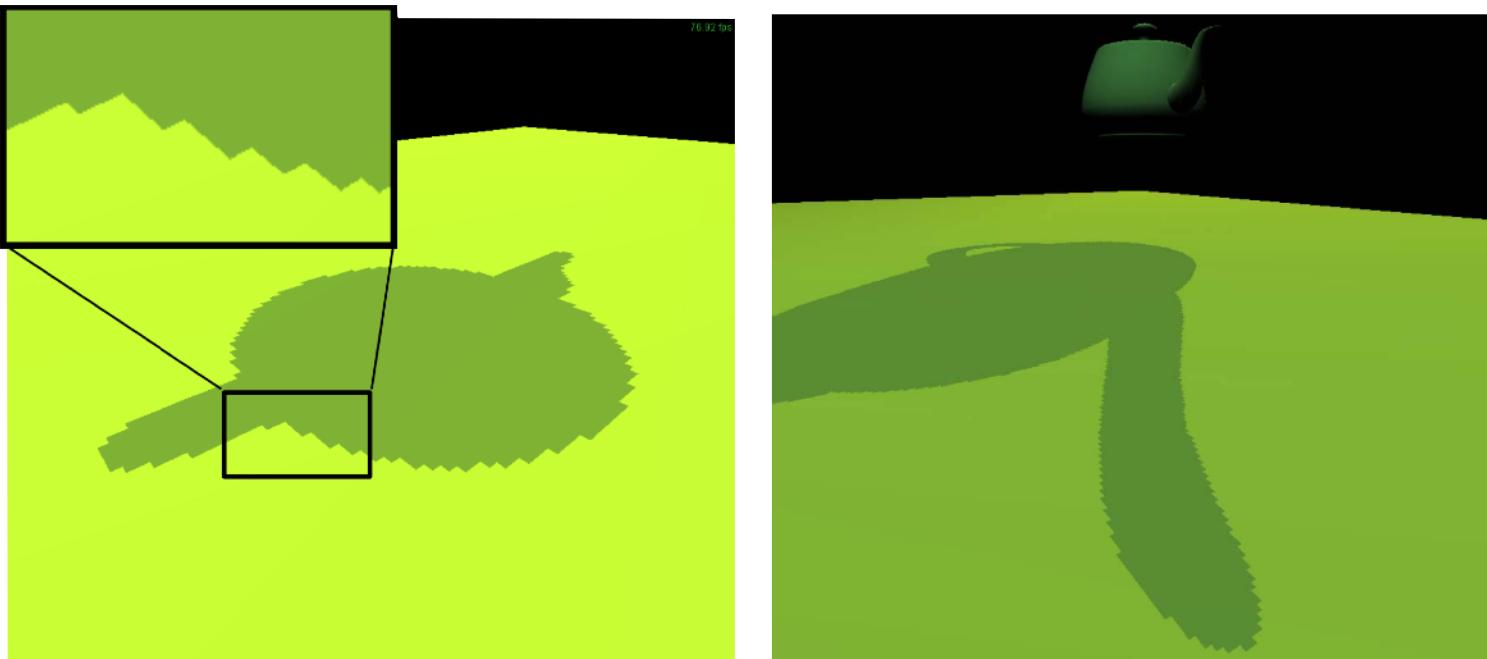


2048x2048



Solutions for Shadow Map Aliasing

- Under-sampling of the shadow map – Jagged shadow edges
- Solution: Filtering: Percentage closer filtering (PCF)



Percentage Closer Filtering (PCF)

- Creating softer shadows
- Unlike normal textures, shadow map textures cannot be pre-filtered to remove aliasing
- Weighted average of neighbouring depth values is not meaningful

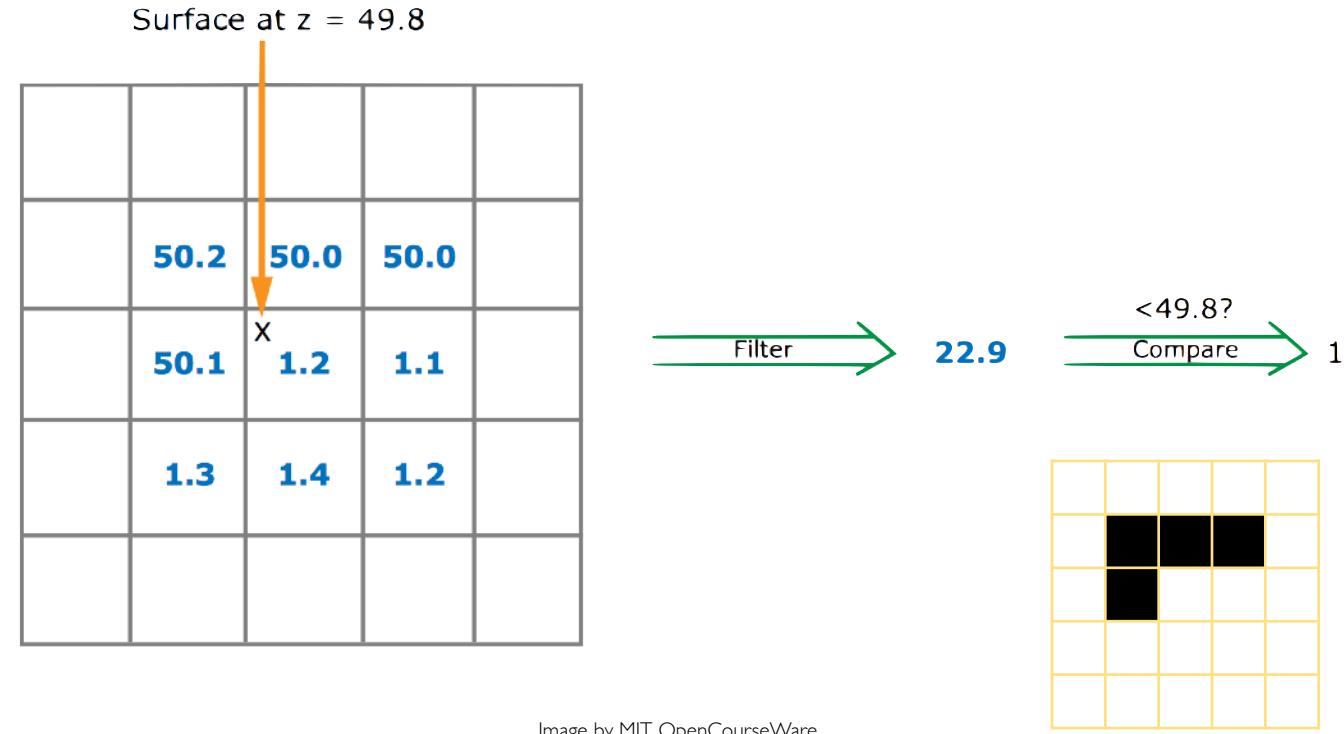
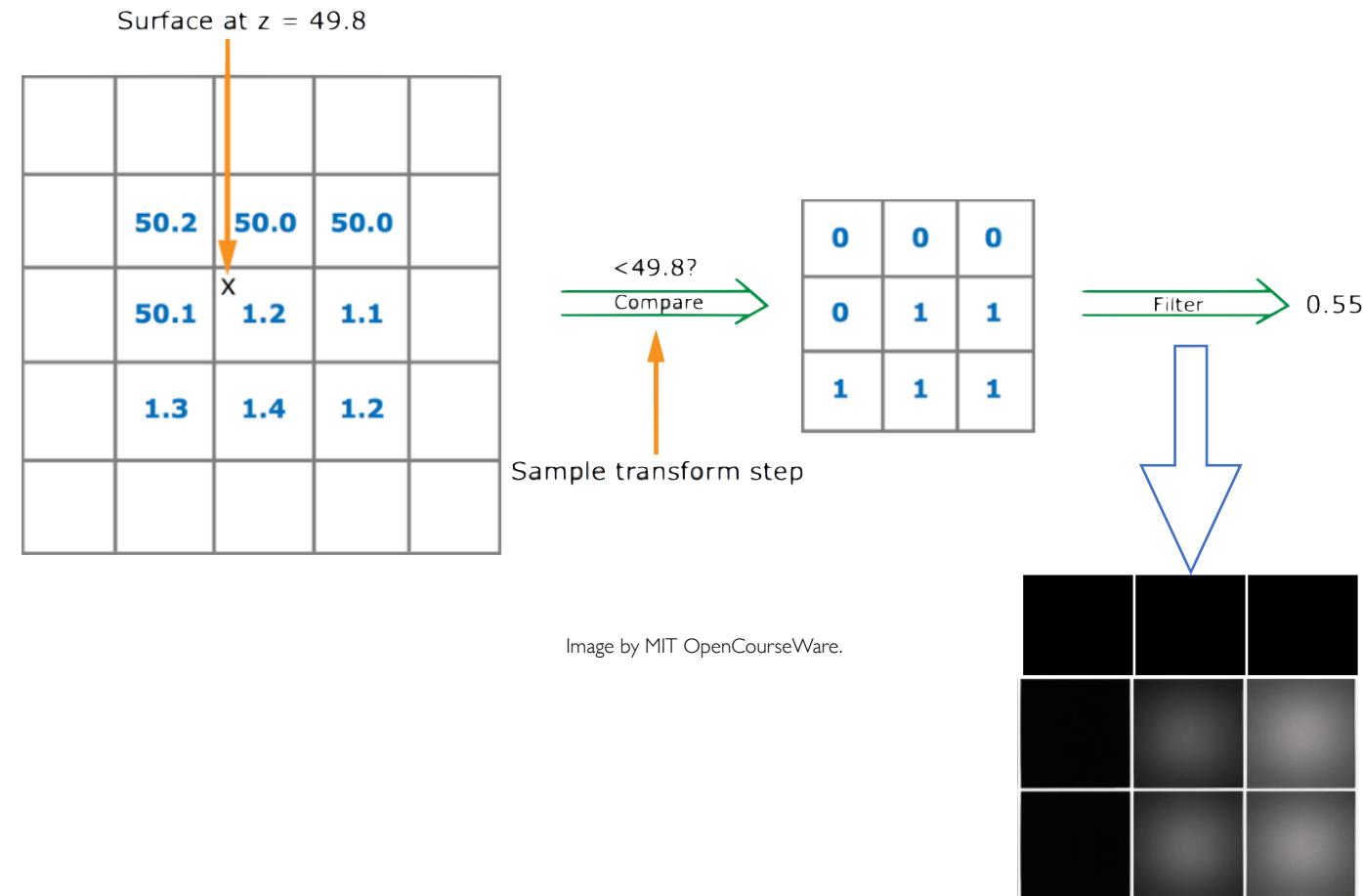


Image by MIT OpenCourseWare.

Percentage Closer Filtering (PCF)

- Creating softer shadows
- Unlike normal textures, shadow map textures cannot be pre-filtered to remove aliasing
- Weighted average of neighbouring depth values is not meaningful

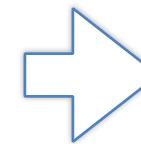


Percentage Closer Filtering (PCF)

- Interpolation of the results of the shadow map computation
- Quality increases with Percentage closer filtering
- Softer appearance of shadows



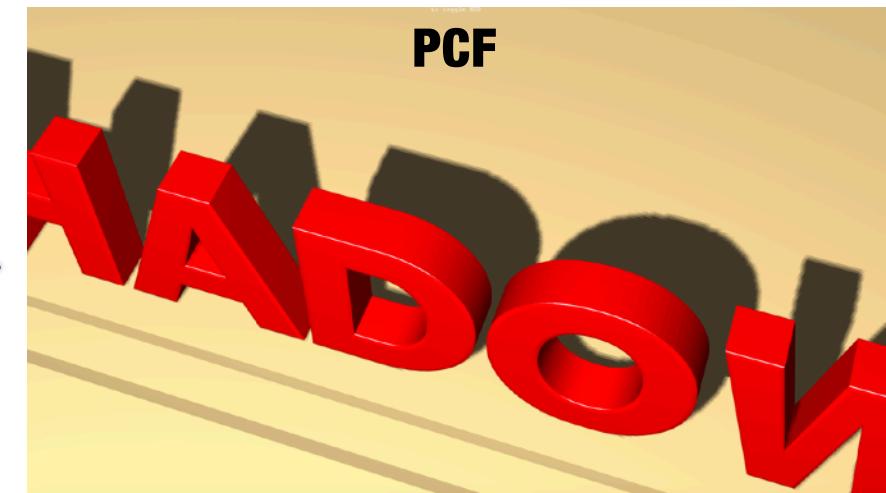
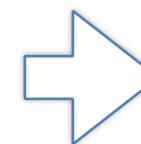
1024x1024



Original Shadow Map



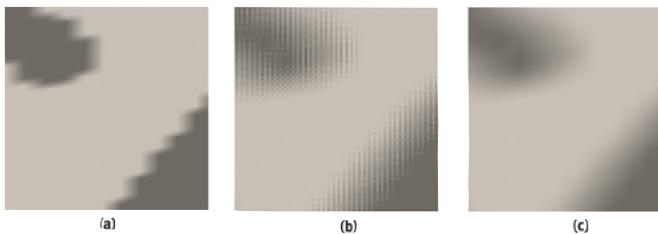
1024x1024



PCF

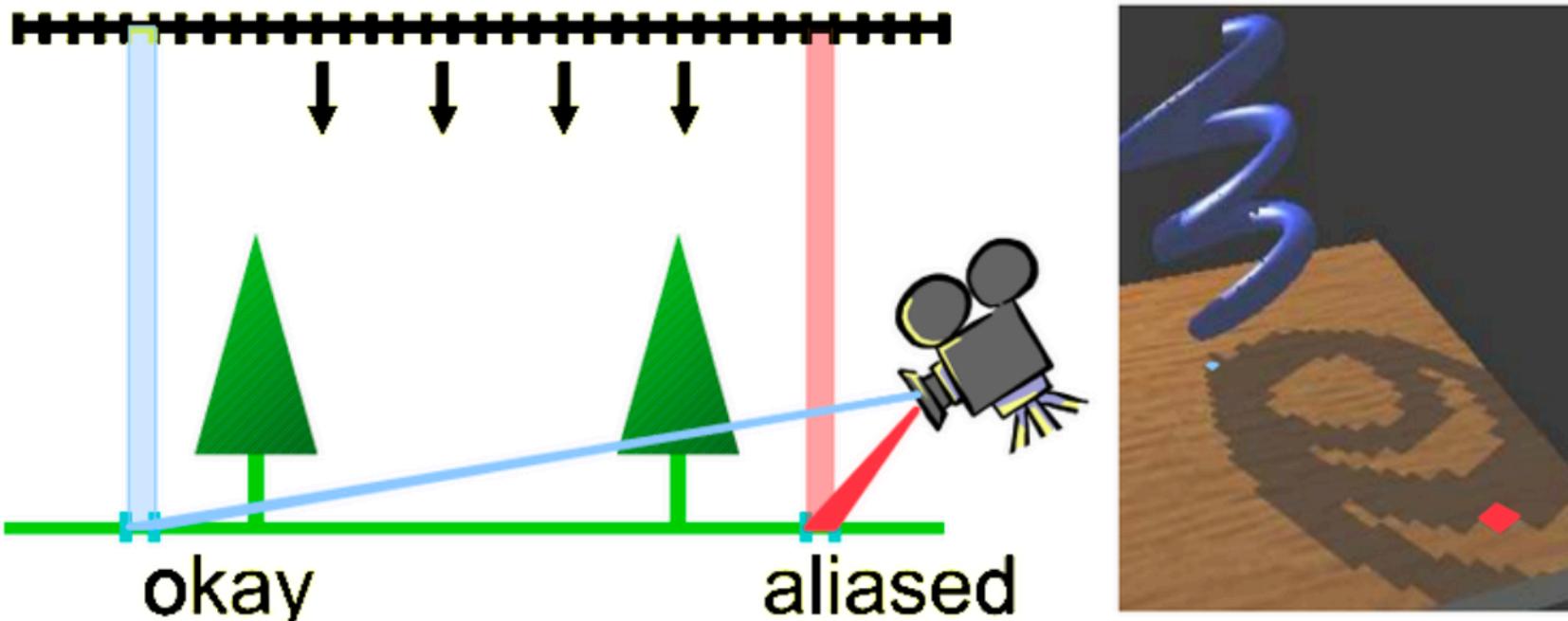
Percentage Closer Filtering (PCF)

- Different filter size



Perspective Aliasing

- View space resolution vs. shadow map resolution

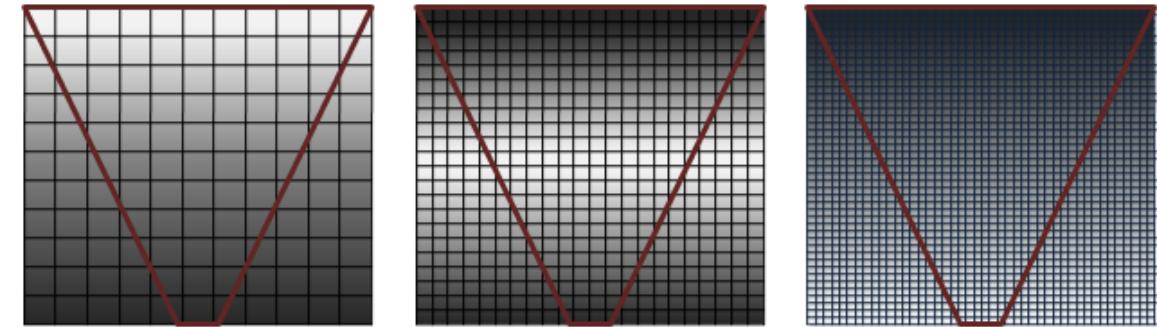


http://www.cg.tuwien.ac.at/~scherzer/files/papers/LispSM_survey.pdf

Cascade Shadow Maps

- Providing higher resolution of the depth texture near the viewer and lower resolution far away

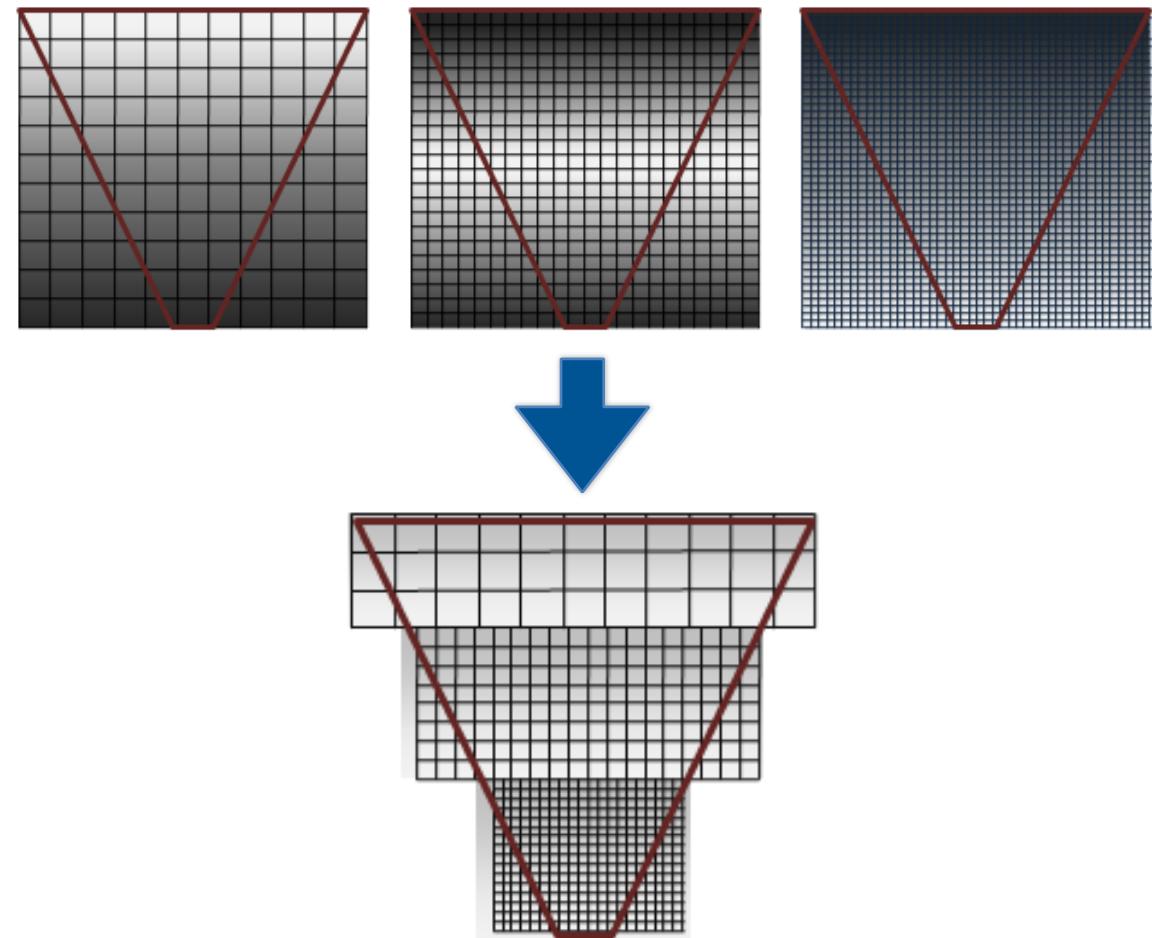
[http://msdn.microsoft.com/en-us/library/windows/desktop/ee416307\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ee416307(v=vs.85).aspx)



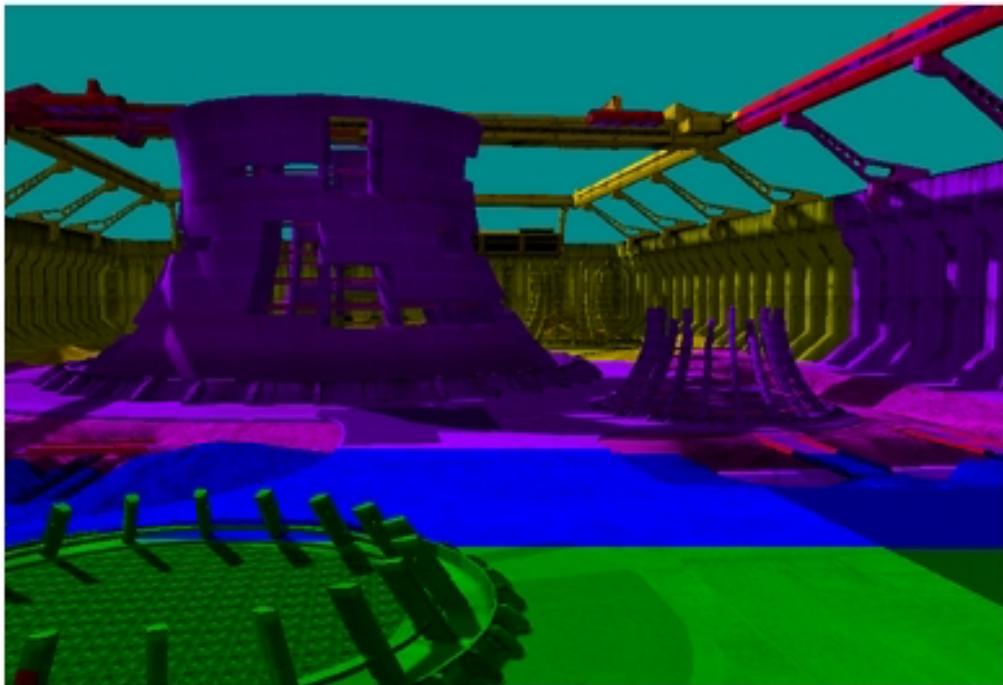
Cascade Shadow Maps

- Providing higher resolution of the depth texture near the viewer and lower resolution far away
- Splitting the camera view frustum and creating a separate depth-map for each partition

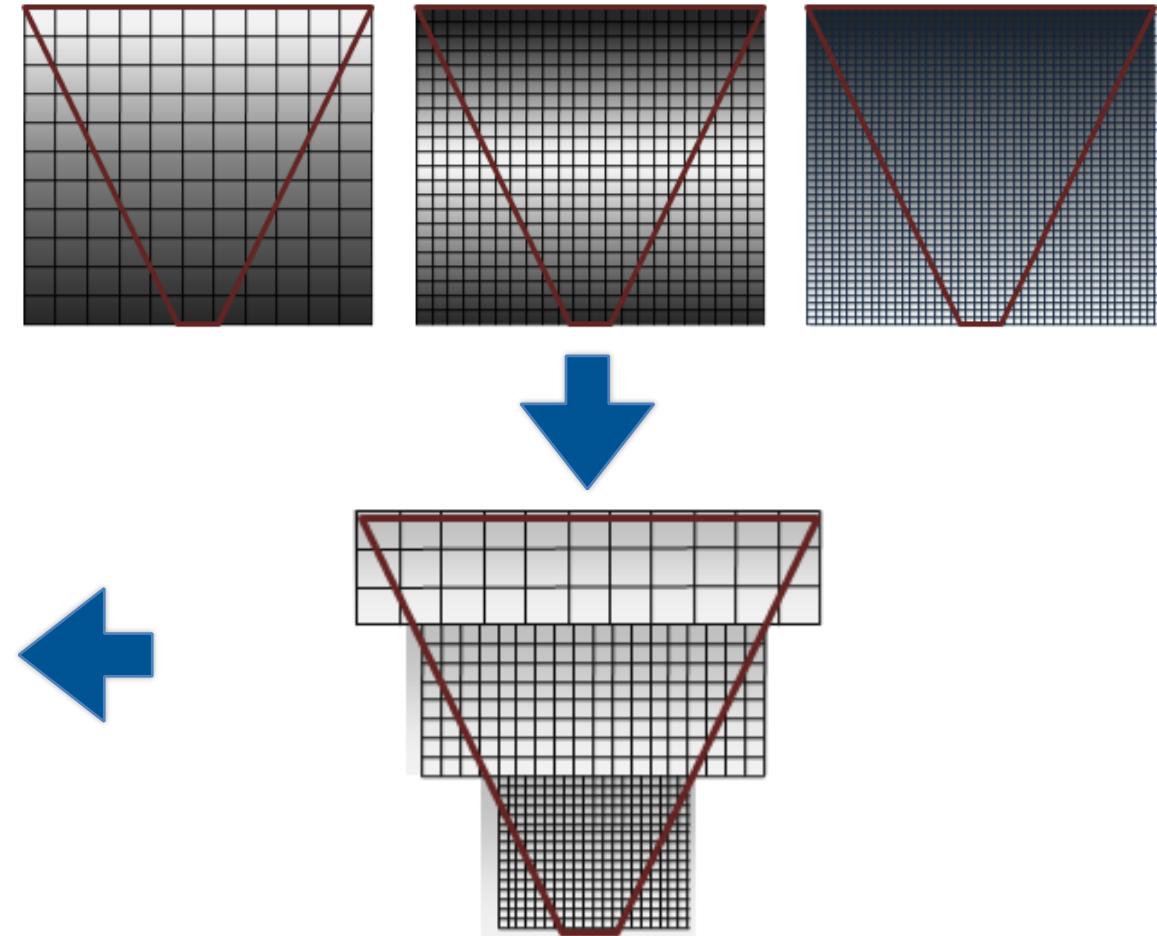
[http://msdn.microsoft.com/en-us/library/windows/desktop/ee416307\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ee416307(v=vs.85).aspx)



Cascade Shadow Maps



[http://msdn.microsoft.com/en-us/library/windows/desktop/ee416307\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ee416307(v=vs.85).aspx)



The end!