Supplemental Material
for
Freeform Shadow Boundary Editing

O. Mattausch[1,2] and T. Igarashi[2] and M. Wimmer[3]
[1]University of Zurich, Switzerland
[2]The University of Tokyo, Japan
[3]Vienna University of Technology, Austria

Abstract

In this document we present additional results for the paper of Mattausch et al. [MIW13]. In particular, we compare silhouette computation methods used as input for the shadow editing algorithm, show some additional editing and animation examples, and show the effect of the ε parameter of the modified shadow volumes which takes a silhouette vertex s_o to s'_o.

1 Silhouette Computation

A comparison of silhouettes of the smooth underlying mesh (computed with the algorithm of Hertzmann et al. [HZ00]) with standard discrete silhouettes and the resulting shadow boundaries are shown in Figure 1.

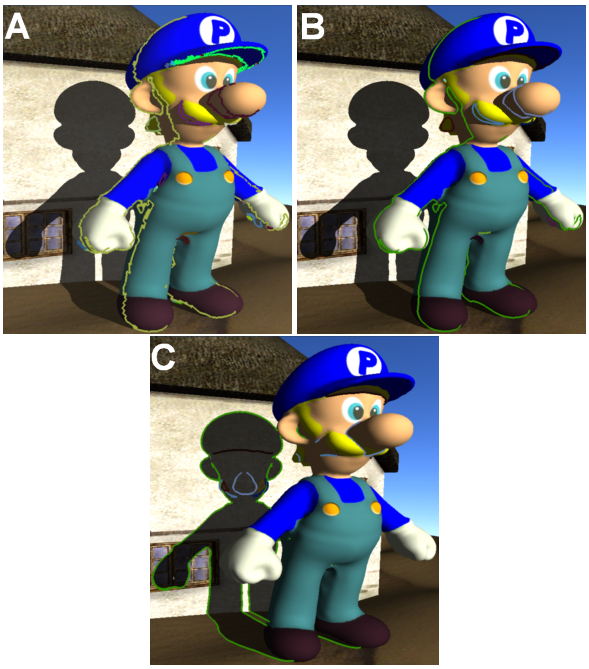

Figure 1: (A) Cluttered silhouettes of the discrete geometry. We use the silhouettes computed from the smooth underlying mesh instead (B), which produce smooth shadow boundary loops (C) and can be edited well. A separate color is used for each silhouette and the same color for the corresponding shadow boundary.

2 More examples

Figure 2 shows how the system behaves in a case with multiple receiver surfaces. Note that the dragged shadow correctly projects on top of the picnic table. In methods that deform the signal on the receiving surface [RTD*10], the edited shadow would be bound to the terrain. Some fun results that can be generated with a few clicks and drags can be seen in Figure 3.

The influence of the ε parameter of the modified shadow volumes can be seen in Figure 4. In this case, we edited the shadow of the nose on the floor (enlarging it and hence also the shadow mesh) and then significantly changed the light configuration. In such a case, setting ε to 0.02 helps to avoid unwanted self shadows due to the shadow mesh.

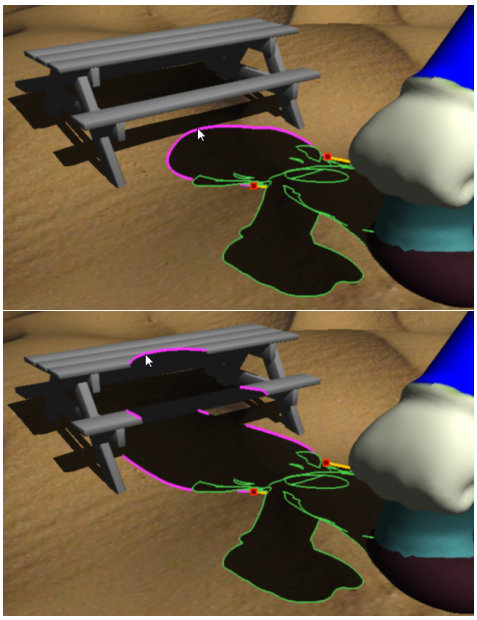

Figure 2: Because we deform the silhouettes of the *shadow caster*, the edited shadow interacts plausibly with the environment. Dragging the shadow will correctly place it on top of the table.

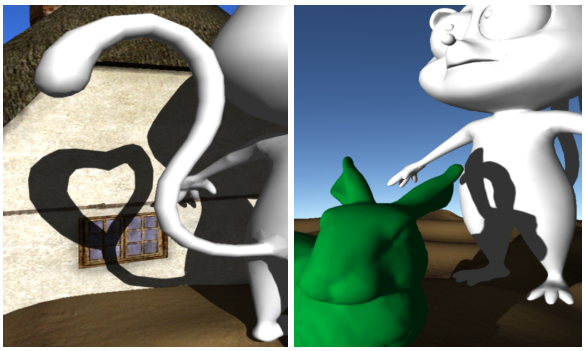

Figure 3: Some fun results.

Animation An animation sequence where only the shadows are moved is shown in Figure 5. The intermediate frames are produced by blending between the start and the end shape.

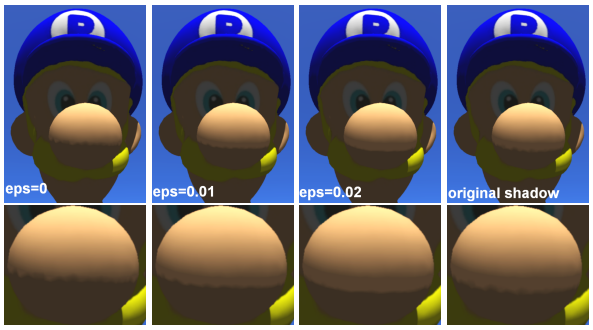

Figure 4: Self shadow (note the difference between the first and the last column) due to the shadow mesh after changing the light configuration. This can be avoided by setting the ε parameter to 1% (respectively 2%) of the total distance between caster and receiver.

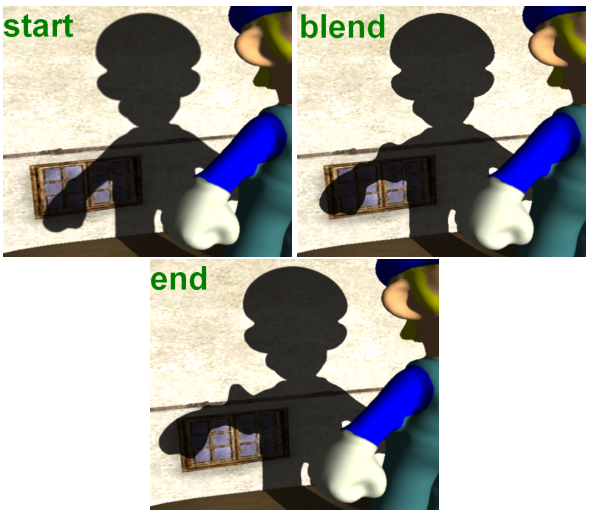

Figure 5: Shadow-only animation (of Paolo flexing). Using the shadow mesh, the middle pose is blended between the initial and the final pose.

References

[HZ00] HERTZMANN A., ZORIN D.: Illustrating smooth surfaces. In *Proceedings SIGGRAPH* (2000), pp. 517–526.

[MIW13] MATTAUSCH O., IGARASHI T., WIMMER M.: Freeform shadow boundary editing. *Computer Graphics Forum 32*, 2 (2013).

[RTD*10] RITSCHEL T., THORMÄHLEN T., DACHSBACHER C., KAUTZ J., SEIDEL H.-P.: Interactive on-surface signal deformation. *ACM Trans. Graph. 29*, 4 (2010), 36:1–36:8.