Hard Shadows
Aliasing and Remedies

Michael Wimmer

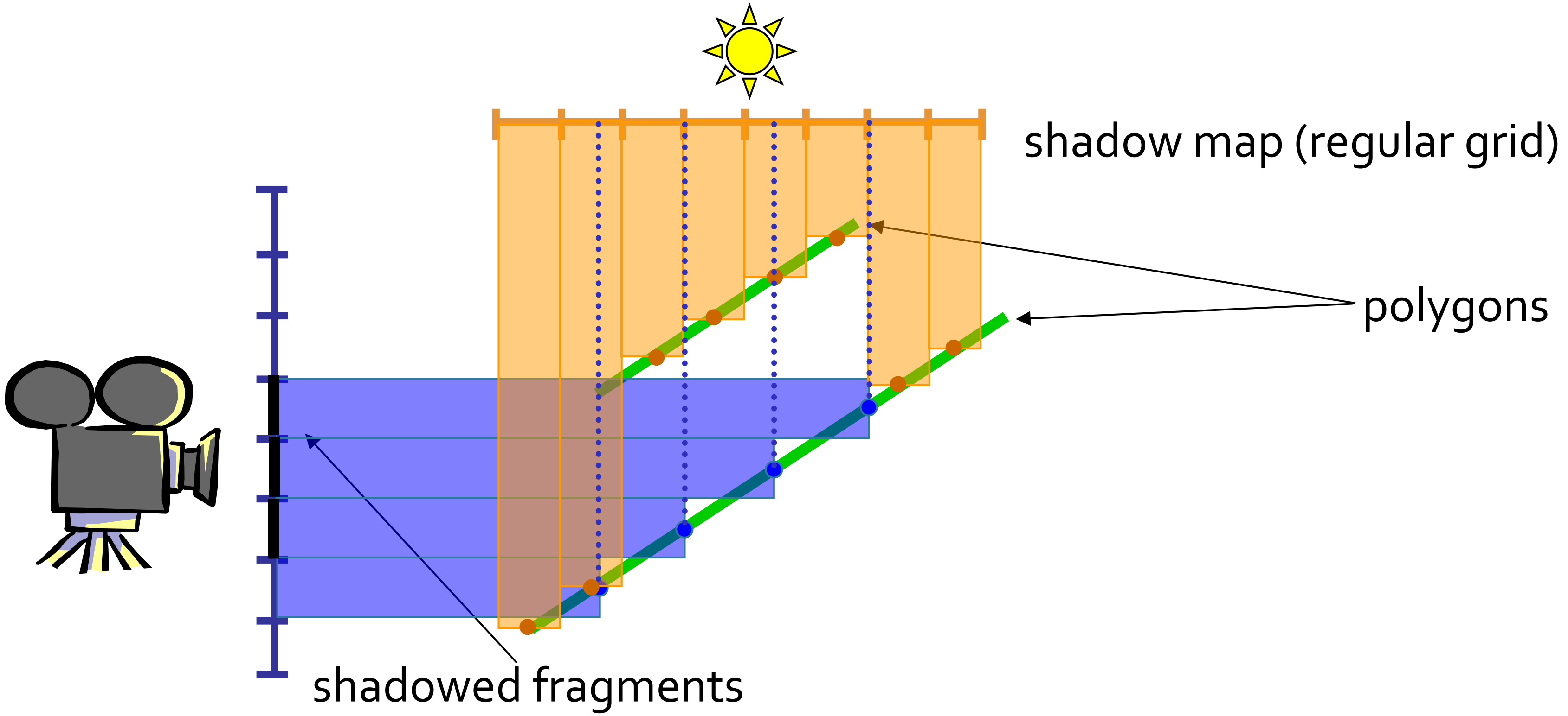
www.realtimeshadows.com

SIGGRAPH2013

Shadow Map as Signal Reconstruction

SIGGRAPH2013

- Initial sampling: shadow-map rendering
- Resampling: determined by view
- Reconstruction: nearest neighbor, PCF, …



Main Types of Error

Initial sampling:

Undersampling

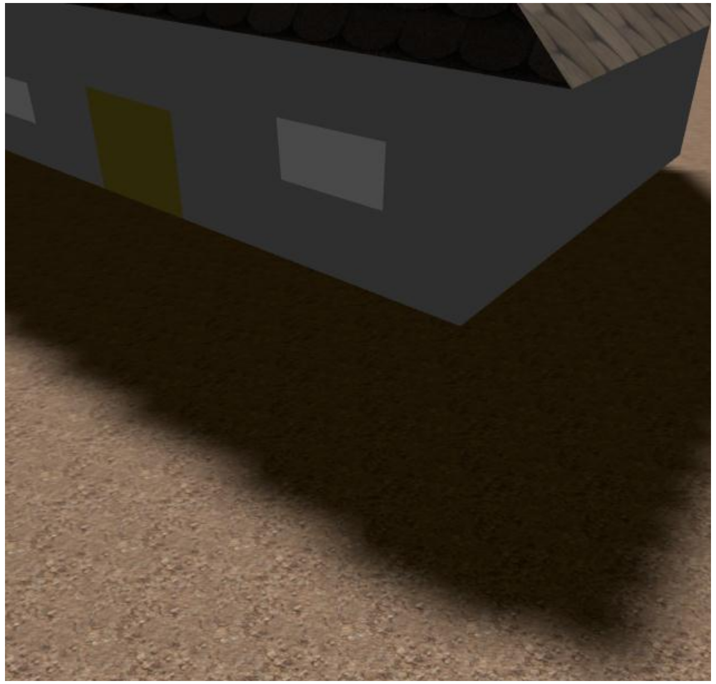
Resampling:

Oversampling

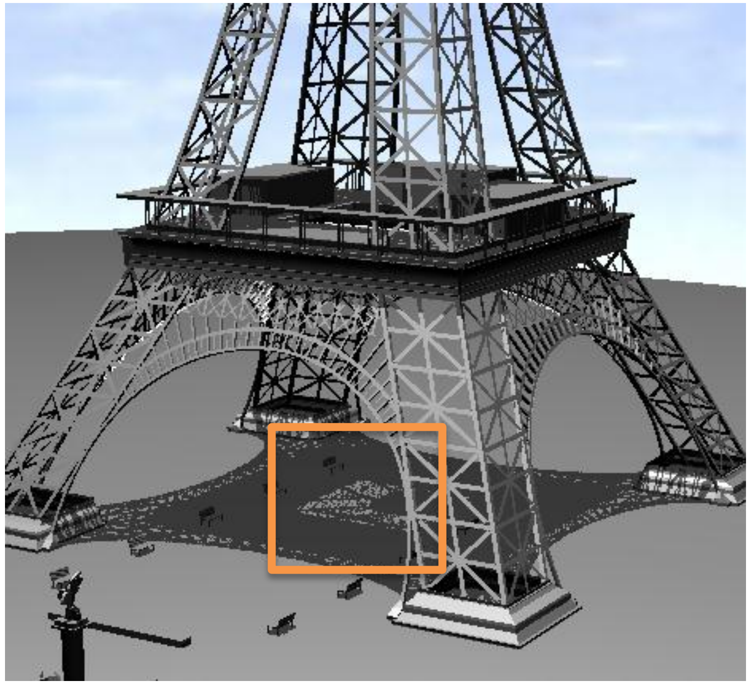
Reconstruction:

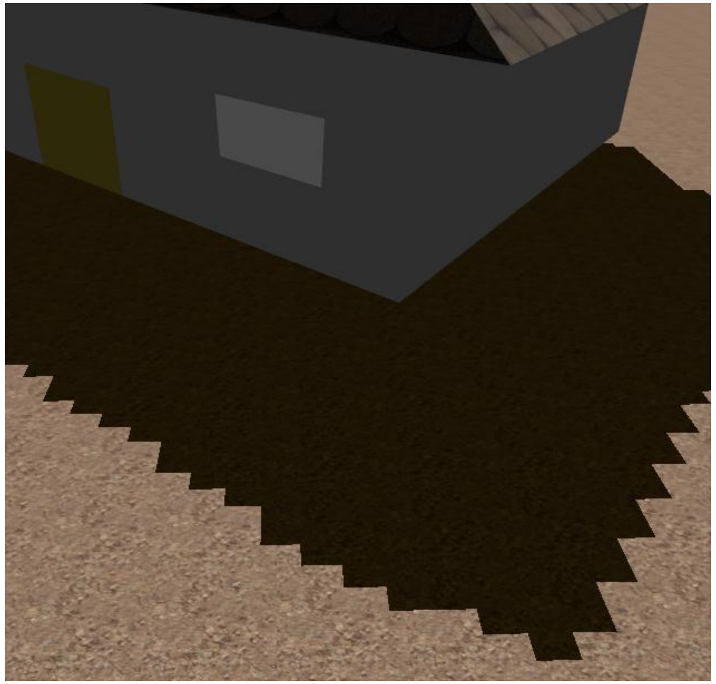
Reconstruction error

(with undersampling)



25 sample Poisson PCF



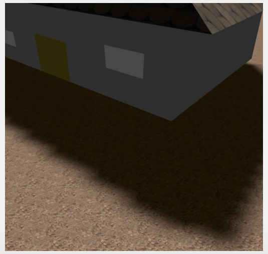


nearest neighbor

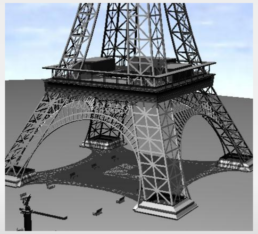
Main Types of Error

- Undersampling
  - No bandlimiting (e.g., „super-sampled shadow map“, possible!)
  - Improve initial sampling!

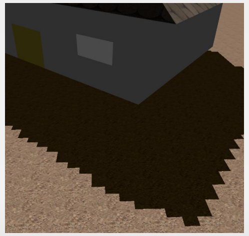
Main topic for next 30 minutes!



- Oversampling
  - Use bandlimiting filters in reconstruction (VSM, CSM, … → later)

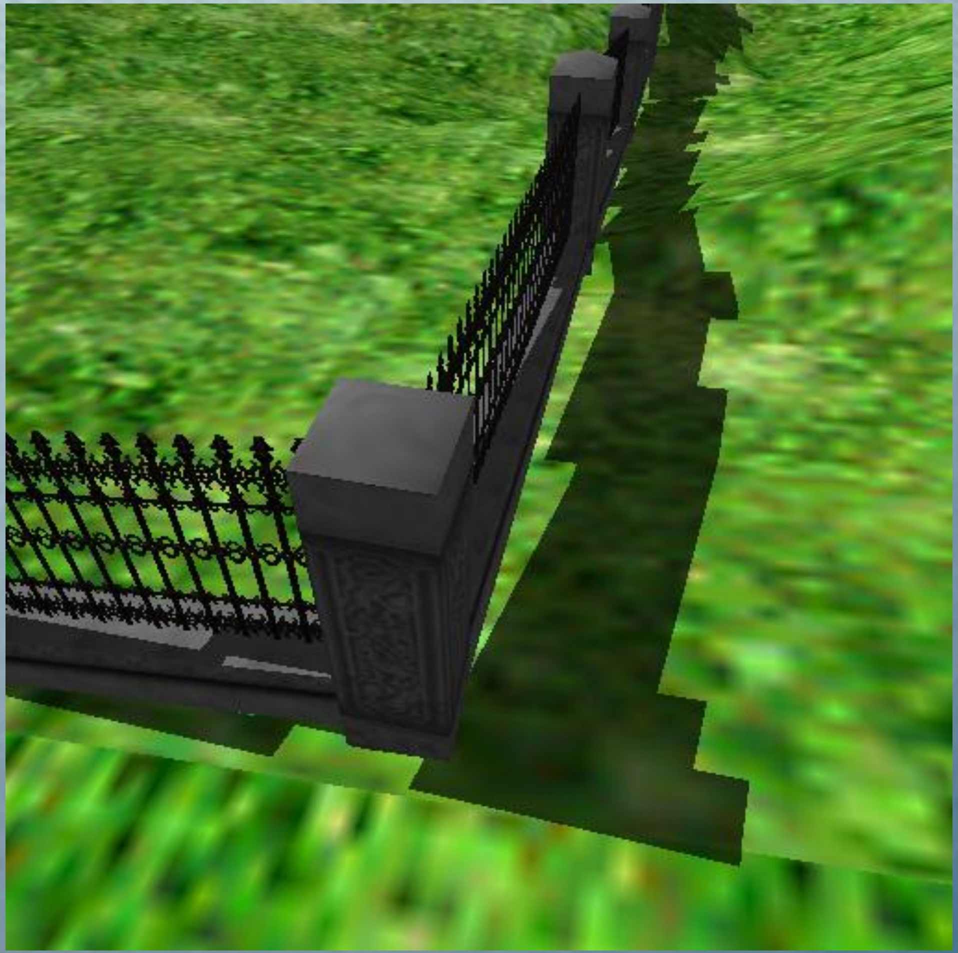


- Reconstruction error
  - Use better reconstruction filters (PCF, …)
  - Use different reconstruction algorithm (Silhouette Shadow Maps)



Overview

- Improving Initial Sampling/Undersampling
  - Fitting (Focusing)
  - Error Analysis
  - Warping
  - Partitioning
  - Irregular Sampling
  - Temporal Coherence
- Better reconstruction



Hard Shadows

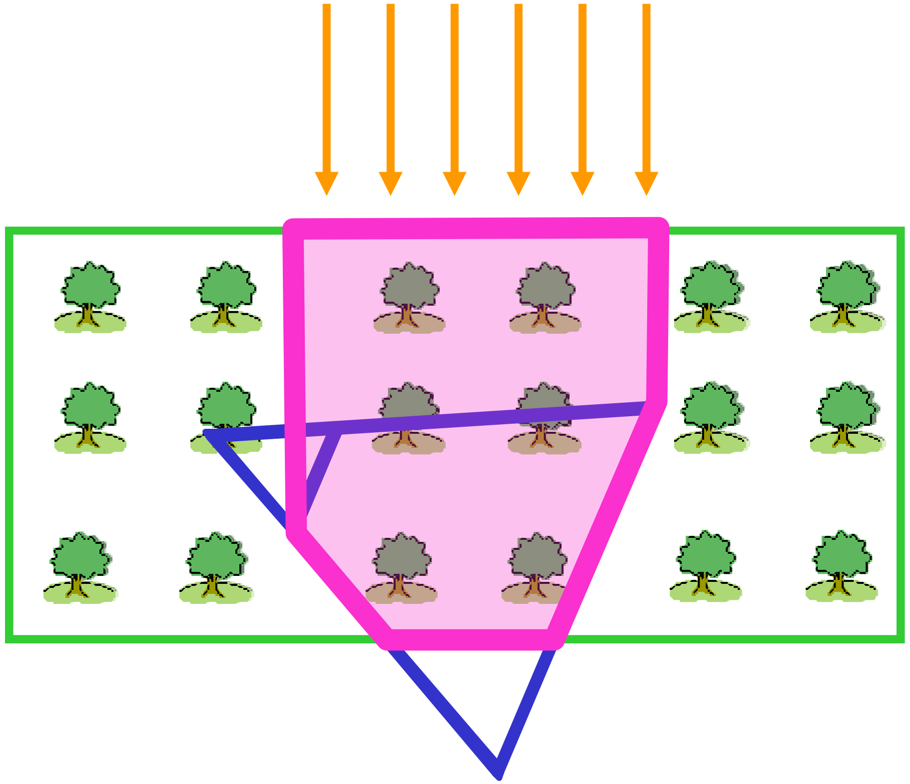
Fighting Undersampling - Fitting

SIGGRAPH2013

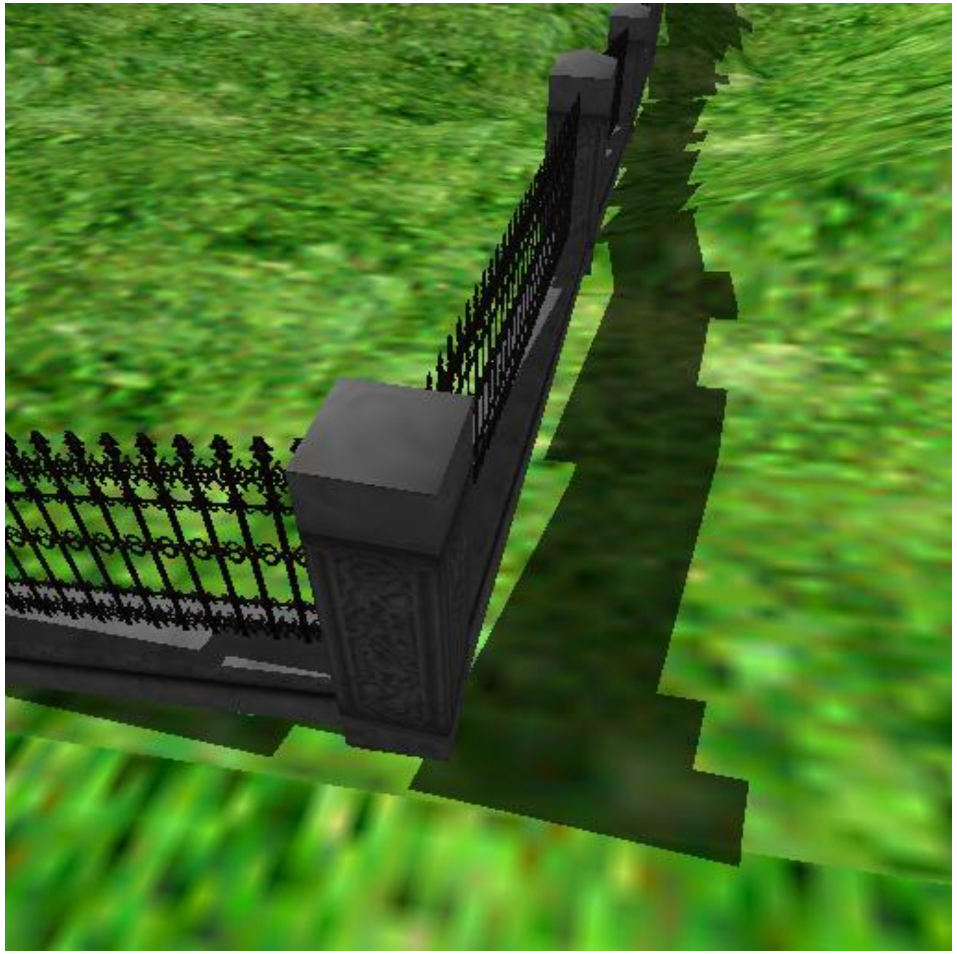
Fitting: Focus the Shadow Map

[Brabec et al. 2002]

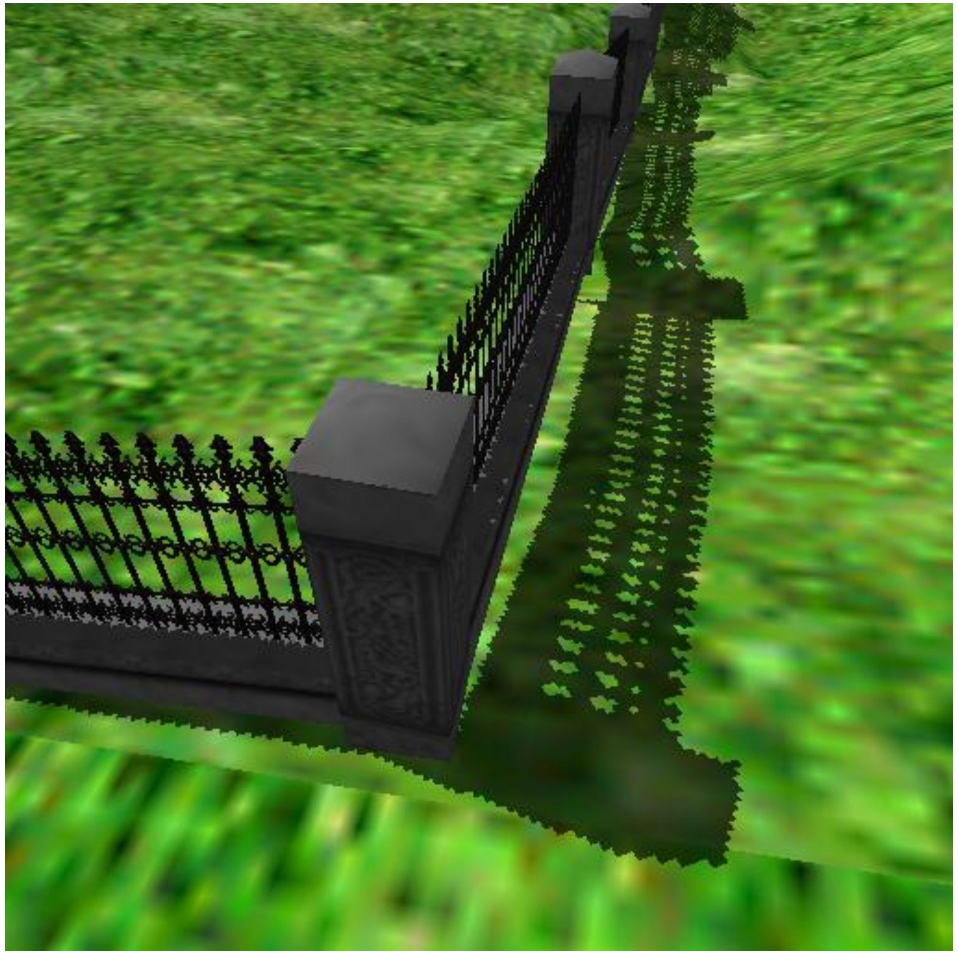
- Only include relevant objects
- Intersection body determined by:
  - Shadow casters
  - Light source frustum
  - View frustum



Fitting: Focus the Shadow Map



Unfocused



Focused

Fitting Calculation

- Model matrix **M**, light view matrix **Lv**, light proj. matrix **Lp**
- Transform intersection body by **Lp Lv**
- Calculate bounds (xmin,ymin,xmax,ymax)
- Calculate fitting matrix **F** („viewport transform“):

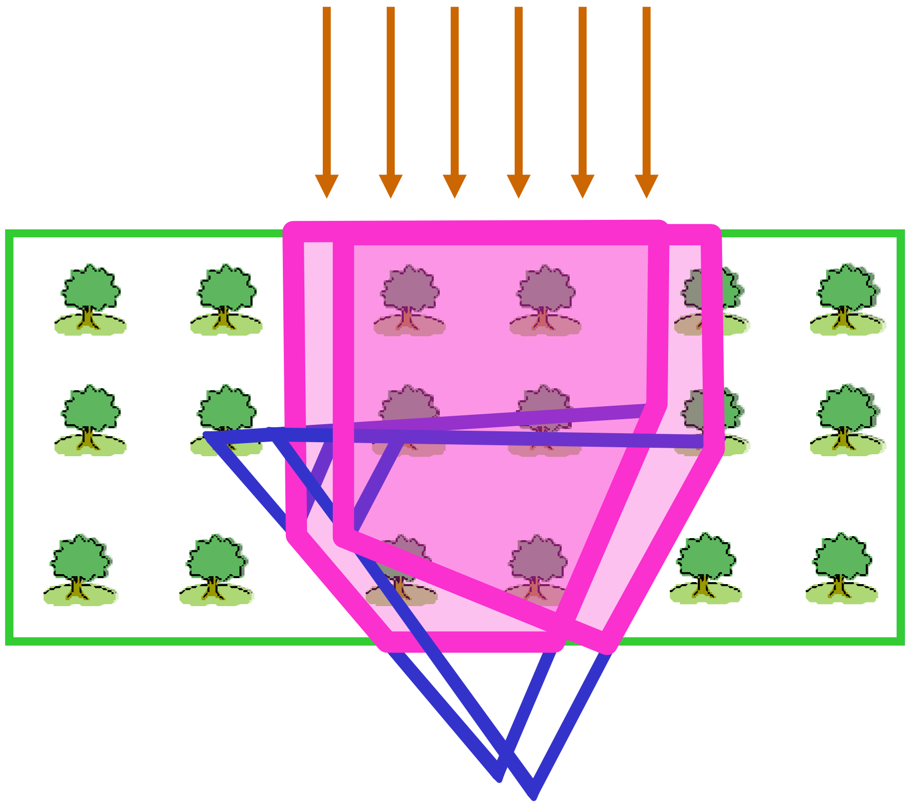
$$\mathbf{F} = \begin{pmatrix} s_x & 0 & 0 & o_x \\ 0 & s_y & 0 & o_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Shadow matrix:

  S = F Lp Lv M

$$s_x = \frac{2}{x_{\max} - x_{\min}}, \quad o_x = -\frac{s_x(x_{\max} + x_{\min})}{2}$$

Fitting: Temporal Aliasing

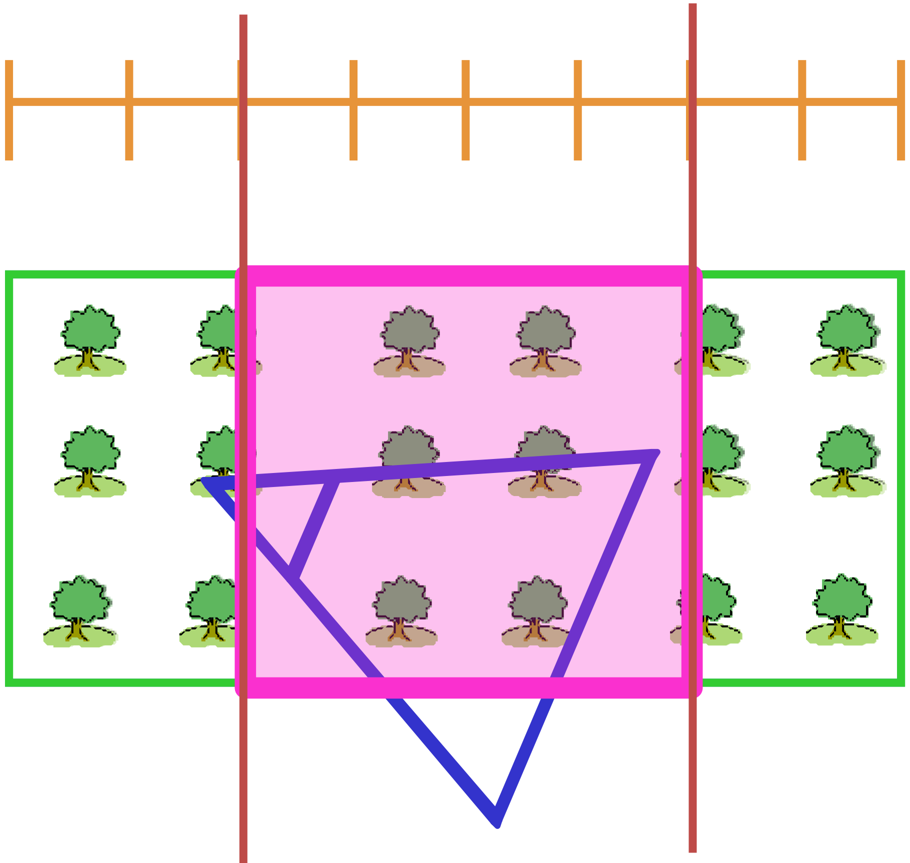


Fitting: Temporal Aliasing

- Solutions:
  - Increase initial sampling frequency
  - Make fitting adhere to texel boundaries:
    - Adjust fitting matrix **F**:
      - r: half shadow-map resolution
      - o: ox/oy

$$o' = \frac{\mathrm{ceil}(or)}{r}$$

    - Will not work for warping



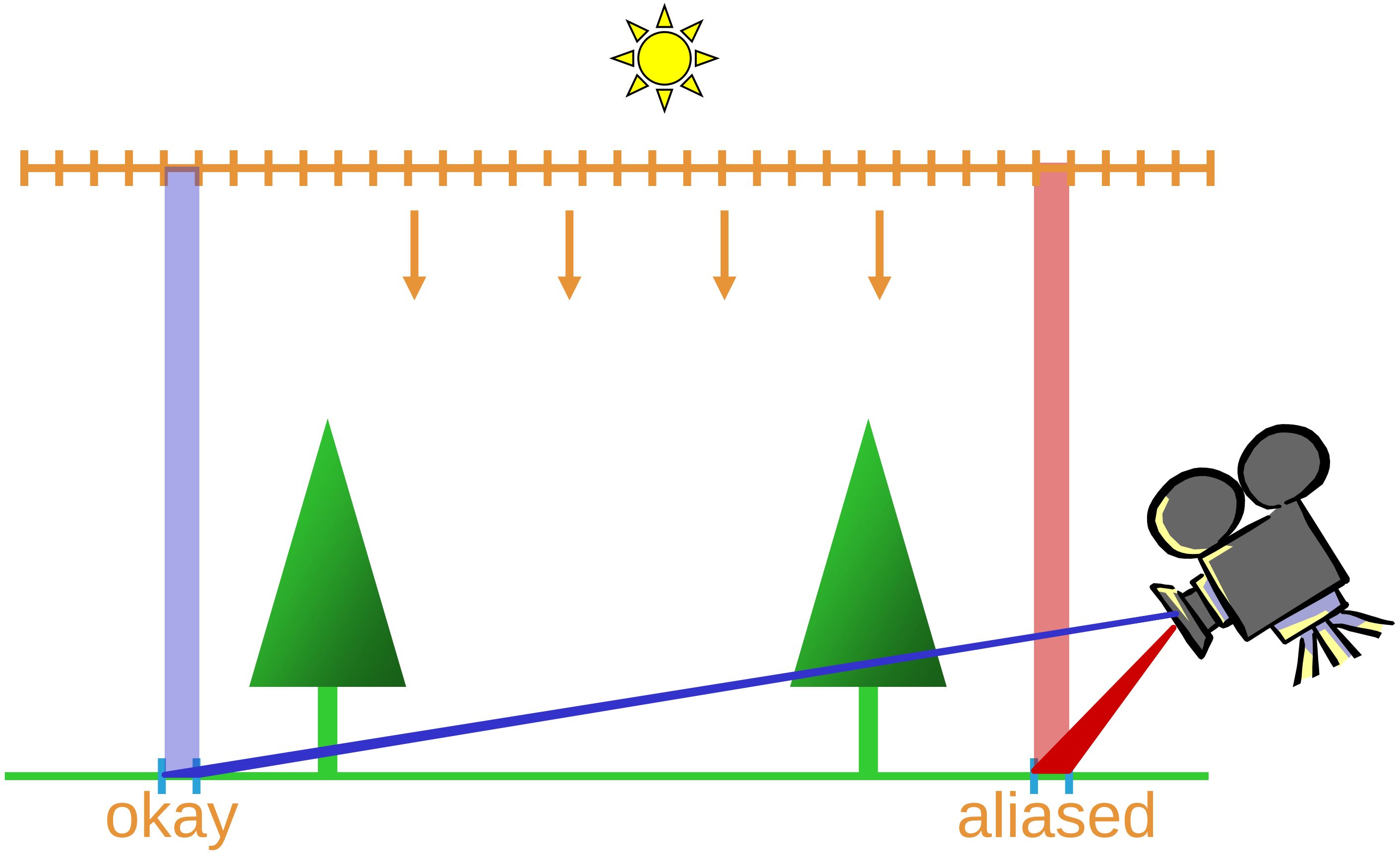
$$\frac{dp}{ds} = \frac{z_n}{z} \frac{dz}{ds} \frac{\cos\alpha}{\cos\beta}$$

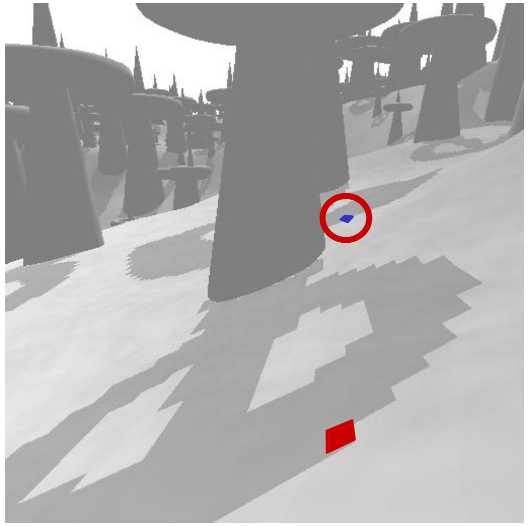
Hard Shadows

Error Analysis

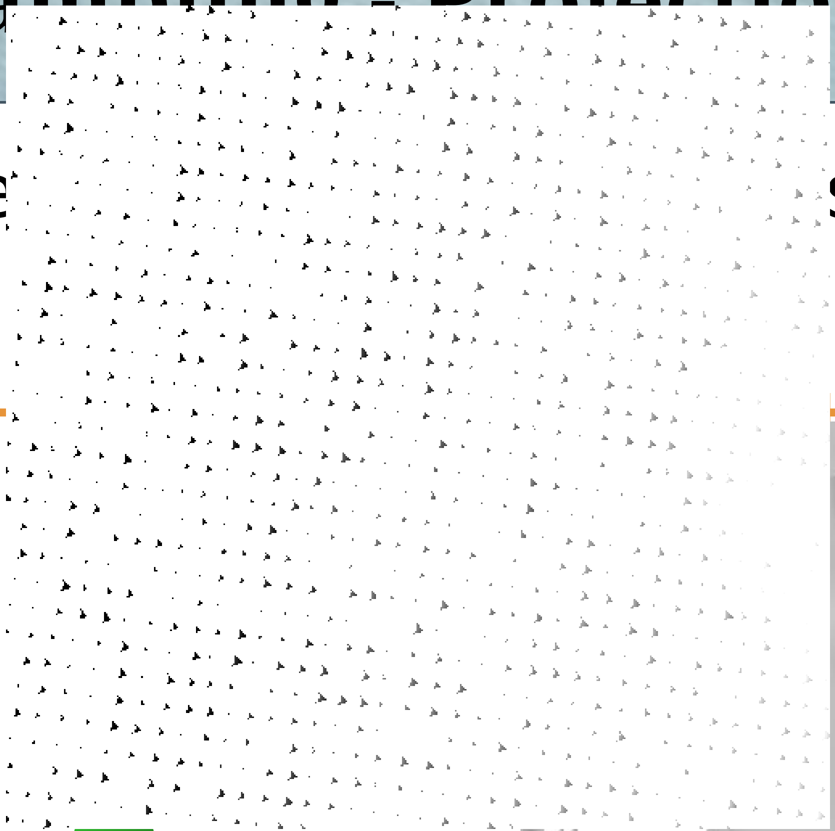
Initial Sampling - Perspective Aliasing

- Sufficient resolution far from the observer
- Insufficient resolution near the observer

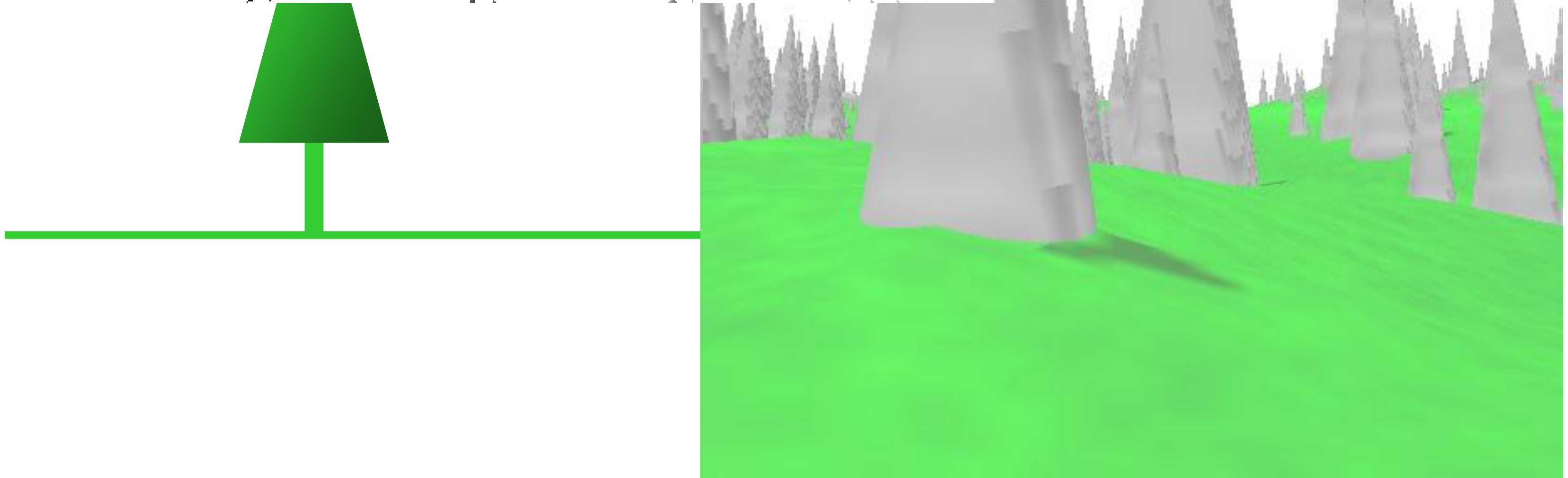




Initial Sampling - Projection Aliasing



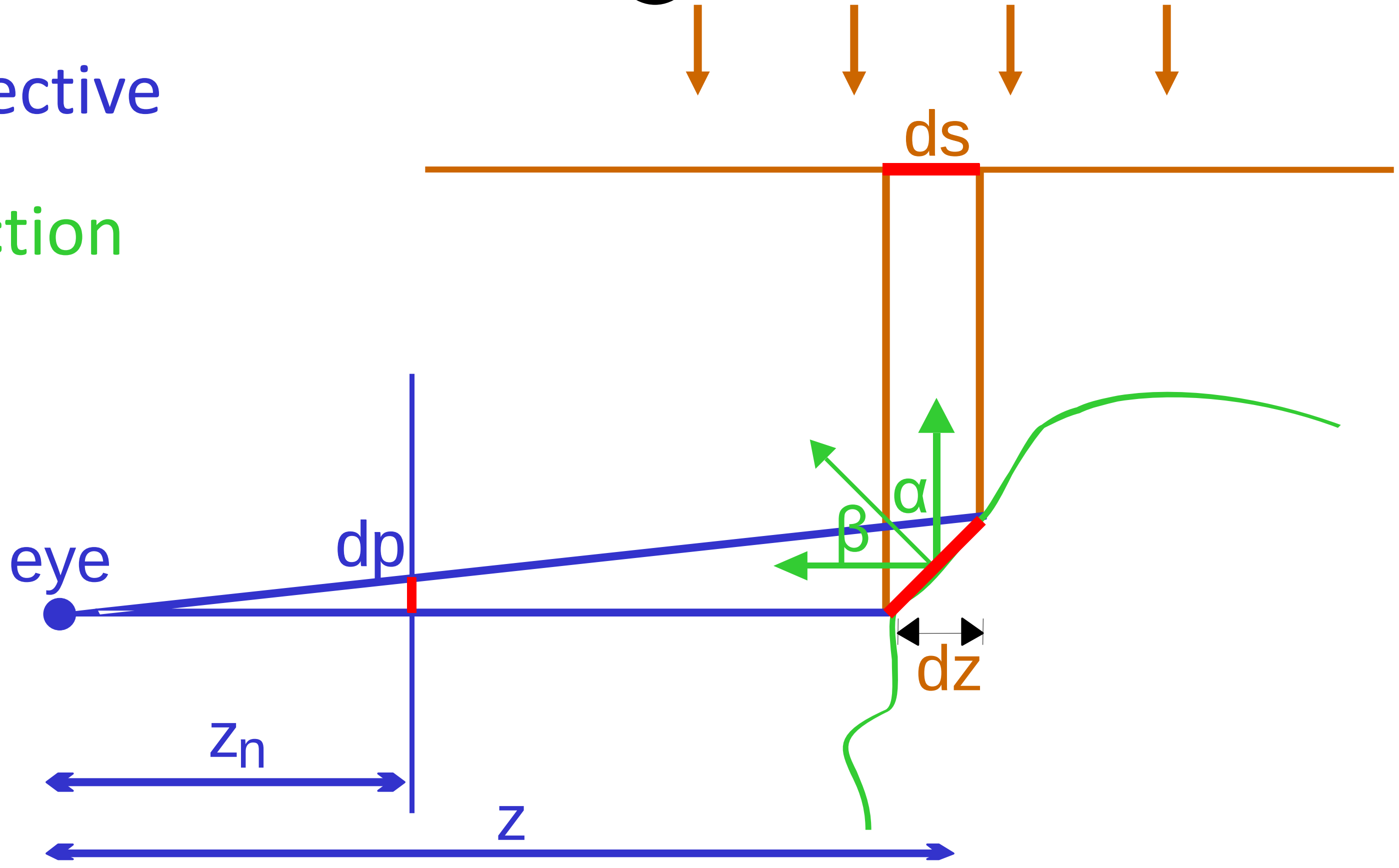
- Shadow re … shadow map

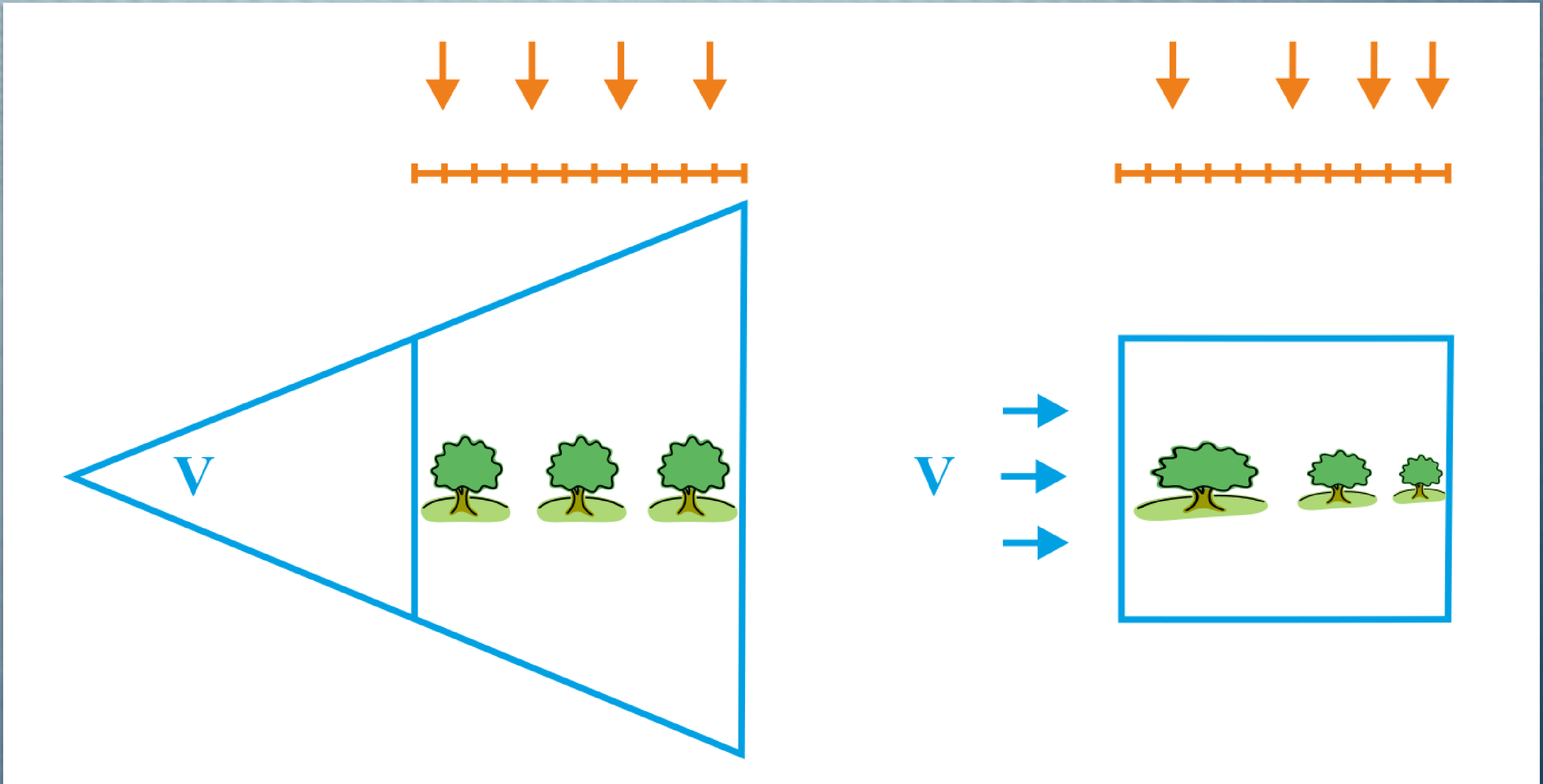


(Simple) Initial Sampling Error Analysis

- Aliasing error:
  - Parameterization
  - Perspective
  - Projection

$$\frac{dp}{ds} = \frac{z_n}{z}\frac{dz}{ds}\frac{\cos\alpha}{\cos\beta}$$



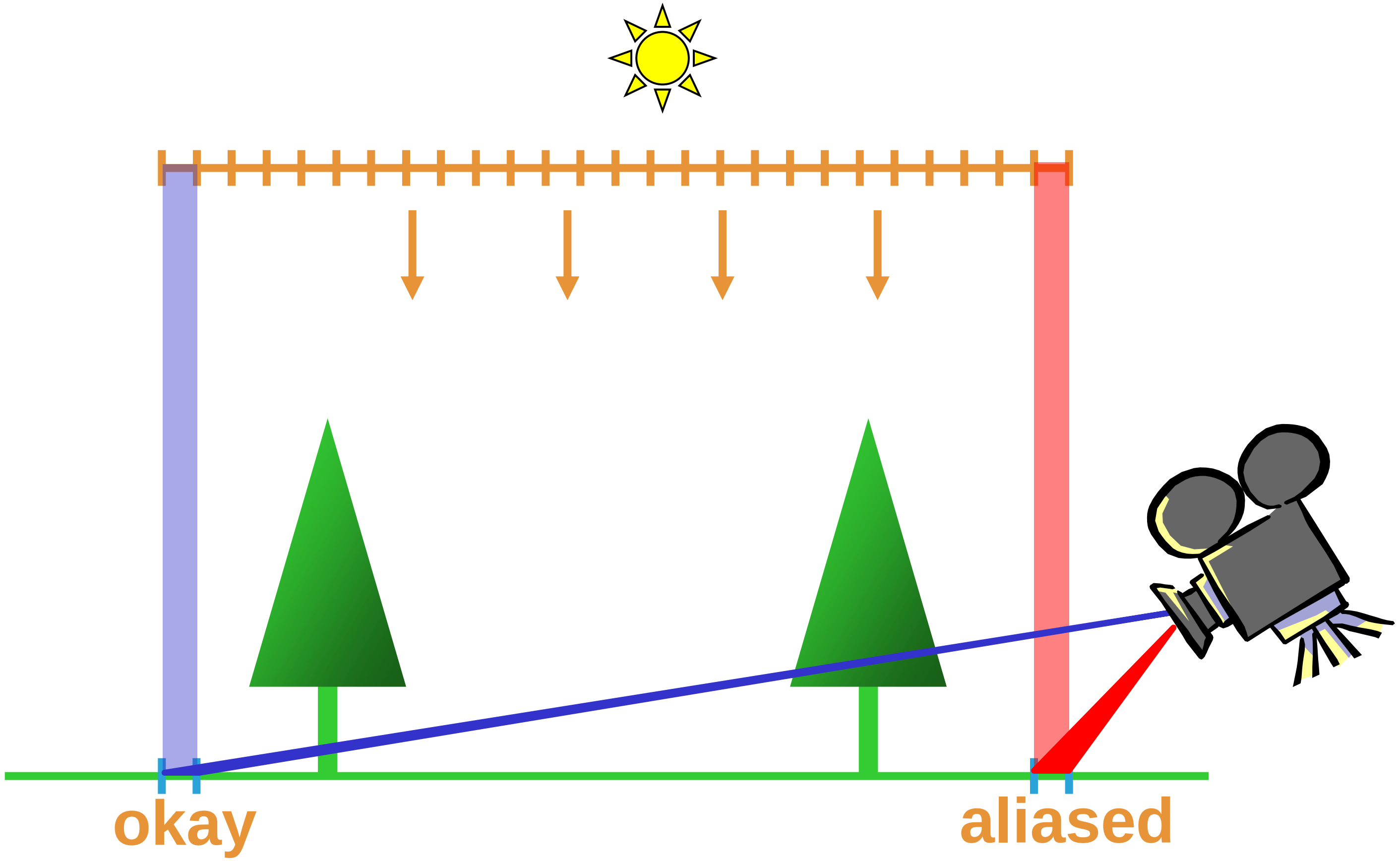


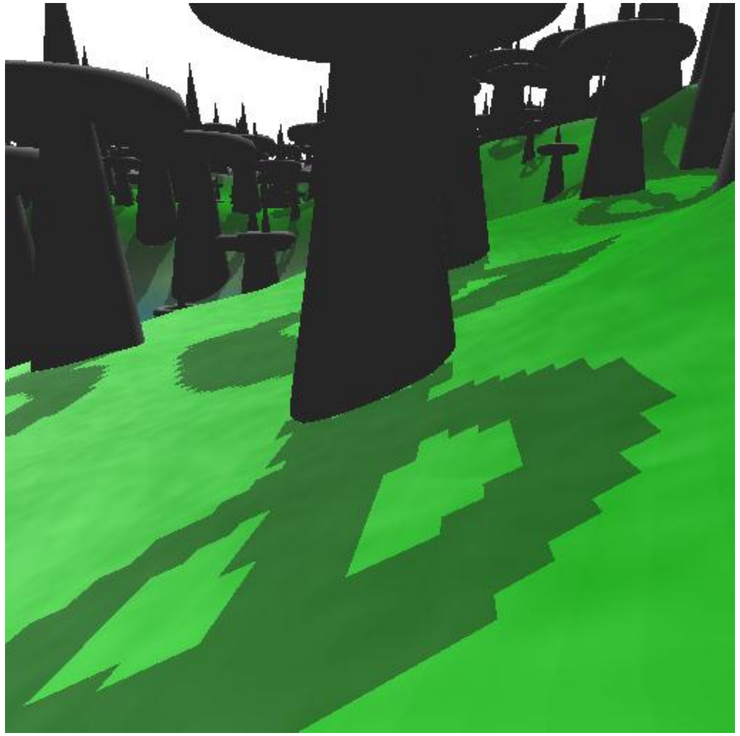
Hard Shadows

Fighting Undersampling – Warping

Solution for Perspective Aliasing

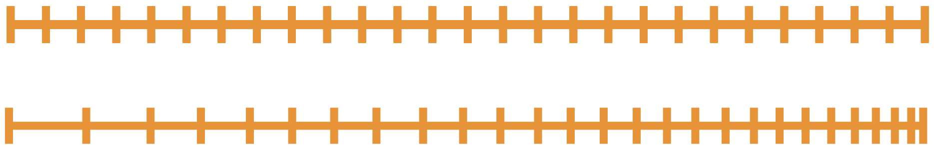
- **Insufficient** resolution near eye





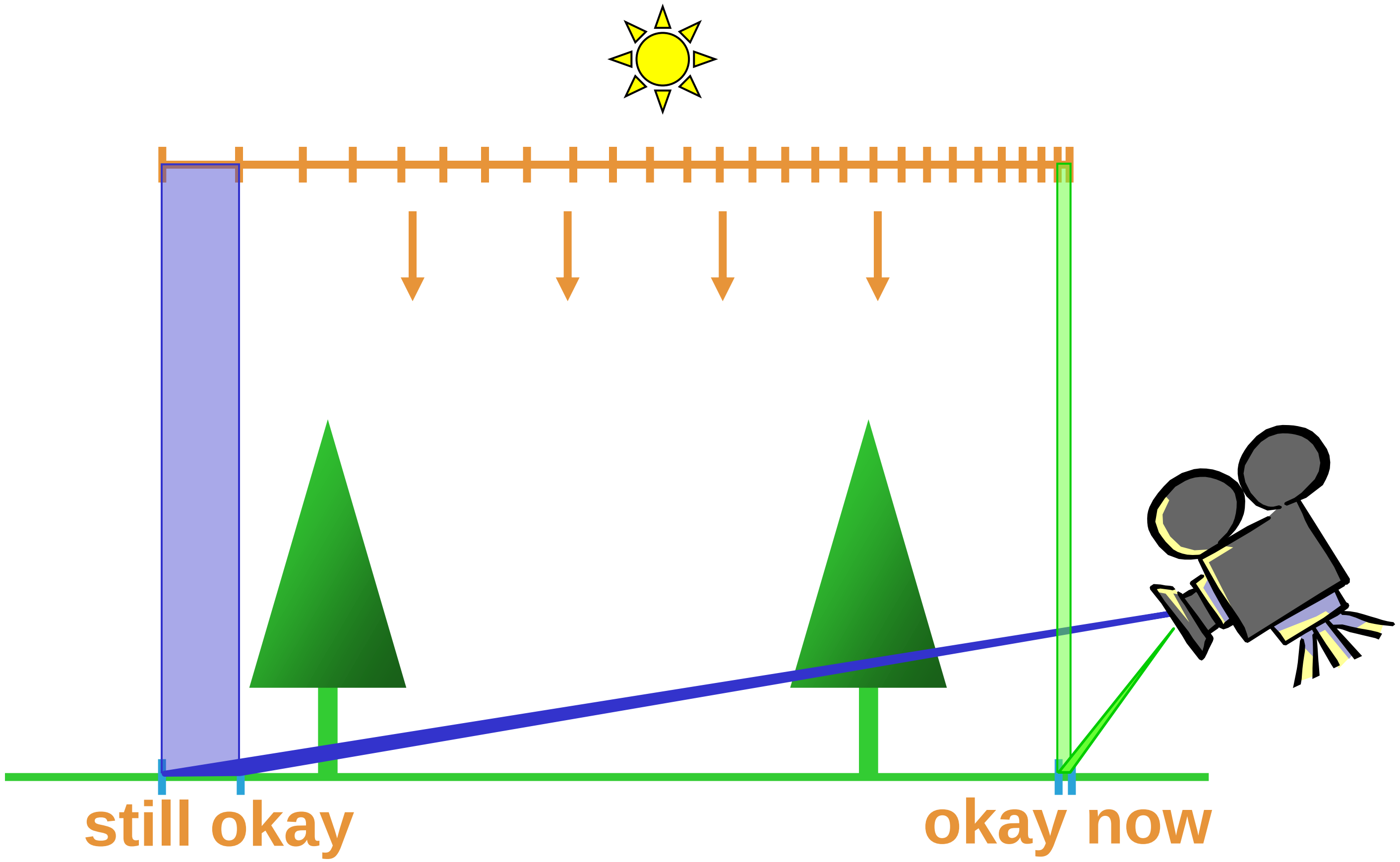
Solution for Perspective Aliasing

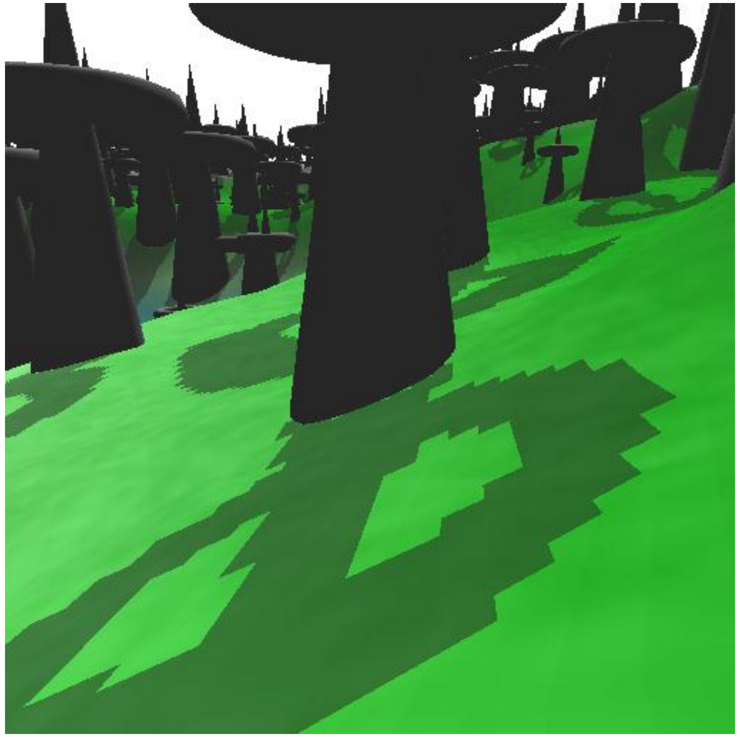
- **Insufficient** resolution near eye
- **Redistribute** values in shadow map



Solution for Perspective Aliasing

- **Sufficient** resolution near eye
- **Redistribute** values in shadow map

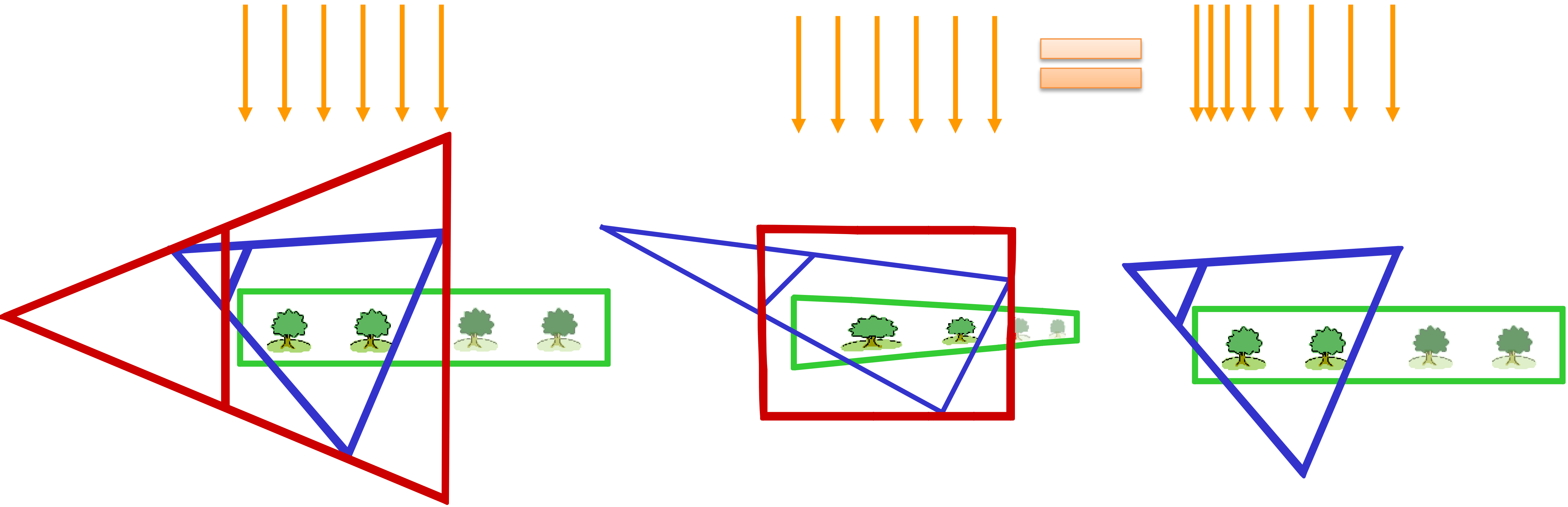




Shadow Map Warping

- Use an **additional perspective frustum**
  - **Perspective Shadow Maps (PSM)** [Stamminger & Drettakis 2002]

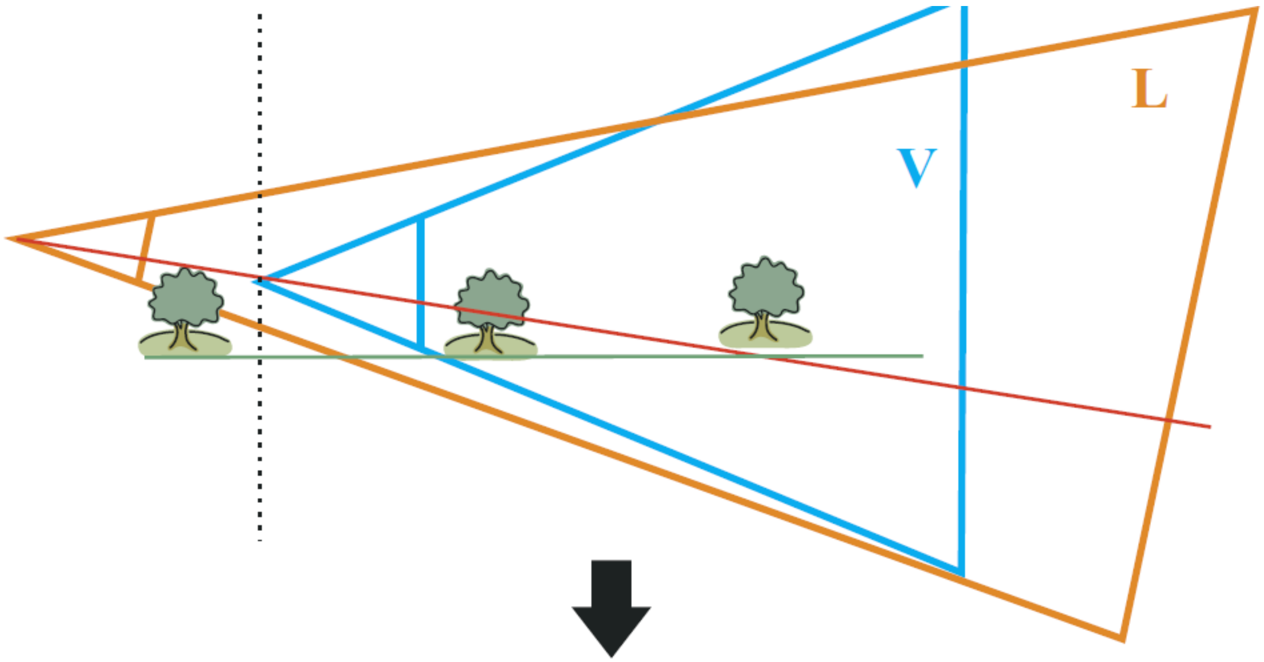
uniform shadow map | warped scene | warped shadow map



PSM: Warping Frustum = View Frustum!

→ Shadows from behind

world space

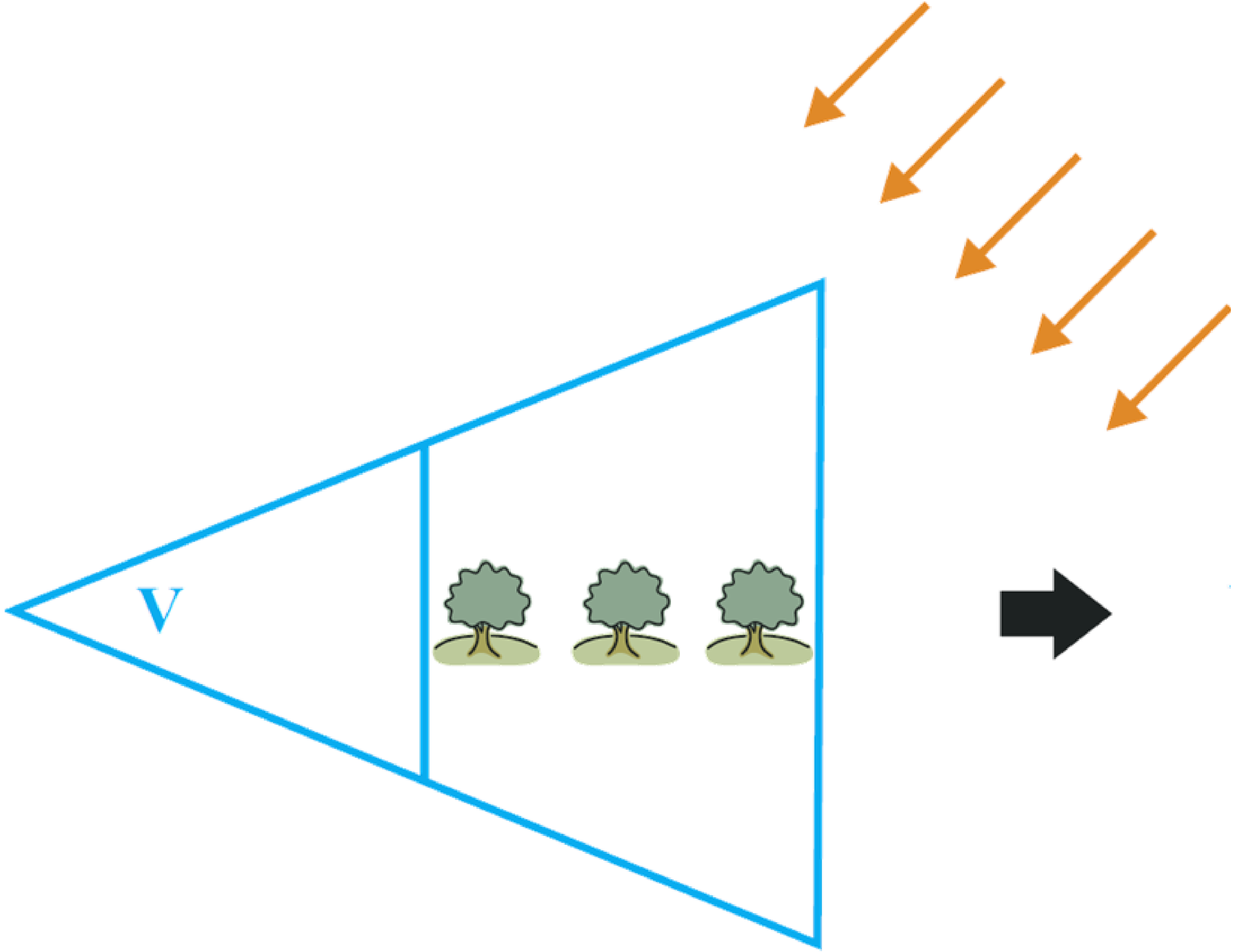


post-perspective space

PSM: Warping Frustum = View Frustum!

SIGGRAPH2013

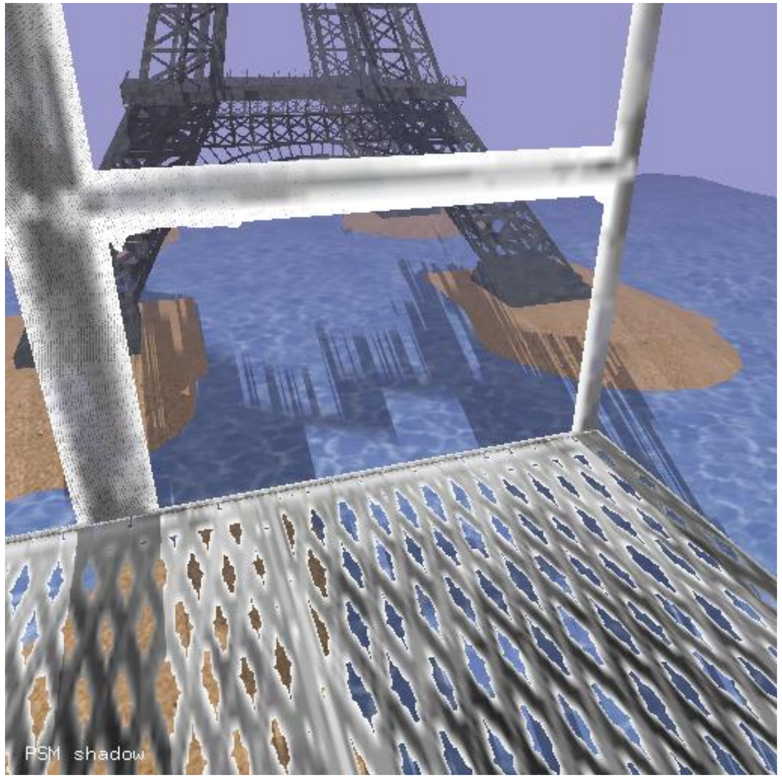
→ Light changes type



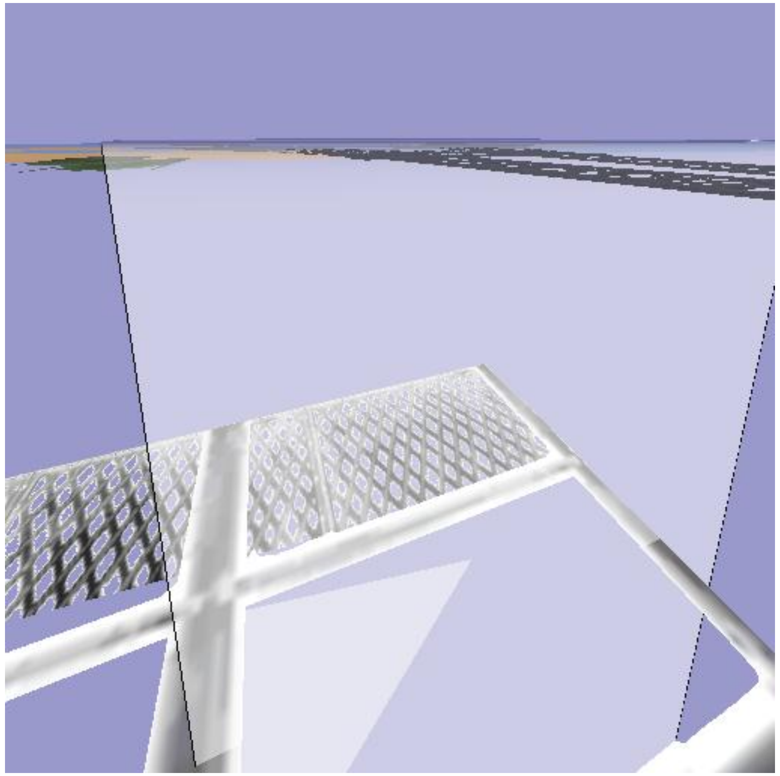
PSM: Warping Frustum = View Frustum!

→ Most severe: uneven z-distribution

- Good near viewer, very bad far away!
- Can be reduced by pushing near plane away



world space



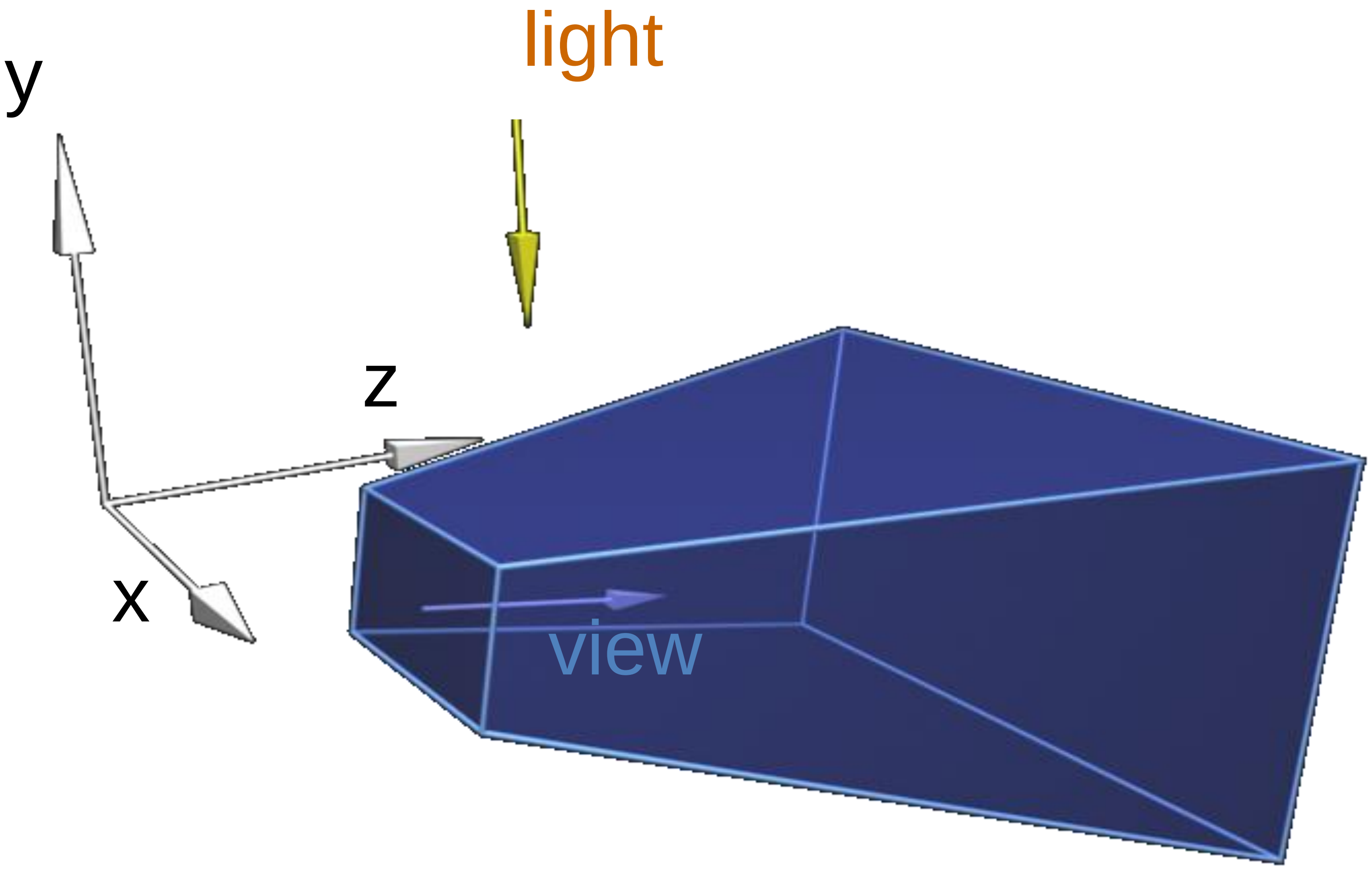
post-perspective space

PSM Problems?

- Enter **Light-Space Perspective Shadow Maps (LiSPSM)** [Wimmer et al. 2004]
  - Perspective frustum defined relative to light space
  - Optimize warping strength to improve z-distribution
  - Similar to Trapezoidal Shadow Maps (TSM) [Martin & Tan 2004]

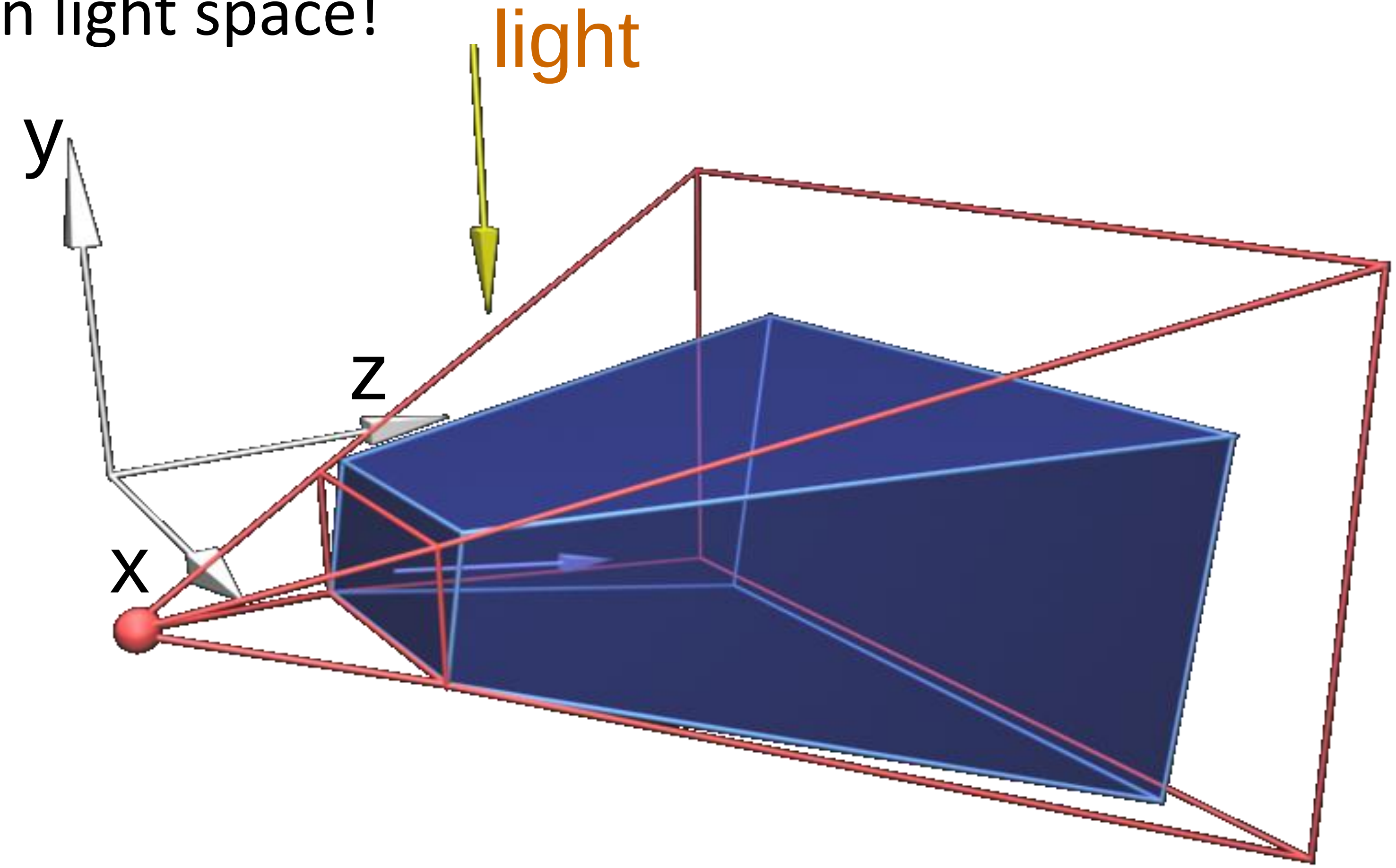
Light-Space Perspective Shadow Mapping (LiSPSM)

- Light and view vector define yz-plane



Light-Space Perspective Shadow Mapping (LiSPSM)

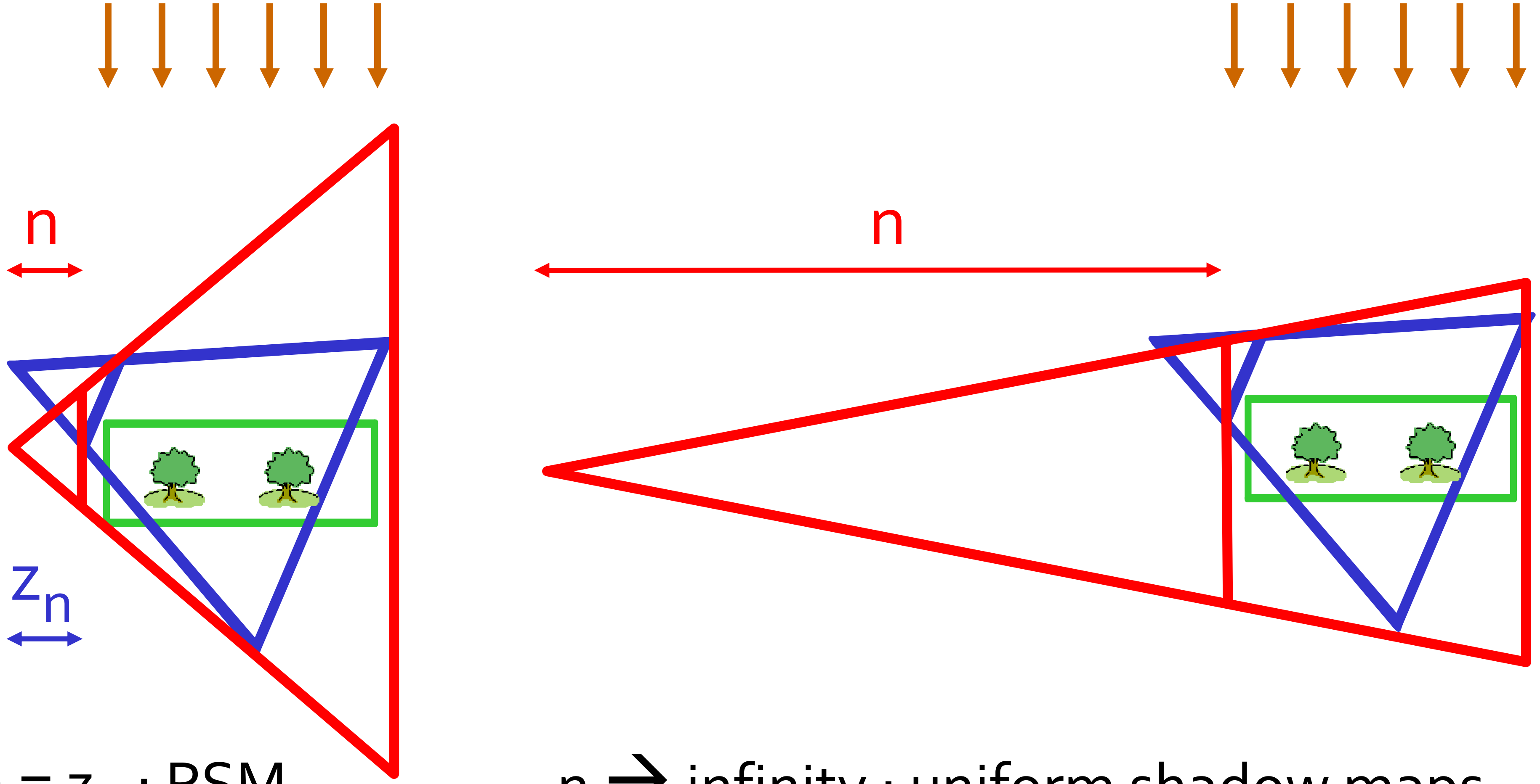
- Find a tight perspective frustum on focused region
  - In light space!



Free Parameter n

SIGGRAPH2013

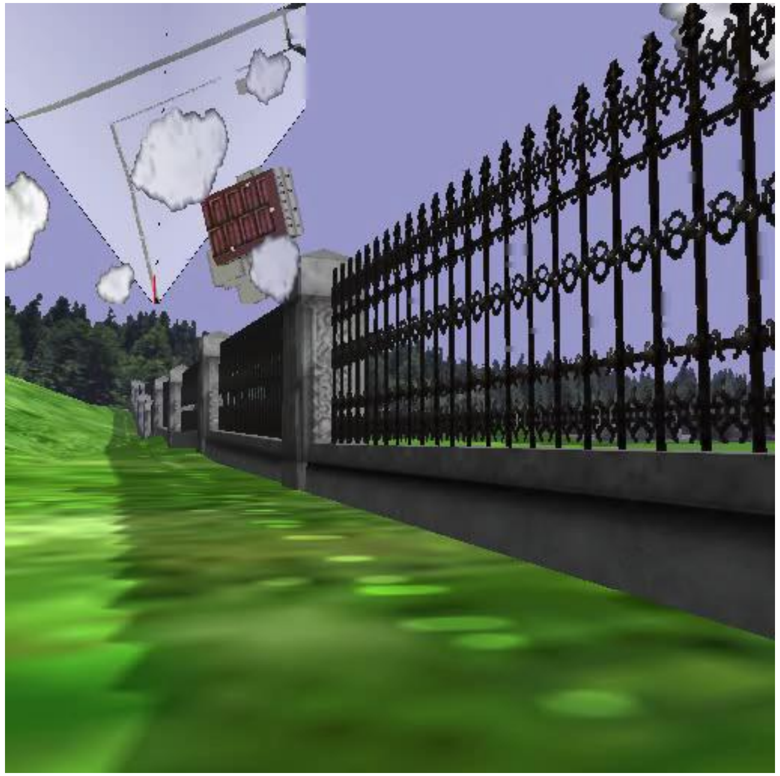
- Controls warping effect



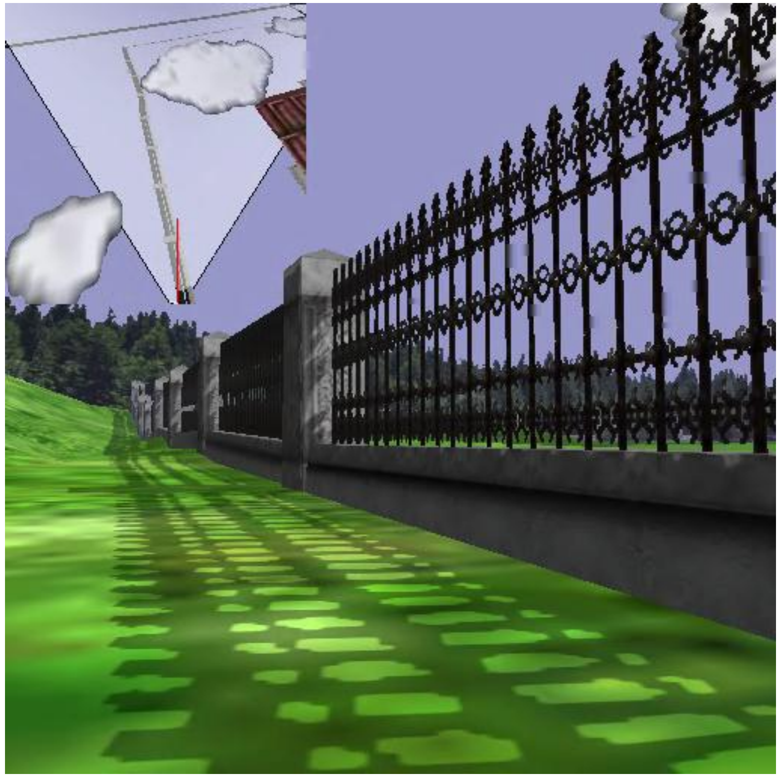
$n = z_n$: PSM

n → infinity : uniform shadow maps

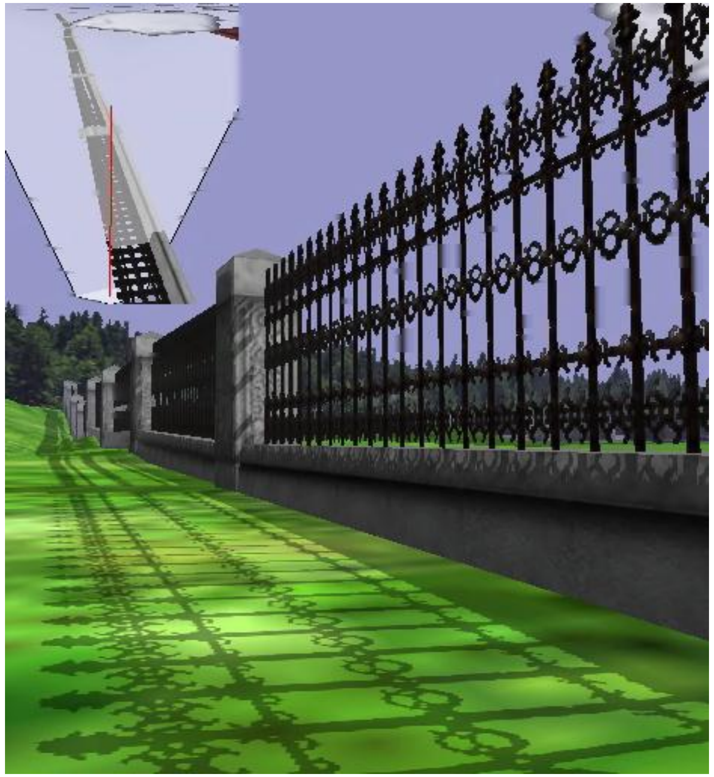
Free Parameter n



very big n



optimal n?



very small n

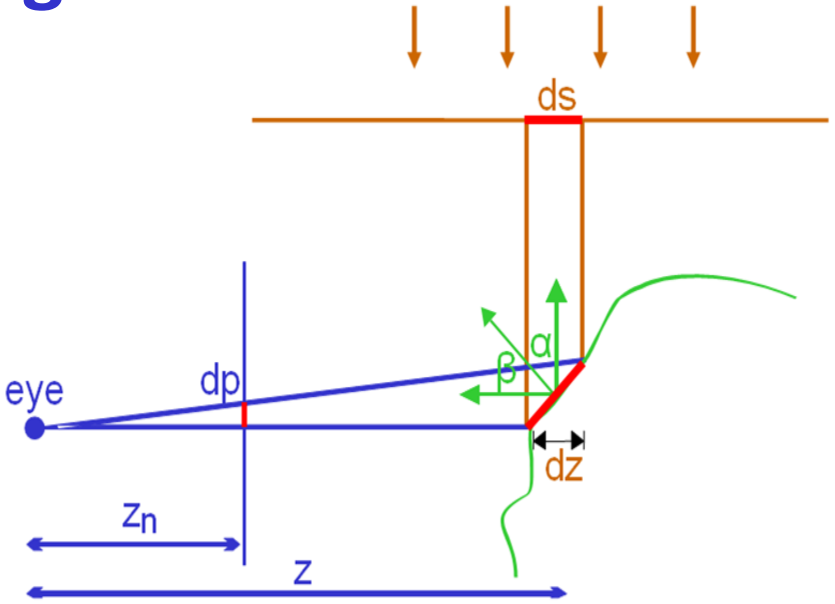
LiSPSM Matrix

- Standard light matrix: **S = F Lp Lv M**
- Orient shadow map along view vector: **R**
- Transform intersection body by **R Lp Lv**
- Find near/far, and choose warping strength **n**
- Calculate warping matrix: **Wp Wv**
- Calculate **F** using **Wp Wv R Lp Lv**
- Warped shadow matrix **Sw** = **F Wp Wv R Lp Lv M**
- Use **Sw** both for shadow-map generation and rendering

How to Choose the Free Parameter n?

- Recall error analysis
- $\frac{dp}{ds} > 1$ → shadow map undersampling
- **Projection aliasing** cannot be changed
- Counter **perspective aliasing** with new shadow map **parameterization** $s(z, \ldots)$
- Goal: $\frac{dp}{ds} \sim 1$

$$\frac{dp}{ds} = \frac{z_n}{z} \frac{dz}{ds} \frac{\cos\alpha}{\cos\beta}$$



Error Analysis

$$\frac{dp}{ds} = \frac{z_n}{z} \frac{dz}{ds} \frac{\cos\alpha}{\cos\beta}$$

- Perfect: logarithmic re-parameterization

$$s \sim \log z \quad \Rightarrow \quad \frac{ds}{dz} \sim \frac{1}{z} \quad \Rightarrow \quad \frac{dp}{ds} \sim 1$$

  - Hardware support? [Lloyd 2007, 2008]

- Uniform shadow maps

$$s \sim z \quad \Rightarrow \quad \frac{ds}{dz} \sim 1 \quad \Rightarrow \quad \frac{dp}{ds} \sim \frac{1}{z}$$

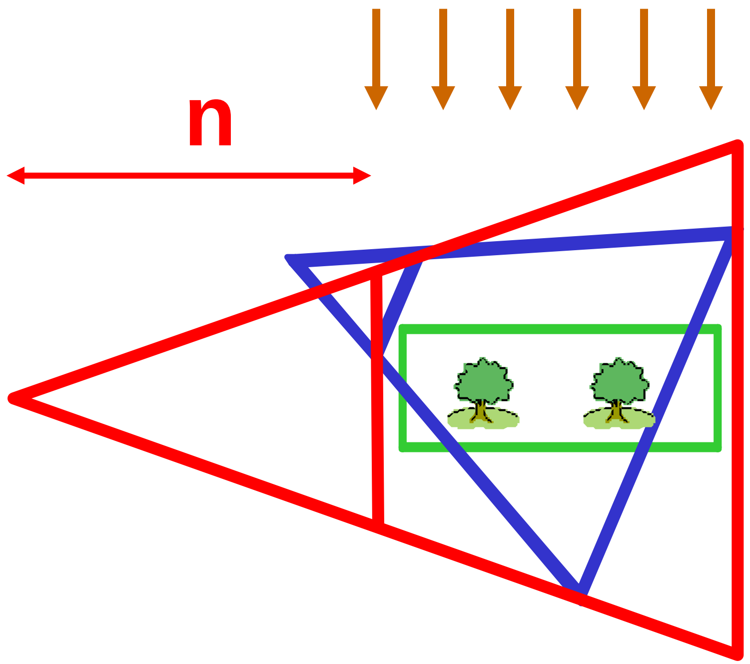
- Perspective shadow maps

$$s \sim \frac{1}{z} \quad \Rightarrow \quad \frac{ds}{dz} \sim \frac{1}{z^2} \quad \Rightarrow \quad \frac{dp}{ds} \sim z$$

  - Linear increase in error!

Error Analysis: LiSPSM Optimal Choice

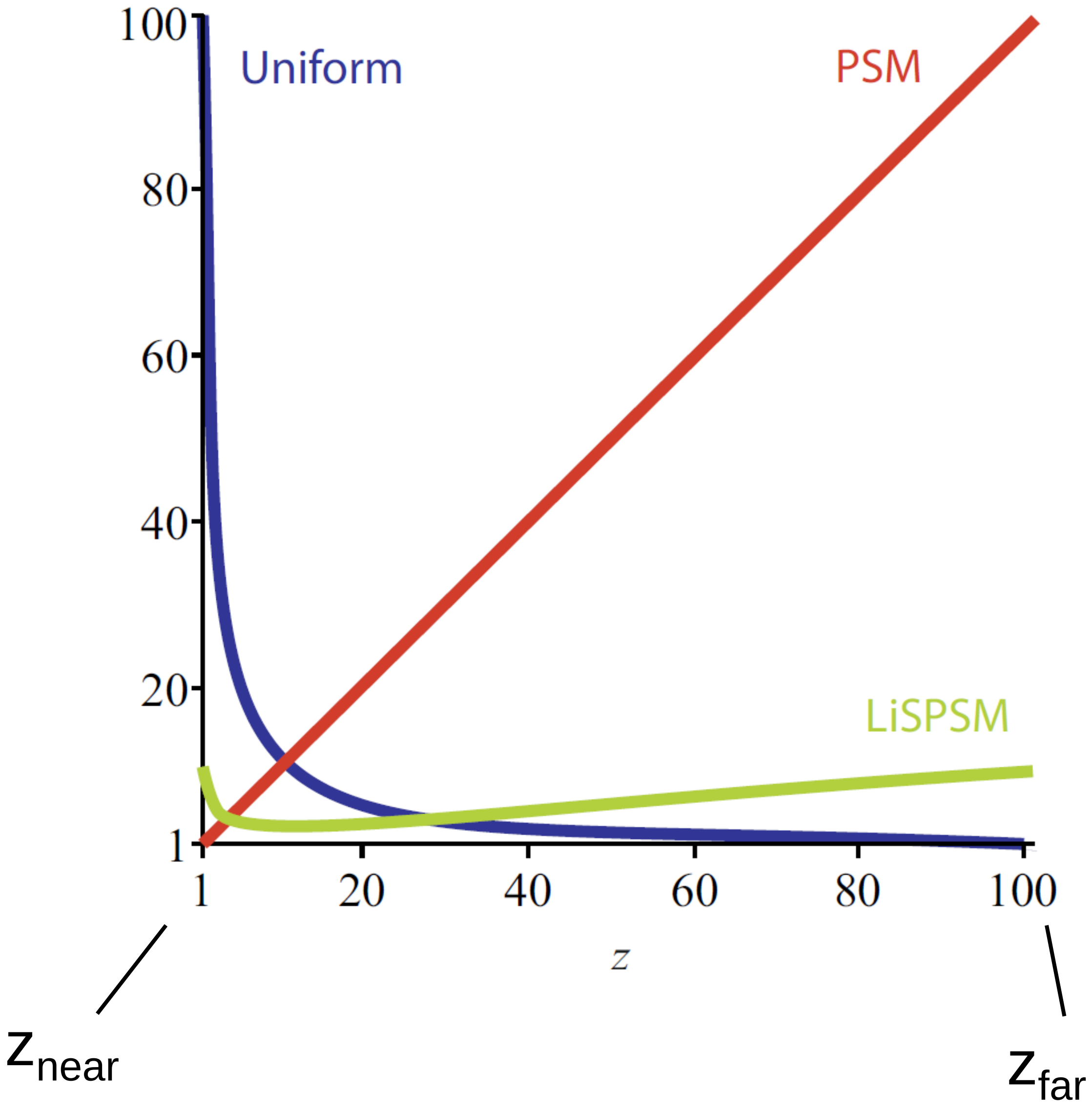
- For LiSPSM, $\frac{dp}{ds}$ depends on n
  - Gives $\frac{dp}{ds}$ between uniform and perspective
- Optimal choice:



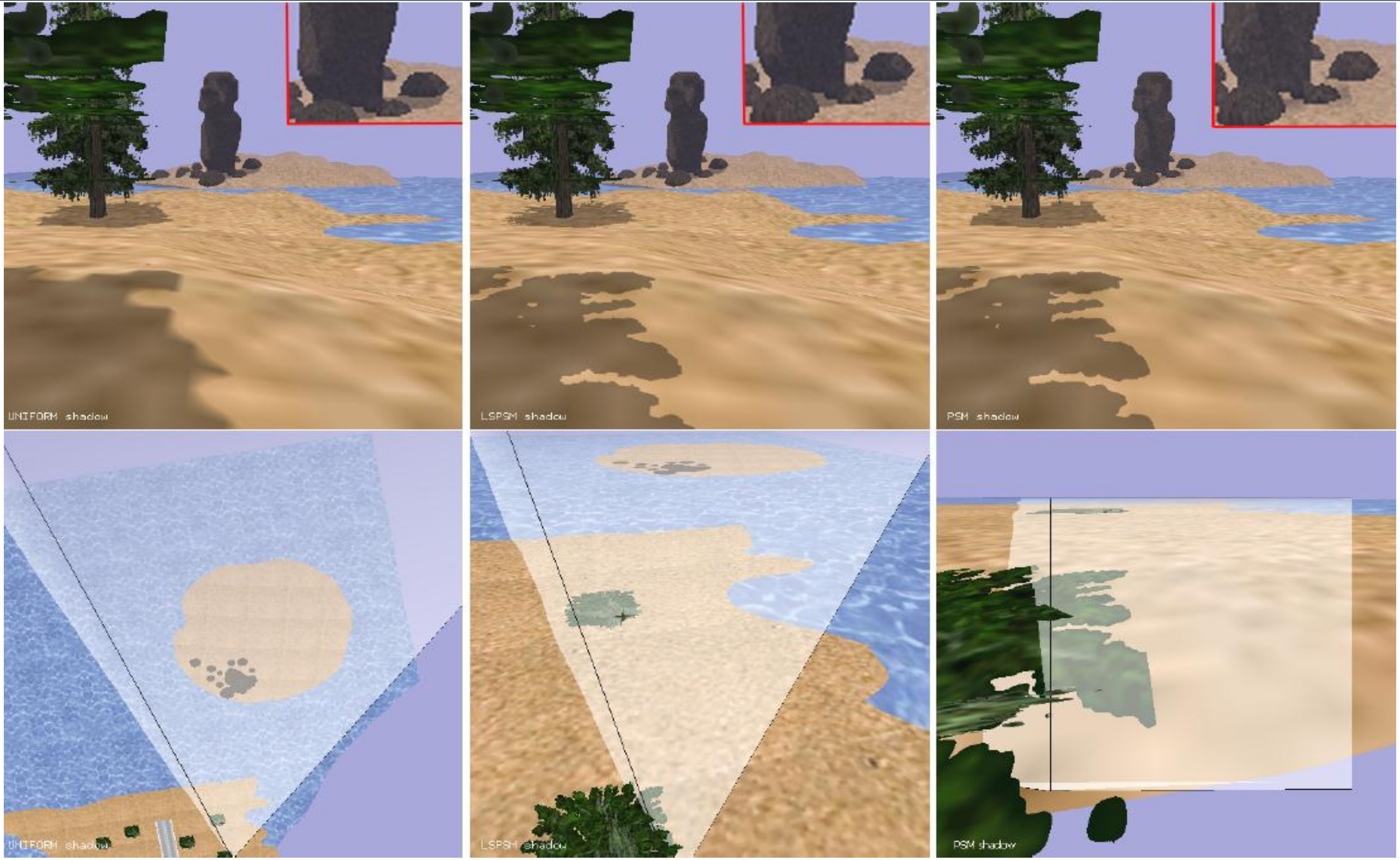
$$n_{opt} = z_n + \sqrt{z_f z_n}$$

Error Comparison

- LiSPSM optimal choice
- Measured along view dir
- LiSPSM vs PSM
  → for same depth range, LiSPSM error much lower



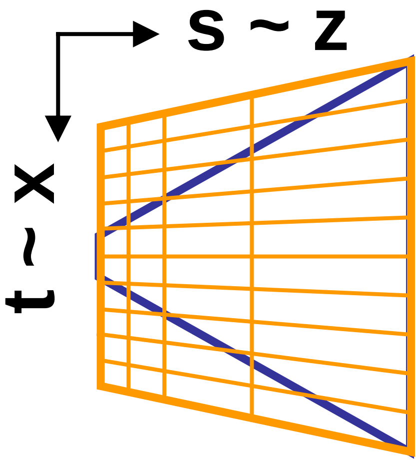
Comparison



Uniform LiSPSM PSM

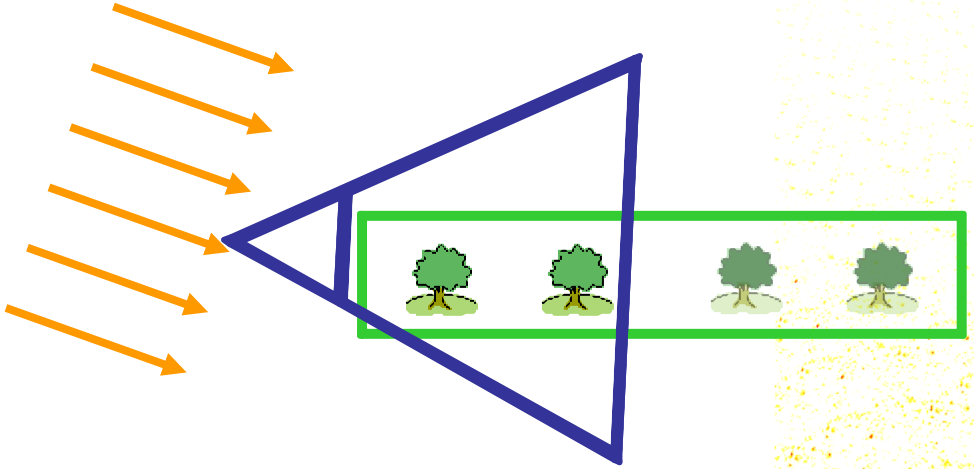
Error Comparison

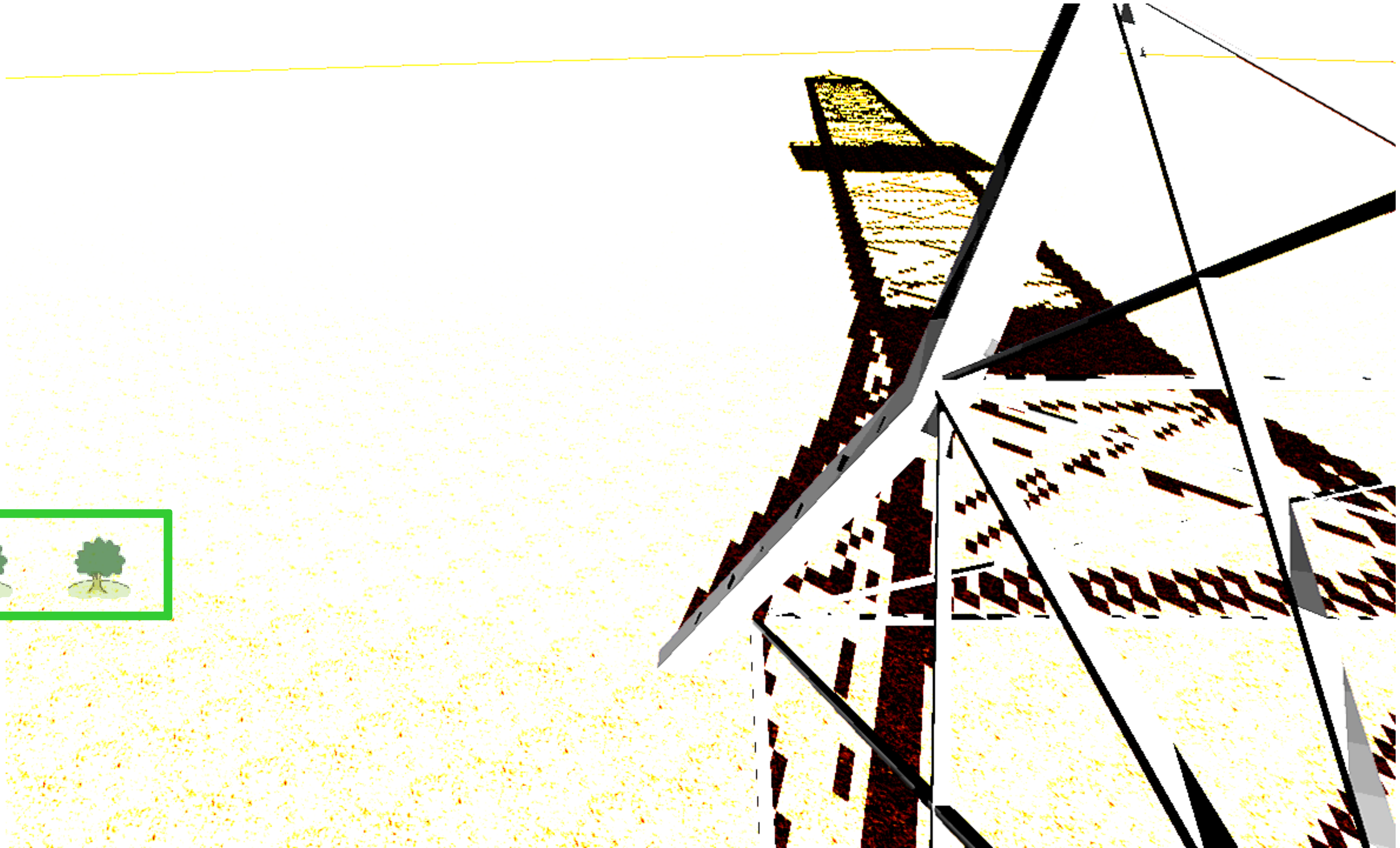
- Caveat: only measured along view direction
- What about $\frac{dp}{dt}$?
  - $\frac{dp}{dt} \sim 1$ for PSM, slightly worse for LiSPSM
- More advanced analysis was done in [Lloyd 2006]
  - Result: “storage factor” constant for n in [z_n, n_{opt}]
  - But: best error distribution for nopt

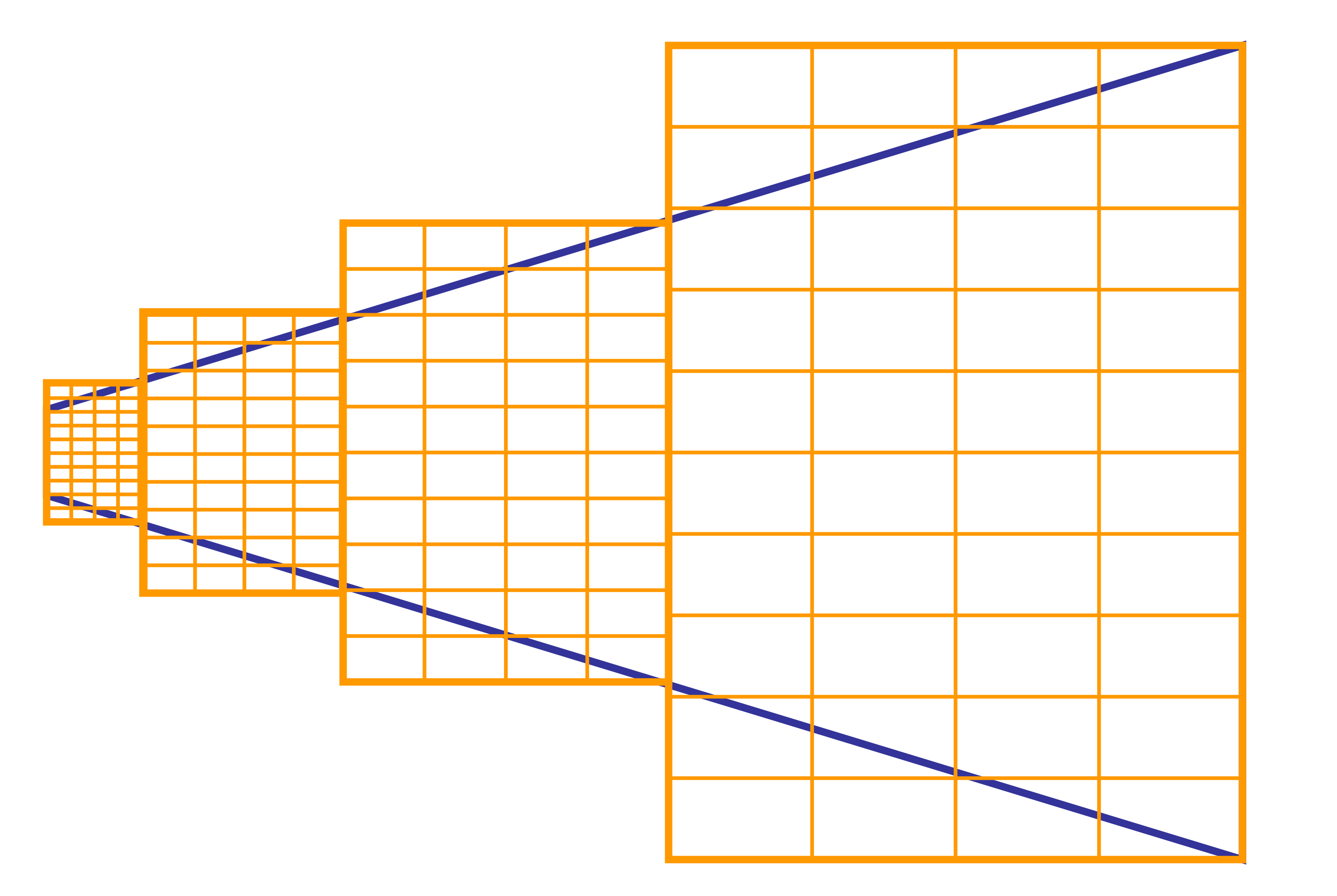


Warping: Problems

- Only works if large z-range visible from light
- Dueling frusta case





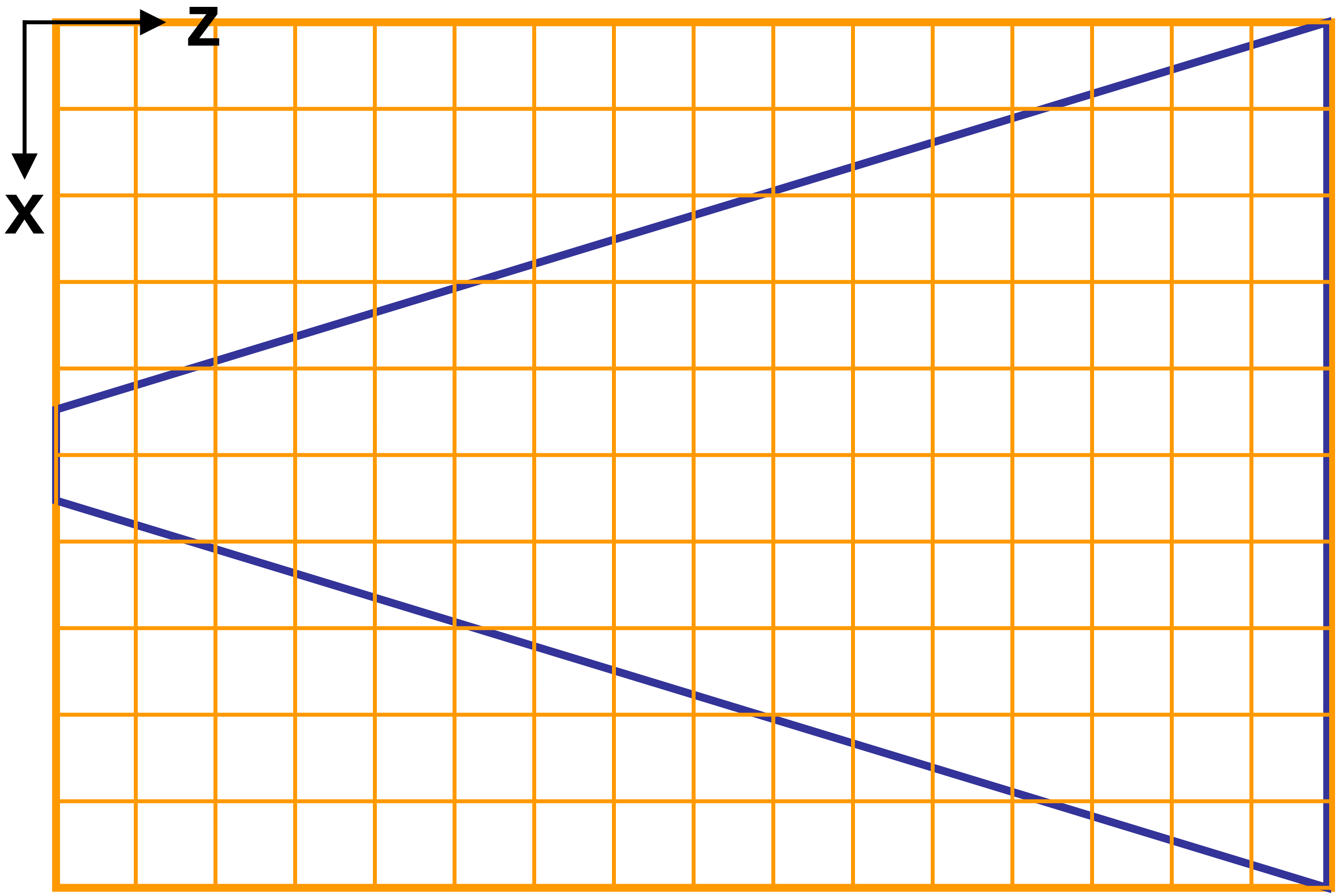


Hard Shadows

Fighting Undersampling – Partitioning

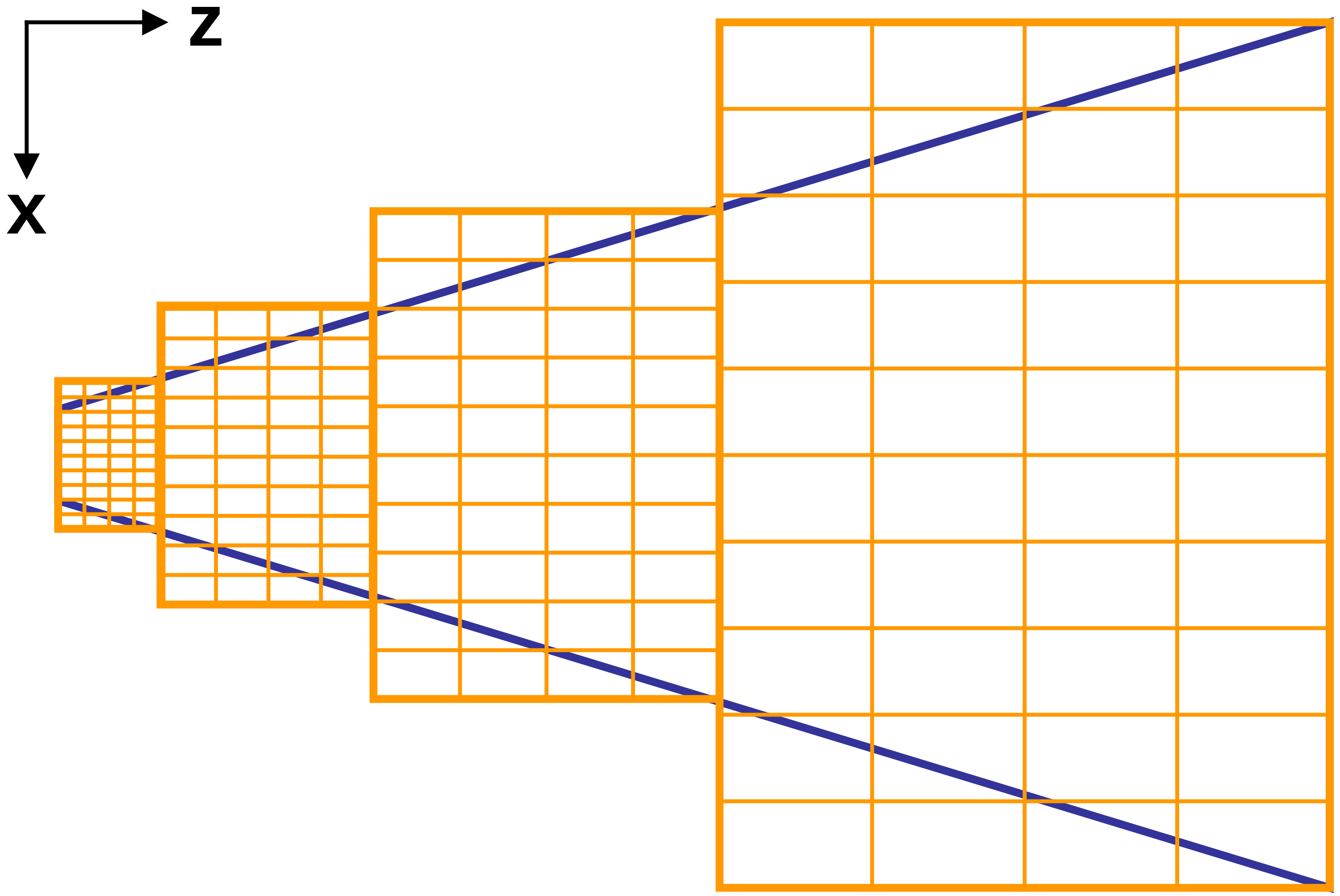
Z-Partitioning: Idea

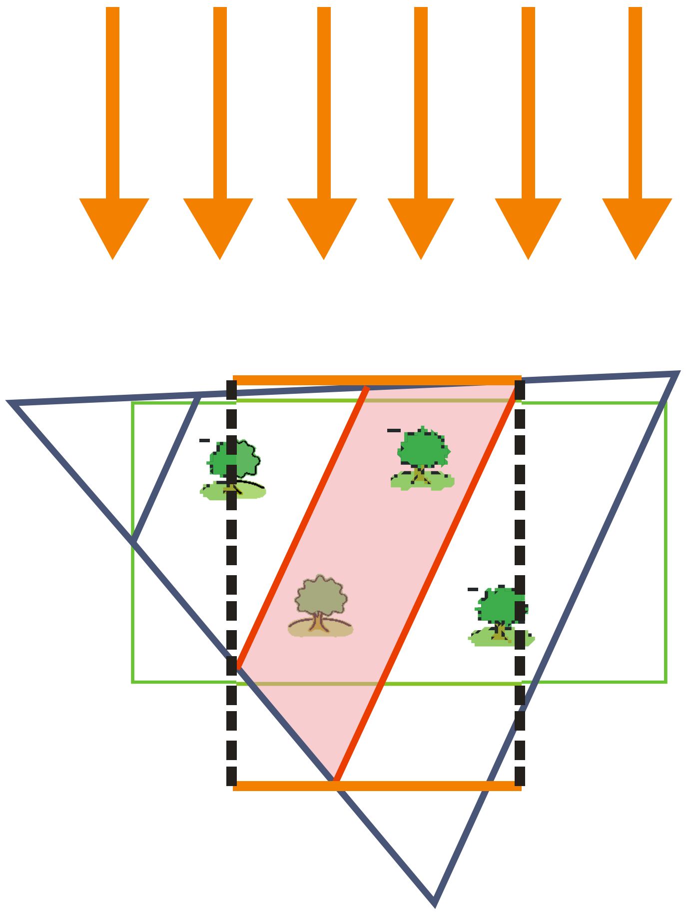
- Parallel Split Shadow Maps [Zhang 2007]
- Cascaded Shadow Maps [Engel 2007][Zhang2009]



Z-Partitioning

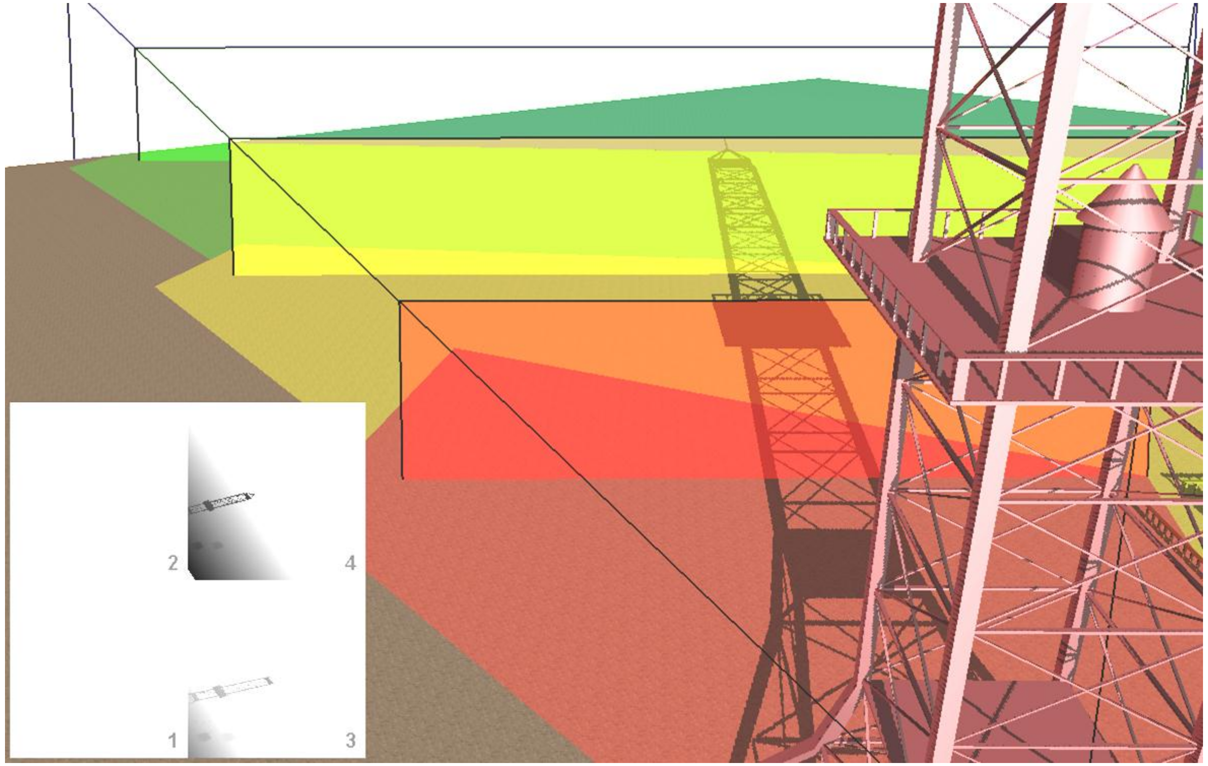
- Partition view frustum into n sub-frusta
- Calculate separate shadow map for each

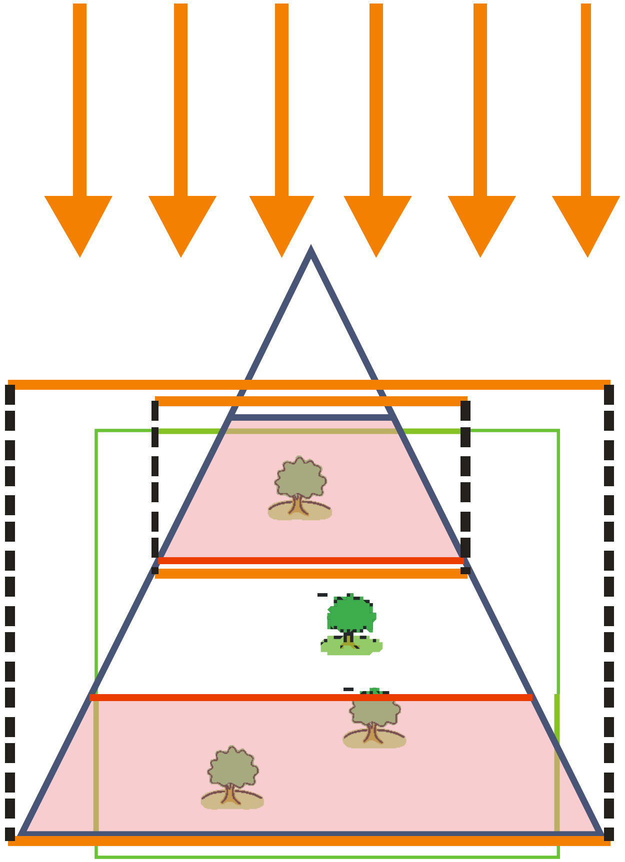




Z-Partitioning

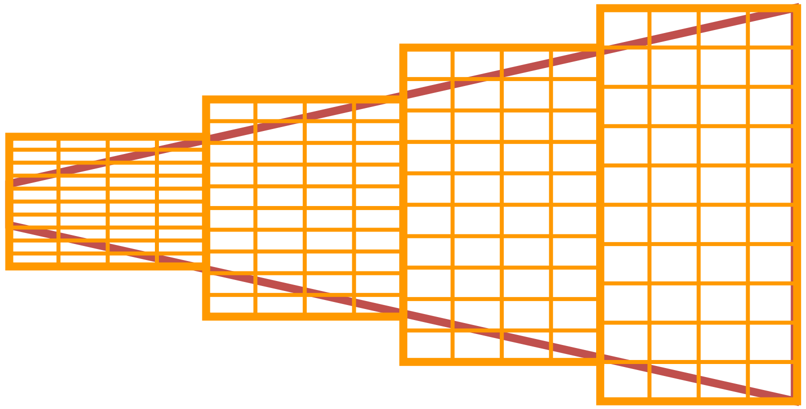
- Works even in cases where warping fails
  - Light from behind, dueling frusta





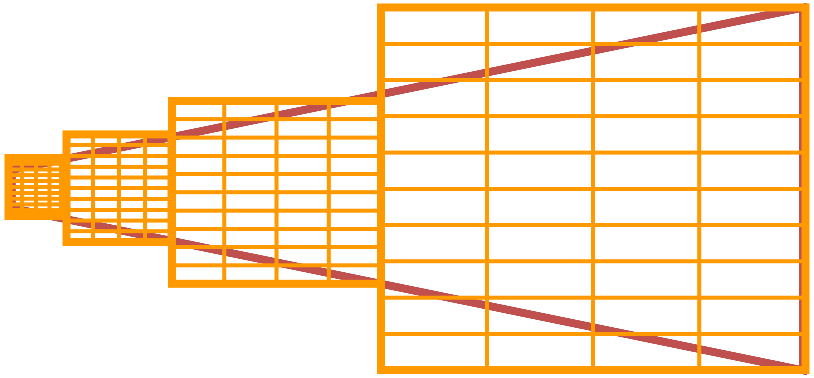
Z-Partitioning

- How to choose partition sizes?
  - Uniform

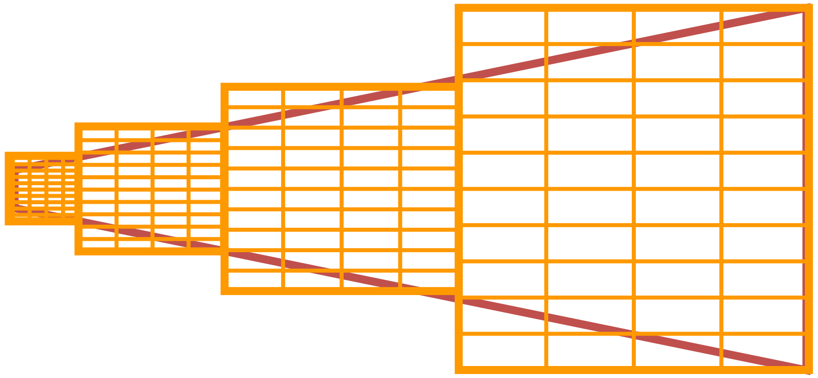


  - Logarithmic/self-similar

$$C_i = z_n \left(\frac{z_f}{z_n} \right)^{\frac{i}{m}}$$

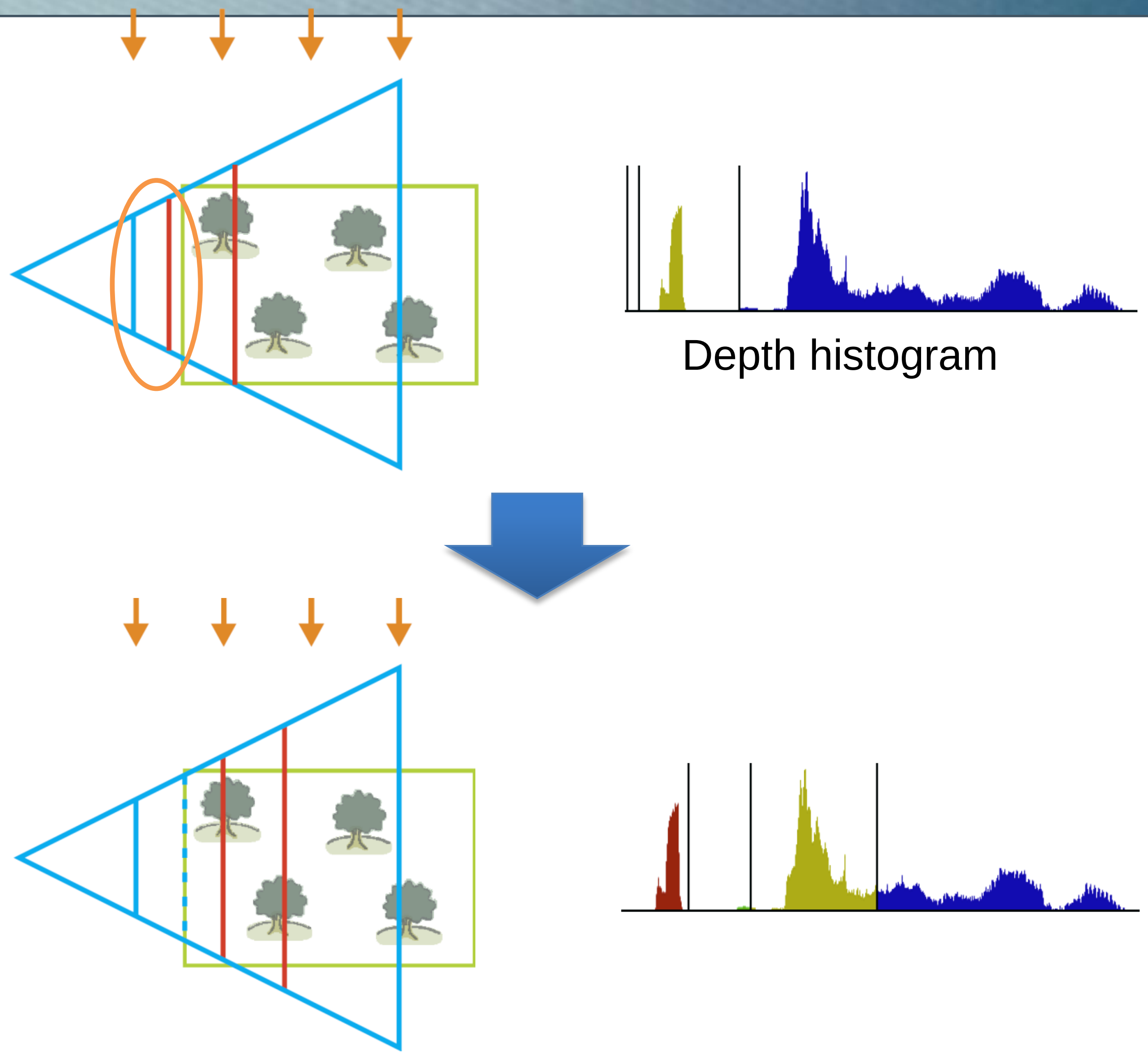


  - Linear blend between the two



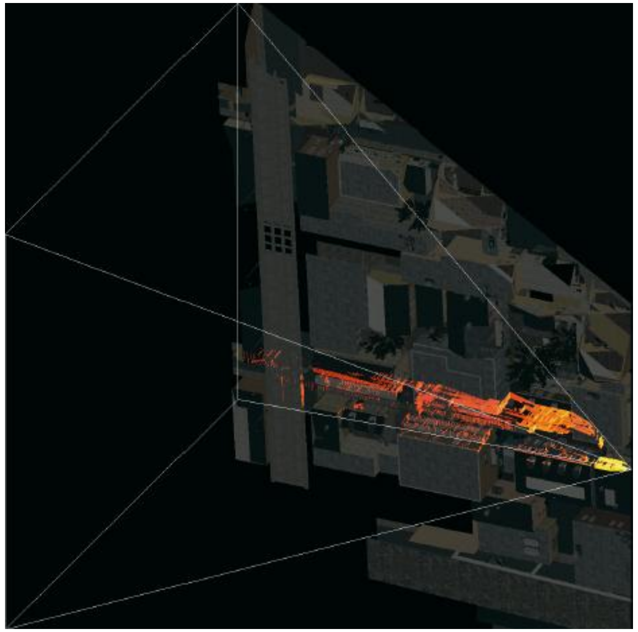
Z-Partitioning – Sample Distribution Maps [Lauritzen et al. 2011]

- Optimal partition: logarithmic
  - Problem: empty areas
- Solution: scene analysis
  - Get min/max-z
  - Adjust depth bounds

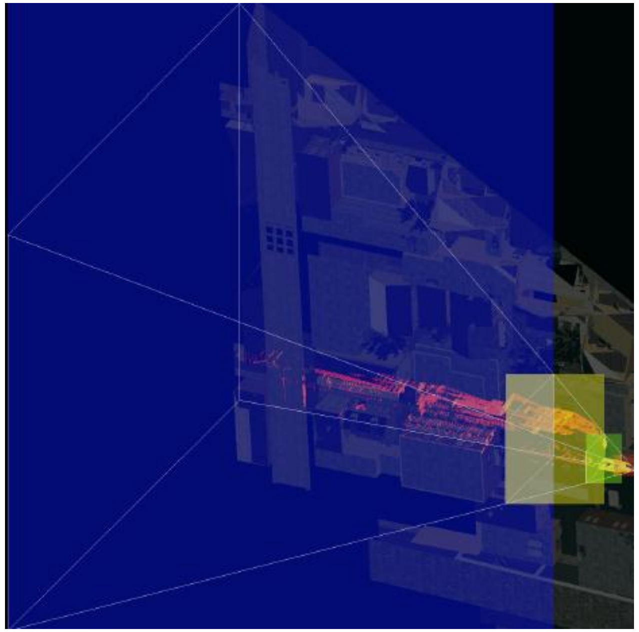


Z-Partitioning – Sample Distribution Maps [Lauritzen et al. 2011]

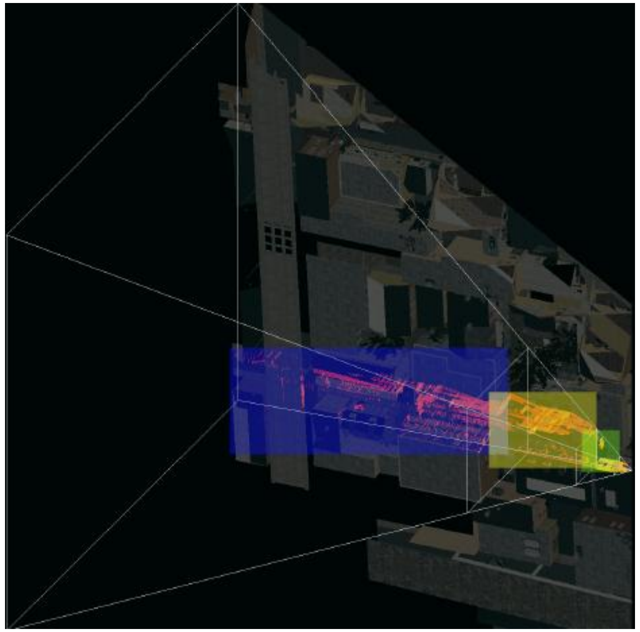
- Extension: analyze fragment distribution in light space
  - Tightly fit partitions in s and t



Shadow map



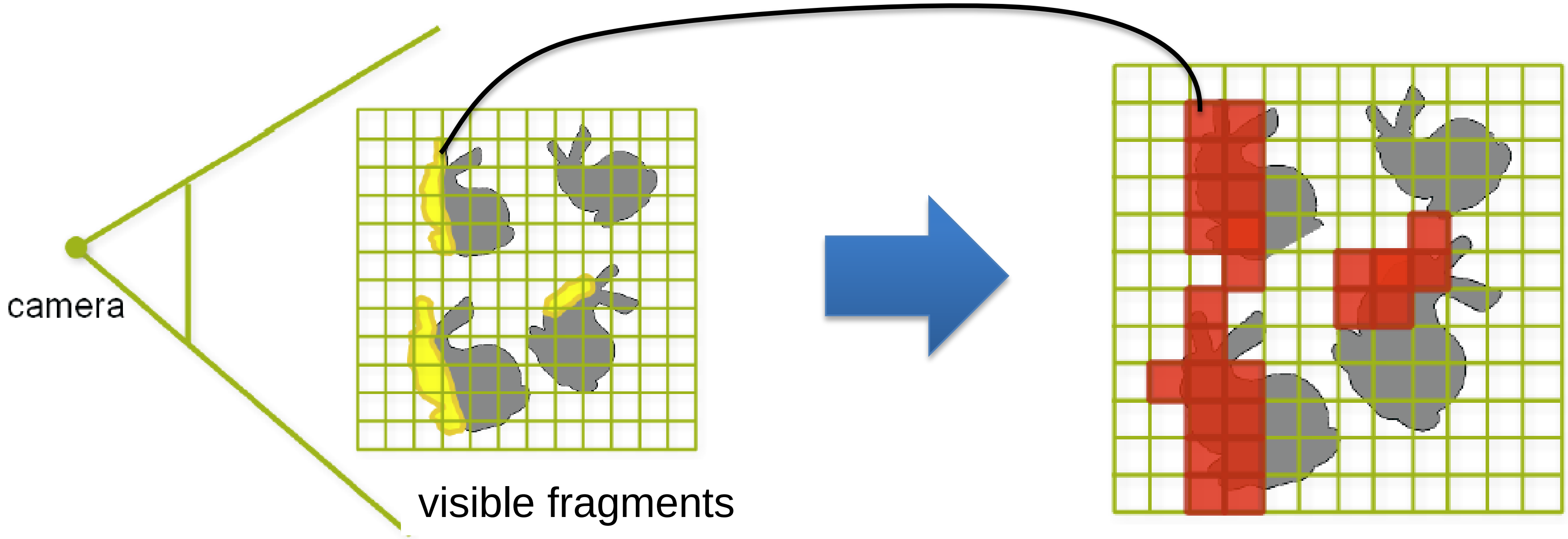
Standard Z-partitions



Bounded partitions

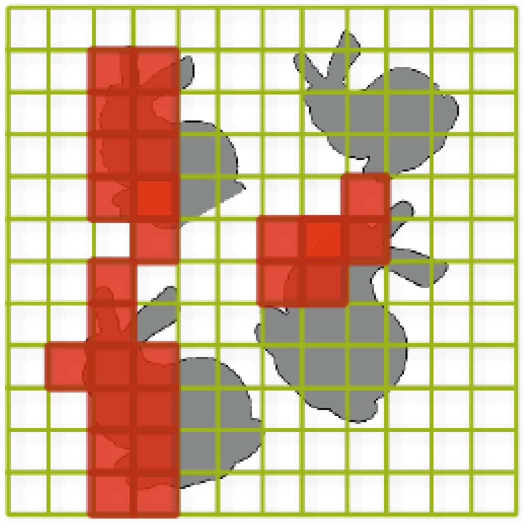
Shadow Caster Culling [Mattausch et al. 2011]

- How to determine relevant fragments in light space?
  - Determine shadow receivers (camera pass)
  - Render shadow receivers into light-space mask (light pass)
    - Fragment-level check using reverse shadow lookup

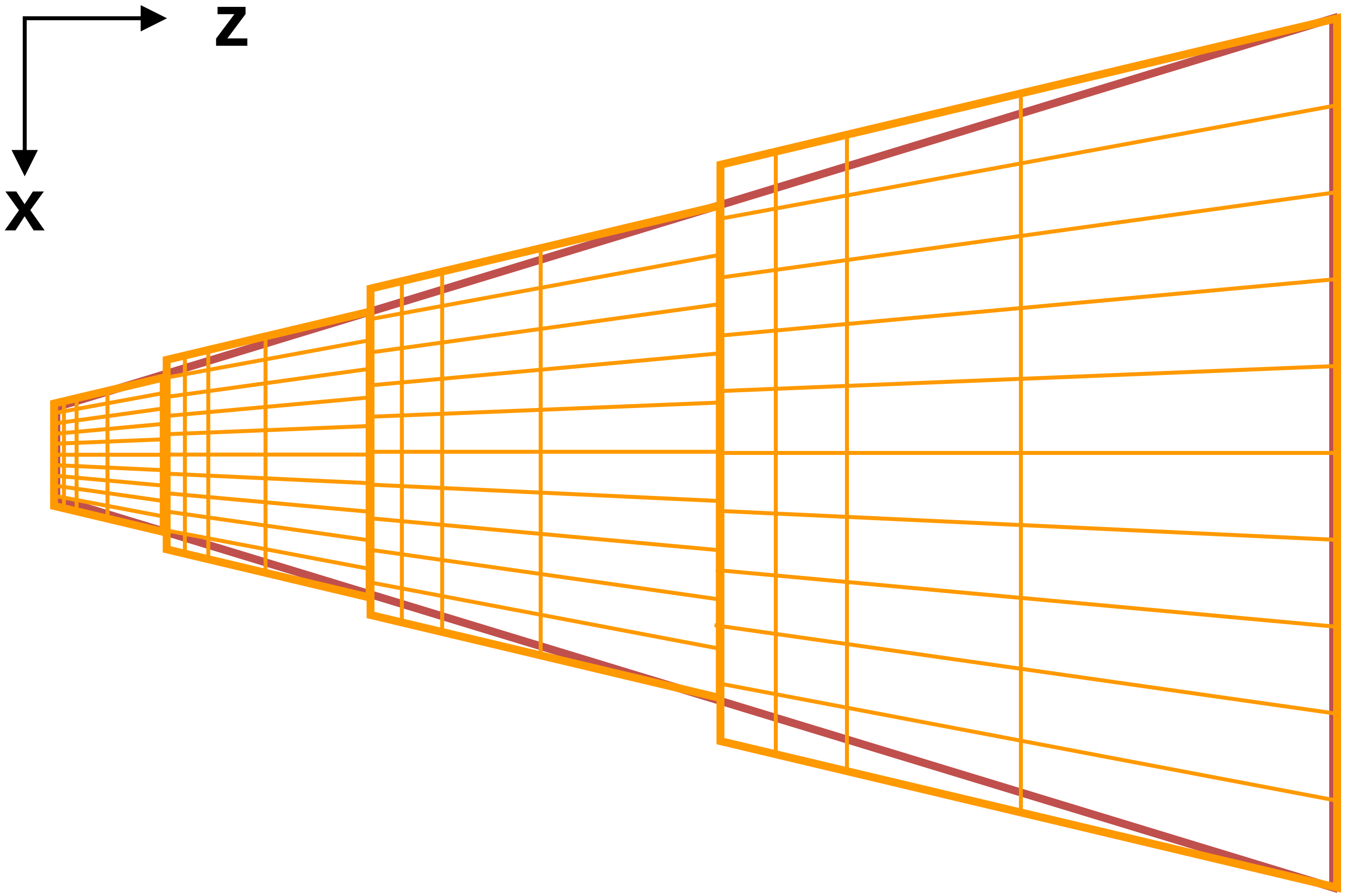


Shadow Caster Culling [Mattausch et al. 2011]

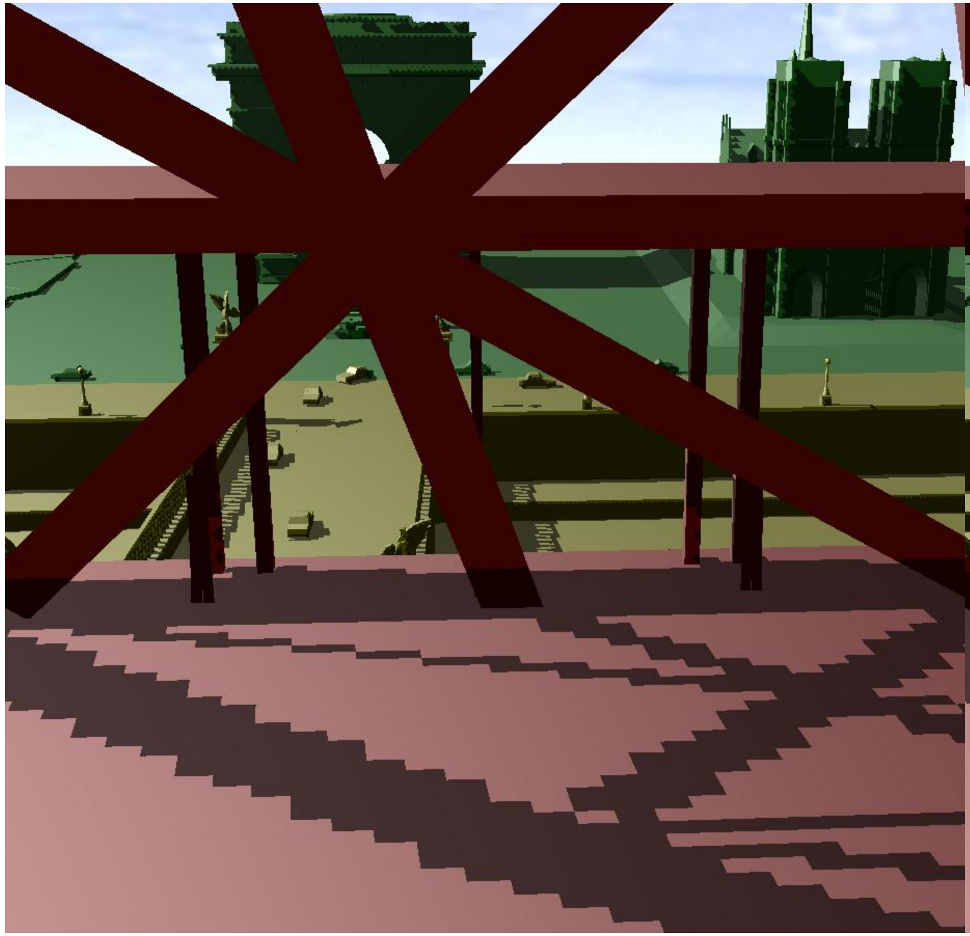
- Culling:
  - Use mask for occlusion culling for depth map
  - Large performance gain for outdoor scenes!



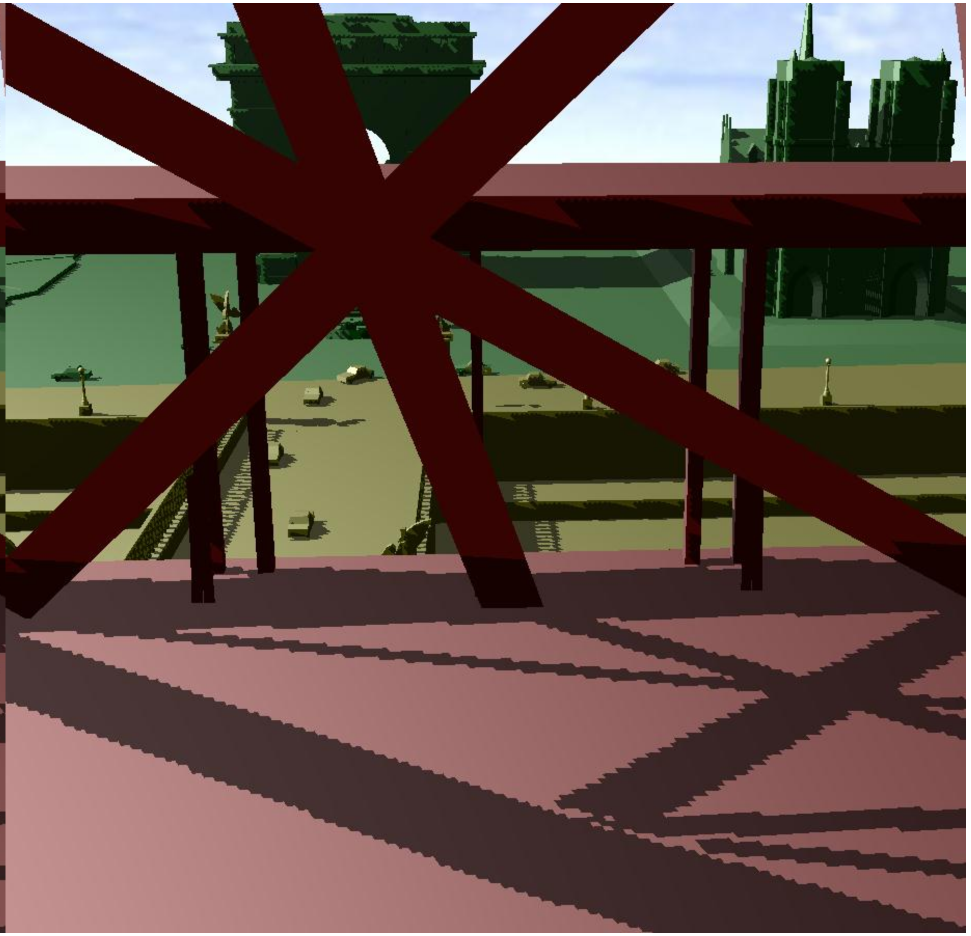
Z-Partitioning and Warping



Z-Partitioning and Warping



Partitioning

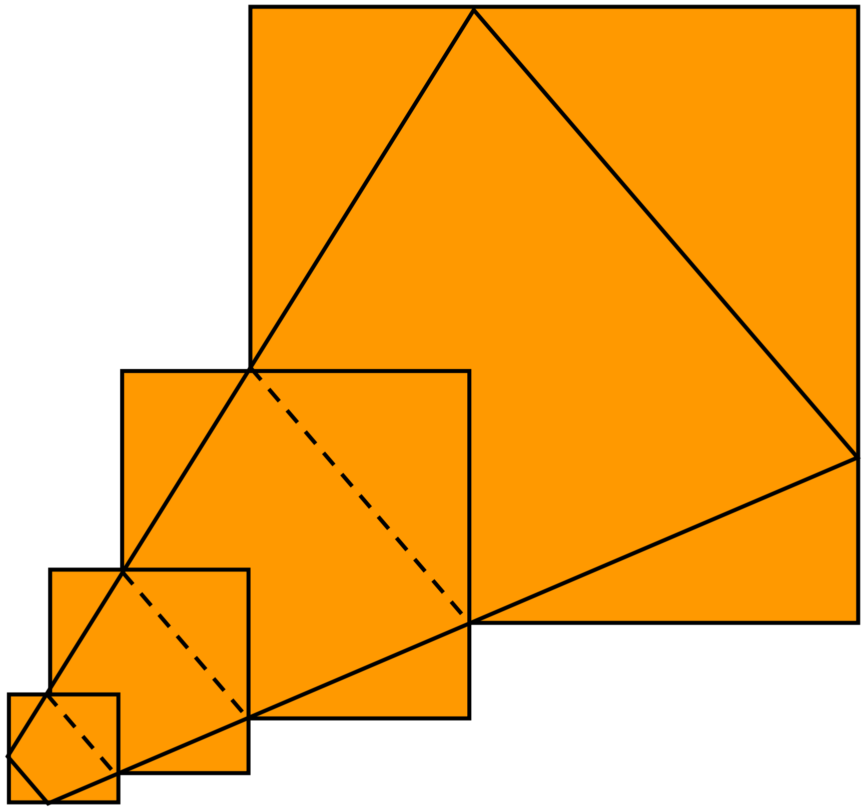


Partitioning + warping

Temporal coherence

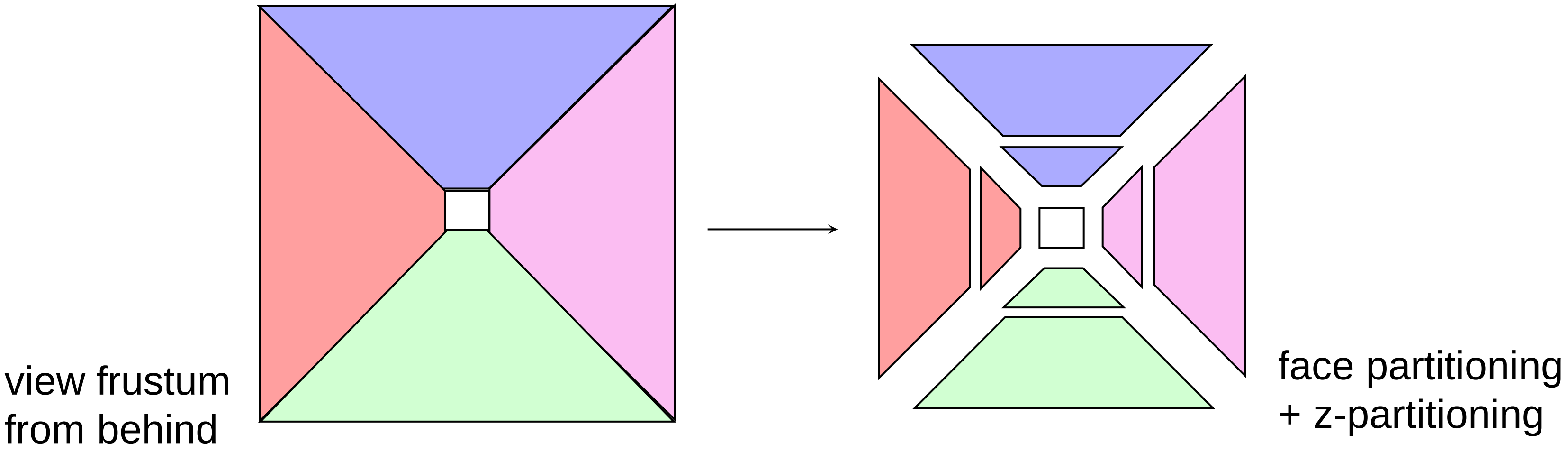
SIGGRAPH2013

- Fix coordinate system orientation in world space
- Shadow maps move at integral multiples of a texel width (see fitting)
- Gives up warping

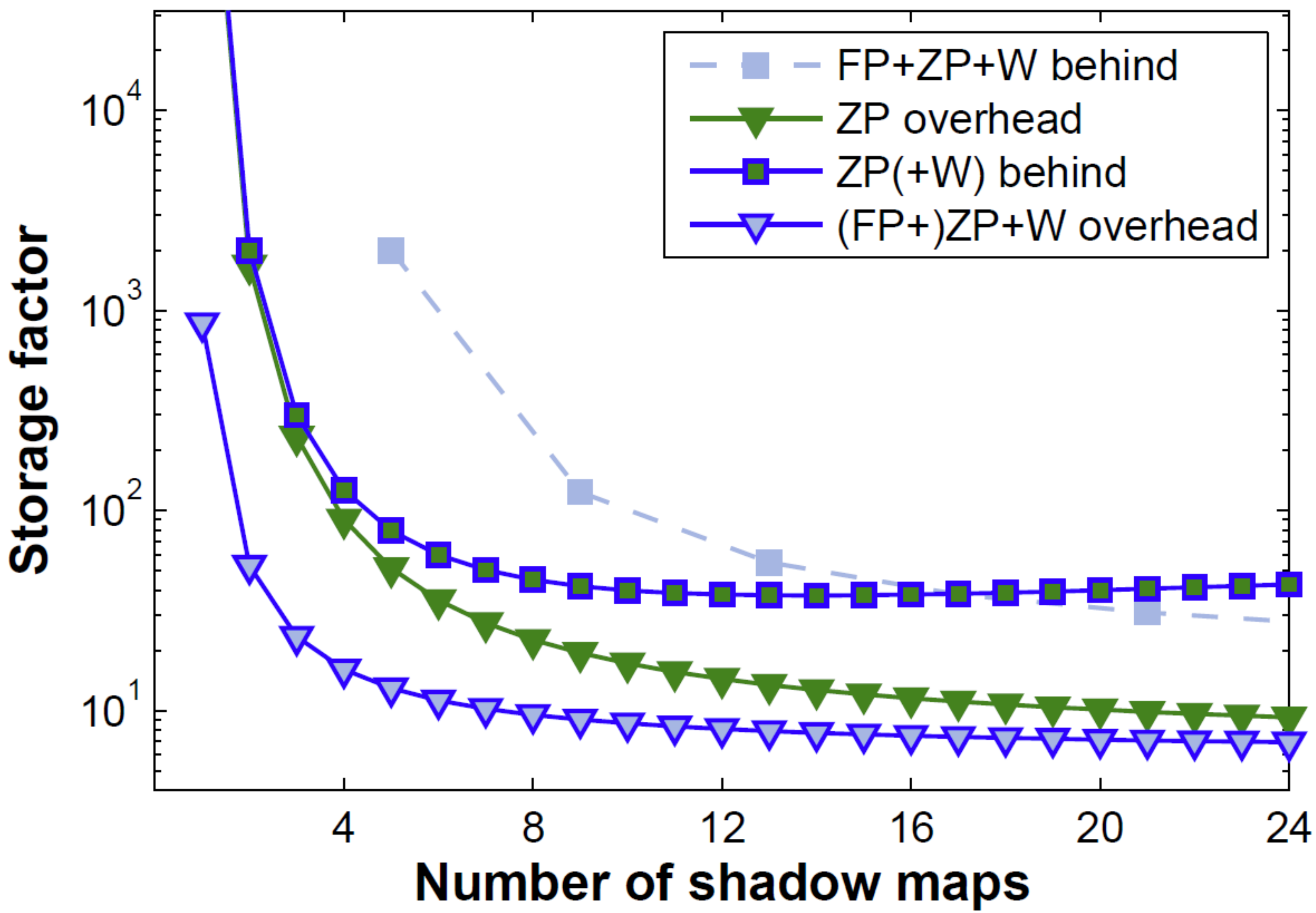


Face Partitioning [Lloyd et al. 2006]

- Alternative to z-partitioning
  - Partition frustum according to faces
- Can be combined with z-partitioning (see later)
  - Reduces redundancy
  - Also works without warping ("oblique projection")



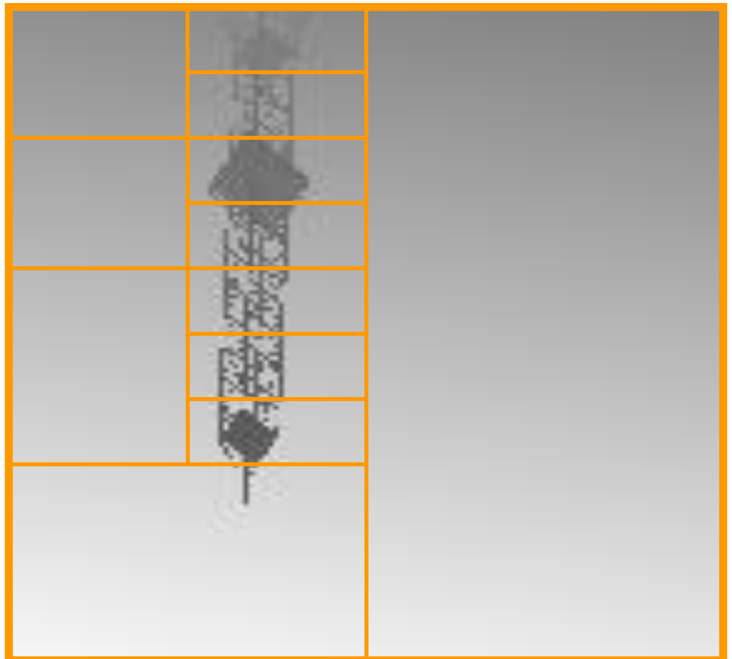
Full Error Analysis [Lloyd et al. 2006]



Adaptive Partitioning

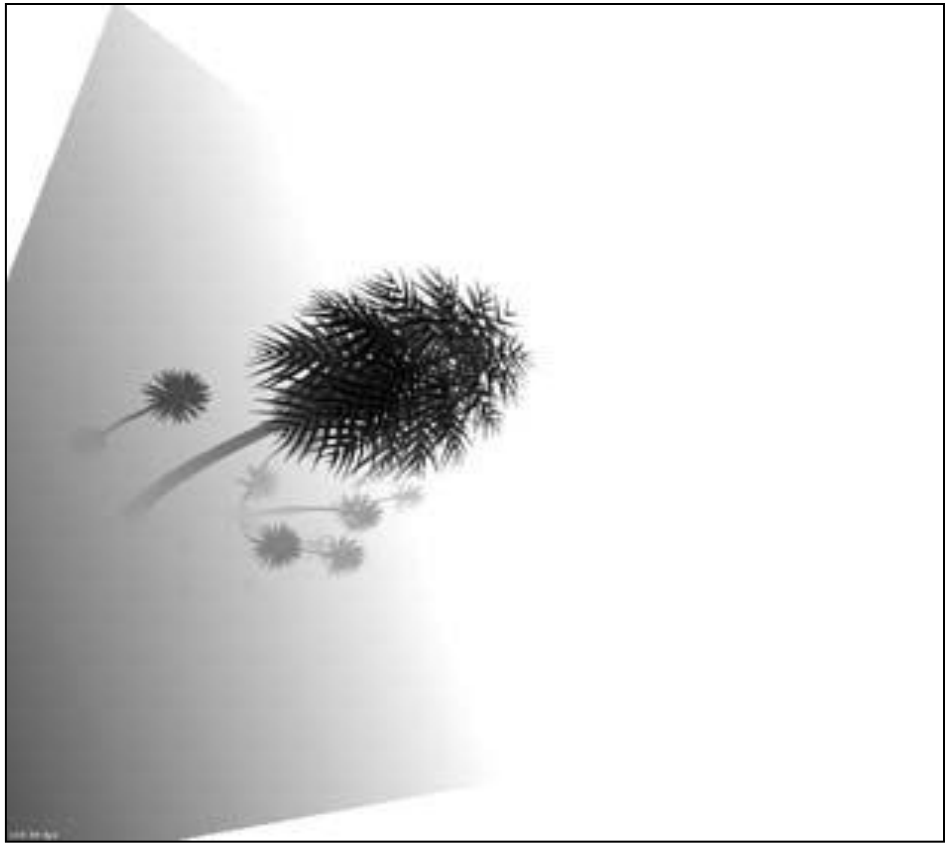
SIGGRAPH2013

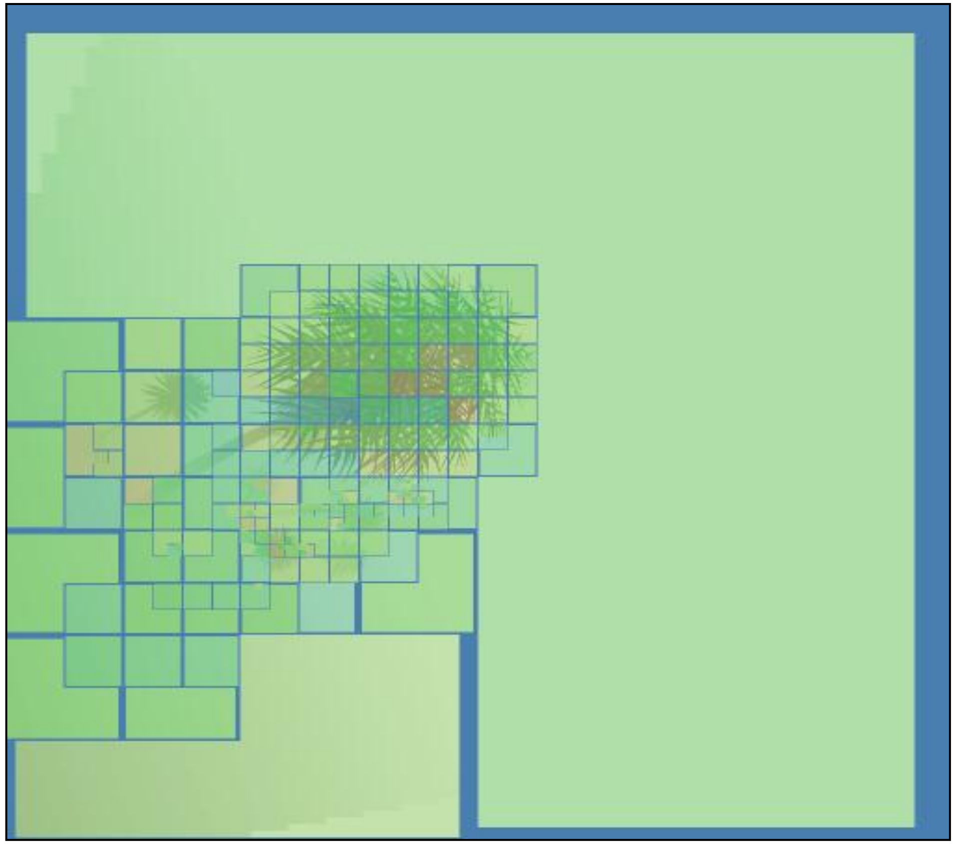
- Warping and z-partitioning are **global** resampling schemes
  - Deal with perspective aliasing
  - Projection aliasing needs local scene adaptive resampling!
- Adaptive partitioning adaptively splits shadow map
  - Usually quad-tree subdivision
- Algorithms mainly differ in termination criteria



Adaptive Shadow Maps
[Fernando et al. 2001; Lefohn et al. 2006]

- High resolution only needed at edges
- Search for edge (slow)
  - If edge → split



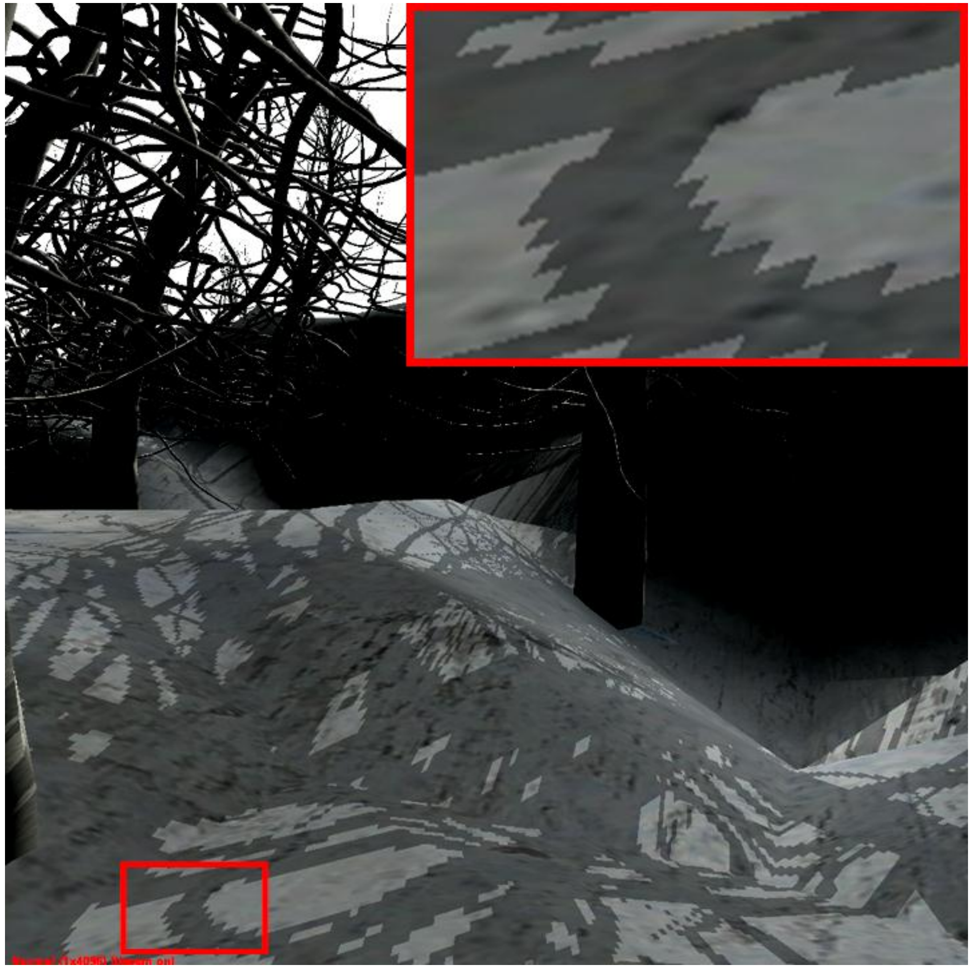


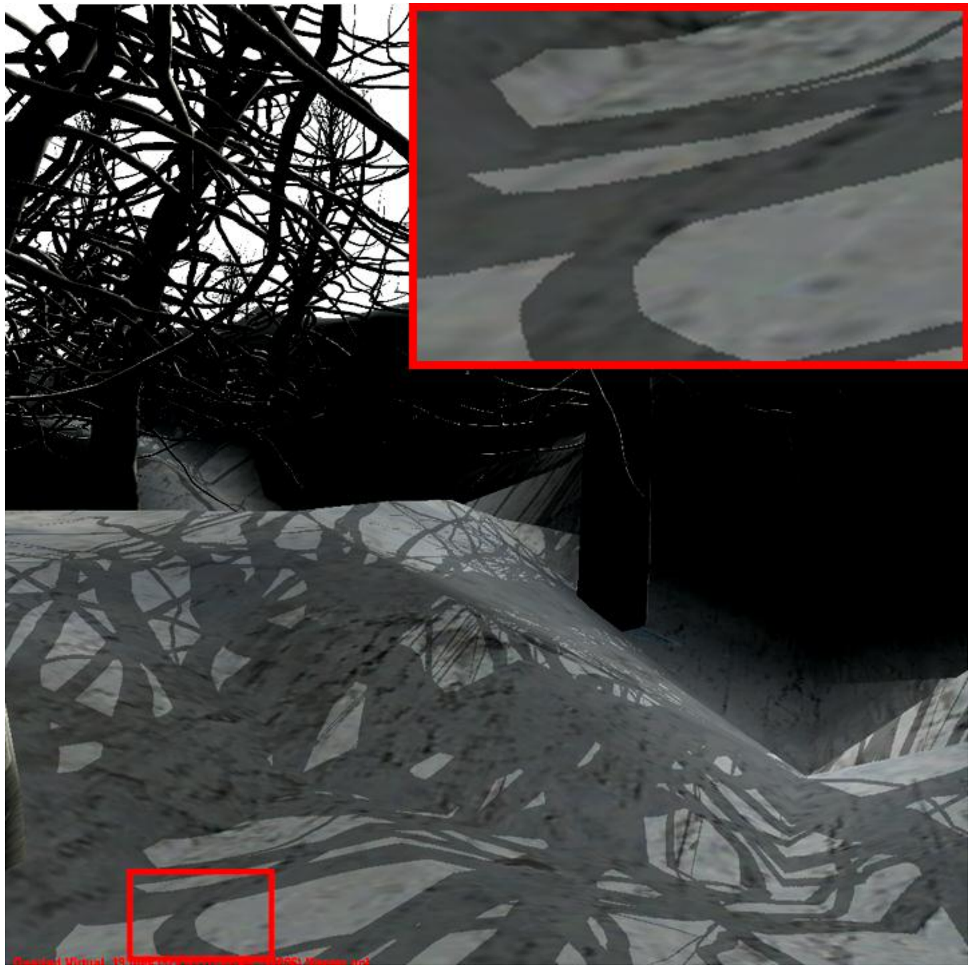
Fitted Virtual Shadow Maps/Resolution Matched SM

[Giegl & Wimmer 2007], [Lefohn et al. 2007]

- Do not calculate all quad-tree levels, but determine finest levels necessary
  - Camera prepass
  - Analyzed on CPU (FVSM) or GPU (RMSM)

Fitted Virtual Shadow Maps
[Giegl & Wimmer 2007]

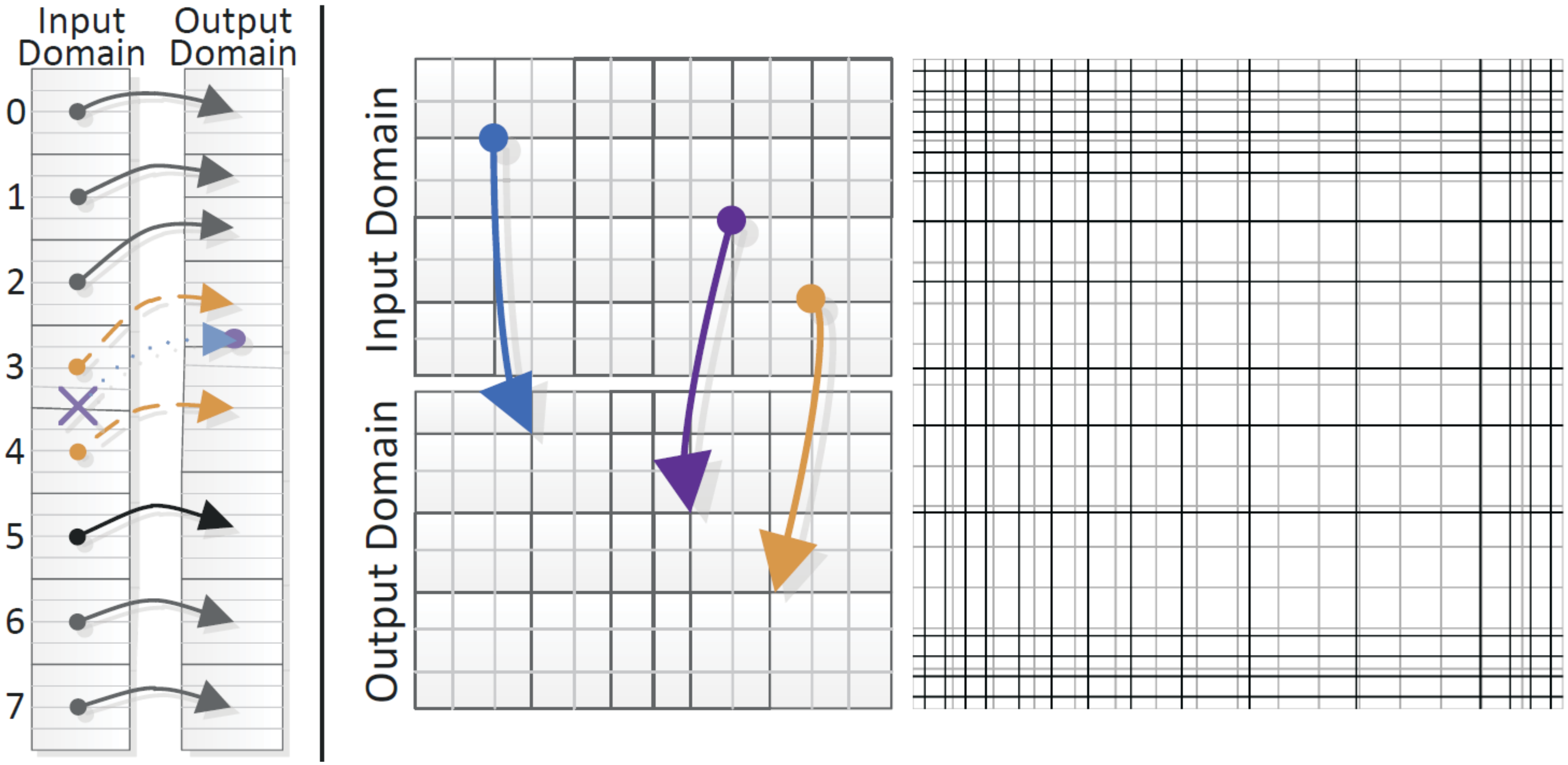




Rectilinear Warping [Rosen et al. 2012]

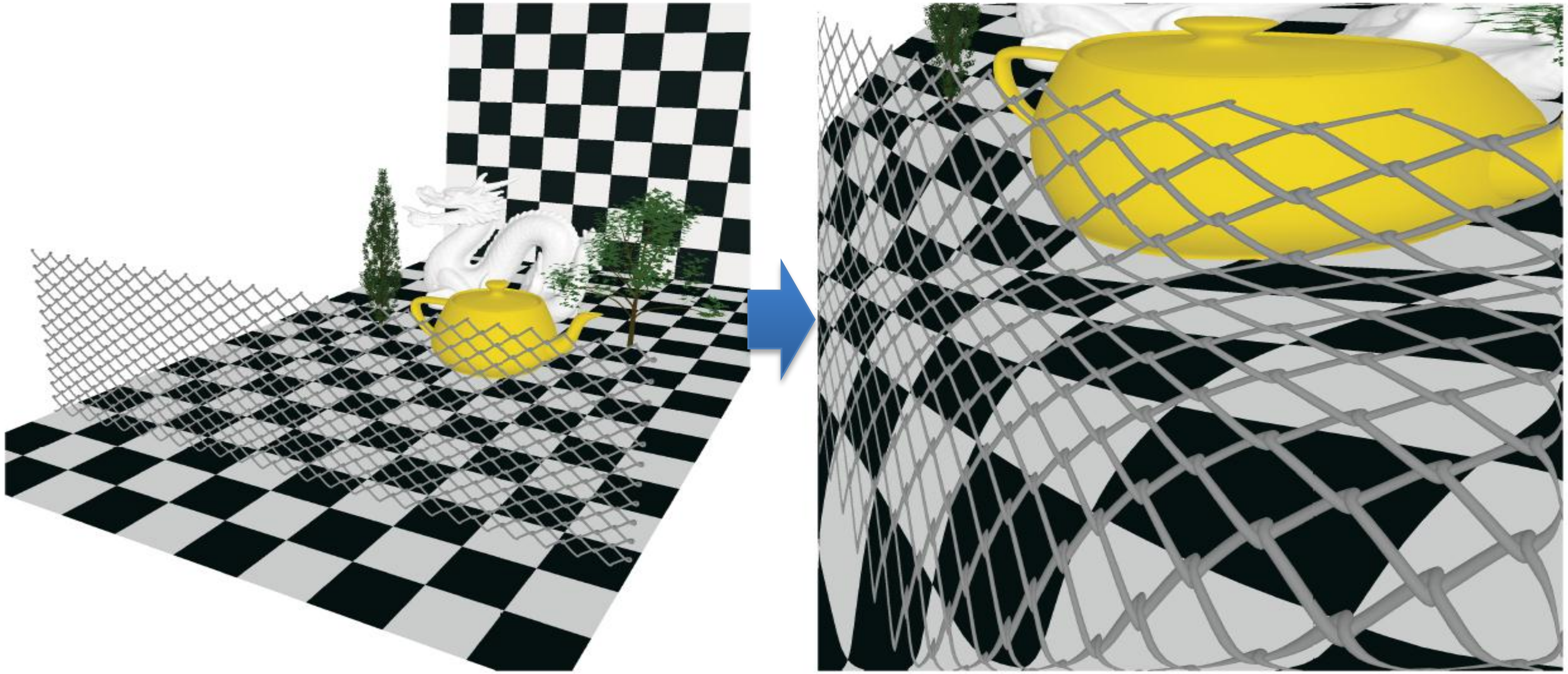
SIGGRAPH2013

- Apply 2 separate 1D-warps based on scene content



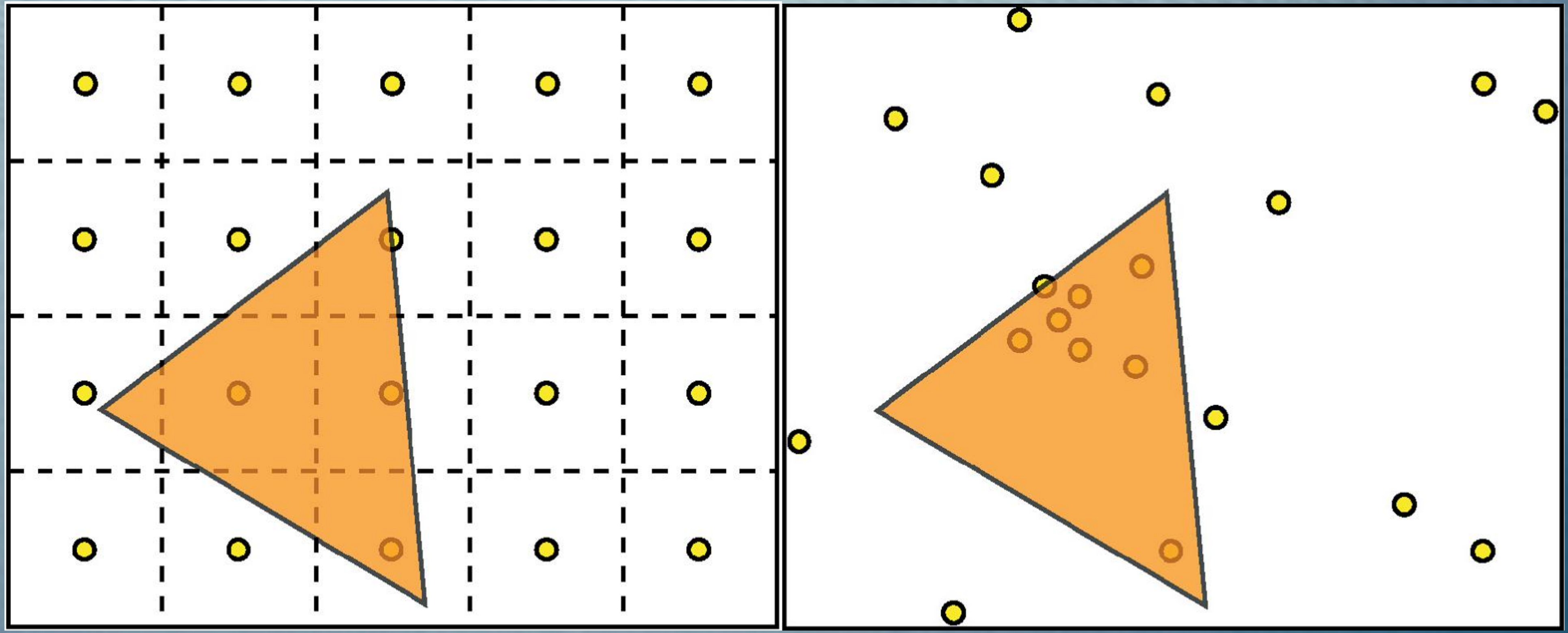
Rectilinear Warping [Rosen et al. 2012]

- Requires scene analysis pass
- Requires tessellation to create shadow map



Standard shadow map

Rectilinearly warped shadow map

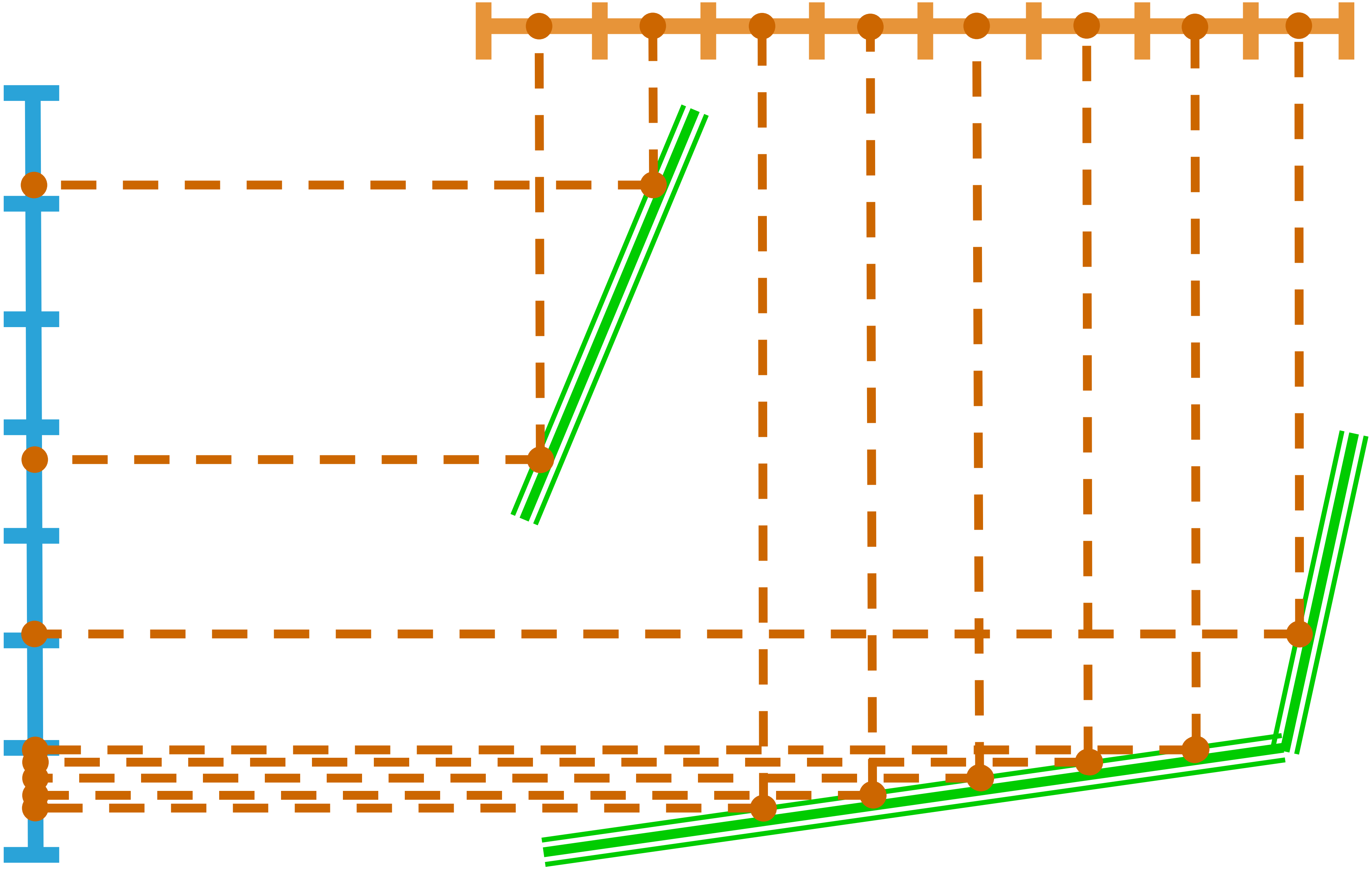


Hard Shadows

Fighting Undersampling – Irregular Sampling

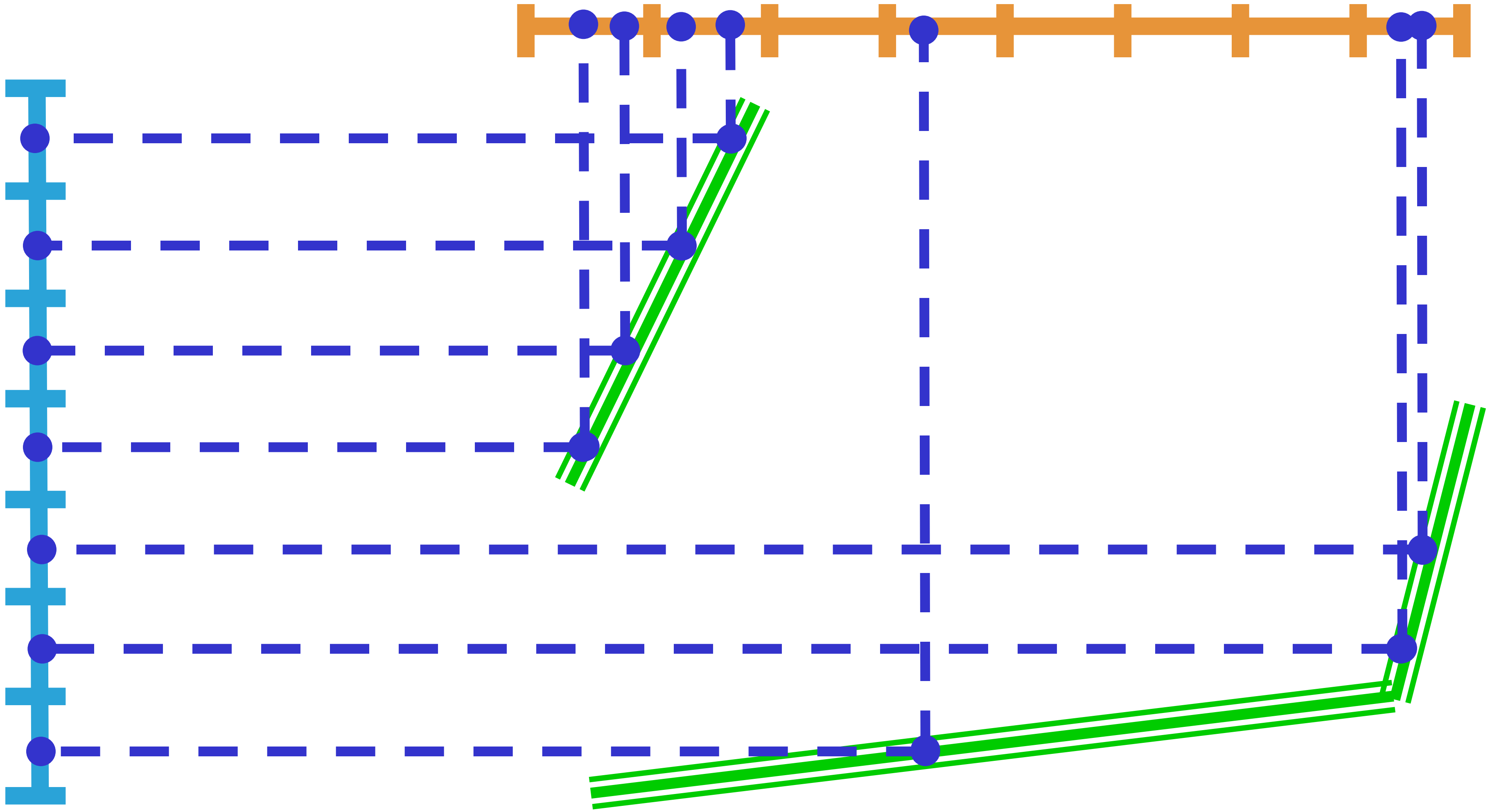
SIGGRAPH2013

Shadow Mapping Sampling



What Samples Do We Want?

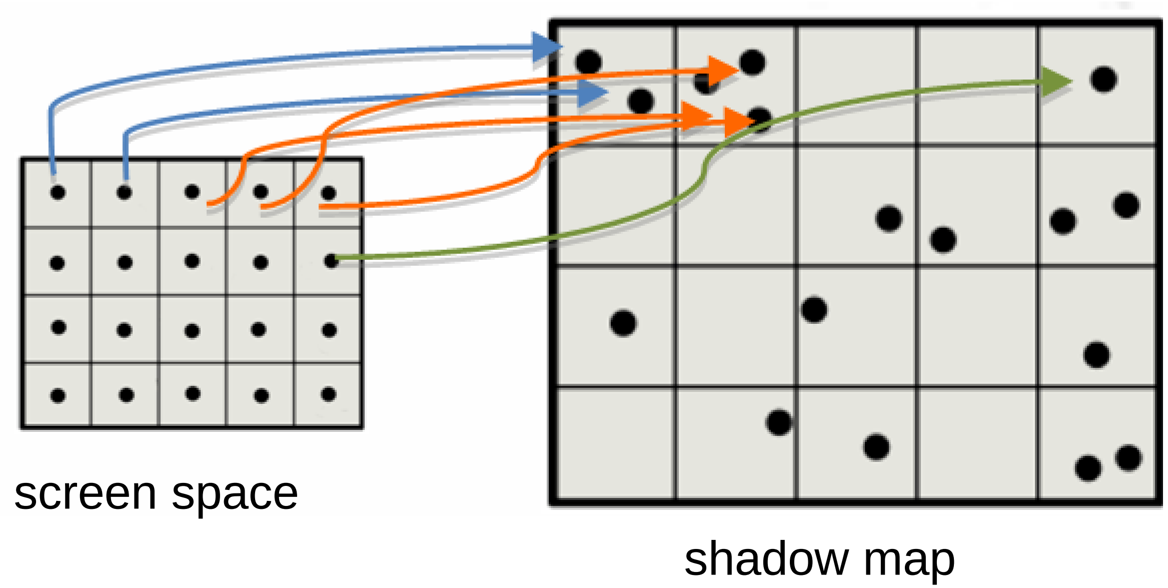
- Idea: use eye space samples to generate shadow map samples

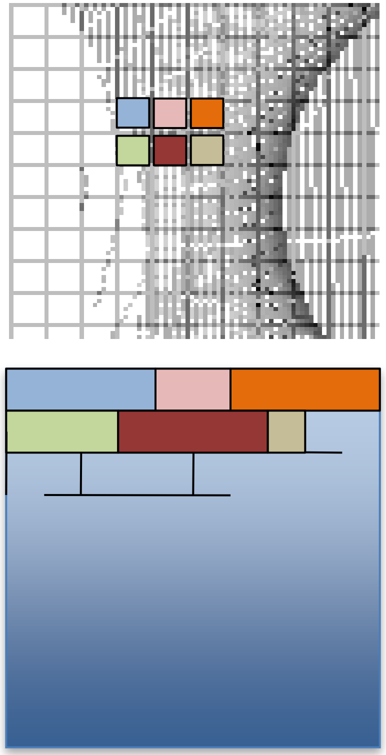


Alias Free Hard Shadows

[Aila and Laine 2004][Sintorn et al. 2008]

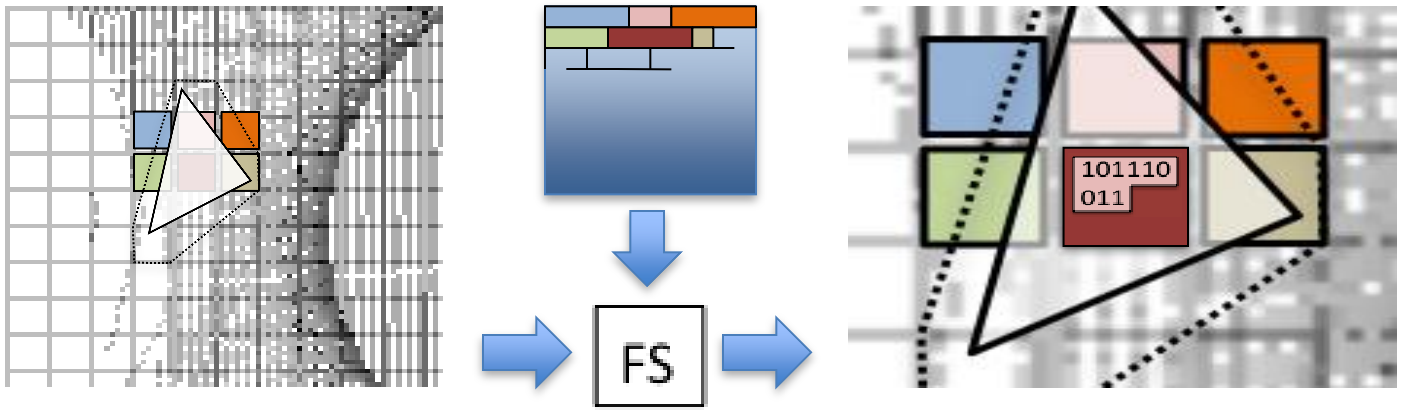
- Camera pass: transform and project view-samples into light-space
  - Store in a compact data structure with a list per light-space texel





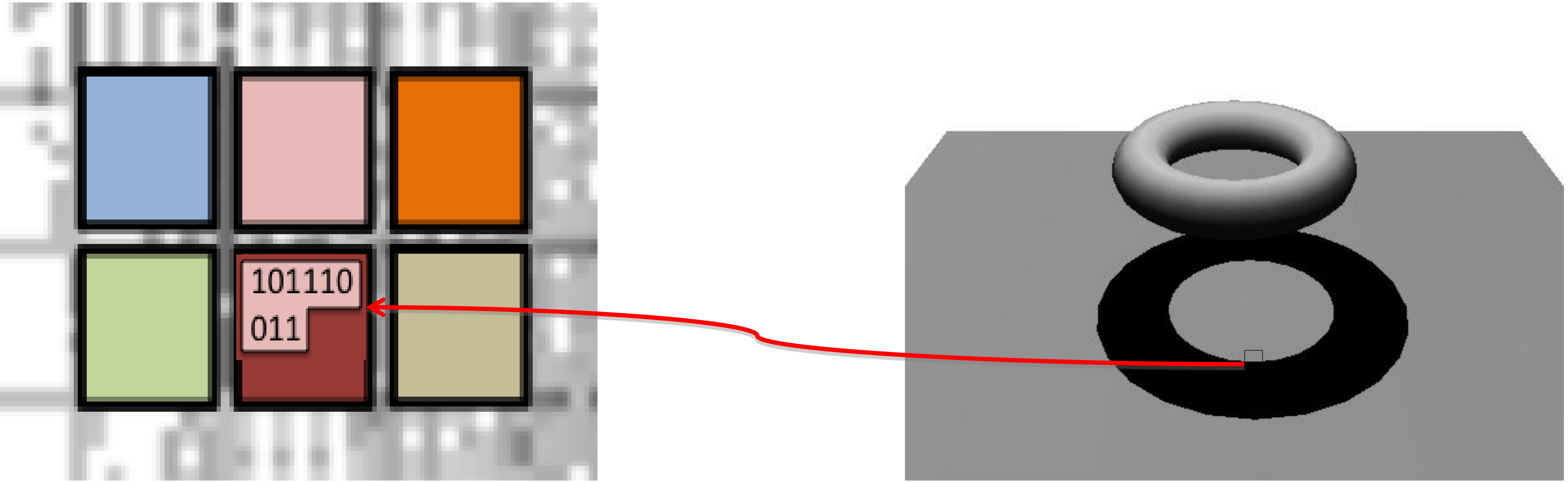
Alias Free Hard Shadows
[Sintorn et al. 2008]

- Light pass: render all geometry (conservatively)
  - For each generated fragment, test all view-samples in list against triangle
  - Set corresponding output bit depending on occlusion



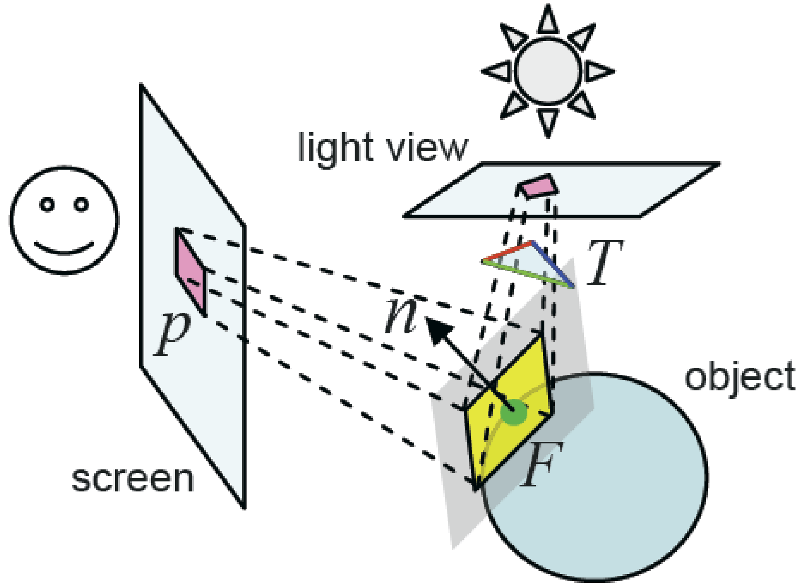
Alias Free Hard Shadows
[Sintorn et al. 2008]

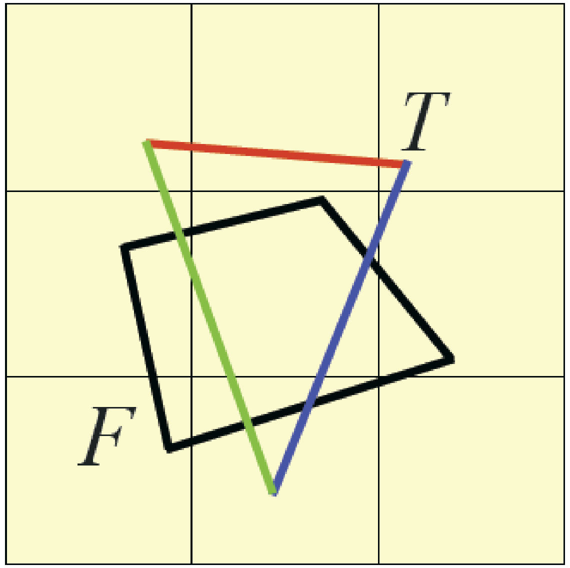
- Final screen-space pass
- Use bitmask from previous pass for shadowing

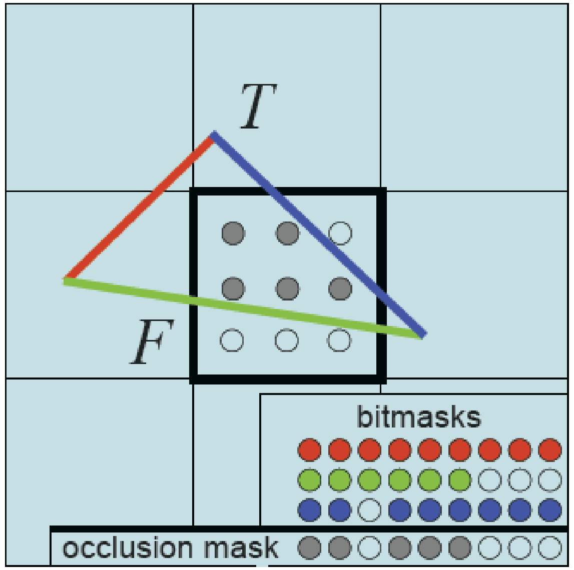


Subpixel Alias Free Shadow Maps
[Zhou et al. 2009]

- Problem: no antialiasing (supersampling)
- Solution:
  - Project facets instead of samples
  - Precompute subsample tests (constant time [Eisemann 2007])







$$cache(n) := \mathbf{conf} * s(n) + (1 - \mathbf{conf}) * cache(n-1)$$

Hard Shadows

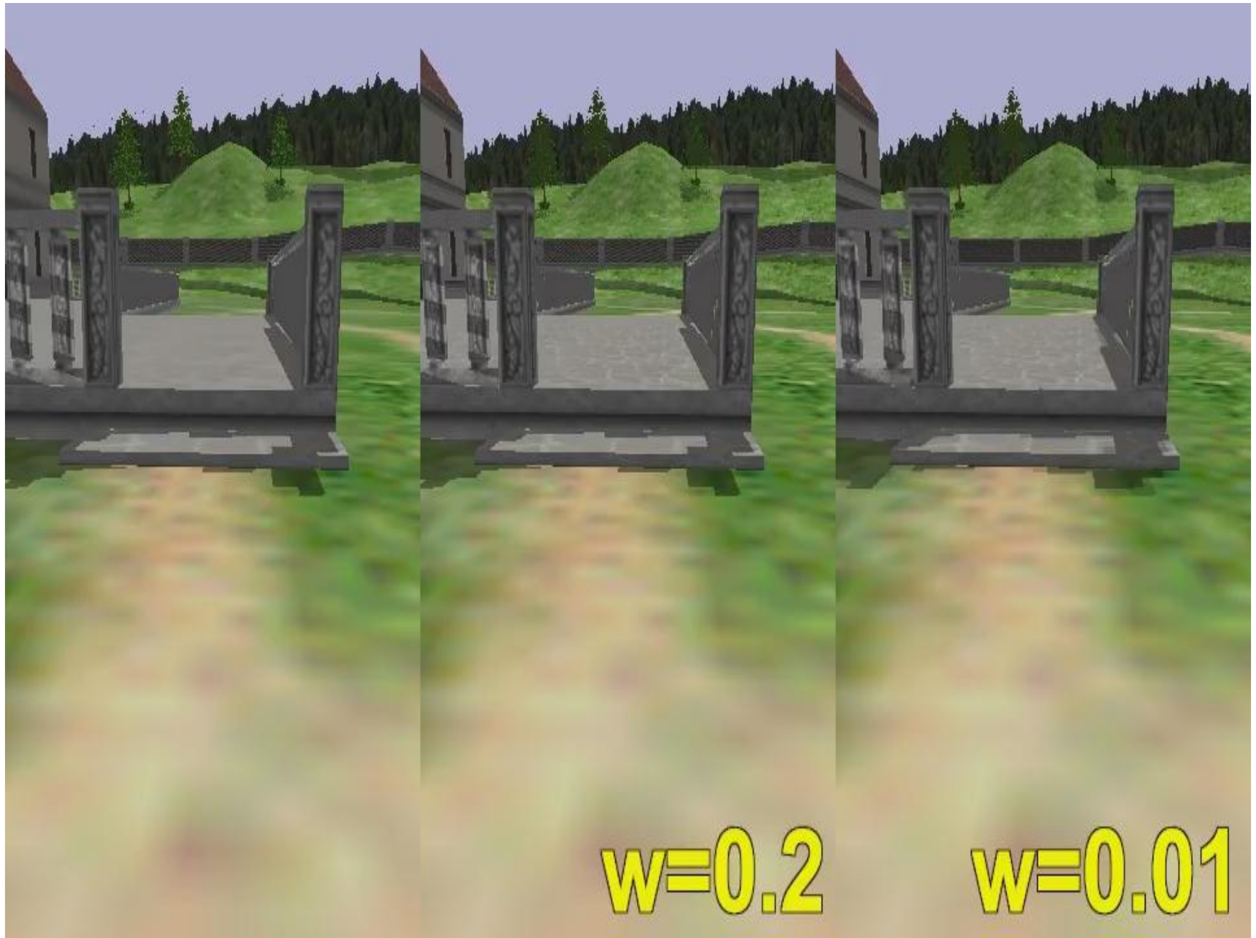
Temporal Reprojection

Temporal Smoothing

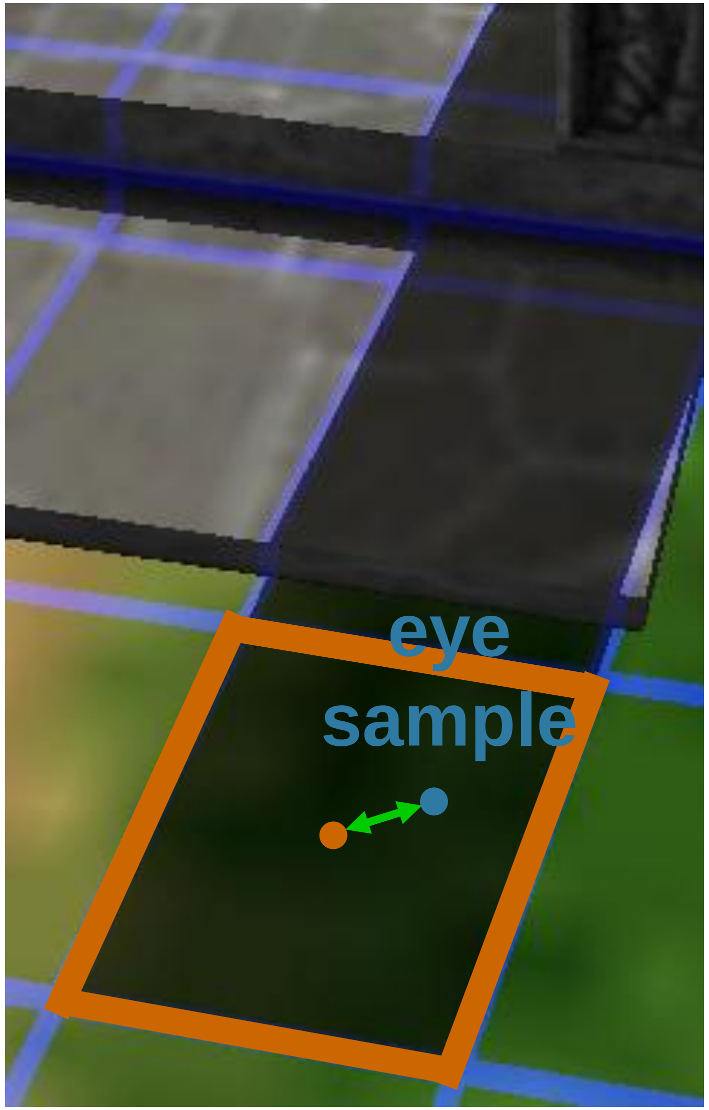
- Shadowing result of previous frame (n–1) is stored in *cache*

Temporal Smoothing

- Shadowing result of previous frame (n–1) is stored in *cache*
- Current frame (n): $cache(n) := w * s(n) + (1-w) * cache(n-1)$



Weight? Confidence Estimation!

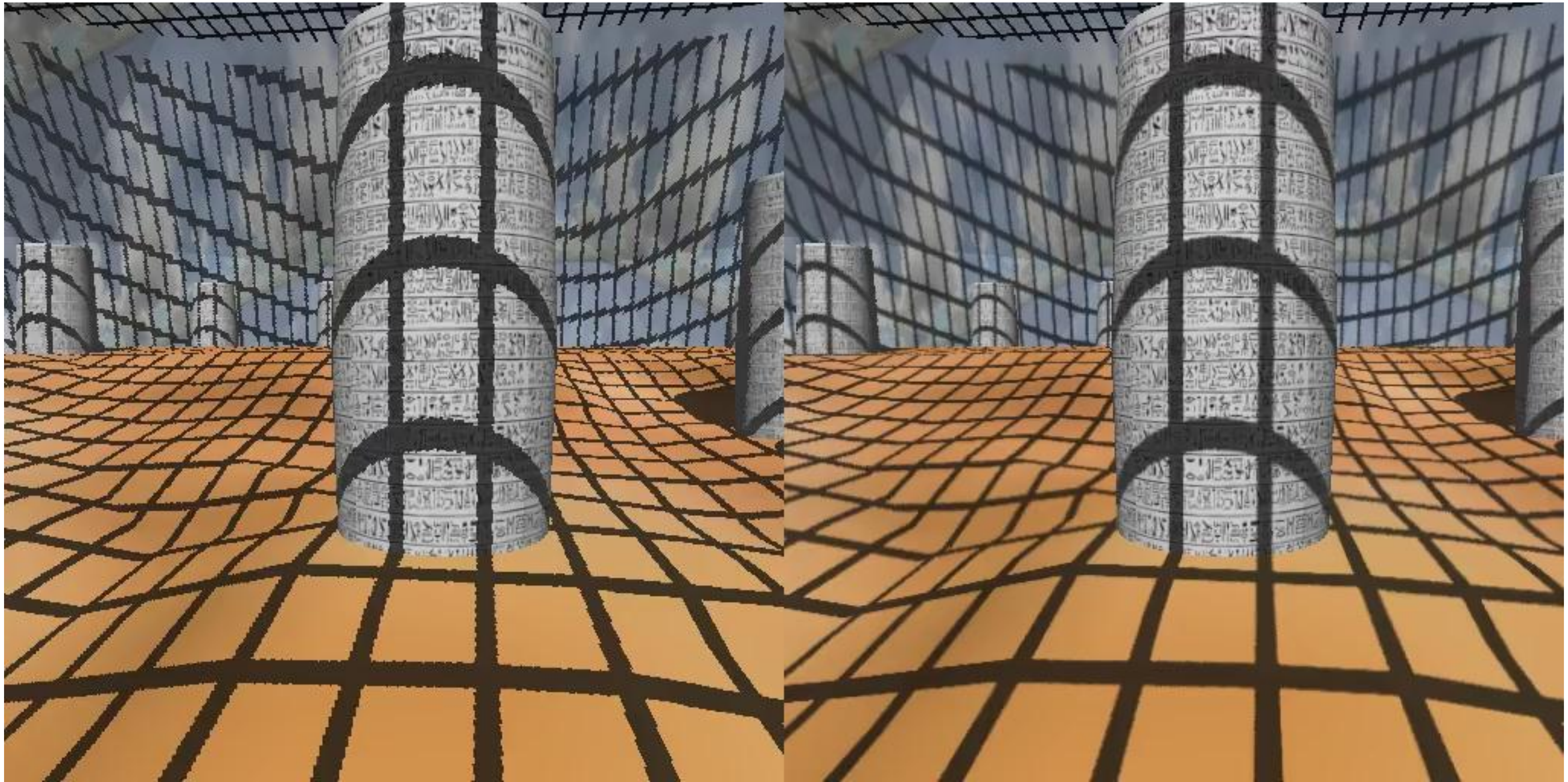


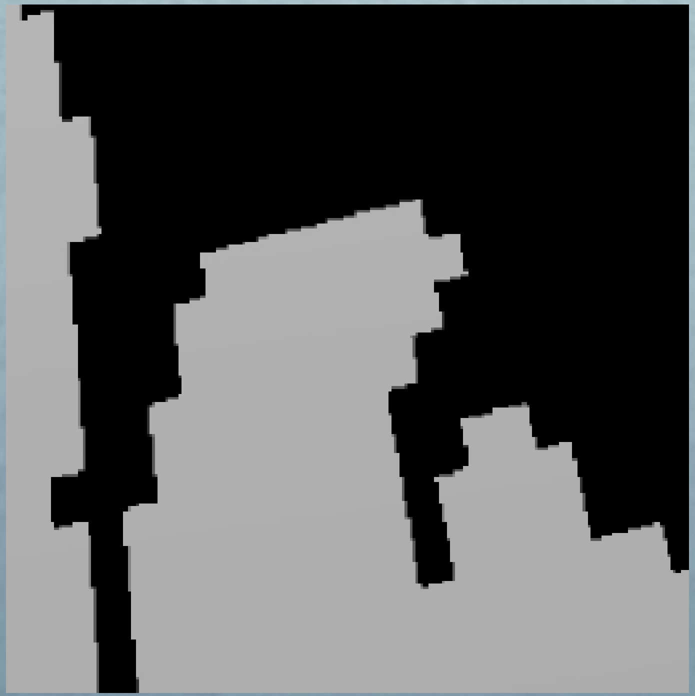
- Shadow map test probably more correct closer to texel center
- Confidence:
  conf := 1- *dist*(**eye**,**texel**)
- Greater confidence
  → greater impact

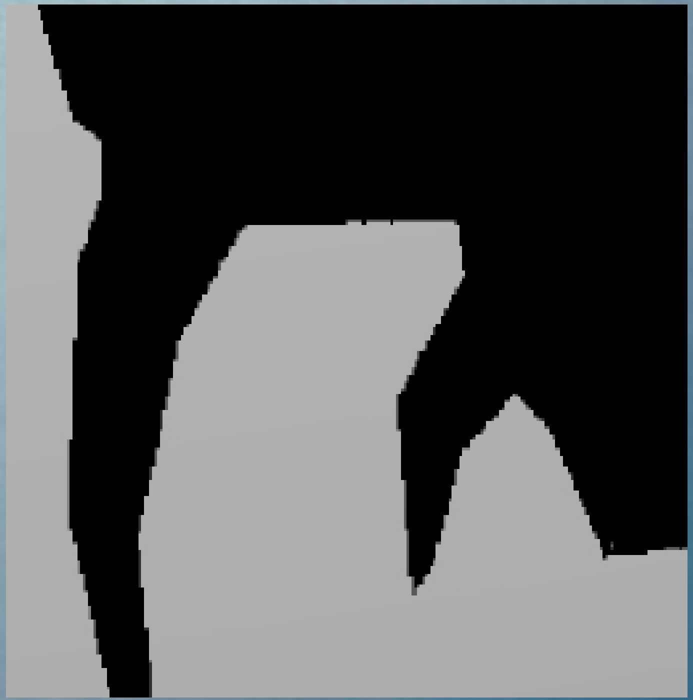
Confidence and Temporal Smoothing

- Confidence is weight for temporal smoothing

$$cache(n) := \mathbf{conf} * s(n) + (1-\mathbf{conf}) * cache(n-1)$$





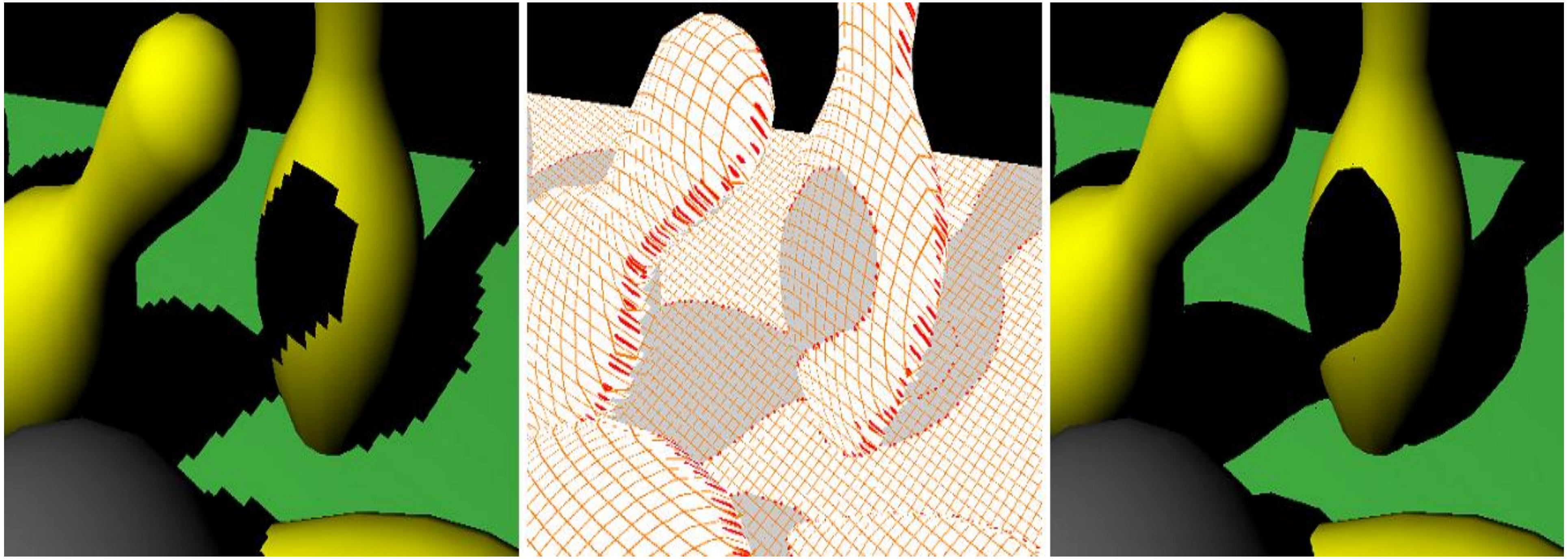


Hard Shadows

Better Reconstruction

Shadow Silhouette Maps
[Sen et al. 2003]

- Use a better silhouette approximation
- Store additional information of shadow edge

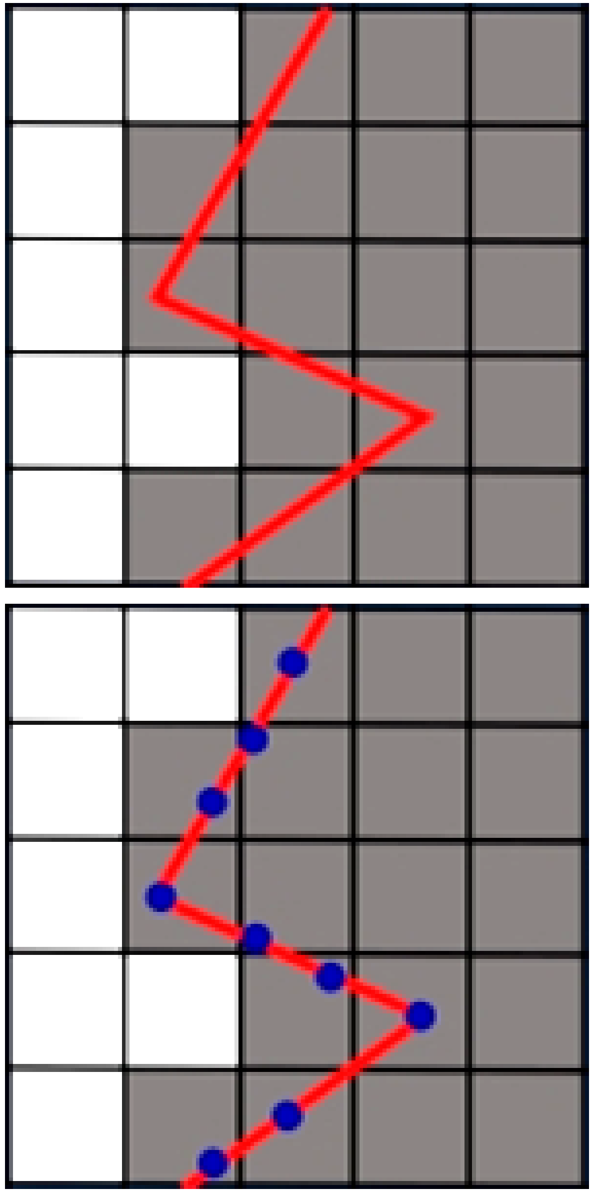


Shadow Silhouette Maps

[Sen et al. 2003]

SIGGRAPH2013

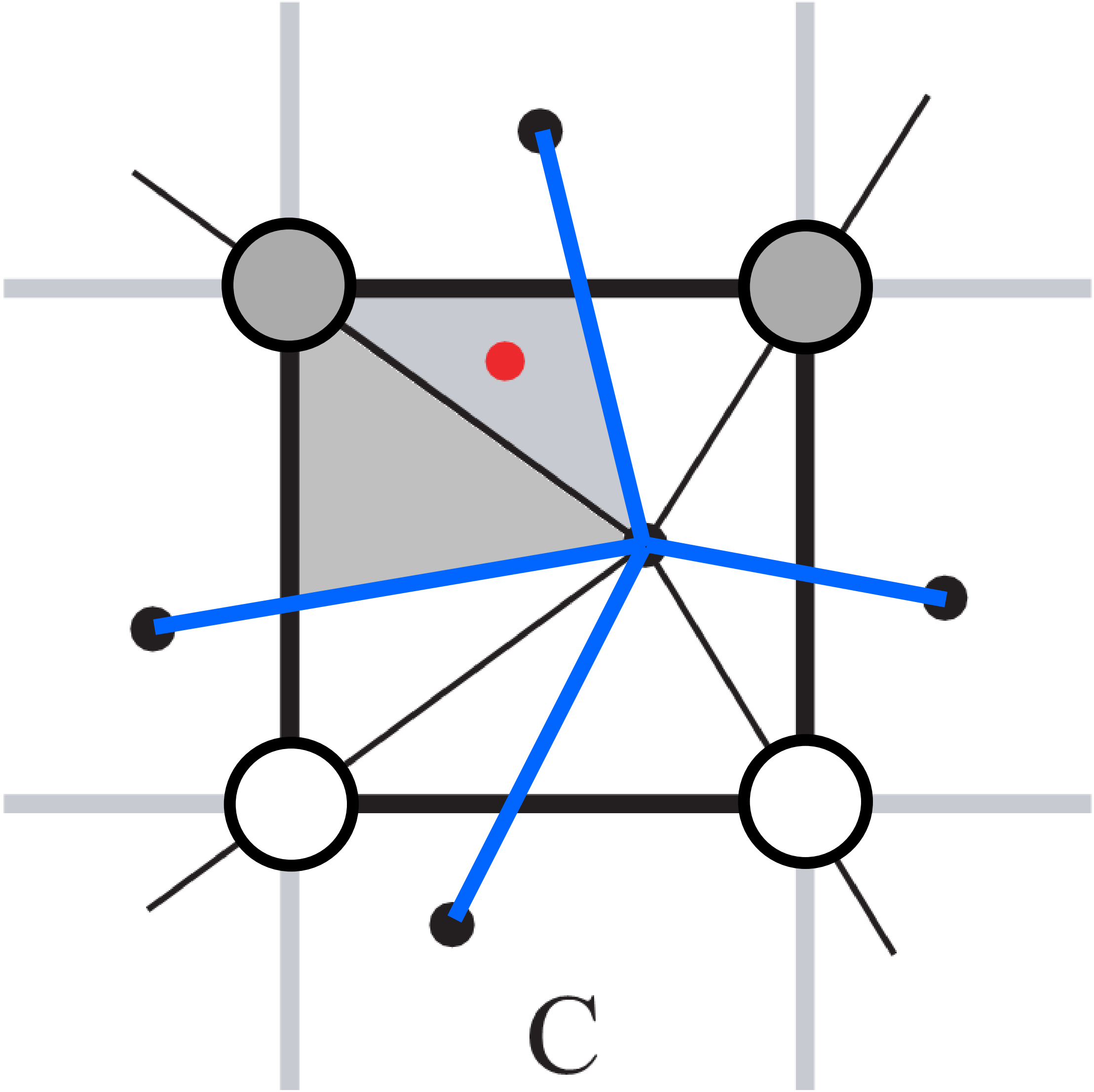
1. Create shadow map
2. Find silhouette edges
3. Rasterize silhouettes
   a. Find points that lie on silhouette edges
   b. Store such points into silhouette map
4. Compute shadows
   a. Non-silhouette pixels use standard shadow map
   b. Silhouette pixels use silhouette map



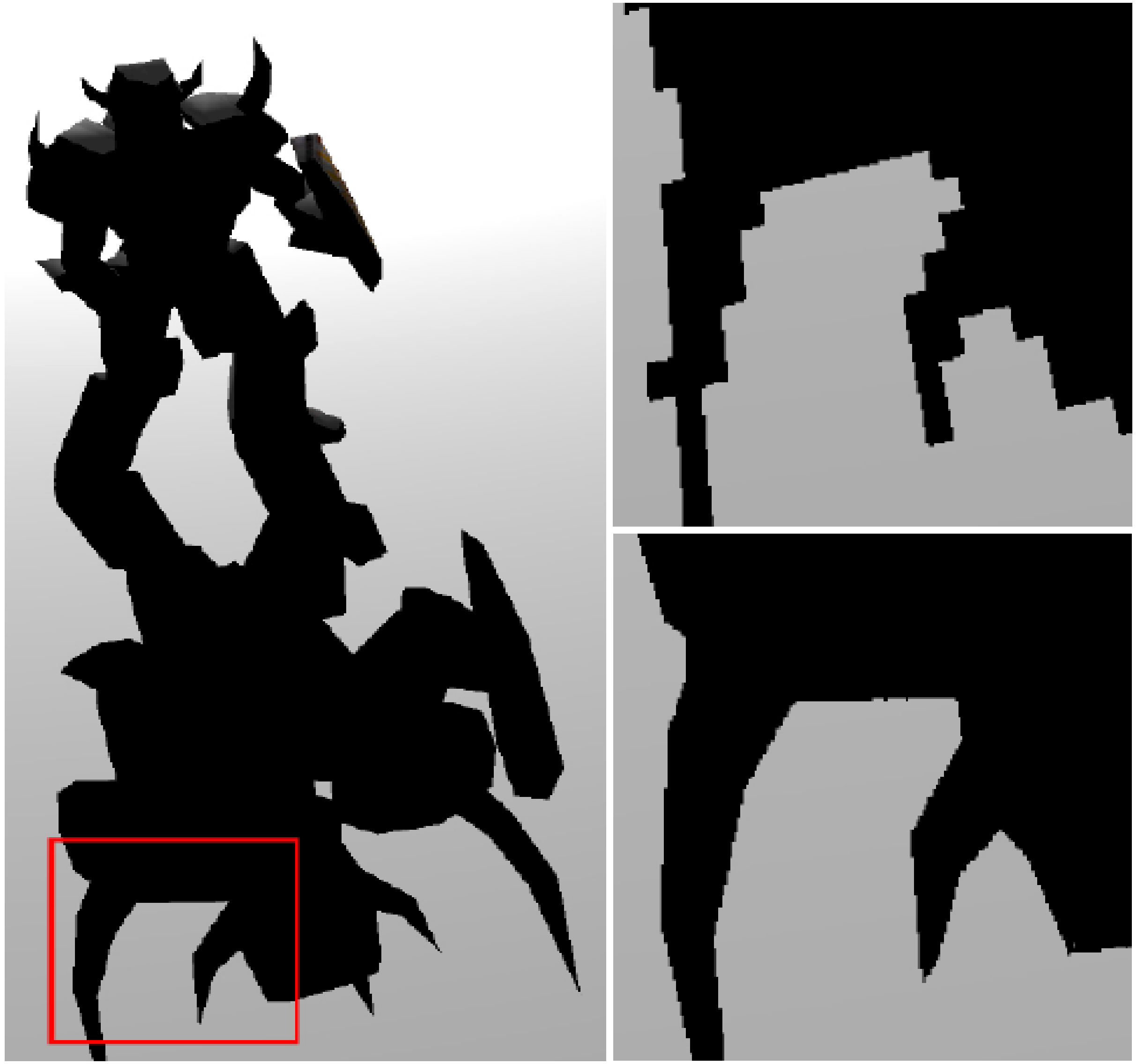
Shadow Silhouette Maps

[Sen et al. 2003]

- Fragment
- Silhouette points
  - 1 interior + 4 external
- Create quadrants
- Shade fragment according to shadow test result at corner point

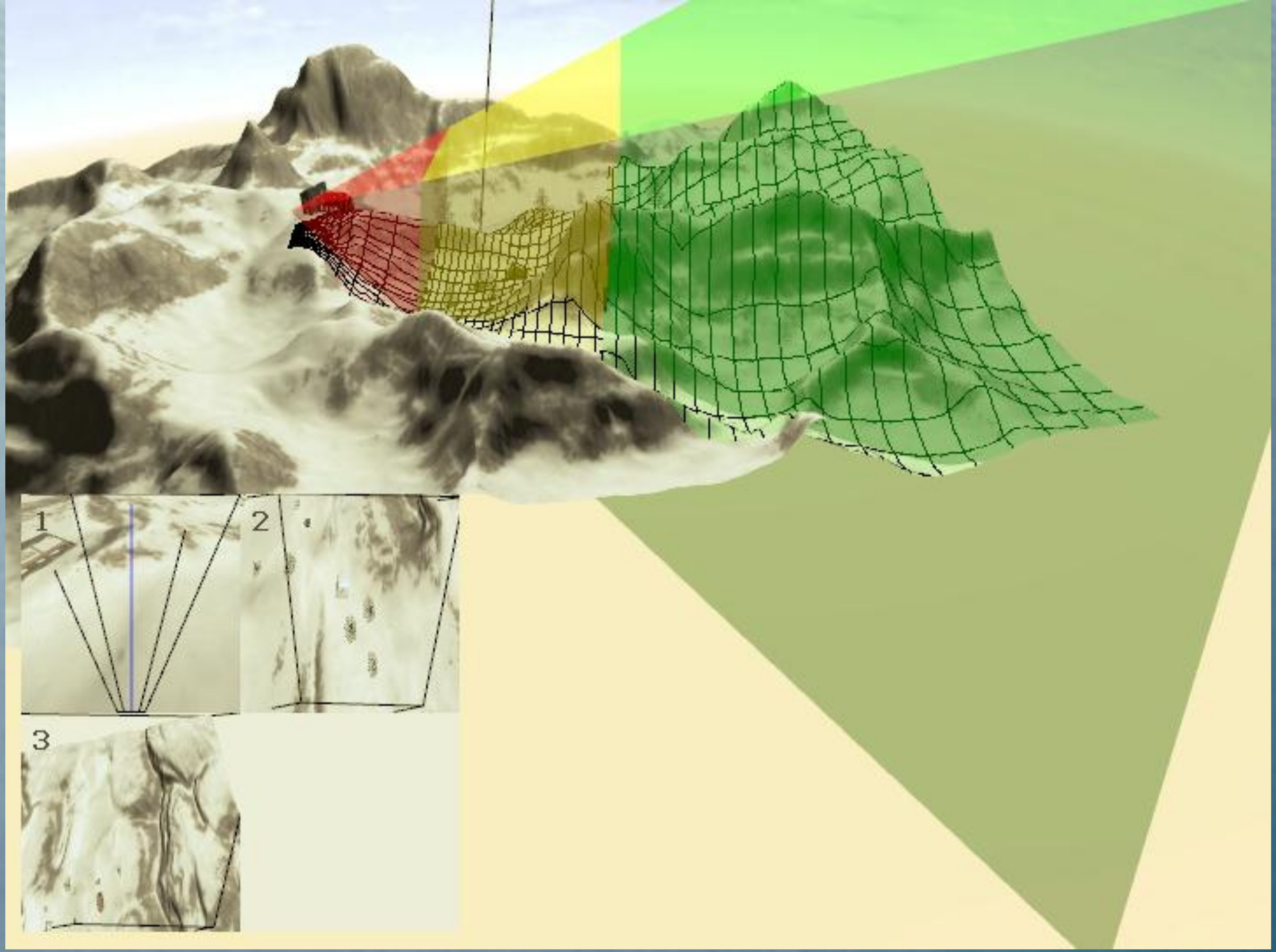


Shadow Silhouette Maps
[Sen et al. 2003]



shadow map

silhouette map



Hard Shadows

Conclusion

Conclusions

www.realtimeshadows.com

SIGGRAPH2013

- Fastest speed, single shadow map: **warping**
  - Good for outdoor
- Fast speed, better quality, >1 shadow maps: **z-partitioning (+warping)**
  - Add "lightweight" scene analysis to tightly bound partitions and cull shadow casters
- High quality, lower speed: **adaptive partitioning**
- Reference quality, even slower: **irregular sampling**
  - Combine with subpixel sampling