

## Screen-Space Percentage-Closer Soft Shadows

Mahdi Bagher, Jan Kautz, Nicolas Holzschuch, Cyril Soler

► **To cite this version:**

Mahdi Bagher, Jan Kautz, Nicolas Holzschuch, Cyril Soler. Screen-Space Percentage-Closer Soft Shadows. ACM SIGGRAPH 2010 Posters, Jul 2010, Los Angeles, CA, United States. ACM, pp.133, 2010, <10.1145/1836845.1836987>. <inria-00536256>

**HAL Id: inria-00536256**

**<https://hal.inria.fr/inria-00536256>**

Submitted on 15 Nov 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Screen-Space Percentage-Closer Soft Shadows

Mahdi MohammadBagher<sup>†‡§</sup> Jan Kautz<sup>†</sup> Nicolas Holzschuch<sup>‡§</sup> Cyril Soler<sup>‡§</sup>

<sup>†</sup> University College London <sup>‡</sup> INRIA Grenoble - Rhône-Alpes <sup>§</sup> CNRS and Université de Grenoble, Laboratoire Jean Kuntzman

## Abstract

We present an algorithm for computing Percentage-Closer Soft Shadows inside a screen-space rendering loop. Our algorithm is faster than traditional soft shadows based on percentage closer filtering, while providing soft shadows of similar visual quality. It combines naturally with a deferred shading pipeline, making it an ideal choice for video games. This algorithm is not only faster, but allows the use of larger shadow maps without dramatically affecting the rendering speed.

## 1 Method

Our algorithm, SSPCSS performs all the computations in screen space, while PCSS [Fernando 2005] performs them on the shadow map, i.e., in light space. Our algorithm performs the following steps:

- **Preparation:** we render the auxiliary buffers: the scene depth map, the shadow map, the hard shadow map (the set of points in screen space that are in shadow) and the projected shadow map (for each point in shadow in screen space, it stores the corresponding closest blocker distance from the shadow map).
- **Blocker search:** we scan the projected shadow map and, for each pixel in screen space, find the average distance of the objects blocking the light source (non-blocker points should just be ignored).

$$W_{search} = \frac{(d_{point} - z_{near})}{d_{point}} W_{light}$$

- **Penumbra estimation:** we estimate the penumbra size for each pixel, using the average blocker depth computed at the previous step.

$$W_{screen\ penumbra} = \frac{W_{penumbra} \cdot d_{screen}}{d_{eye}}$$

$$, \text{ where } d_{screen} = \frac{1}{2 \tan(\frac{\text{fov}}{2})}, W_{penumbra} = \frac{(d_{receiver} - d_{blocker})}{d_{blocker}} W_{light}.$$

- **Shadow filtering:** we filter the hard shadow map by a variable-size filter, whose diameter is the penumbra size computed in the previous step.

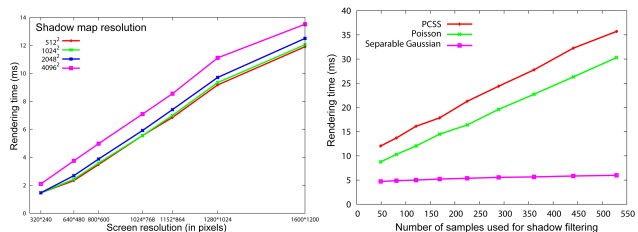
## 2 Screen-Space Edge-Aware Filtering

The filtering performed for blocker search and shadow filtering steps need to be edge-aware, since we lose information about edges in screen space. We have used cross-bilateral filters [Tomasi and Manduchi 1998] for this purpose, but any other edge-aware filter could also be used.

We used a set of pre-computed 1D separable Gaussian weights or a 2D random Poisson disk as the domain filter in the cross-bilateral filter. The range filter is a function of the depth difference in camera space between two points. We used two different bilateral filters:

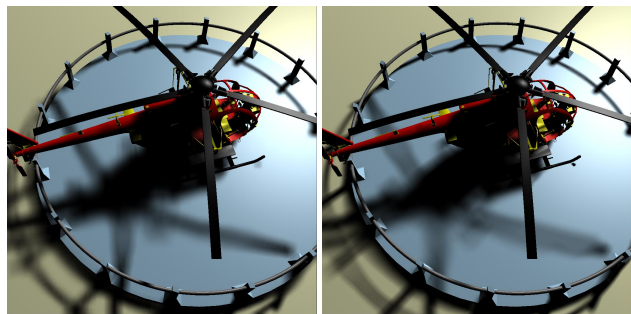
**Separable Cross-Bilateral Gaussian Filter:** a highly efficient, but less accurate filter [Pham and van Vliet 2005].

**Cross-Bilateral Poisson Filter:** an accurate but more computation-intensive filter using a random Poisson disk. It is still faster than



(a) time as a function of screen resolution

(b) SSPCSS compared to PCSS



(c) SSPCSS

(d) PCSS

the PCSS algorithm using the same Poisson disk on the shadow map test results.

Separable Gaussian filtering is much faster than 2D Poisson filtering because a separable Gaussian filter takes  $2(n+1)$  samples, compared to  $n^2$  samples for a 2D Poisson filter.

## 3 Discussion and Results

Our work uses almost the same idea as in [Robison and Shirley 2009], which has focused on image space gathering by ray-tracing, but we have focused on rasterization on GPU to achieve higher frame rates for real-time rendering applications such as games.

The computation time for our Screen-Space PCSS algorithm varies linearly with the total number of pixels on the screen. The computation time is almost independent from the shadow map resolution. The cross-bilateral separable Gaussian filter is much faster than the cross-bilateral Poisson filter. More importantly, its performances are almost independent from the filter radius.

## References

- FERNANDO, R. 2005. Percentage-closer soft shadows. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Sketches*, ACM, New York, NY, USA, 35.
- PHAM, T. Q., AND VAN VLIET, L. J. 2005. Separable bilateral filtering for fast video preprocessing. *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*.
- ROBISON, A., AND SHIRLEY, P. 2009. Image space gathering. In *HPG '09: Proceedings of the Conference on High Performance Graphics 2009*, ACM, New York, NY, USA, 91–98.
- TOMASI, C., AND MANDUCHI, R. 1998. Bilateral filtering for gray and color images. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, IEEE Computer Society, Washington, DC, USA, 839.