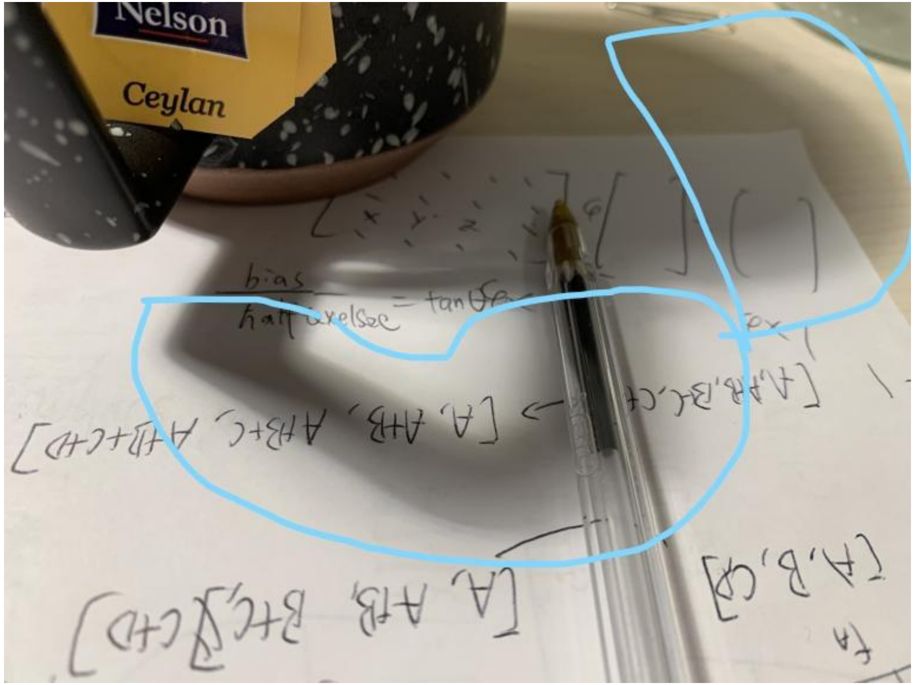
IG3DA-Project: Variance Soft Shadow Mapping

Putian YUAN

What Variance Soft Shadow Mapping(VSSM) does?

To render soft shadow(e.g. penumbra)

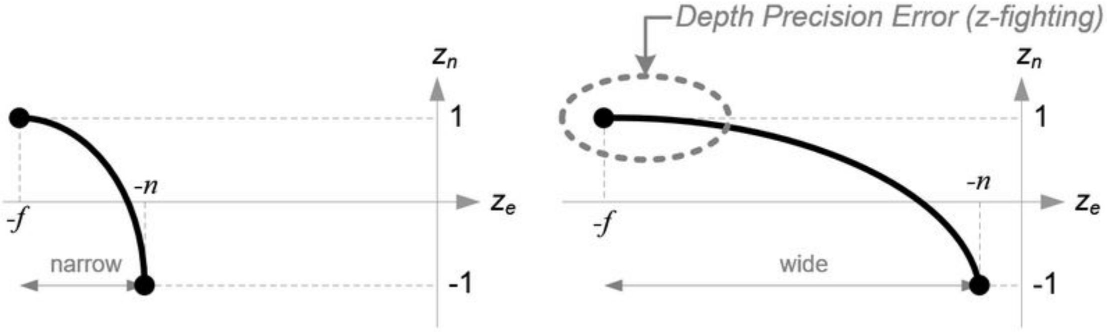


Related work

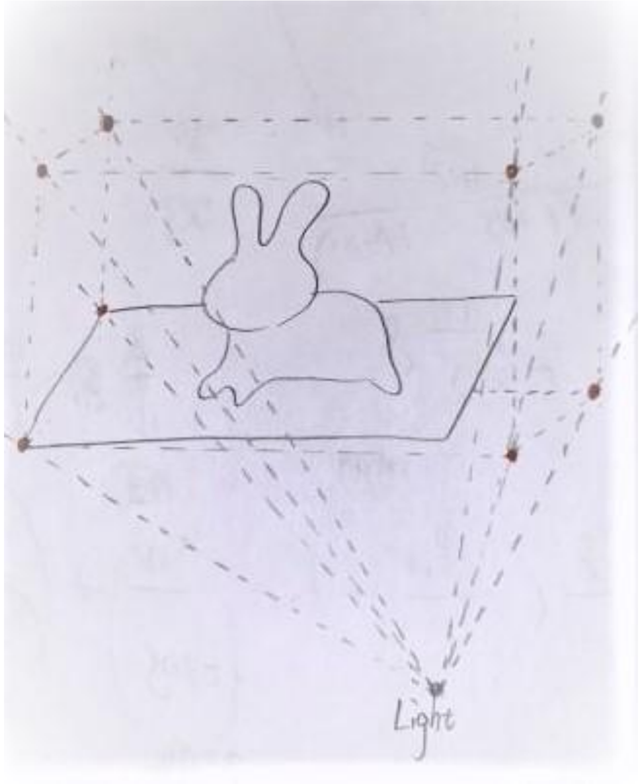
- Shadow Maps
- Percentage-Closer Filtering(PCF)
- Percentage-Closer Soft Shadow(PCSS)
- Variance Shadow Maps(VSM)
- Summed-Area Table(SAT)
- Summed-Area Table Variance Shadow Maps(SAT-VSM)

Implementation

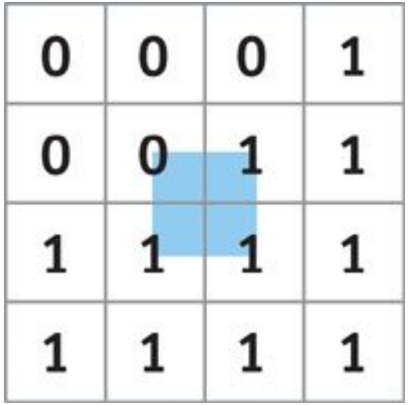
Shadow Maps



Comparison of Depth Buffer Precisions



PCF



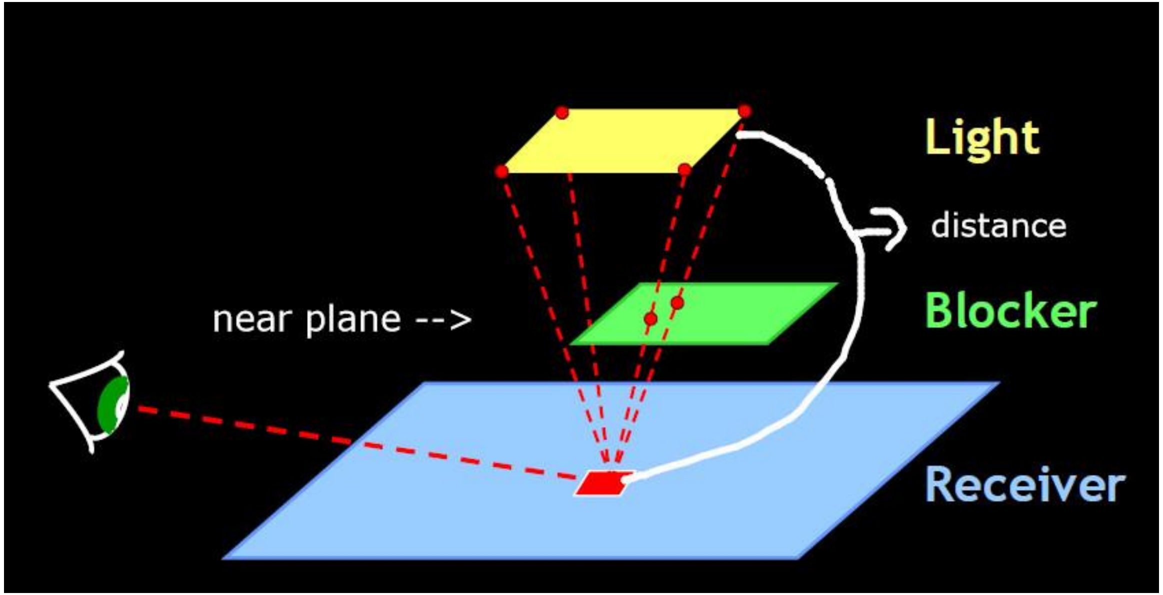
0	0	0	1
0	0	1	1
1	1	1	1
1	1	1	1

2*2 kernel

Averaged result = (0+1+1+1)/4 = 0.75

PCSS

- Compute averaged blocker depth
- Penumbra size estimation
- Use PCF with penumbra size to filter depth-test results



Blocker Search Size = (distance-near)/distance * ~~LightSize~~

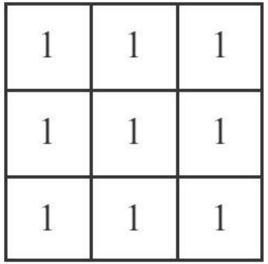
= (distance-near)/distance * MaxSearchSize

$$\omega_{penumbra} = \frac{\omega_{light}(d_{receiver} - d_{blocker})}{d_{blocker}}$$

$$= \omega_{light}\left(\frac{d_{receiver}}{d_{blocker}} - 1\right)$$

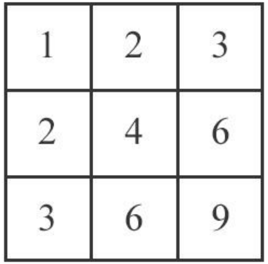
Clamp the penumbra size into user-defined range [min, max]

SAT



1	1	1
1	1	1
1	1	1

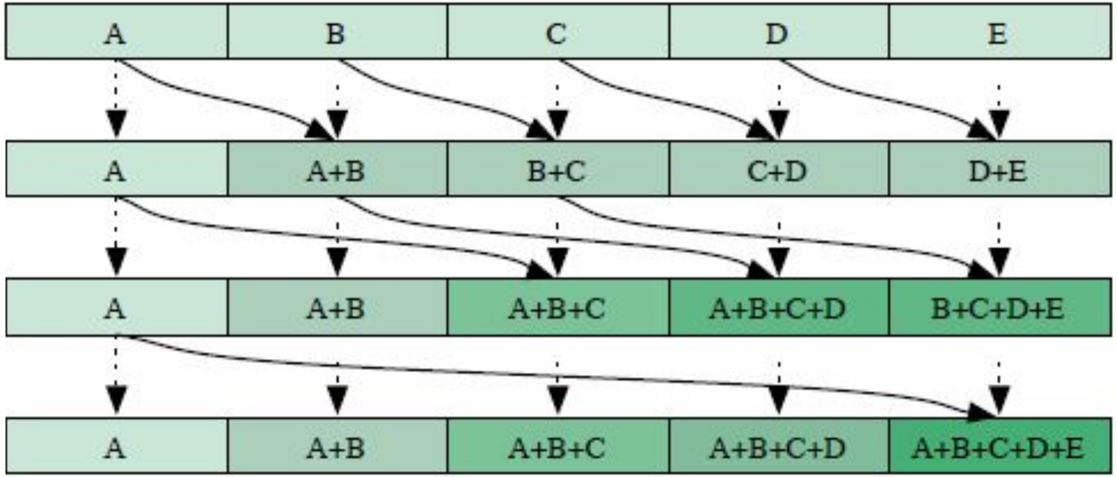
Input image



1	2	3
2	4	6
3	6	9

Integral image

Origin(0,0) is at left-top



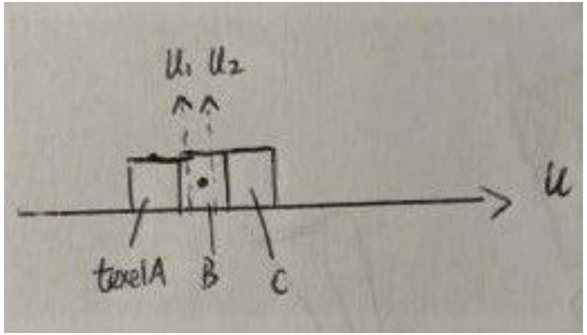
recursive doubling algorithm

- 32-bits Texture for storage
- Reading 8 texels at the same time
- Minus 0.5 for each texel before generating SAT

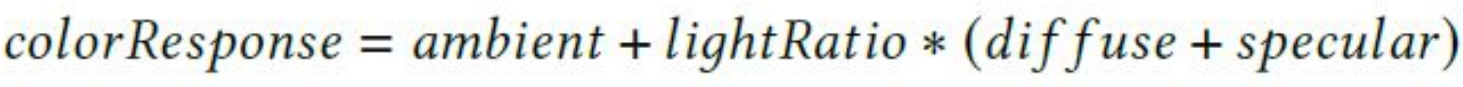
VSM

$$M_2 = \mu^2 + \frac{1}{4}\left(\left[\frac{\partial f}{\partial x}\right]^2 + \left[\frac{\partial f}{\partial y}\right]^2\right).$$ **Instead of** $$M_2 = E(x^2) = \int_{-\infty}^{\infty} x^2 p(x) dx$$

- Linear depth scaled into [0,1] not projected depth(no z-fighting)
- Minimum variance to eliminate shadow acne
- Minimum pMin to reduce light bleeding
- 32-bits floating-point RG-color texture(mipmap, linear interpolation)
- Four neighbor texels for bilinear interpolation

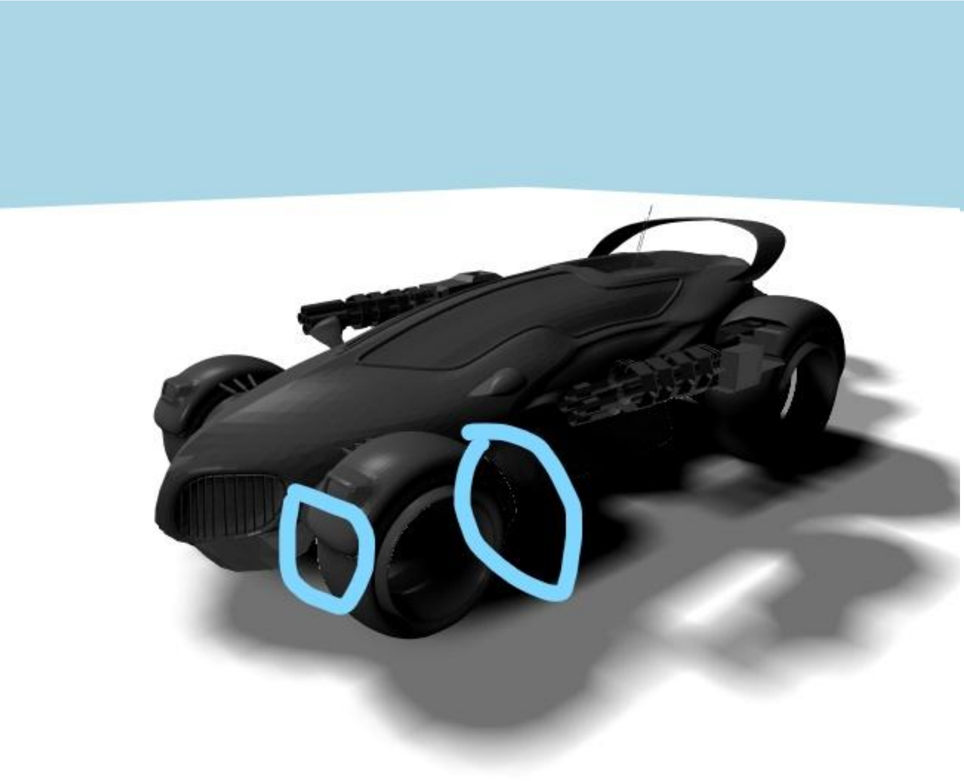


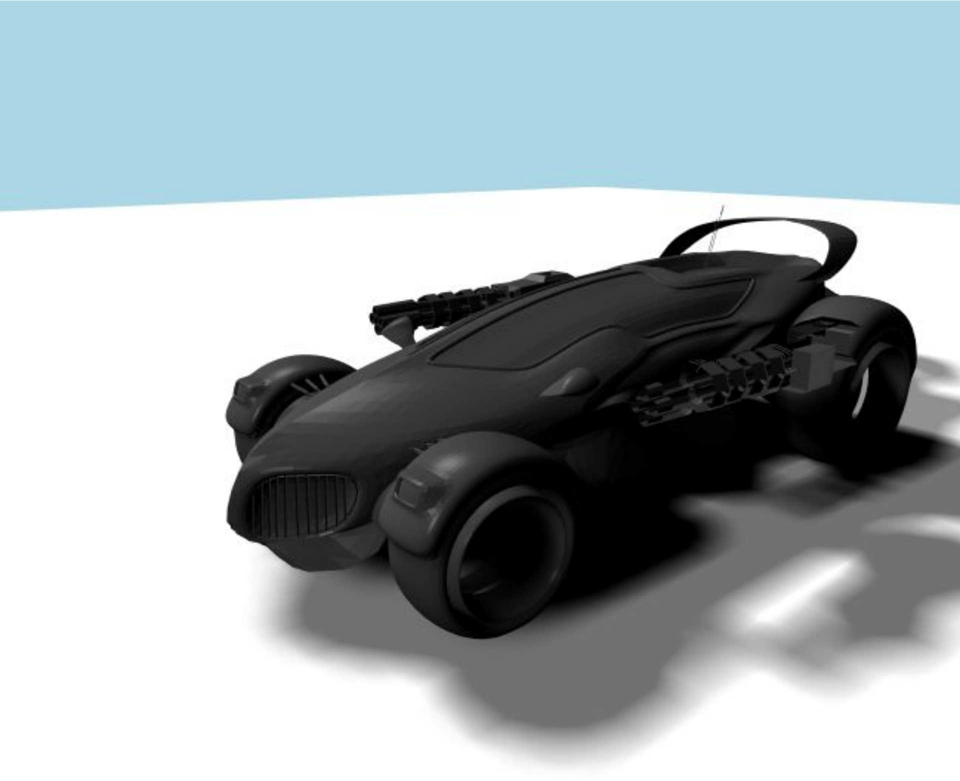
An attempt of separating lighting and shadowing



$$colorResponse = ambient + lightRatio * (diffuse + specular)$$

- Can we can use two separate passes: one for light ratio computation(shadowing), one for lighting(diffuse, specular), then use any filter function to filter the light ratio(shadow)?
- Light ratio texture(32-bits floating point) → crack issue
- Not every decimal has a terminating representation in binary format. (0.25 or 0.5 ☑)   (0.3 ☒ -> be truncated)
- Possible solution: Floating-value represented by two integers can solve.





VSSM ≈ (SAT-VSM+PCSS)

- Novel formula for averaged blocker depth $$z_{Occ} = \frac{z_{Avg} - P_{max} z_{unocc}}{1 - P_{max}}$$

- Subdivision scheme to solve “non-planarity” issue

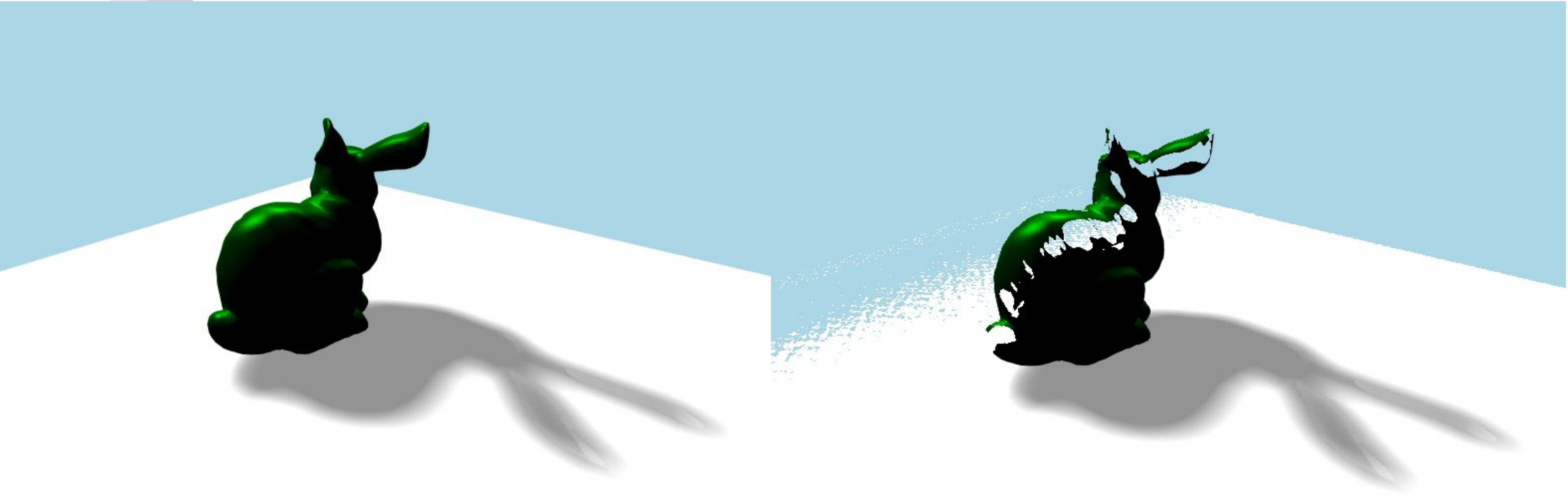
$Z_{Avg} \geq d$ “Non-planarity” condition

when $Z_{Avg} < d$(they call it as planar case)

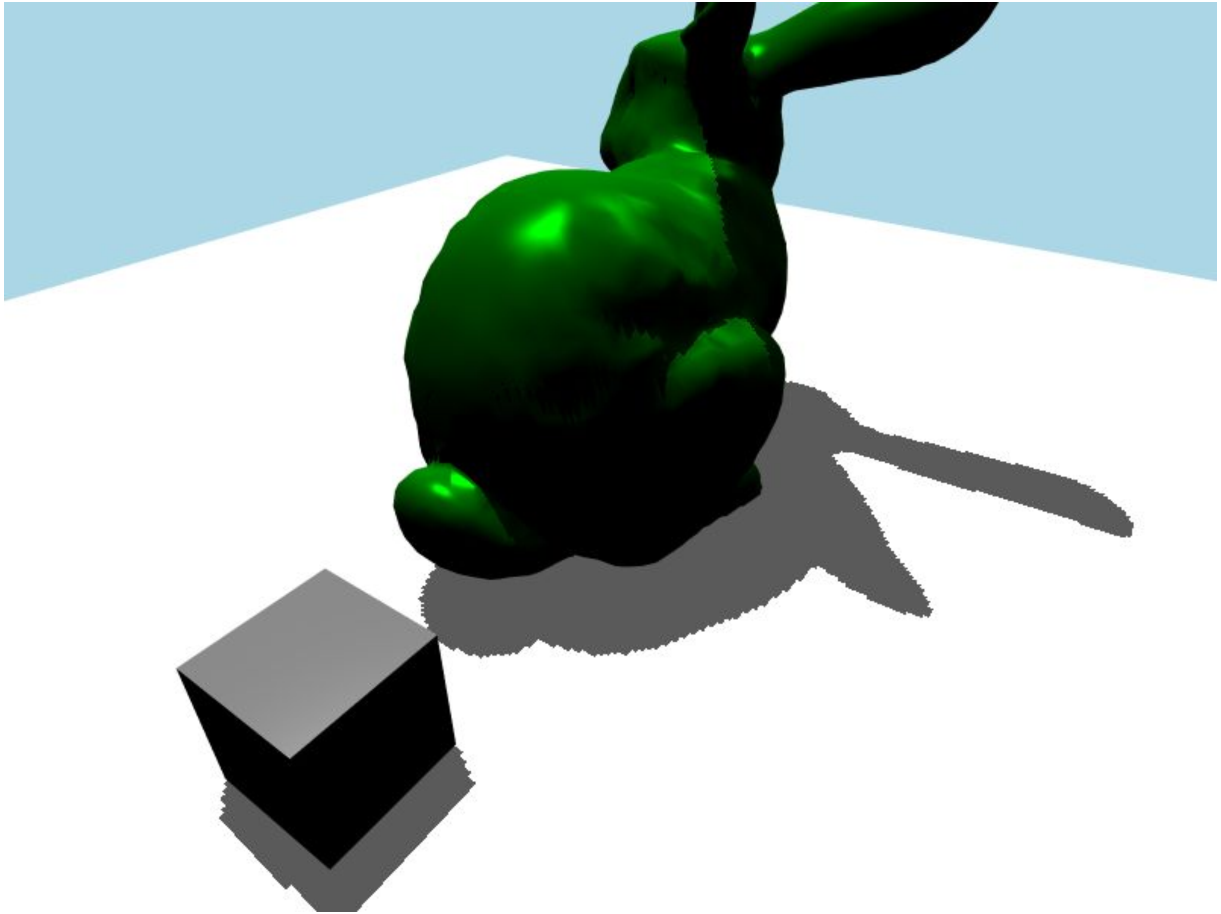
they use $Z_{Occ} = \dfrac{Z_{Avg} - P_{max} Z_{unocc}}{1 - P_{max}}$ and assume $Z_{unocc} = d$

$Z_{Avg} < d$, we can let $Z_{Avg} = P_0 d$, and P_0 must less than 1.

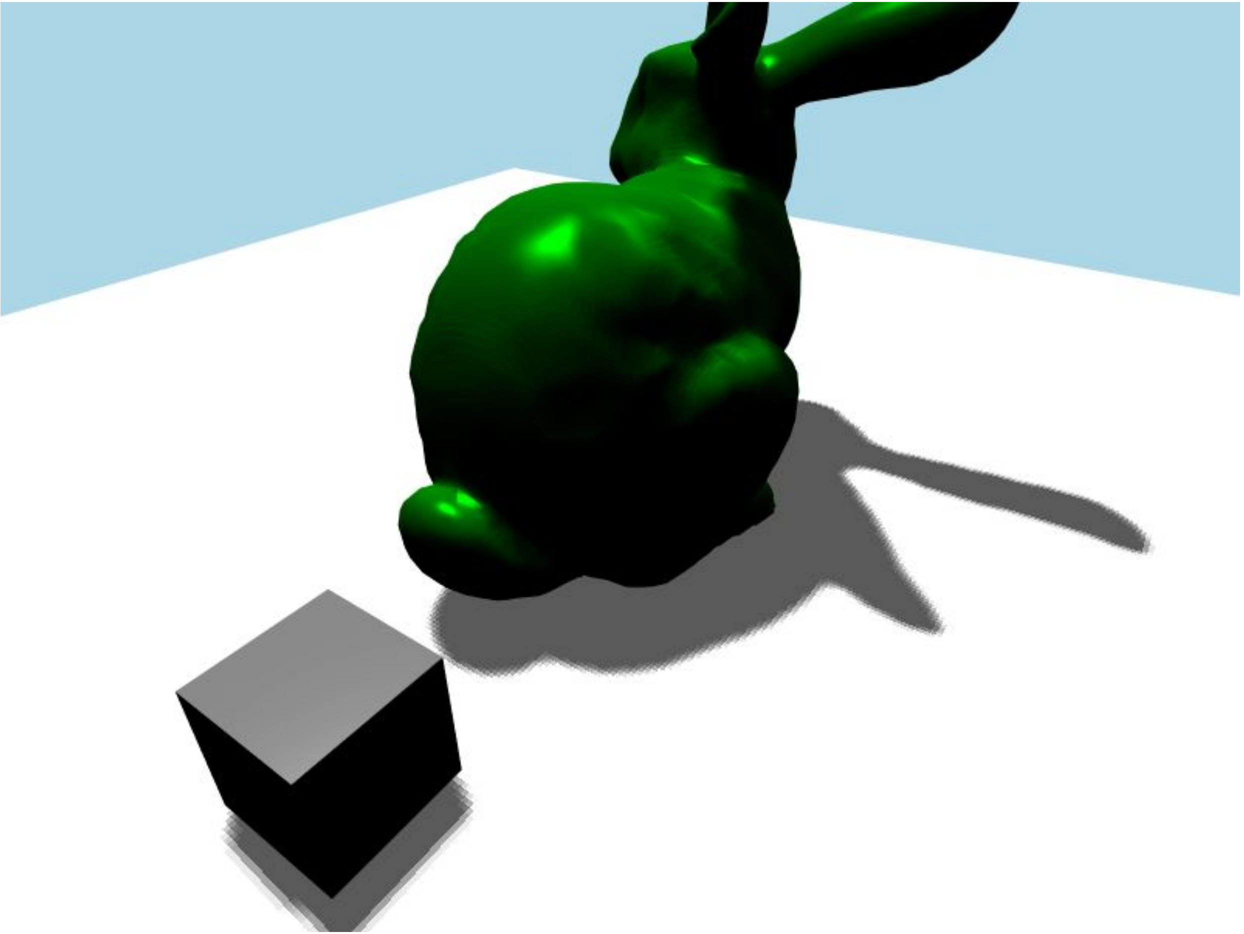
$Z_{Occ} = (P_0 - P_{max})(\dfrac{d}{1 - P_{max}})$ —> Z_{Occ} can be negative!!!



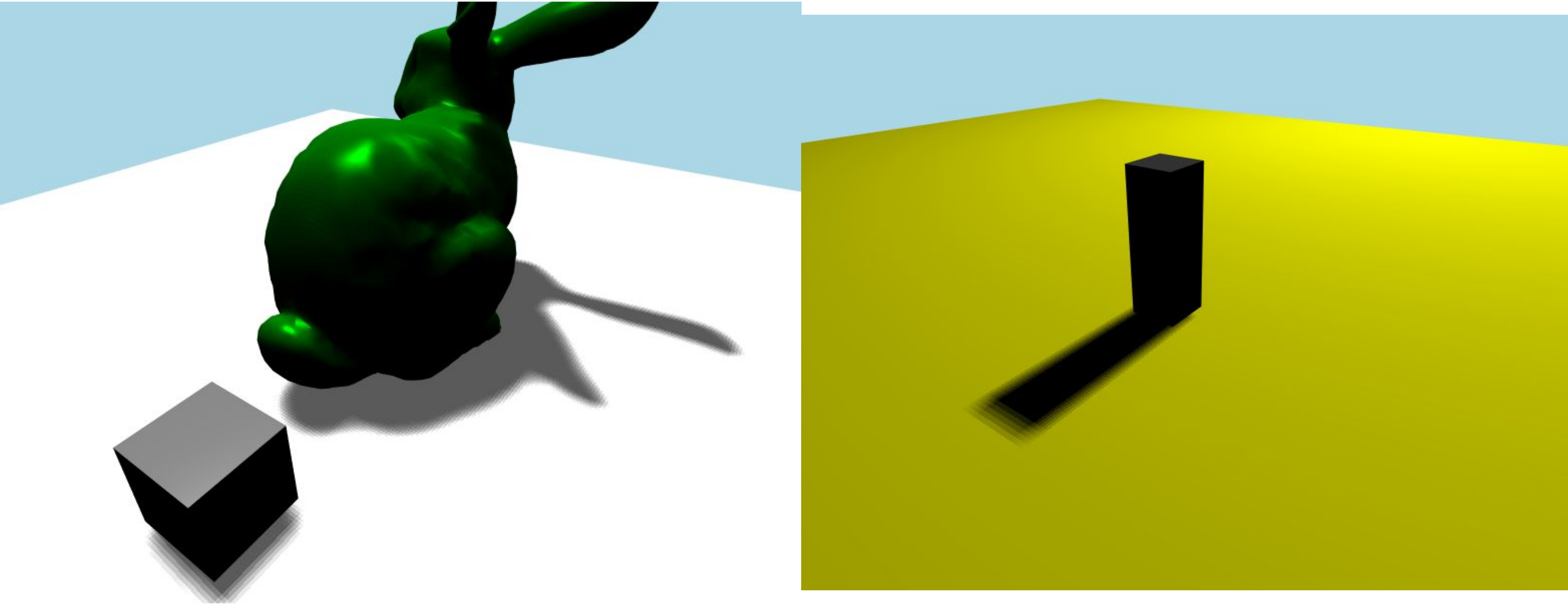
Results



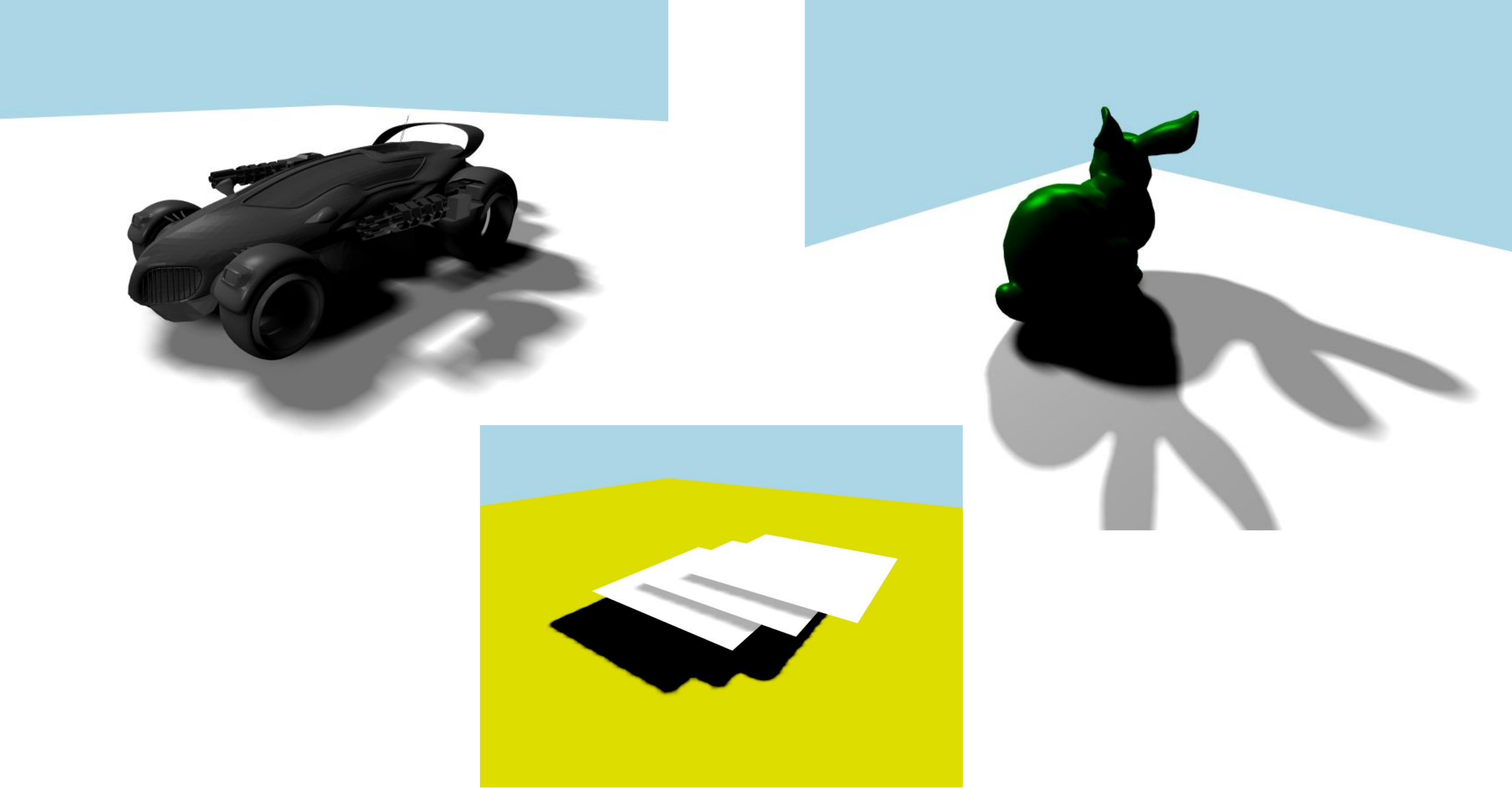
Basic Shadow Maps



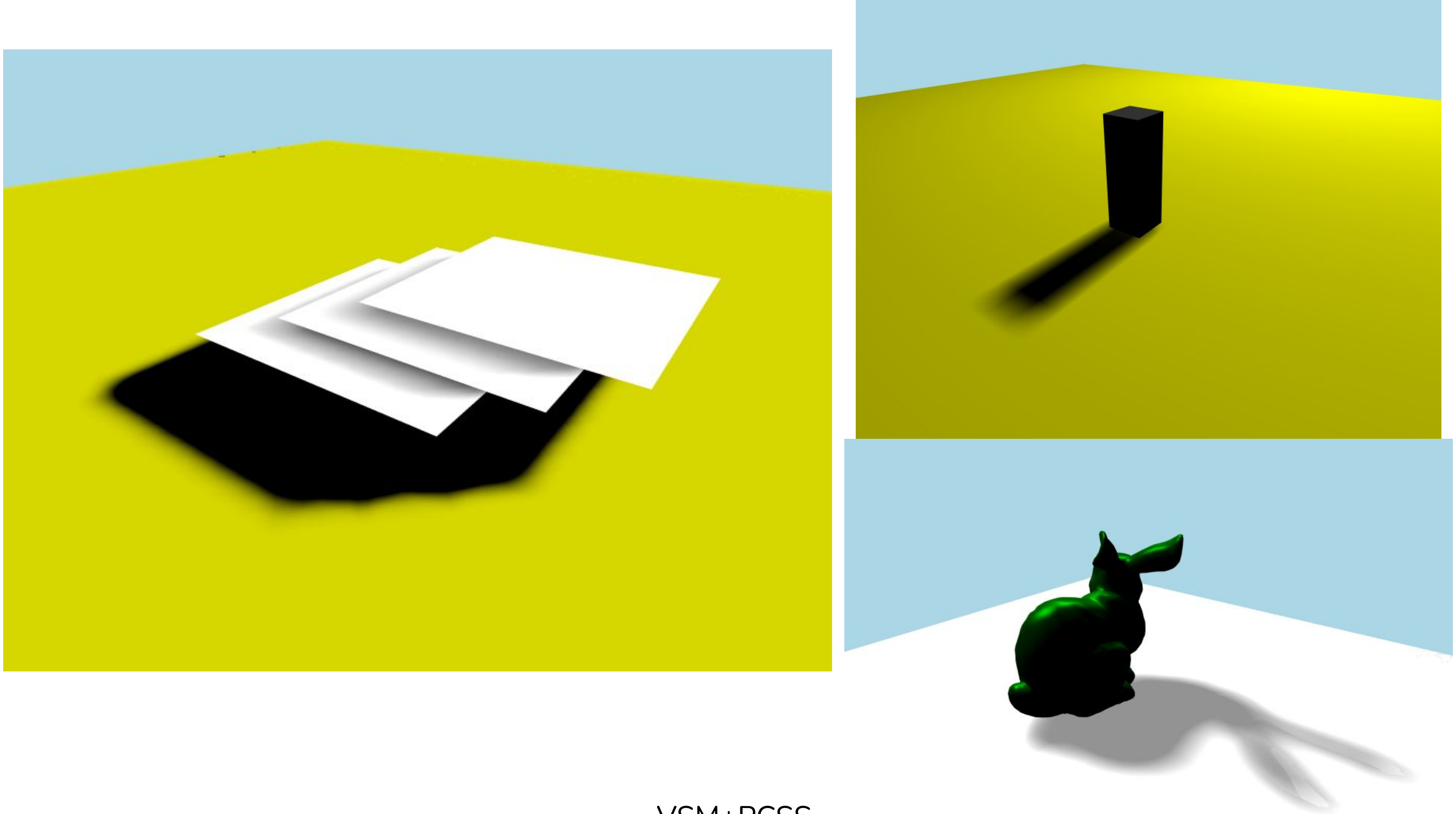
Shadow Maps + PCF



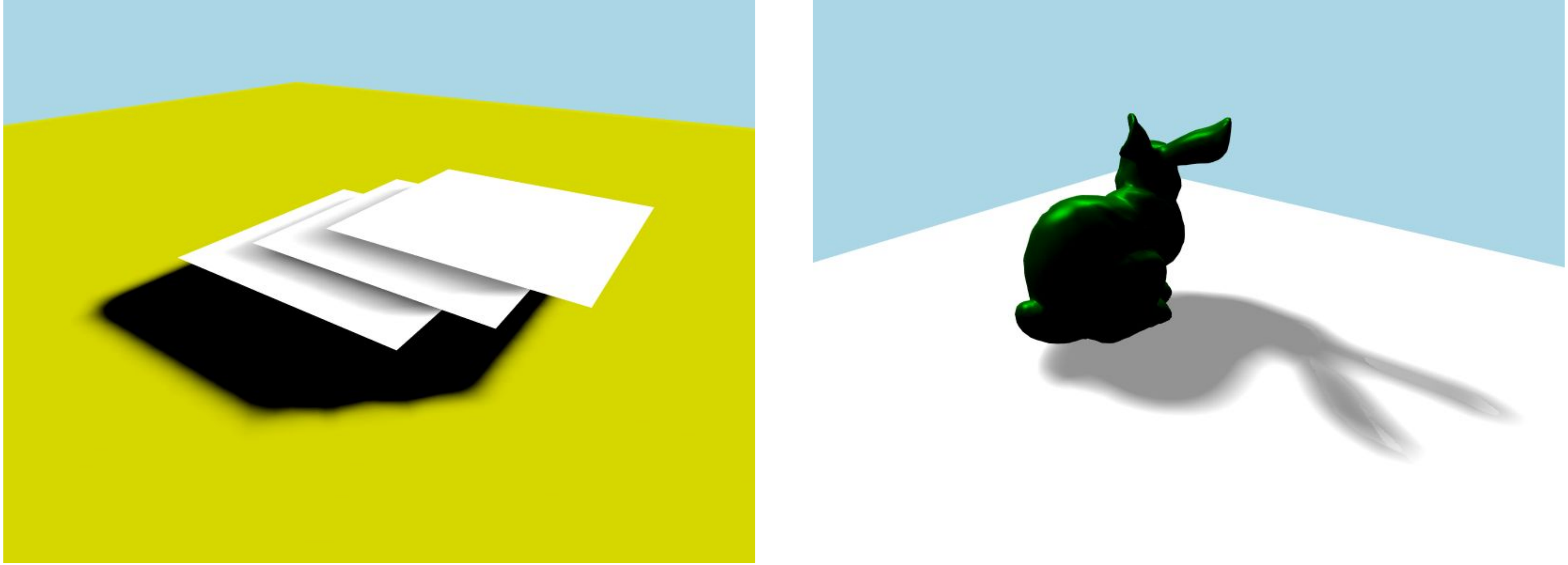
Shadow Maps + PCSS



VSM



VSM+PCSS



VSSM

Thank you