Interactive Pen-and-Ink Rendering for Implicit Surfaces

Ryan Schmidt*
University of Calgary

Tobias Isenberg*
University of Calgary

Brian Wyvill*
University of Calgary

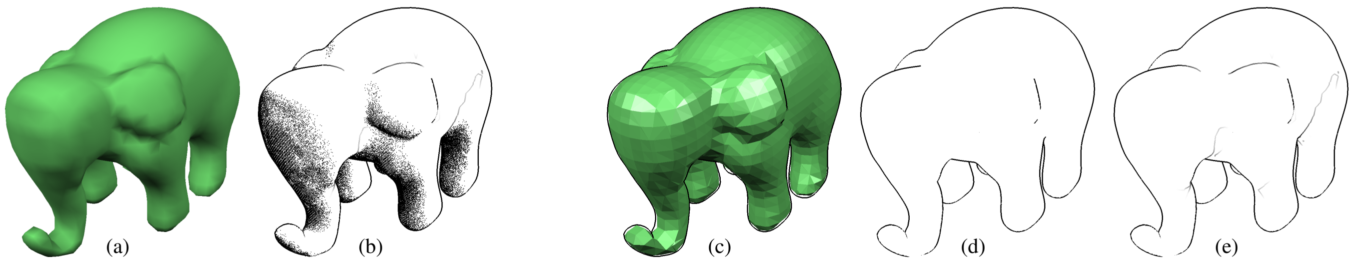


Figure 1: An implicit surface model visualized using a smooth-shaded triangle mesh (a) and a pen-and-ink drawing (b). The generated silhouettes (d) are much smoother than base mesh (c) and render at interactive rates. Suggestive contours (e) add important additional detail.

1 Introduction

The shape of a complex surface can often be conveyed with just "a few good lines". Line rendering is particularly beneficial when visualizing implicit surfaces, as the cost of extracting sufficiently accurate iso-surfaces can be very high. We are exploring the use of low-resolution iso-surface meshes generated in real-time [Schmidt et al. 2005] to interactively render implicit surfaces in a pen-and-ink style. Our approach utilizes silhouettes and feature contours extracted from coarse meshes as an initial approximation to the actual surface contours. These coarse feature lines are then iteratively refined using the implicit functions, producing smooth and highly interactive results.

Rapid visualization is critical in interactive modeling systems, where designers require real-time feedback as they manipulate and deform the implicit surface. Recent pen-and-ink approaches for functional implicit surfaces [Foster et al. 2005] generate high-quality but time-consuming results. The relatively high cost of implicit function evaluation prohibits interactive volume dataset generation, ruling out related methods [Burns et al. 2005]. Our approach, while demonstrated using implicit surfaces, is applicable to any functional smooth surface, including NURBS surfaces. The renderer is used for interactive visualization in a sketch-based modeling system, where it has particular aesthetic value.

2 Dynamic Silhouette Refinement

We begin with a coarse base mesh (Fig. 2(a)). Silhouettes are extracted using standard brute-force sub-polygon methods. Accurate surface normals and curvatures are computed from the implicit function, not the mesh. The silhouette loops are then projected onto the surface (Fig. 2(b)). On implicit surfaces this is done using a few steps of gradient descent, on NURBS patches the surface points can be directly evaluated. Then, the linear silhouette segments are repeatedly subdivided and projected onto the surface until an error threshold is reached (Fig. 2(c)).

Our algorithm gives very good results for regular silhouettes ($N \cdot V = 0$; see Fig. 1). The iterative refinement converges to the correct silhouette under relatively weak conditions on the base mesh and implicit function. We have also experimented with suggestive contours (radial curvature $k_r = 0$), using a finite-difference approach which does not require third derivatives. Mesh resolution appears to have more effect on suggestive contours, although we have not carefully tuned the various available parameters.

Hidden-line removal presents a challenge, as implicit surface ray-intersection is too slow. The coarse mesh can be used, however feature lines will be erroneously clipped in areas of negative curvature. Instead, we again utilize the base mesh. Points are distributed on each triangle and projected to the surface. A small tangent disc is rendered into the z-buffer at each point, providing fast and reasonably accurate hidden-line removal. In addition, these discs can be used to efficiently render stippling points (Fig. 1(b), 2(d)).

The base mesh also provides an efficient means for hierarchically organizing data, simplifying dynamic generation and visibility culling. As a result there is little unnecessary pre-computation overhead, which improves response time during surface deformation. 15–30 fps are achieved for static manipulation of moderately-complex stippled models, and 1–5 fps during interactive editing.

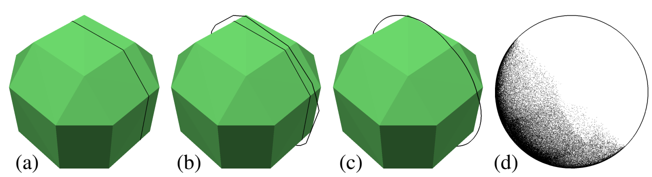


Figure 2: Silhouette projection from the base mesh to the implicit surface, silhouette subdivision, and final result with stippling.

*{rms | isenberg | blob}@cpsc.ucalgary.ca

References

BURNS, M., KLAWE, J., RUSINKIEWICZ, S., FINKELSTEIN, A., AND DECARLO, D. 2005. Line Drawings from Volume Data. *ACM Transactions on Graphics 24*, 3 (July), 512–518.

FOSTER, K., JEPP, P., WYVILL, B., SOUSA, M. C., GALBRAITH, C., AND JORGE, J. A. 2005. Pen-and-Ink for BlobTree Implicit Models. *Computer Graphics Forum 24*, 3 (Sept.), 267–276.

SCHMIDT, R., WYVILL, B., AND GALIN, E. 2005. Interactive Implicit Modeling with Hierarchical Spatial Caching. In *Proc. of Shape Modeling International*, IEEE Inc., 104–113.

Interactive Pen-and-Ink Rendering for Implicit Surfaces
– Additional Figures –

Ryan Schmidt*
University of Calgary

Tobias Isenberg*
University of Calgary

Brian Wyvill*
University of Calgary

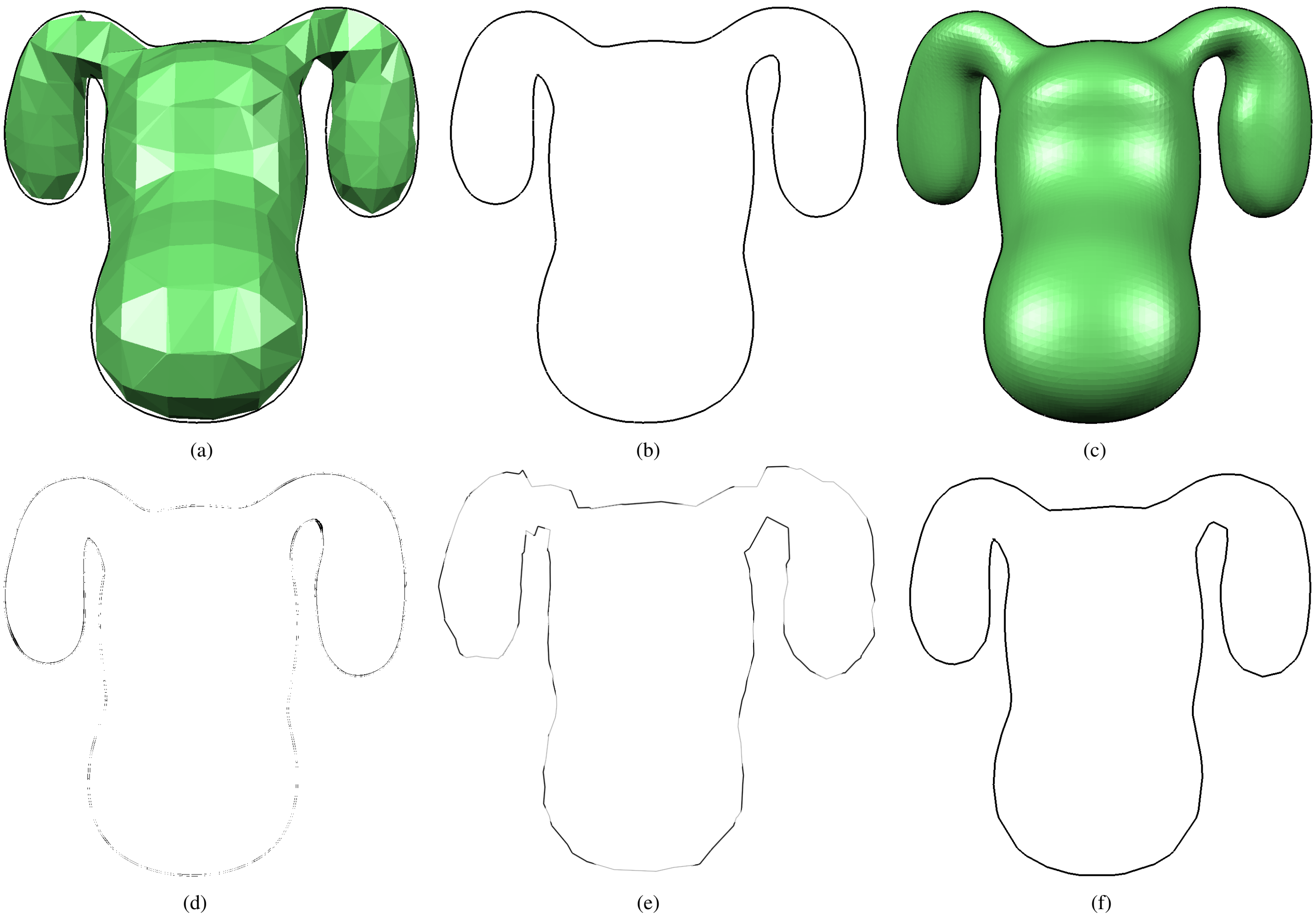


Figure 3: Silhouette example on a low-resolution mesh of a simple implicit surface. A low-resolution base mesh is shown in (a), the smooth silhouette extracted using our method is shown in (b). The silhouette is nearly identical to the high-resolution silhouette in (c), as shown in the difference image (d). For comparison, the low-resolution mesh silhouette is shown in (e), and the low-resolution silhouette after projection but before subdivision is shown in (f). The projection step alone clearly provides a significant improvement in silhouette quality, however subdivision is necessary to impart a real sense of smoothness (b).

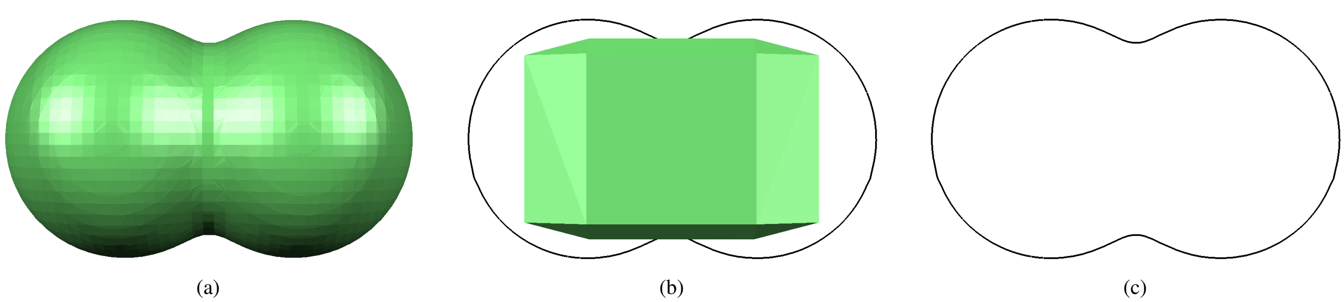
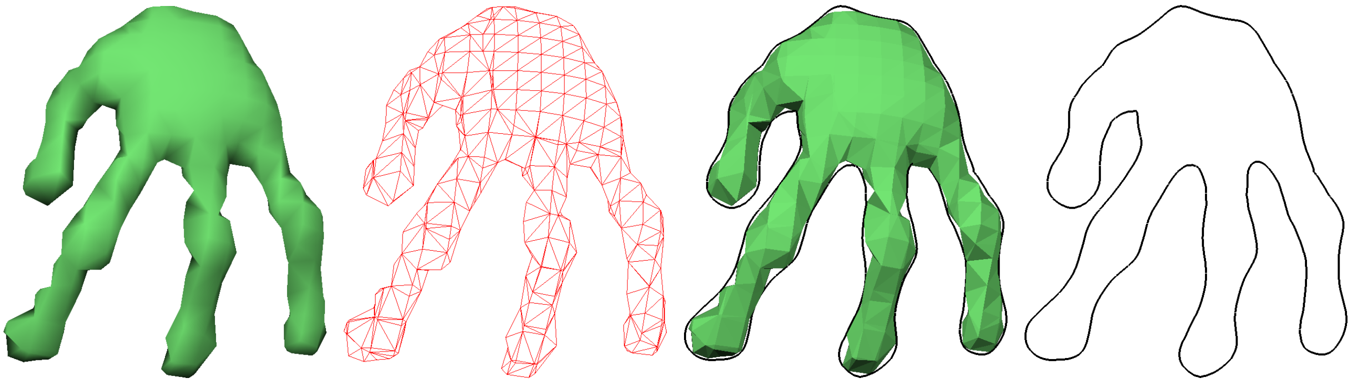


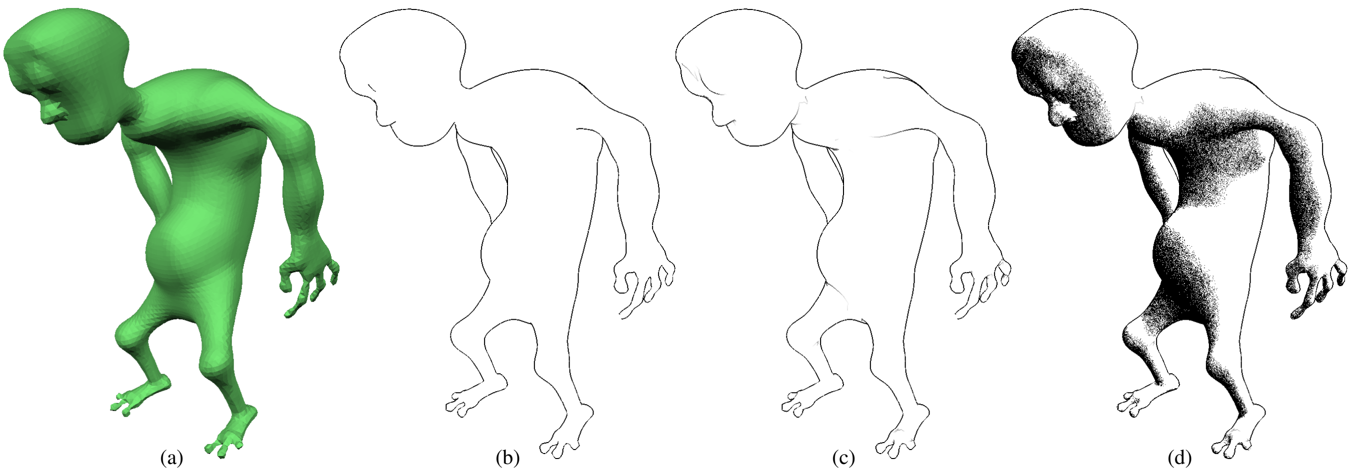
Figure 4: An extreme example of our silhouette refinement algorithm. The base mesh in (b) is a very low-resolution polygonization of the shape in (a). However, the silhouette (c) from this low-resolution base mesh, once projected and refined, is still quite accurate.

*{rms | isenberg | blob}@cpsc.ucalgary.ca



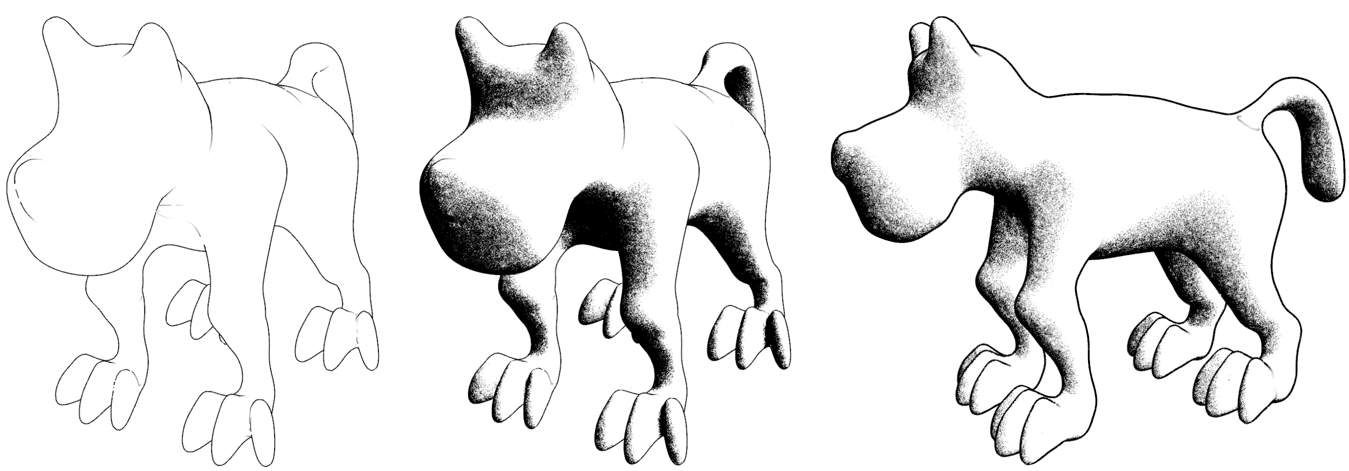
(a) (b) (c) (d)

Figure 5: Implicit surface with relatively fine details: (a) Gouraud-shaded and (b) wireframe images of the implicit surface's low-res base mesh, (c) base mesh with silhouette, and (d) silhouette only.



(a) (b) (c) (d)

Figure 6: Even high-resolution iso-surface polygonization can lead to poor-quality representation of finer details (a). Silhouette refinement (b) improves the representation of small details such as the toes. Suggestive contours (c) and stippling (d) improve perception of surface detail.



(a) (b) (c)

Figure 7: Complex model rendered using our technique with silhouettes and suggestive contours (a) as well as with additional stippling (b). A different view of the model with silhouette lines, suggestive contours, and stippling is shown in (c).