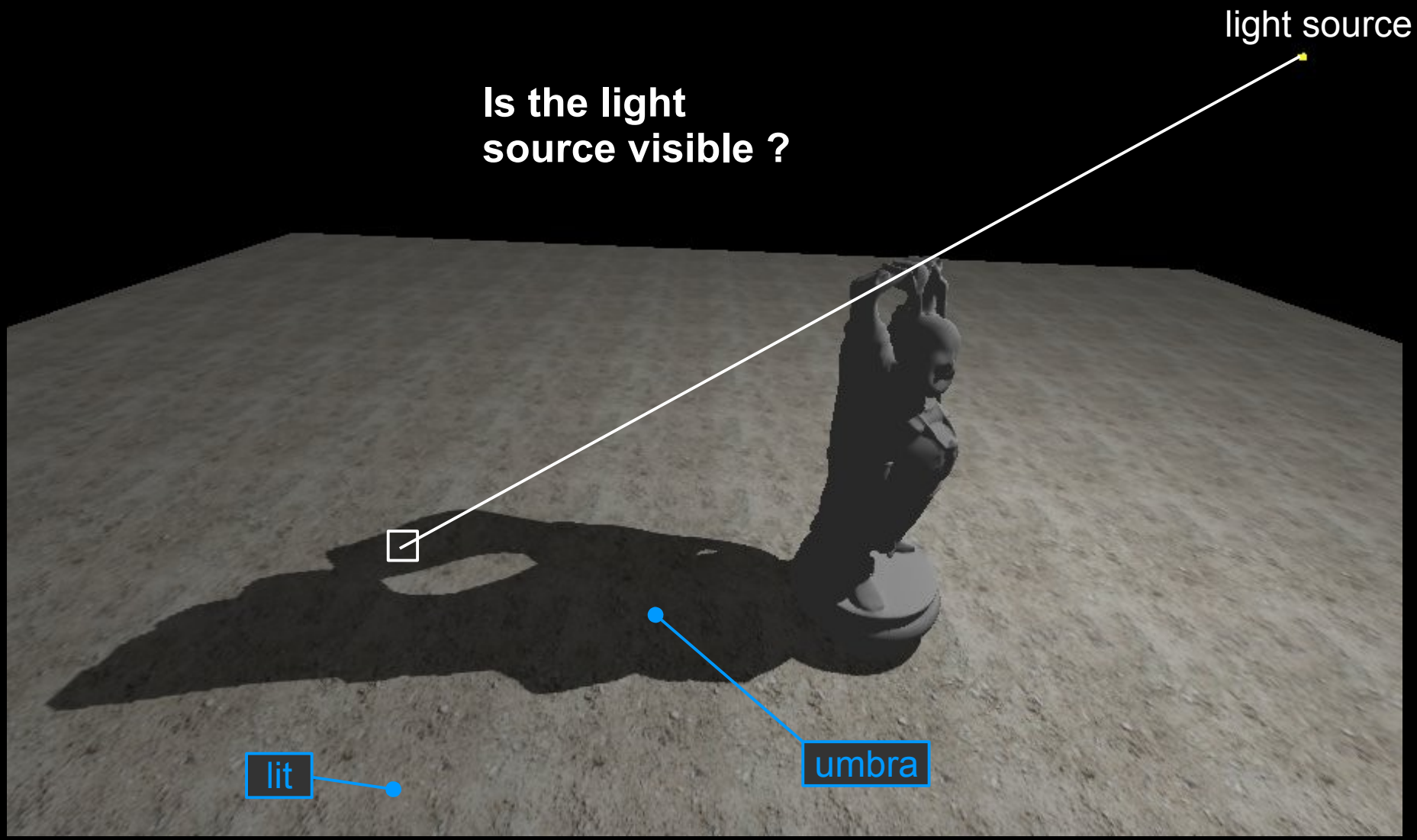
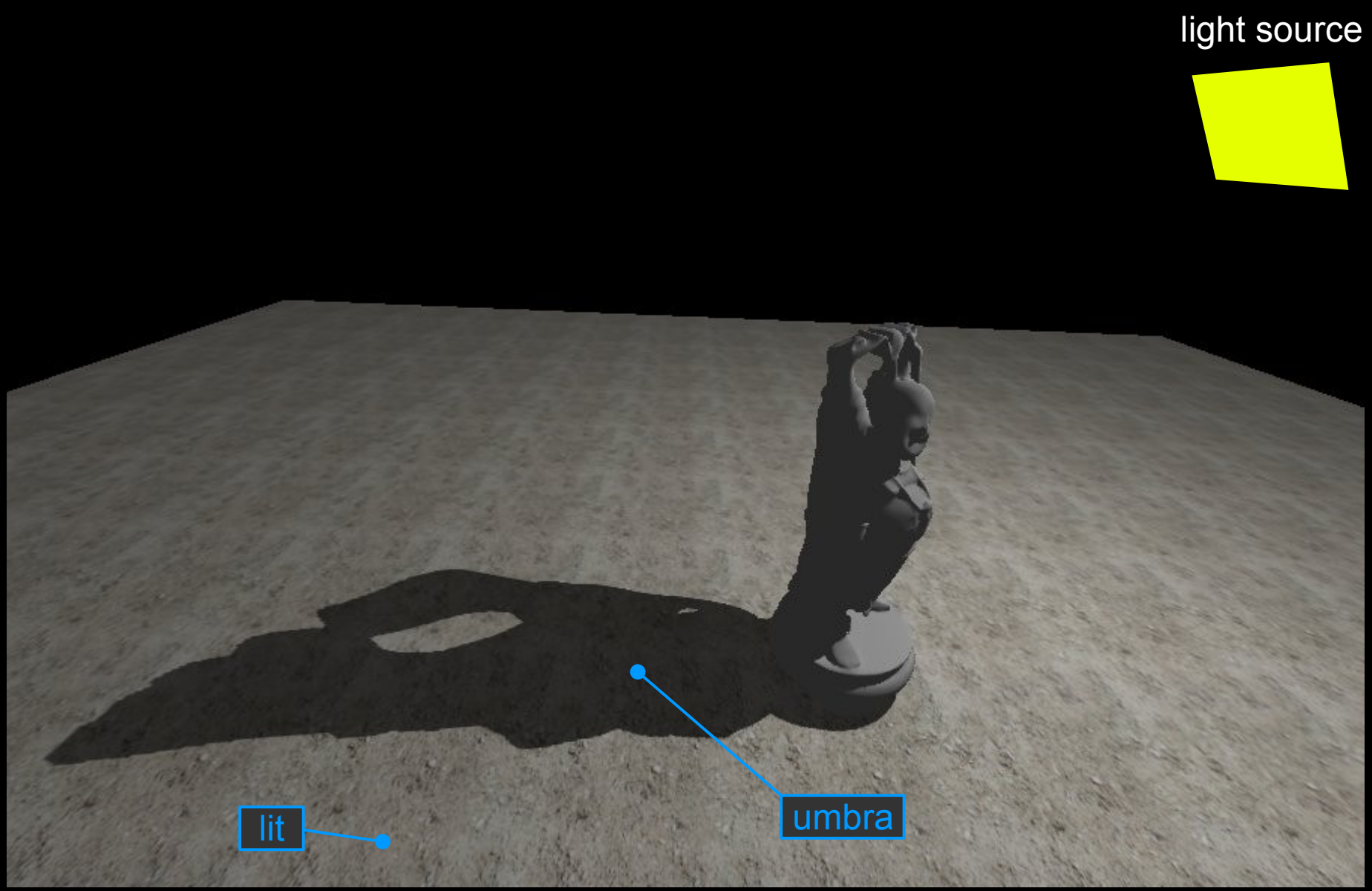
Real-time Soft Shadow Mapping by back-projection

Gaël Guennebaud
Loïc Barthe, Mathias Paulin
IRIT – UPS – CNRS
TOULOUSE – FRANCE
http://www.irit.fr/~Gael.Guennebaud/

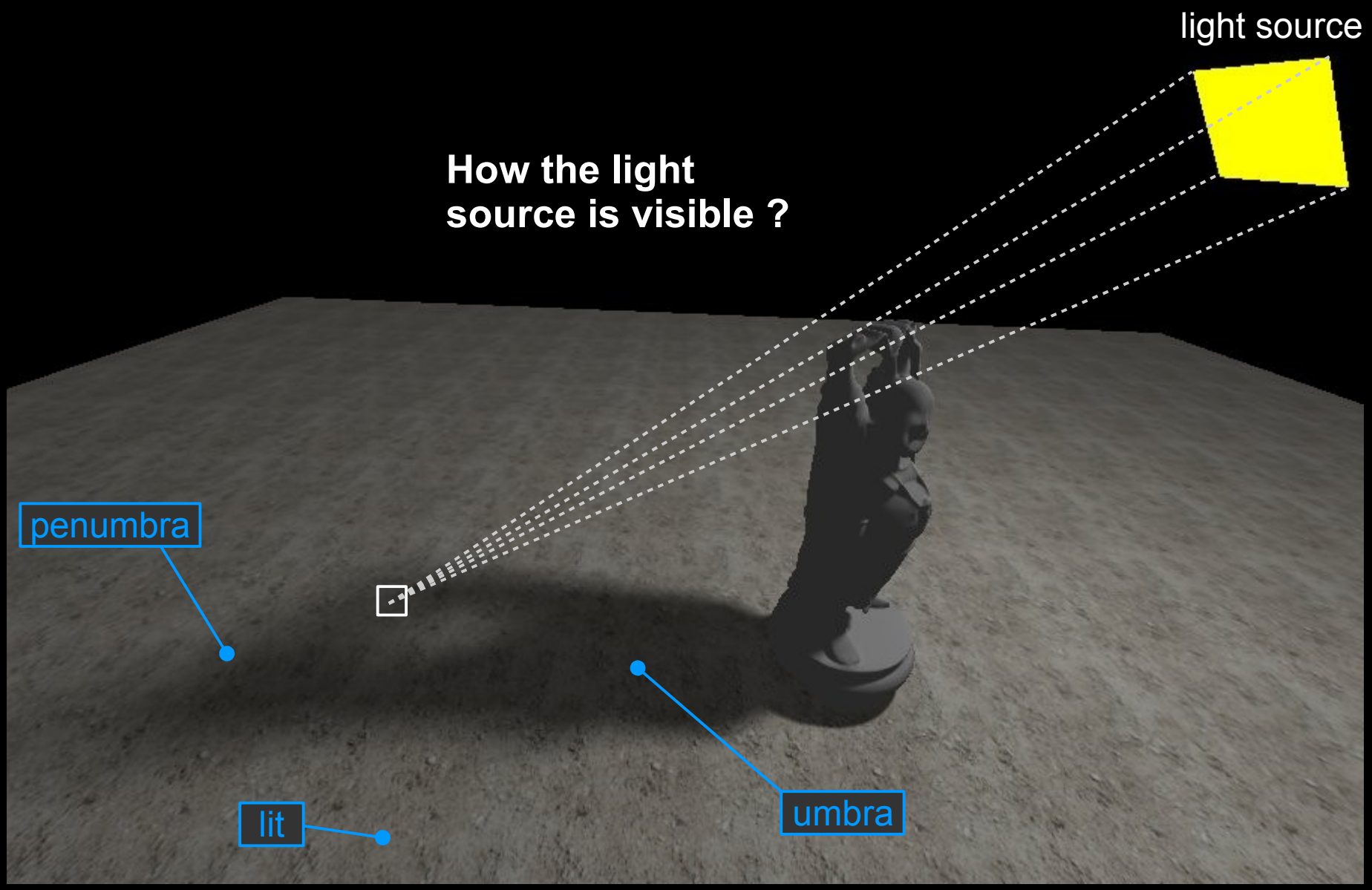
Hard Shadows



Soft Shadows



Soft Shadows



Soft Shadows: purpose

- A soft shadow algorithm should be:
  - **real-time** on complex and dynamic scenes
  - **generic**,
    - any geometry => *rasterizable (point cloud, mesh, image)*
    - any light source => *rectangular*
    - no occluder/receiver differentiation
  - **physically based** => *visually realistic*

Real-time soft shadows
previous works

- Two categories of approaches
  - geometry based *(shadow volume)*
    - object space silhouette extraction
  - image based *(shadow map)*
    - compute one or several shadow maps
- + some hybrids

Real-time soft shadows previous works

- Shadow volume based
  - *penumbra wedges* [Assarsson et al. 03]

    ✔ exact for flat objects without overlapping

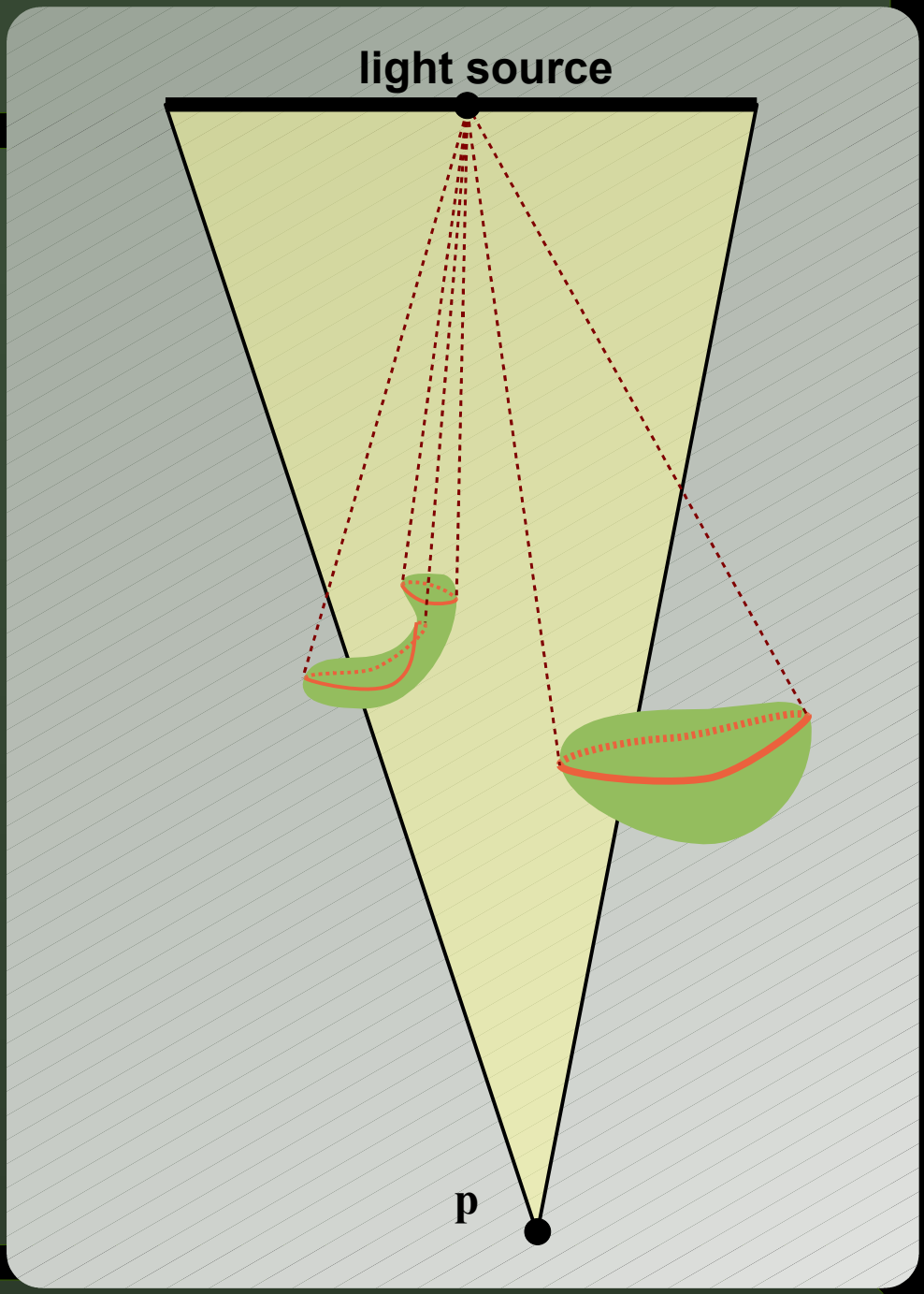
    ✘ wrong occluder fusion
    ✘ not scalable (limited to simple scenes)
    ✘ limited to manifold meshes

Real-time soft shadows previous works

- Image based methods (shadow maps)
  - some require (expensive) pre-computation
    - ✔ good realism
    - ✘ limited to static scenes
  - other based on distance ratio (no visibility computation)
    - ✘ (very) low quality

Principle

- What is the visibility percentage v_p between a point **p** and the light source ?
  - very complex problem
  - => simplifications
- Penumbra wedge approach:
  - occluders -> silhouettes

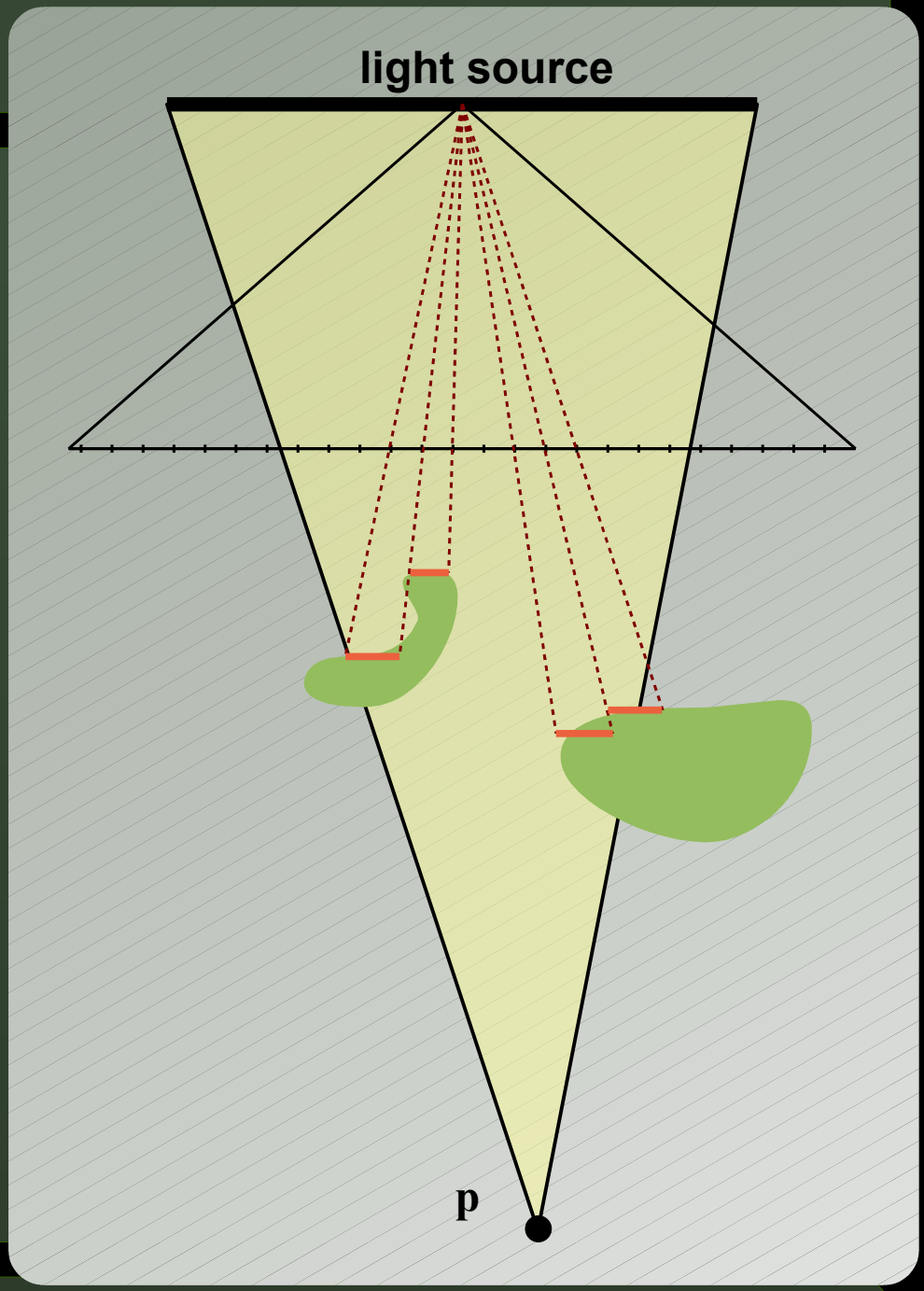


Principle

- What is the visibility percentage v_p between a point **p** and the light source ?

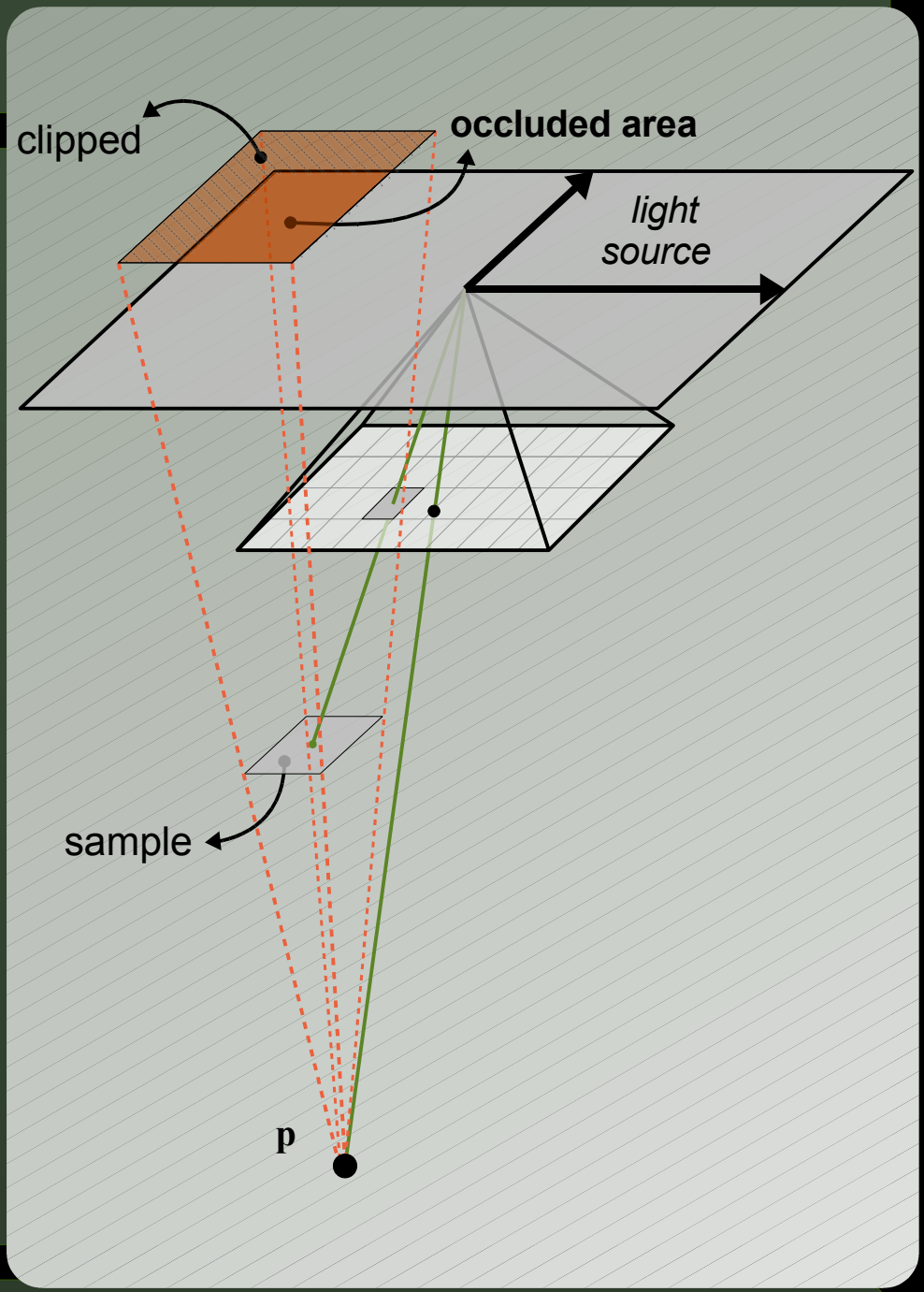
- Our approach:

  key idea: use the **shadow map** as a simplified and discrete representation of the scene



Principle

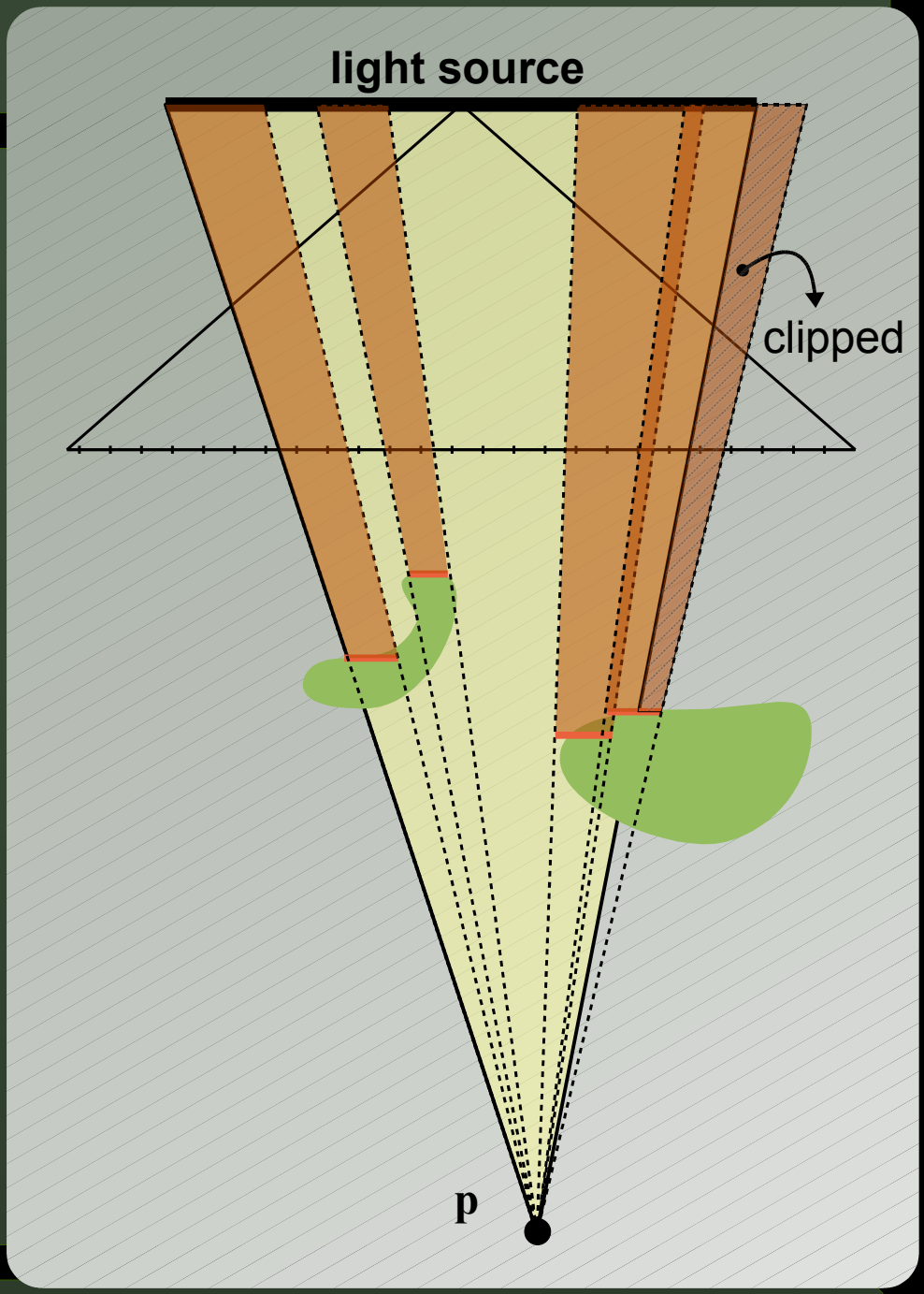
- Area occluded by a shadow map sample ?
  - back-projection on the light source
  - + clipping (trivial)



Principle

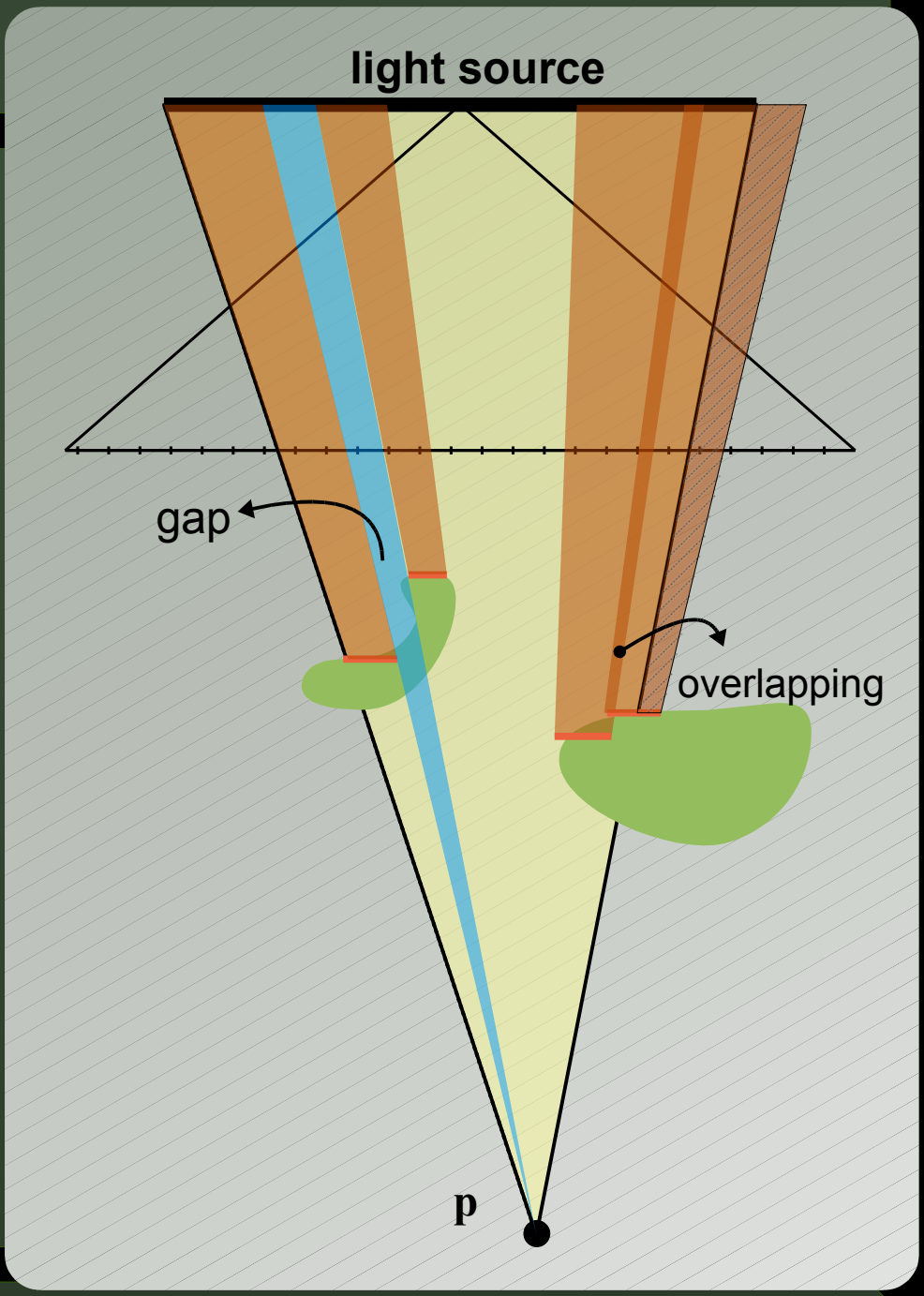
- What is the visibility percentage v_p between a point **p** and the light source ?
  - algorithm:

    subtract the area occluded by each shadow map sample

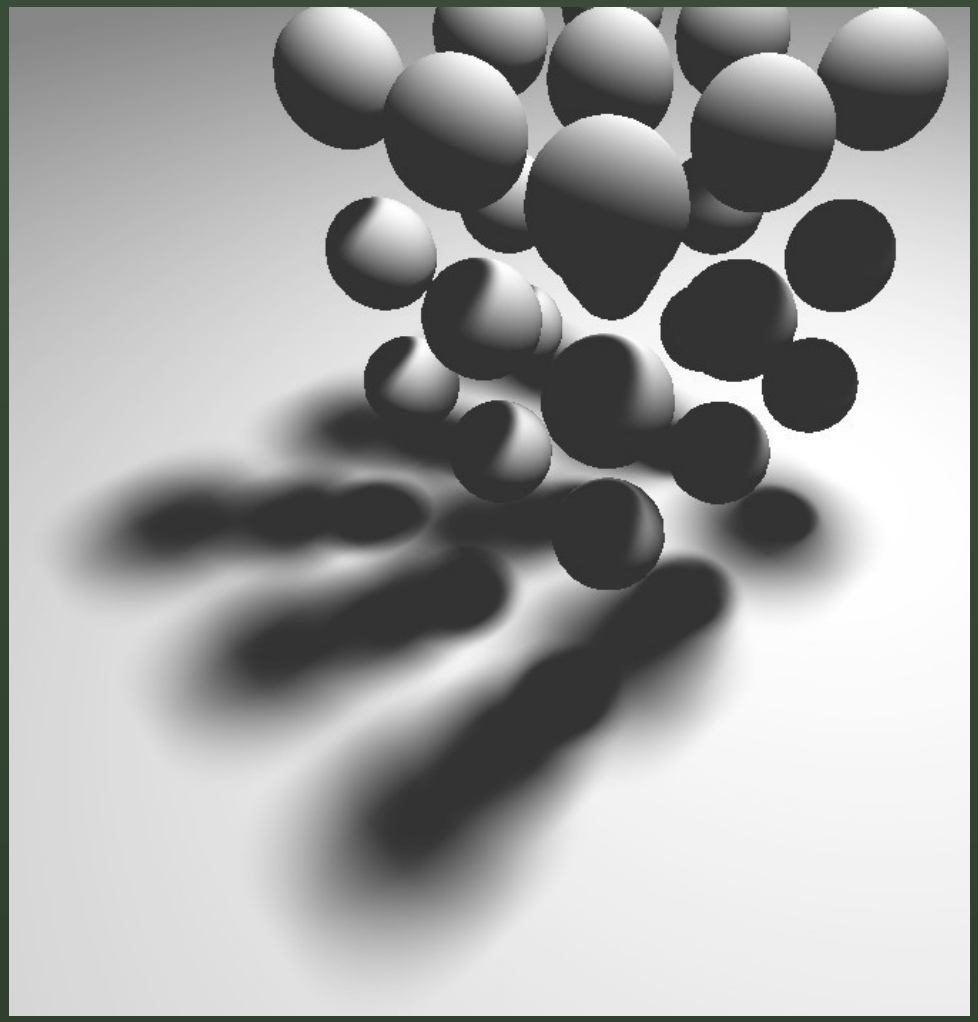


Main issue

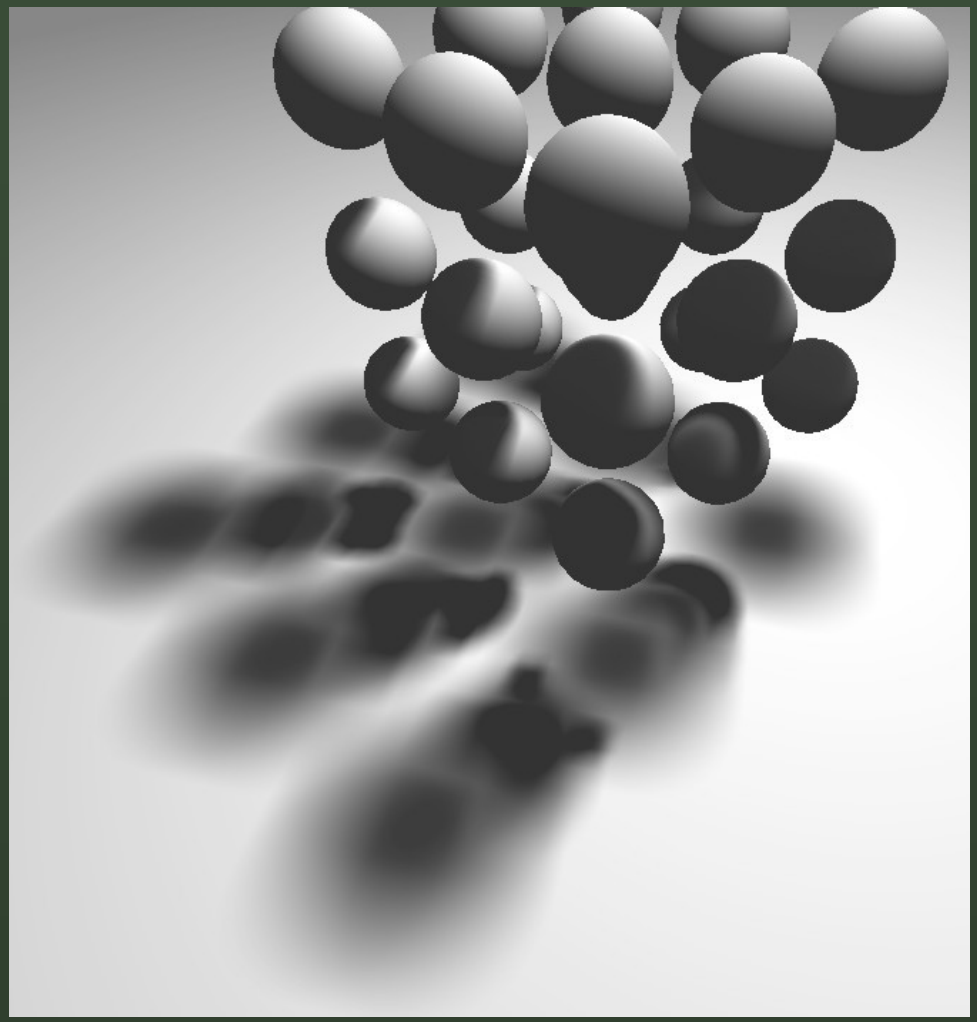
- gaps & overlaps
  - simple in 1D
  - very complex in 2D



Gaps & Overlaps



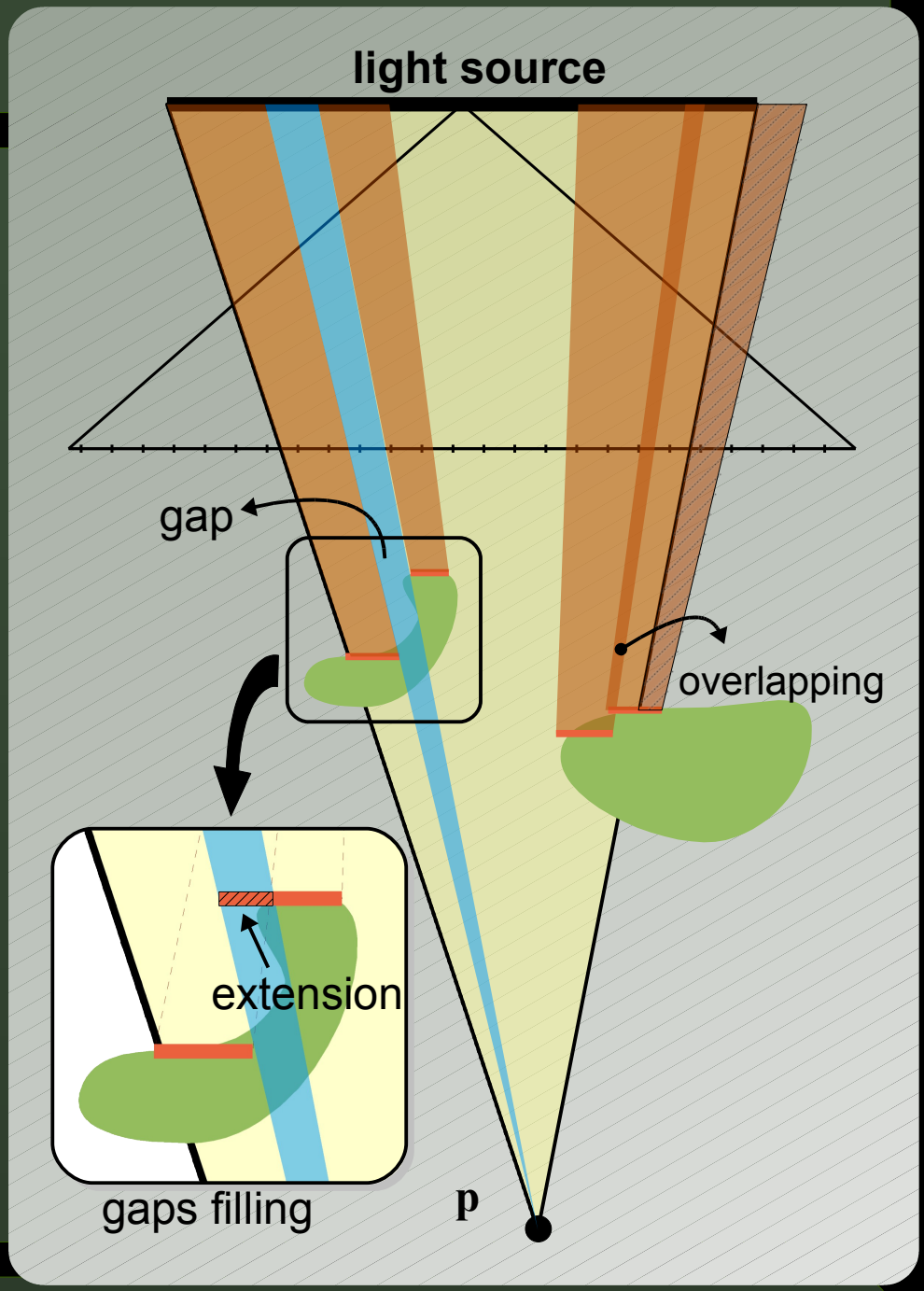
reference



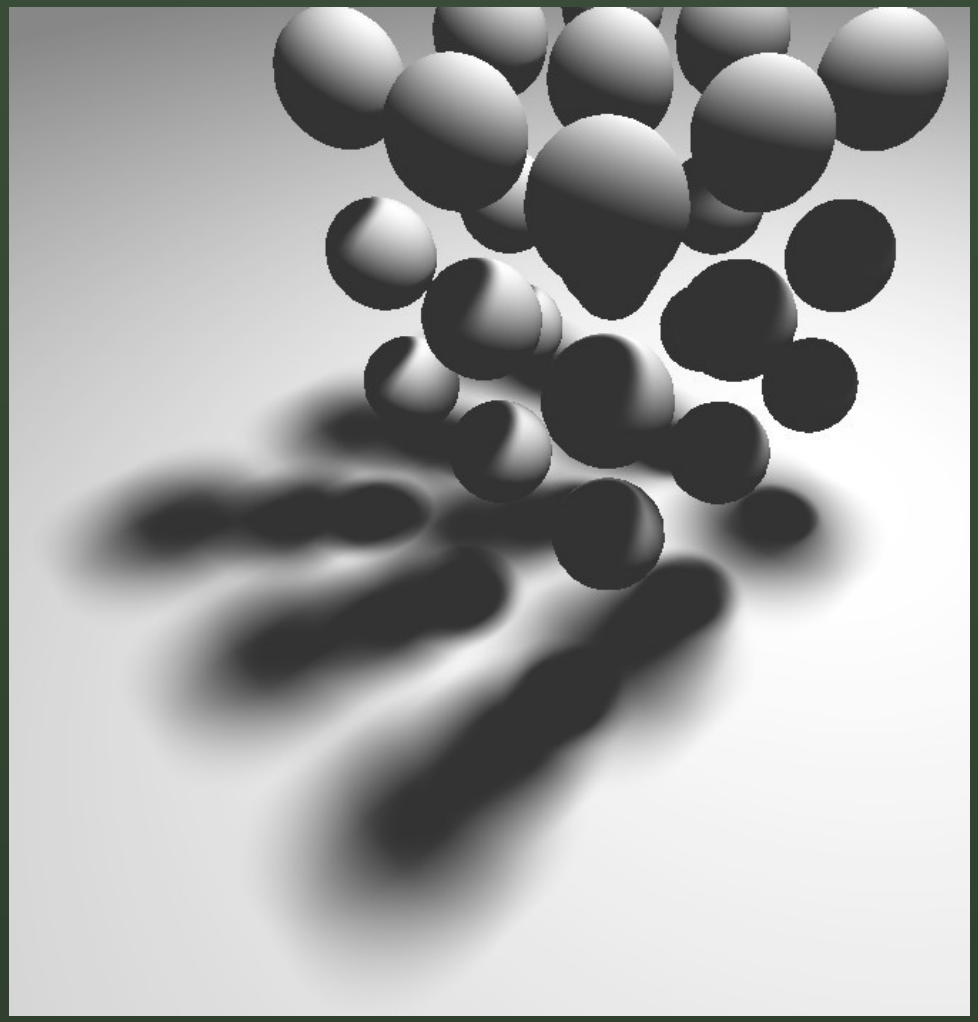
naive algorithm

Gaps filling

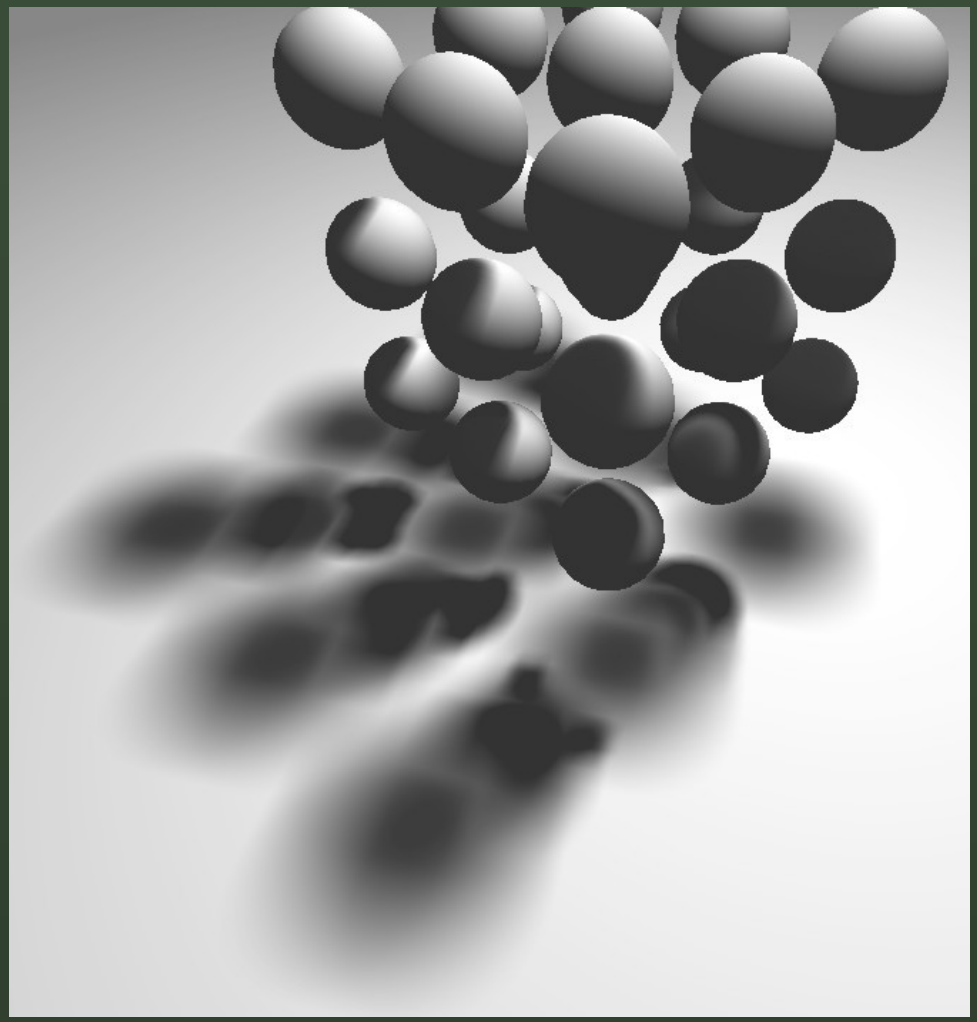
- gaps & overlaps
  - simple in 1D
  - very complex in 2D
- overlap artifacts are acceptable
- => at this time, we just fill the gaps



Gaps filling

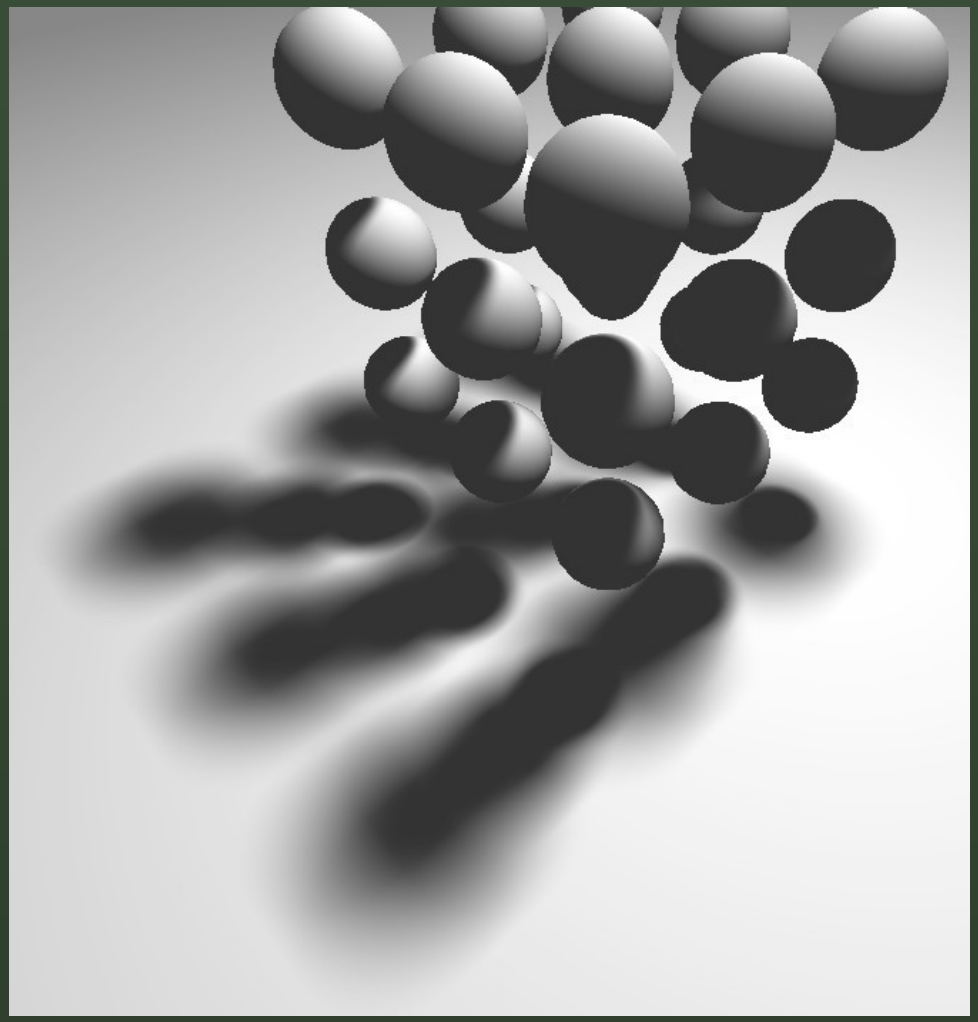


reference

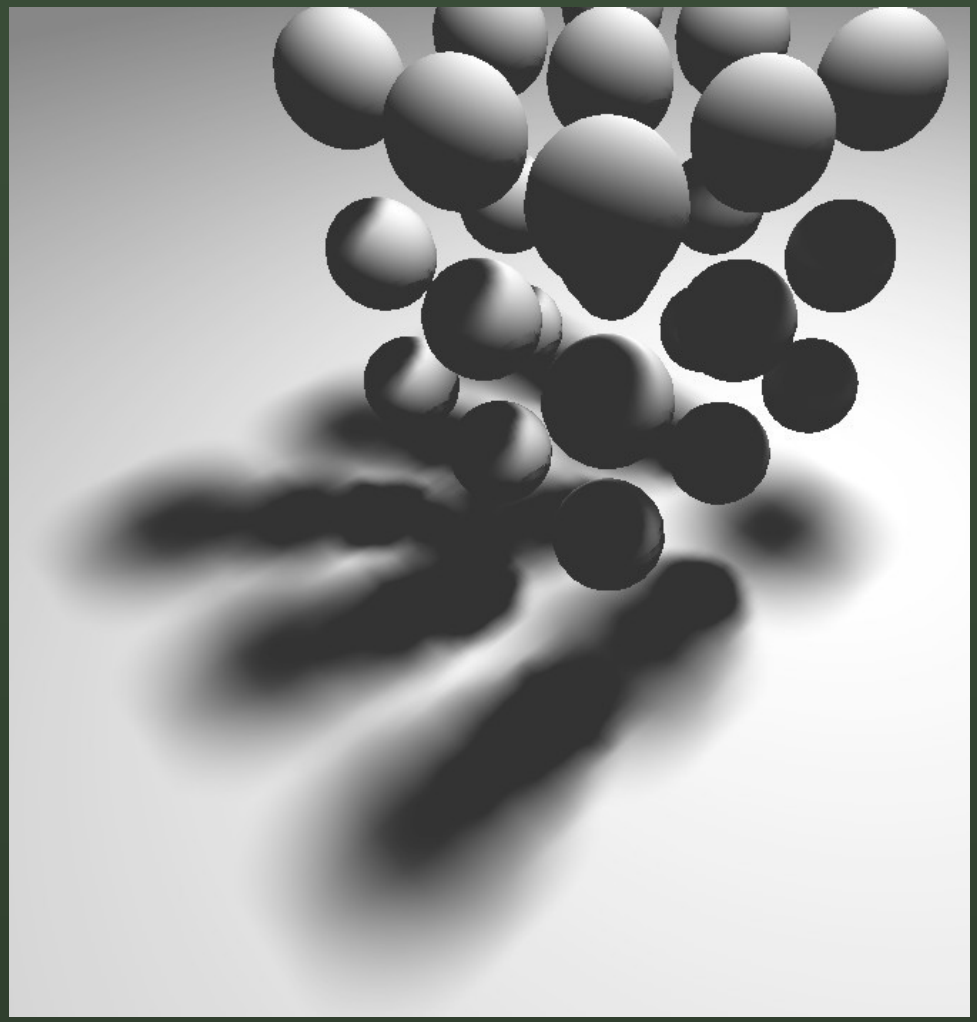


naive algorithm

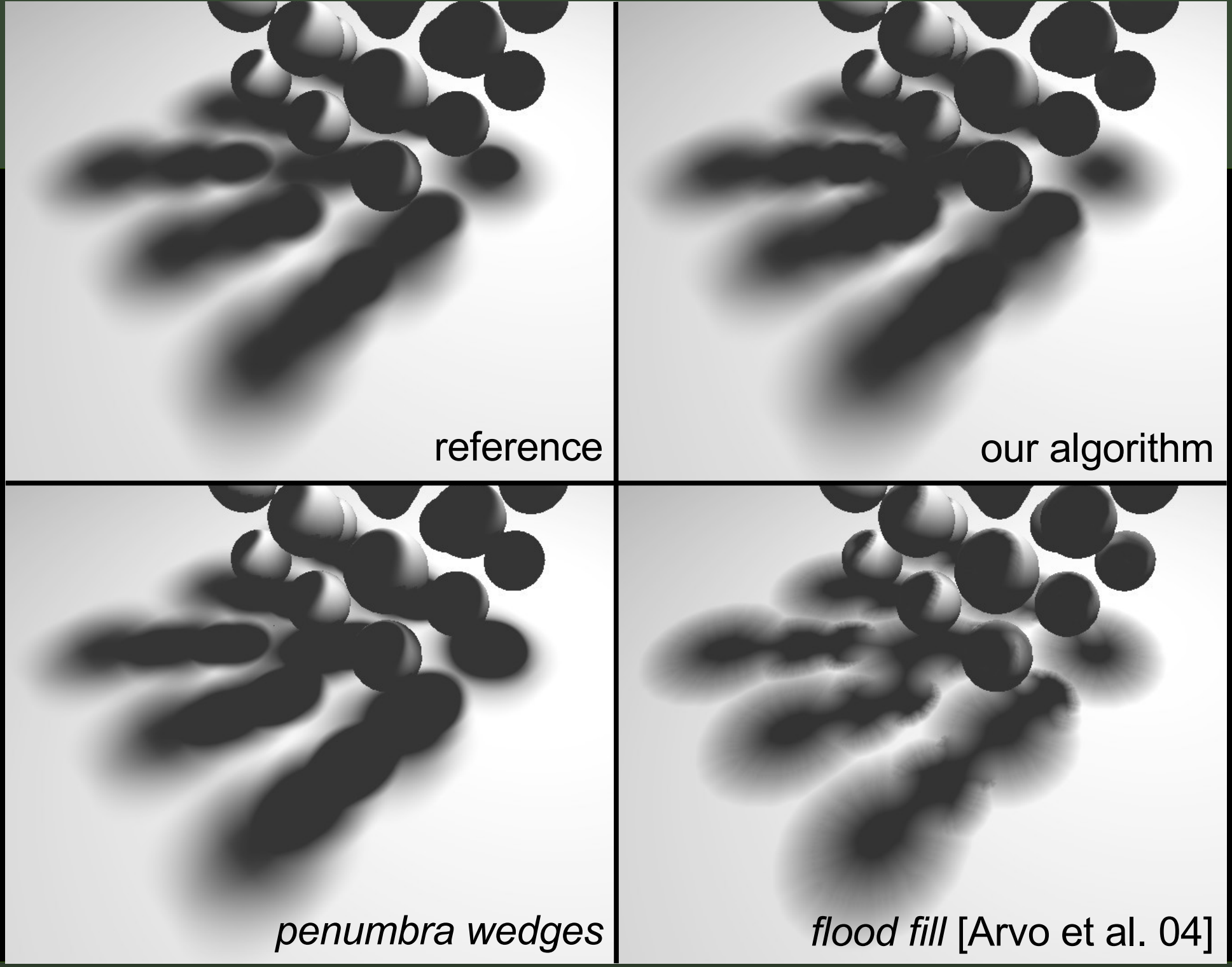
Gaps filling

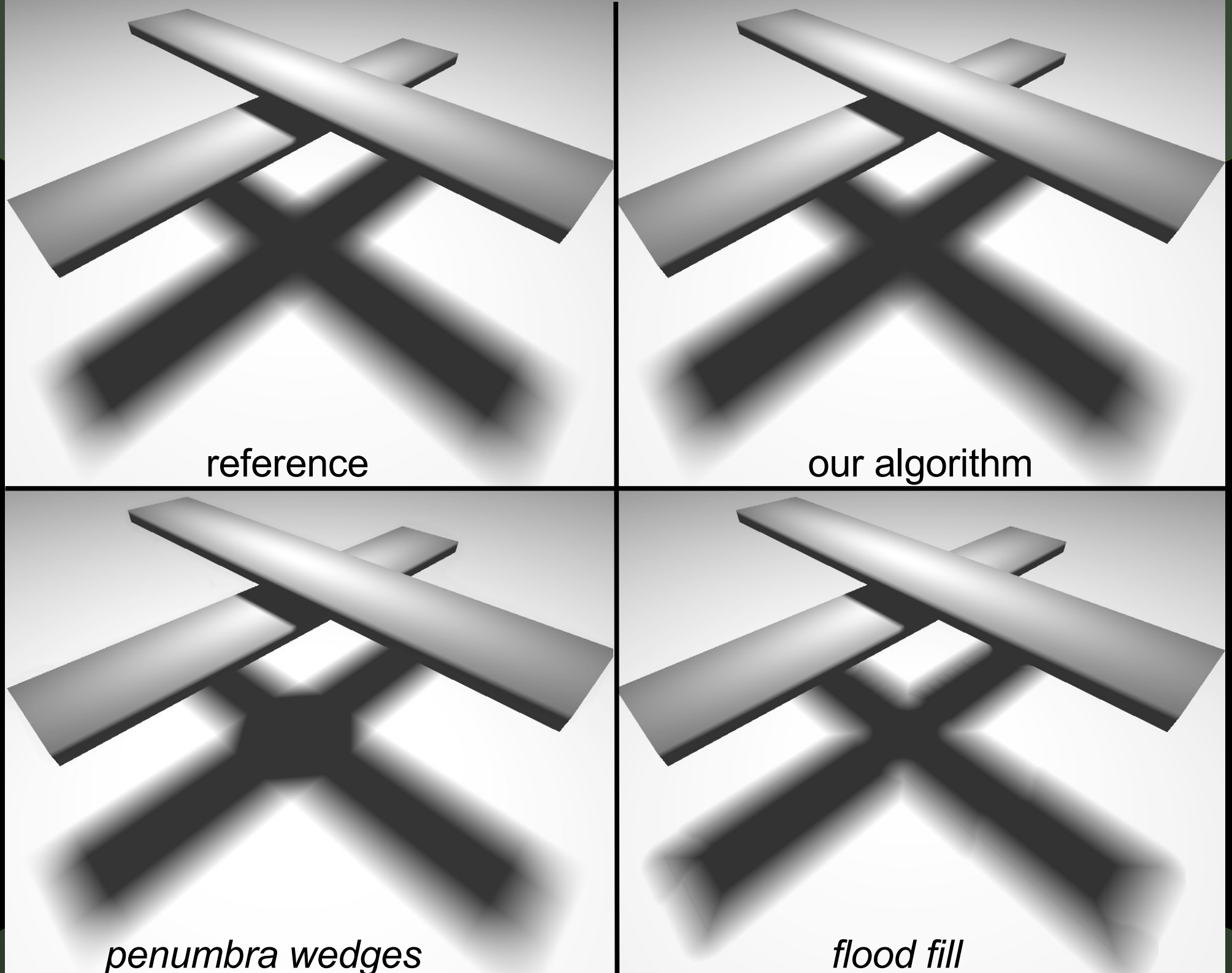


reference



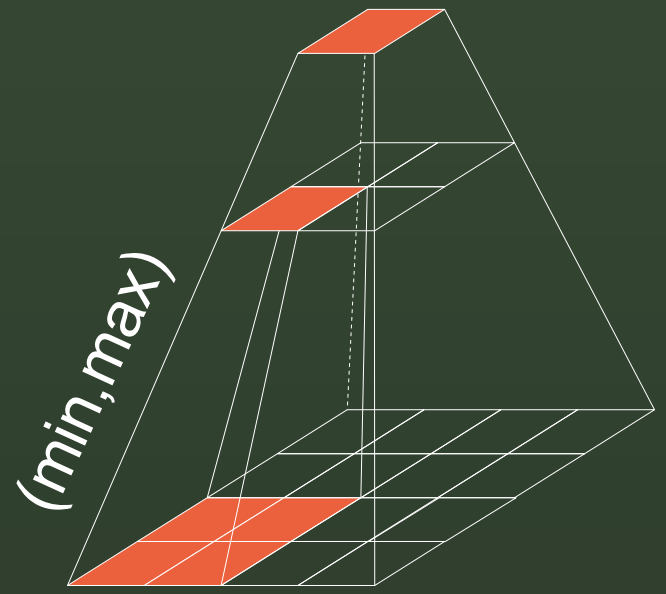
with gap filling


reference
our algorithm
penumbra wedges
flood fill [Arvo et al. 04]


reference
our algorithm
penumbra wedges
flood fill

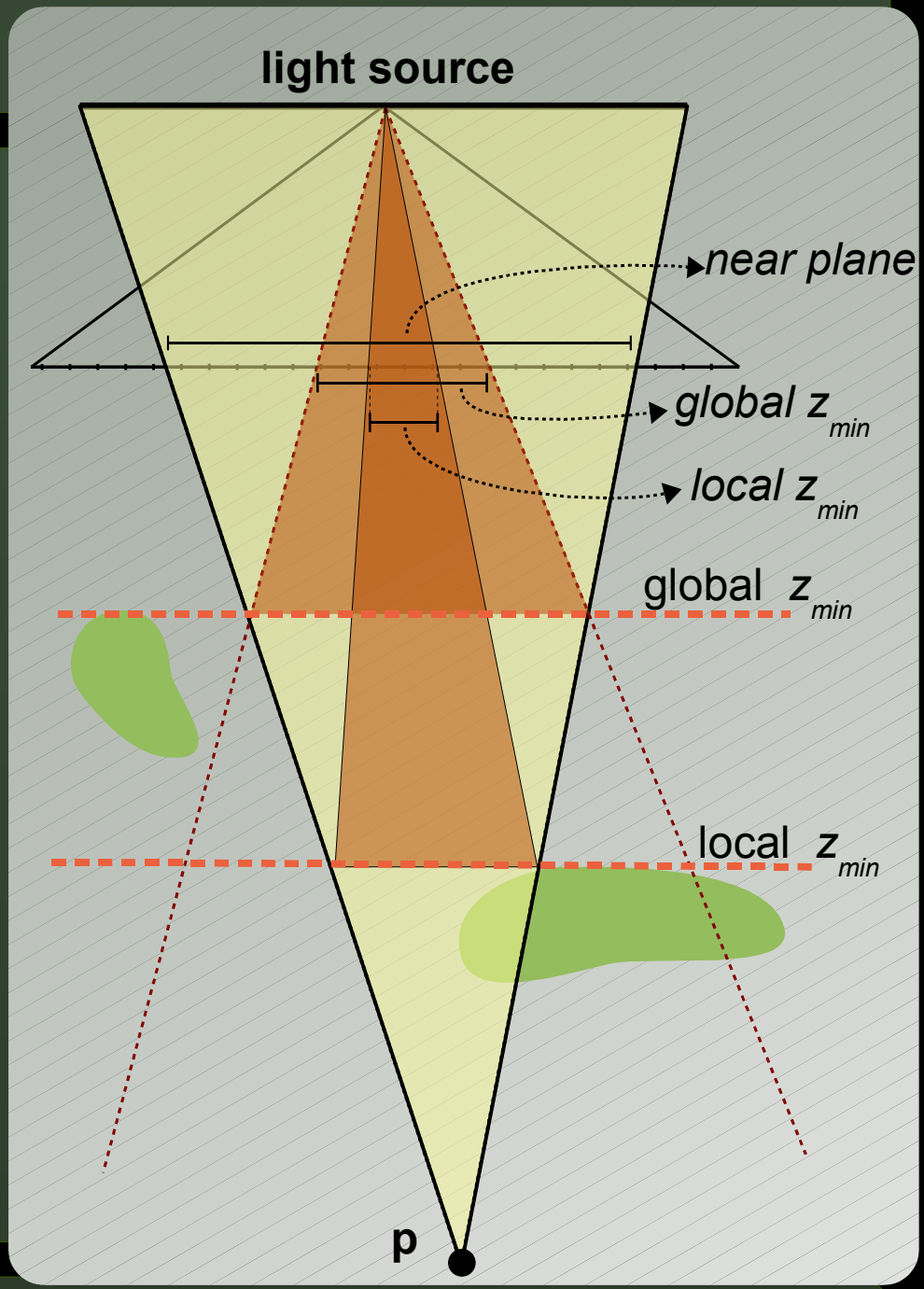
Optimizations
hierarchical shadow map (HSM)

- shadow map → hierarchical shadow map (HSM)
  - similar to mipmaps
  - each pixel stores the min and max depth values



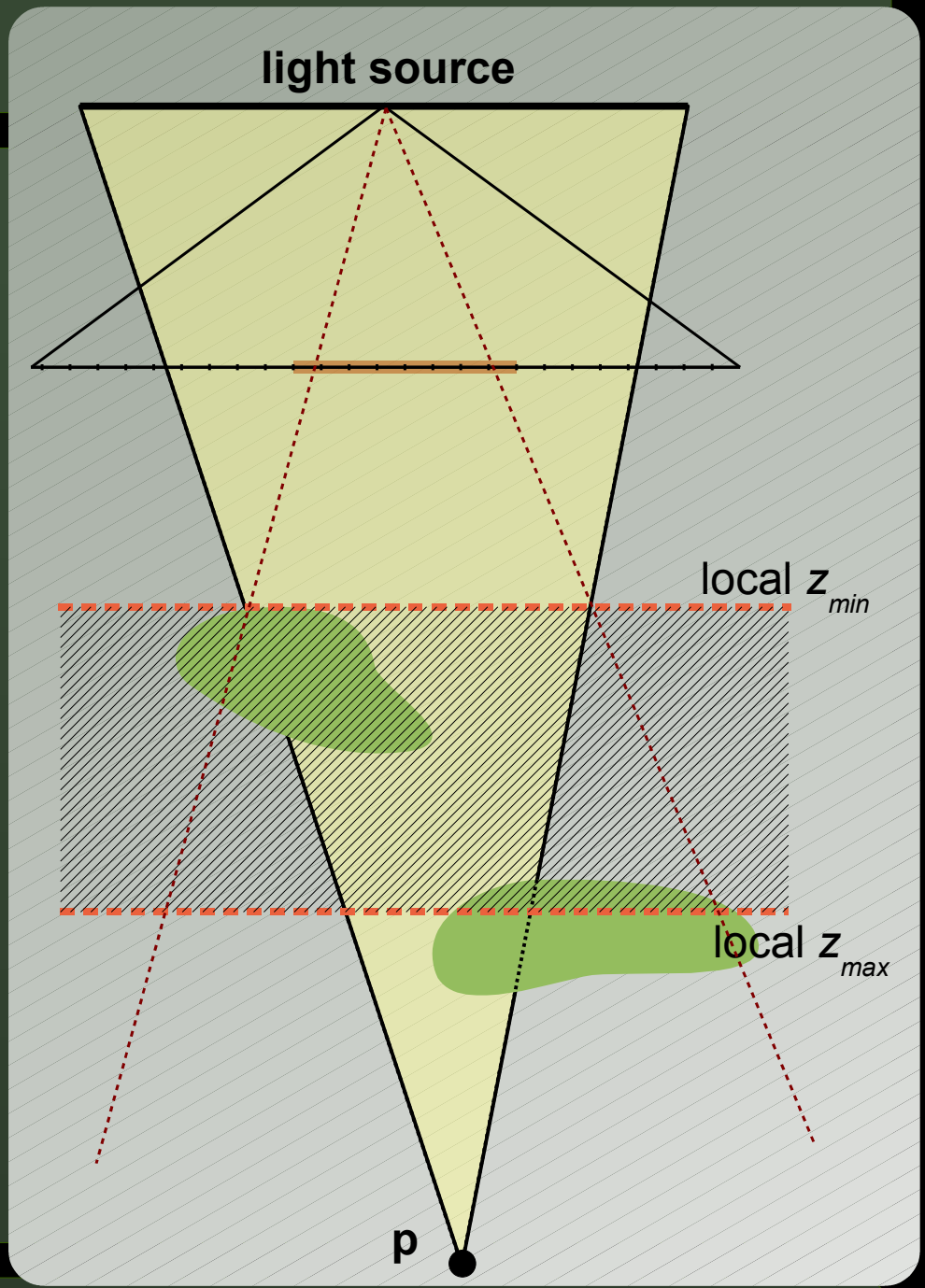
Optimizations - I

- **Occluder search area reduction**
  (treat only samples which may occlude the light)
- Clip the pyramid **p**-light by:
  - the **near plane**
  - the **global z_{min}**
    (last HSM level)
  - iteratively by the **local z_{min}**
    (HSM access)



Optimizations - II

- **Penumbra classification**
  (treat only visible pixel which are in the penumbra)
- uses the local z_{min} and z_{max}
  - if $z_p \leq z_{min}$ then $v_p = 1$
  - if $z_p \geq z_{max}$ then $v_p = 0$
  - else $v_p \in [0,1]$



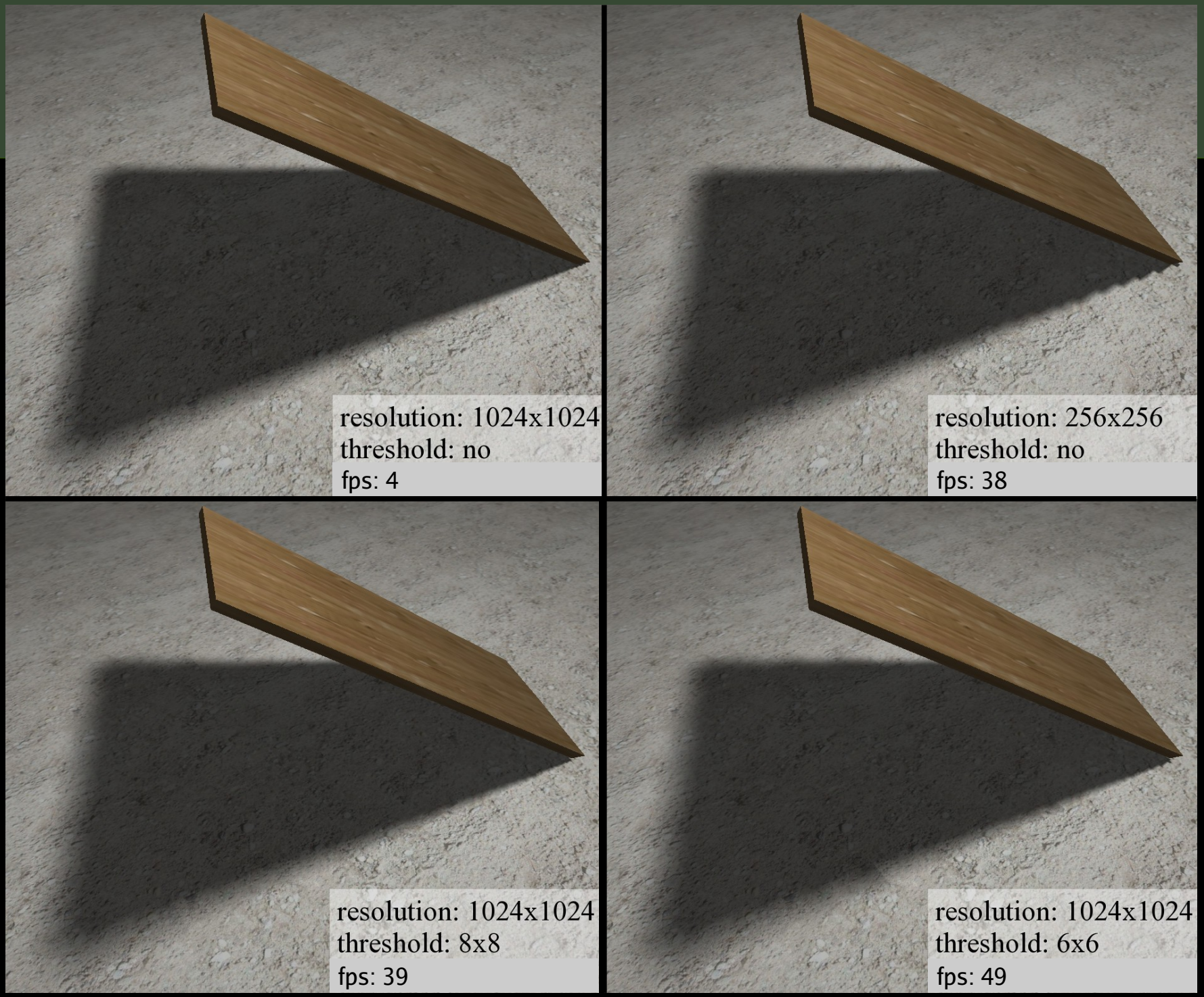
Optimizations - III

- Rendering cost depends on the shadow map resolution

  => adaptive resolution

Optimizations - III

- **Adaptive precision**
  (use low resolution for large penumbra)
  - if the occluder region is too large
    => use a low level of the HSM
  - \+ guaranty the real-time
  - \- slight artifacts at the level transitions


resolution: 1024x1024
threshold: no
fps: 4
resolution: 256x256
threshold: no
fps: 38
resolution: 1024x1024
threshold: 8x8
fps: 39
resolution: 1024x1024
threshold: 6x6
fps: 49

Summary of the algorithm

- Draw the scene in the shadow map
- Compute the HSM (GPGPU, ~3 ms)
- Draw the scene from the view point in a depth buffer
  - ~ deferred shading

- Compute the visibility buffer:
  - **for** each pixel **p** *(draw a quad)*
    - estimate the occluder search area (HSM)
    - if p is lit or in the umbra then **OK**
    - **else** loop over the occluder samples...
      - ~ 15 instructions / sample

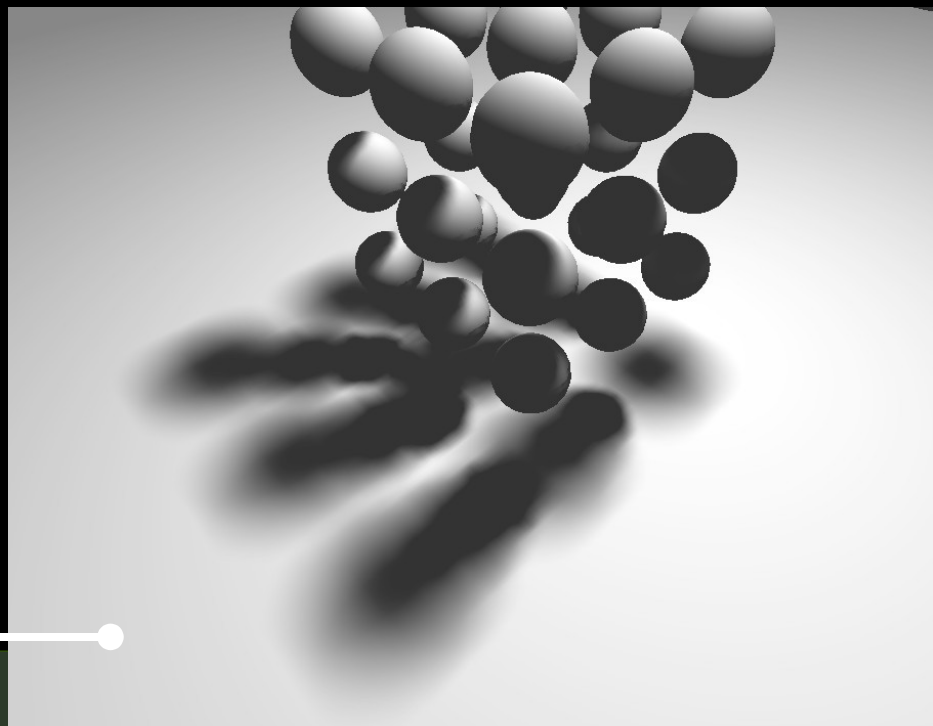
dynamic branching

- Draw the scene with lighting and soft shadows !

performances

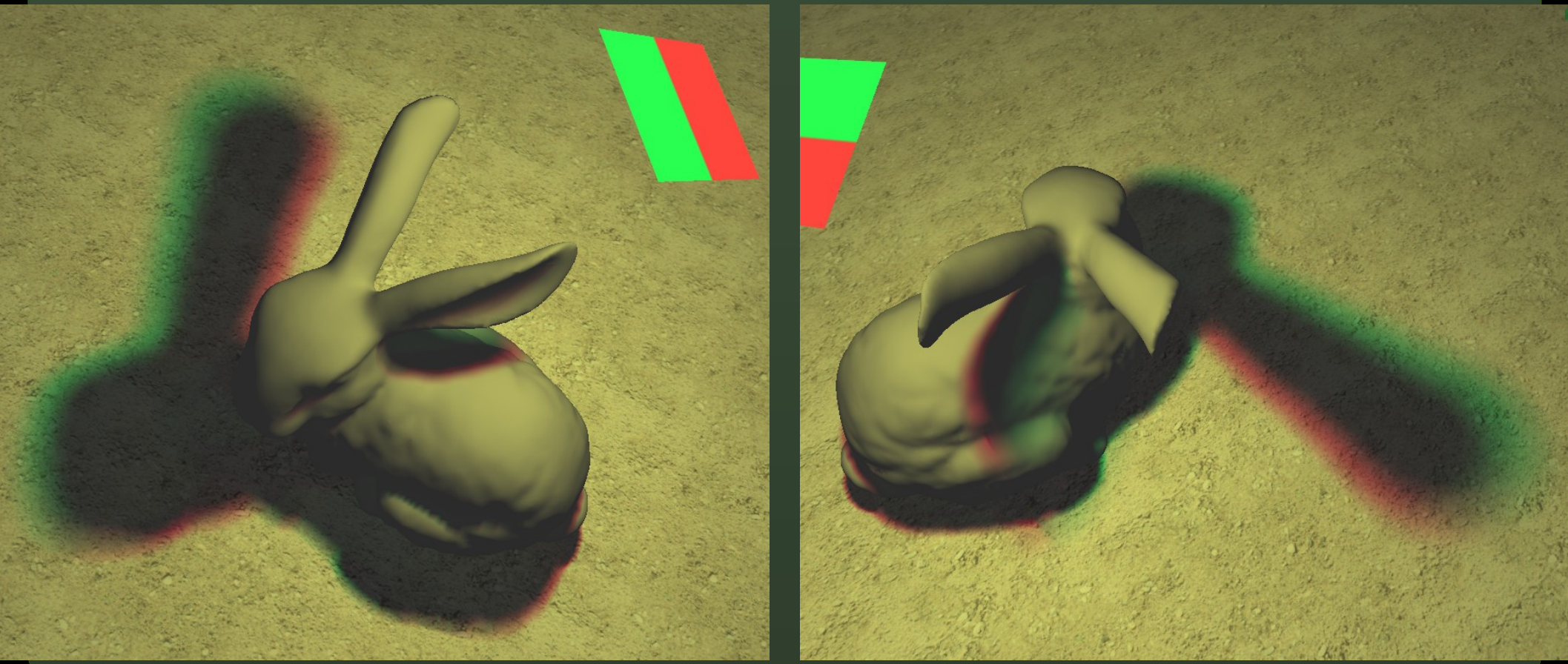
(on a GeForce 7800)

Scene	Fig. 7	Fig. 1	Fig. 8
Shadow map	1.7	2.6	8.7
Camera depth map	0.7	1.3	7.6
HSM construction	3.1	3.1	3.1
Visibility pass 1	0.9	0.9	0.9
Visibility pass 2	39	28	15
Final rendering pass	0.8	1.6	8.2
Total (ms)	46.2	37.5	43.5
fps	21.6	26.6	23



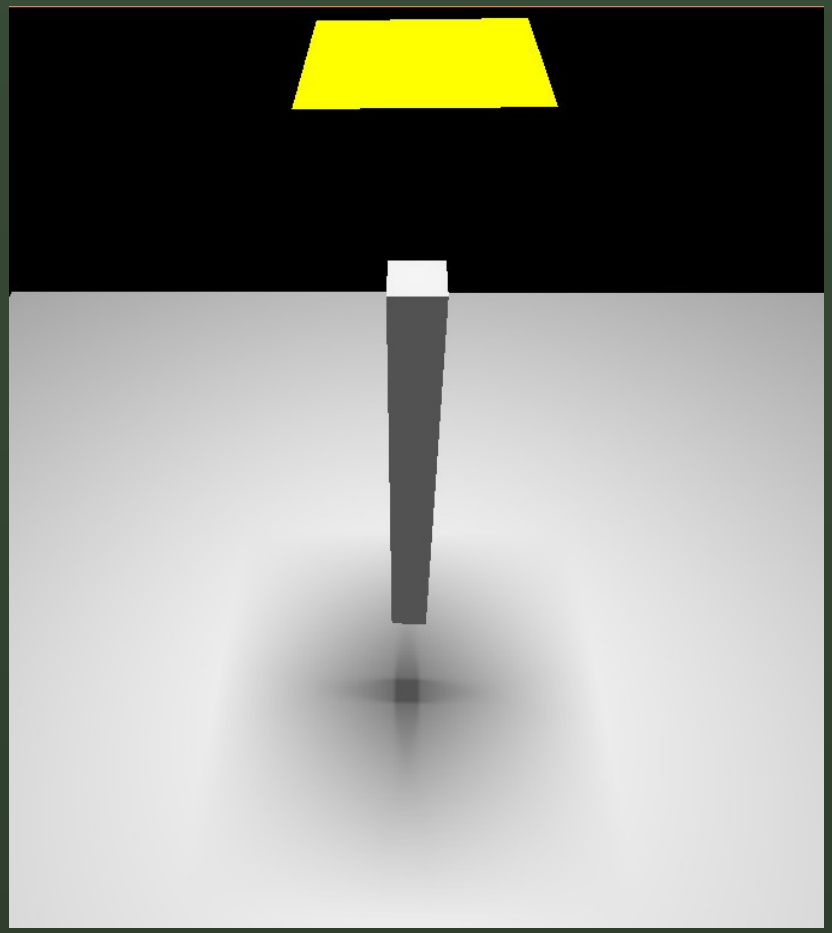
Textured light source

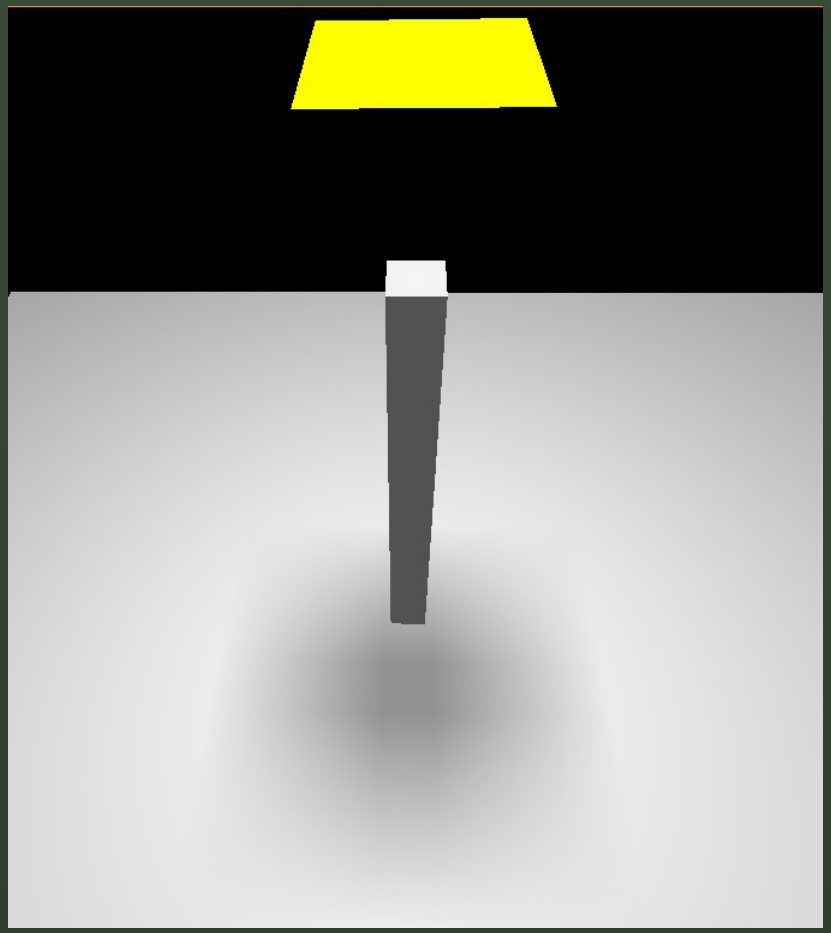
- via 4D texture (expensive and coarse)
  or via a « Summed Area Table » (SAT)



Other limitations

- Only parts visible from the light center
  are taken into account in the visibility computation





Comparison with [Atty et al. 2006]

- Recent work done in parallel
  - similar visibility computation
- Main differences:
  - all computations are done in the light space
  - loops are swapped:
    - `for each shadow map samples` **`s`** *(CPU)*
      - `for each point` **`p`** `of the scene` *(quad rasterization)*
        - `remove from` $\mathbf{v}_p$ `the area occluded by` **`s`**
          (fragment program)

Comparison with [Atty et al. 2006]

- General consequences:
  - ✗ occluders & receivers must be distinct set
  - ✗ higher complexity
  - ✗ no « gap filling »
  - ✔ 2 passes approach reducing the “single light sample artifacts” but...
- Current consequences: (GPU limitations)
  - ✔ no dynamic branching at the fragment level
  - ✗ limited to low shadow map resolutions (200x200)

Soft shadow mapping conclusion

- Summary
  - provides high quality soft shadows in real-time
    - not physically exact, but close in most cases
  - has all the advantages of shadow maps
    - suitable for complex scenes
    - suitable for any rasterizable geometry
    - no pre-computation => dynamic scenes

Soft shadow mapping future works

- More accuracy
  - overlap error
  - single light sample error
- Aliasing
  - increase the effective resolution (e.g. ASM, PSM...)
- Performances
  - adaptive strategy without discontinuity

Soft shadow mapping future works

- More accuracy
  - overlap error — *almost OK*
  - single light sample error — *almost OK*
- Aliasing
  - increase the effective resolution (e.g. ASM, PSM...) — *...*
- Performances
  - adaptive strategy without discontinuity — *OK*

END