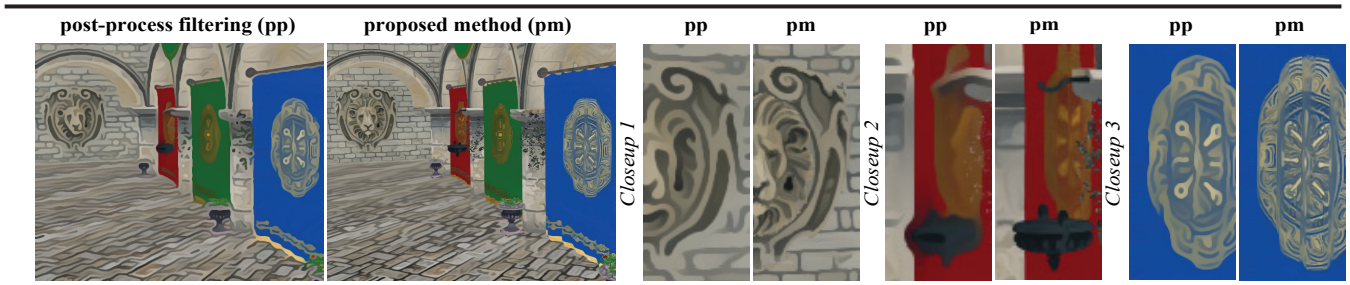
Image Filtering for Interactive Level-of-Abstraction Visualization of 3D Scenes

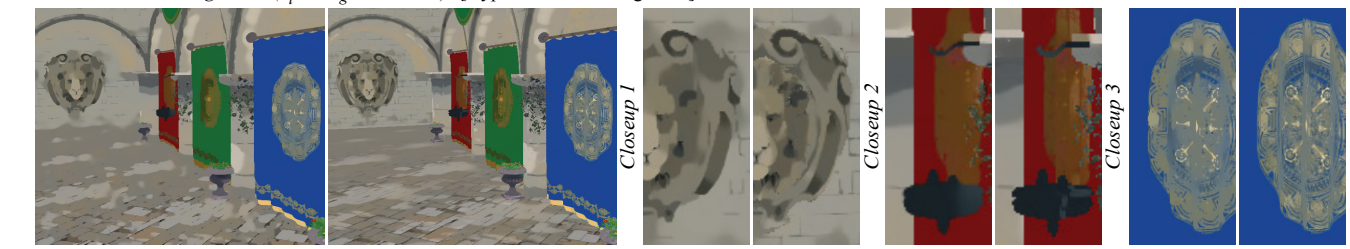
– Supplementary Materials –

Amir Semmo Jürgen Döllner

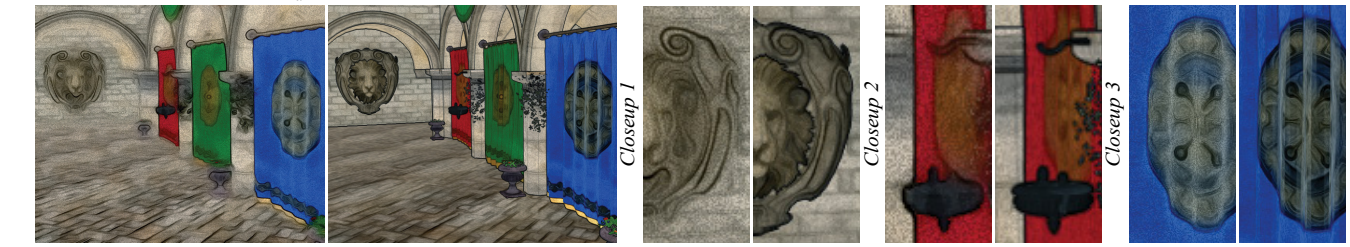
Hasso Plattner Institute, Germany



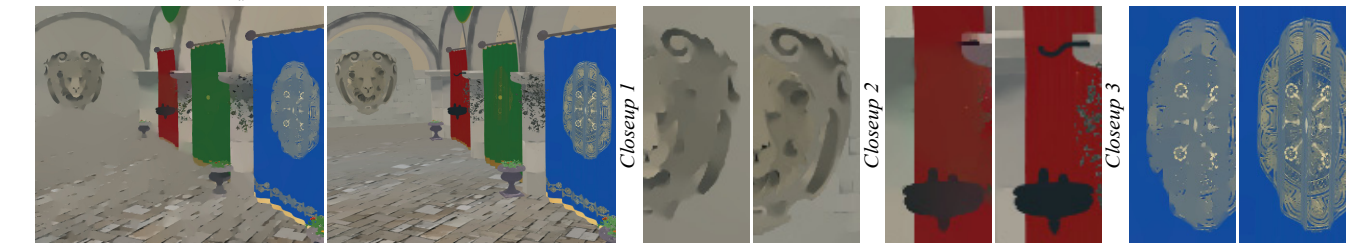
Coherence-Enhancing Filter ($\sigma_i = 0$, $\sigma_g = 1.5$, $N = 5$) [Kyprianidis and Kang 2011]



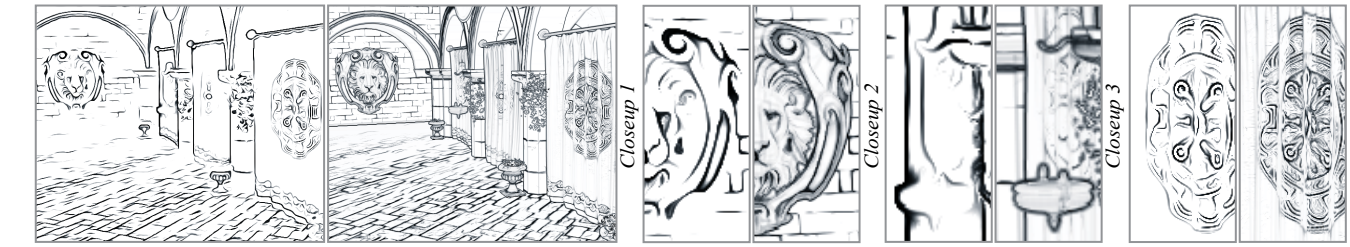
Real-Time Video Abstraction ($\sigma_d = 3$, $\sigma_r = 4.25\%$, $n_e = 1$, $n_a = 4$, $q = 2$, $nbins = 8$) [Winnemöller et al. 2006]



Watercolor Rendering ($\sigma_d = 5$, $\sigma_r = 20\%$, $n_e = 1$, $n_a = 4$, $q = 2$, $nbins = 16$)



L_0 *Smoothing* ($\lambda = 0.04$, $\kappa = 2$) [Xu et al. 2011]



Flow-based Difference of Gaussians ($\tau = 0.995$, $\sigma_m = 1$, $\sigma_e = 6$, $\varphi = 2$) [Kyprianidis and Döllner 2008]

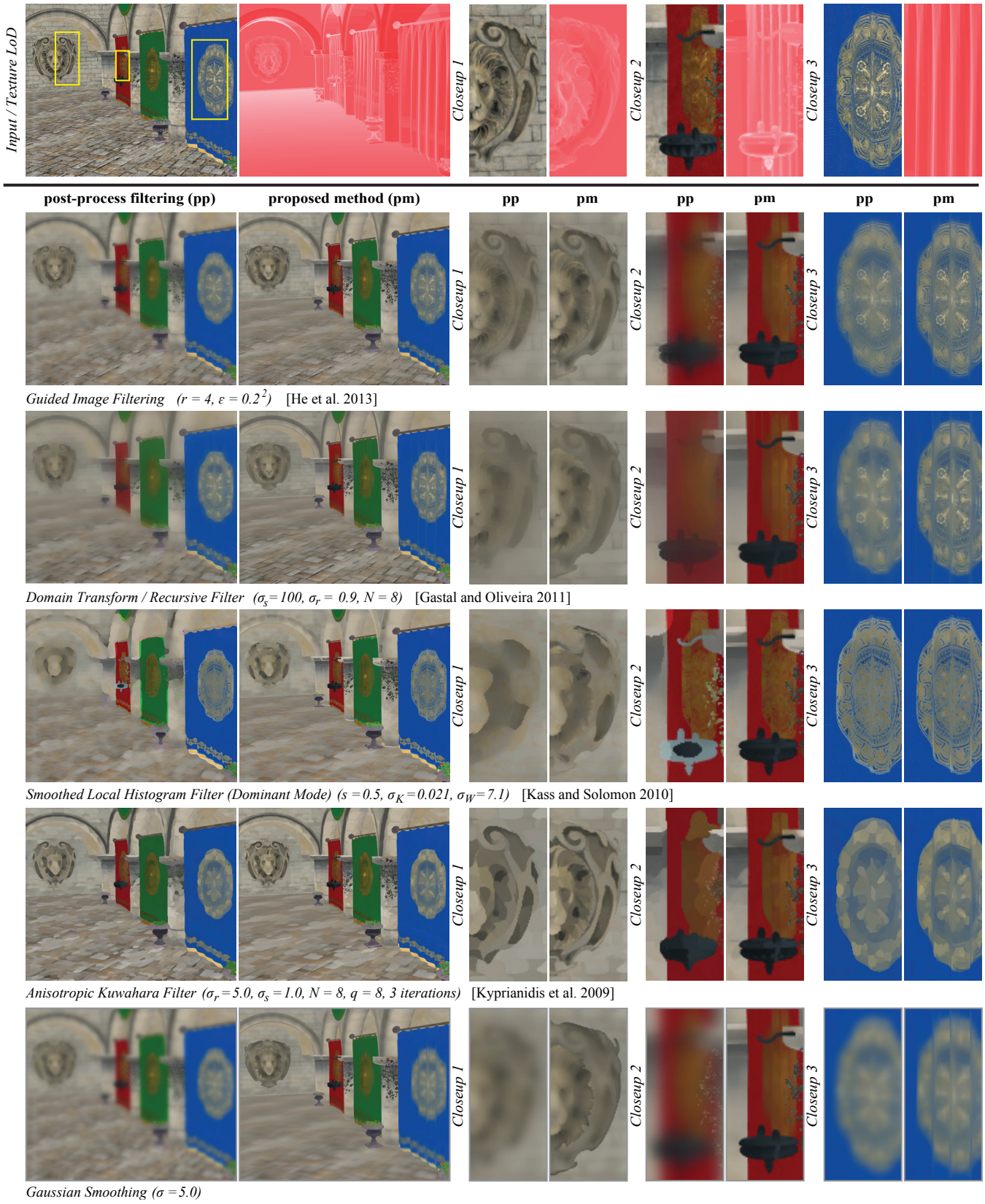


Image filters applied to a 3D scene using our system. The top rows show the original output with texture LoD, and the second columns show our proposed method. Notice how the texture gradients on the floor, the textured banners, and the lion figure in the back are aggressively smoothed when filtered in a post-process stage, while our approach preserves their structures and overall object borders without compromising the filters' qualities. (Sponza Atrium scene © Marko Dabrovic and Frank Meinl from Crytek. All rights reserved.)

Algorithm 1: *Kernel for the per-fragment filtering of texture data*

```
function local_image_filtering: begin
    Input: G-buffer G, texture page table P with color lookup P_C and
           process flag lookup P_F, filtering budget B
    k ← 0                                   /* global number of texels filtered */
    for pixels p ∈ G do in parallel
        (ID, lod, u, v) ← G (p)             /* sample G-buffer */
        if ID = 0 then                      /* early out */
            return
        (T_0, T_1) ← P (ID, ⌊ lod ⌋ and ⌈ lod ⌉)   /* mipmap LoDs */
        forall the (T, u_S, v_S) of textureGather (T_0)
        and        (T, u_S, v_S) of textureGather (T_1) do /* 8 */
            if P (T, u_S, v_S) not marked as processed then
            /* start of critical section */
                if k < B then               /* threshold budget */
                    P_F (T, u_S, v_S) ← mark as processed
                    P_C (T, u_S, v_S) ← filtering (T, u_S, v_S)
                    k ← k + 1               /* filtered color */
                else                        /* progressive filtering */
                    P_C (T, u_S, v_S) ← lookup (ID, lod, u, v)
                end
            /* end of critical section */
        end
    end
end
```

References

GASTAL, E. S. L., AND OLIVEIRA, M. M. 2011. Domain Transform for Edge-Aware Image and Video Processing. *ACM Trans. Graph. 30*, 4, 69:1–69:12.

HE, K., SUN, J., AND TANG, X. 2013. Guided Image Filtering. *IEEE Trans Pattern Anal Mach Intell. 35*, 6, 1397–1409.

KASS, M., AND SOLOMON, J. 2010. Smoothed Local Histogram Filters. *ACM Trans. Graph. 29*, 4, 100:1–100:10.

KYPRIANIDIS, J. E., AND DÖLLNER, J. 2008. Image Abstraction by Structure Adaptive Filtering. In *Proc. EG UK TPCG*, 51–58.

KYPRIANIDIS, J. E., AND KANG, H. 2011. Image and Video Abstraction by Coherence-Enhancing Filtering. *Comput. Graph. Forum 30*, 2, 593–602.

KYPRIANIDIS, J. E., KANG, H., AND DÖLLNER, J. 2009. Image and Video Abstraction by Anisotropic Kuwahara Filtering. *Comput. Graph. Forum 28*, 7, 1955–1963.

WINNEMÖLLER, H., OLSEN, S. C., AND GOOCH, B. 2006. Real-Time Video Abstraction. *ACM Trans. Graph. 25*, 3, 1221–1226.

XU, L., LU, C., XU, Y., AND JIA, J. 2011. Image Smoothing via L_0 Gradient Minimization. *ACM Trans. Graph. 30*, 6, 174:1–174:12.