Subdivision for Line Drawings

Bert Freudenberg*
Otto-von-Guericke-Universität Magdeburg

Abstract

This paper presents approaches of incorporating subdivision techniques in the rendering process for computer-generated line drawings. A new method for accelerating silhouette detection is introduced. This "silhouette propagation" method gains its strength from being intertwined with the 3D subdivision process. Another technique reduces the number of 3D subdivision steps necessary for high-quality renditions of line drawings. It is shown how, by using 2D subdivision on curves, the same line quality can be achieved that the computationally much more expensive 3D subdivision yields.

1 Introduction

The graphics community has always been looking for improved representations of geometric models. Recently, subdivision surfaces are becoming the representation of choice in a lot of application areas. One of those areas is non-photorealistic rendering, which also gained attention in recent years. This presentation investigates the use of subdivision techniques for the purpose of generating non-photorealistic images in the style of line-art illustrations.

The minimalist appearance of line drawings is composed of strokes along the main features of the depicted object: its silhouette, and creases on the surface. The silhouette marks the outer boundary of an object as seen from the view point. Crease lines are placed where facets of the surface are joined at an angle, creating a sharp edge. By emphasizing visual discontinuities and leaving out unnecessary detail, the image's comprehensibility is largely improved, which is the reason why a lot of illustrators use a line drawing style.

Subdivision surfaces have several features that make them especially apt for rendering line drawings. For example, this representation explicitly contains smoothness information, that is, whether an edge in the model is a crease or not. Special rules are used for subdividing these edges. For another example, subdivision surfaces support multi-resolution methods by design. Operations can be carried out at coarse levels and propagated to finer levels of detail.

Amazingly, these features do not seem to have got exploited in non-photorealistic rendering to date. Either subdivision surfaces are treated as polygonal models after performing a few subdivision steps [MKT+97, KGC00], or they are only used as a generic smooth surface representation [HZ00].

*Institut für Simulation und Graphik, 39016 Magdeburg, Germany

This work utilizes subdivision as a tool for rendering line drawings in two ways. Firstly, subdivision *surfaces* are used as the geometric model representation because of the reasons mentioned above. An algorithm for silhouette detection using this representation is presented in Section 3. Secondly, subdivision *curves* are employed for drawing smooth shapes in coarser subdivision levels. This technique is described in detail in Section 4. Some results are presented in Section 5, while Section 6 gives some concluding remarks and shows possibilities for future work.

2 Subdivision

In this work the Modified Butterfly subdivision scheme is used [ZSS96]. It was chosen mainly because it is an interpolating scheme. Although approximating schemes in general achieve a better surface quality, this is irrelevant in the context of non-photorealistic rendering. Much more important is that the coarser subdivision levels—indeed, even the control mesh—roughly resemble the limit surface, which is approached after few subdivision steps (see Figure 1).

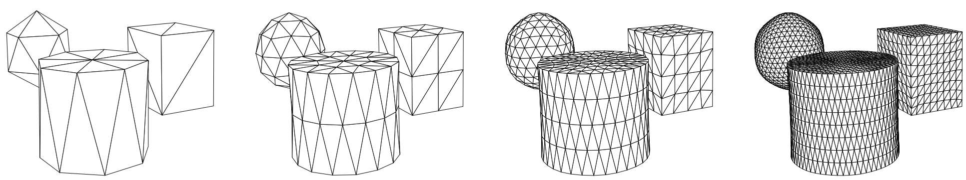

Figure 1: Subdivision of some simple objects. The initial mesh is shown on the left, then three steps of subdivision were performed.

The rule used for subdividing crease edges is different from the normal edge subdivision rule. The surface subdivision only uses vertices on one side of a crease, so sharp edges are preserved in each refinement step. That means that to make a line drawing, the sharp edges "only" need to be drawn along with the silhouettes (Figure 2).

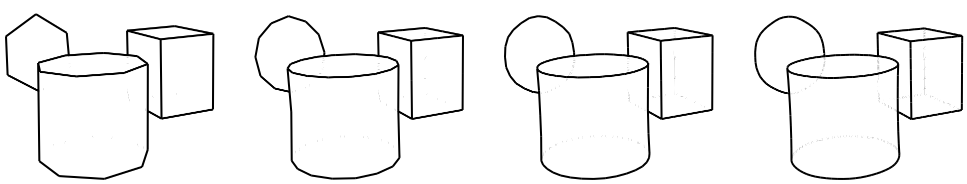

Figure 2: Silhouettes and creases after 0, 1, 2, and 3 subdivision steps.

3 Silhouette propagation

In a fully refined subdivision mesh, there is a huge number of triangles, bordered by an even more impressive number of edges. Silhouette determination is used to select the edges at the perimeter of an object, as seen from the viewer. Only a small portion of all edges are silhouettes, the majority is contributed by inner edges. A number of approaches were taken to rapidly determine silhouette edges, including recent developments like probabilistic selection [MKT+97] or duality maps [HZ00].

The approaches currently used in the field of non-photorealistic rendering operate on the fully refined mesh. But an interesting property of subdivision is that a silhouette edge in subdivision level *l+1* must be the result of subdividing a triangle containing a silhouette edge in level *l*. Expressed the other way around, to determine all silhouette edges in level *l+1* it is sufficient to consider edges in those triangles that were adjacent to a silhouette in level *l*.

This insight leads to a new method for fast silhouette determination that is intertwined with the subdivision process. The idea is to find silhouette edges in one subdivision level, and propagate this information to the next level. That way, most edges in the finer subdivision levels do not have to be processed at all (see Figure 3). Additionally, only smooth silhouettes need to be tested, since sharp edges are drawn anyway.

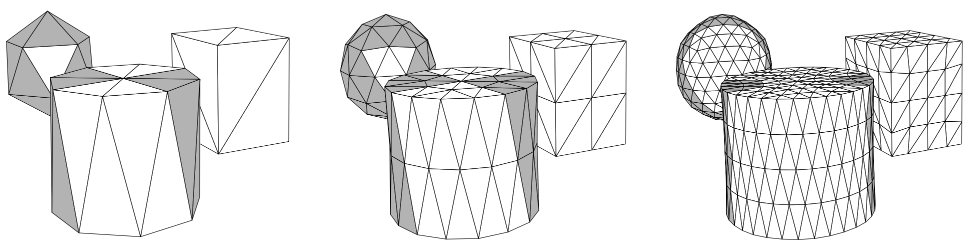

Figure 3: Silhouette propagation example: Triangles adjacent to the silhouette in one level (left) are subdivided and become the silhouette band for the next subdivision level (middle). The propagation process is repeated (right).

In the set-up phase, the silhouette edges in the controlling mesh are gathered. Here, any deterministic algorithm can be used. In the most straight-forward case, the orientation for all faces is determined and the edges sharing both a front-facing and a back-facing triangle are marked as silhouette edges, as illustrated in Figure 4 (left).

Then, in the subdivision step, each edge is subdivided, creating a new vertex for each edge. There is a narrow band of triangles along the previous-level silhouettes that contains the edges of the new silhouette. This band consists of the triangles that emerge from subdividing the triangles that shared at least one vertex with the silhouette (see Figure 4, middle). Triangles outside this band do not need to be considered, because these are known to have the same orientation as the triangles adjacent to them in the band.

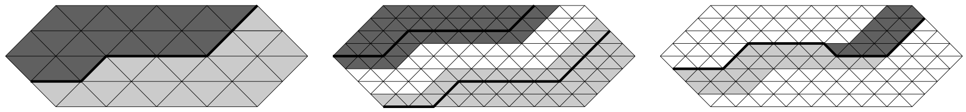

Figure 4: Silhouette propagation scheme. Before subdivision (left), silhouette band (center), refined silhouette (right). Different shades of grey depict front/back faces.

To determine the silhouette for this subdivision level, the triangles in the band are categorized as front-facing or back-facing. Then all edges sharing a front facing and a back facing triangle are finally marked as silhouettes (Figure 4, right). The whole process is repeated until the desired level of subdivision is reached.

4 Drawing by subdivision

While it is certainly possible to keep on subdividing until there are no visual artifacts anymore, this creates an excessive number of triangles. Usually, to overcome this problem, adaptive subdivision techniques are introduced. These adaptive schemes will generate more detail in visually important regions, for example, in areas of higher curvature.

Considering line drawings, "visually most important" are the creases and silhouettes. So, an adaptive scheme can be applied that subdivides areas containing creases or silhouettes with higher resolution than other regions of the mesh. However, why should we spend so much effort on subdividing *faces* when all we want to draw in the end are the *edges*? Instead, we can use *curve subdivision* to refine creases and silhouettes without the need to subdivide surfaces any more than necessary.

For creases, this approach is obviously justified. Crease edges are a special-case in the subdivision process anyway. They get subdivided using the 4-point curve-subdivision scheme [DLG87]. For silhouettes, however, we have to ensure the surface is subdivided with sufficient precision, so that the deviation of the actual silhouette from the curve subdivision silhouette is neglectable.

This is illustrated in Figure 5. The silhouettes and creases are drawn with one level of subdivision. The actual mesh geometry is shown for comparison. On the left, no 3D subdivision was performed. Here, the rim of the cylinder deviates noticeably from the object's silhouette. After performing one (middle) or two (right) surface subdivision steps, however, the interpolated curve is sufficiently close to the actual silhouette.

The curve subdivision process is as follows: First, chains of crease edges are assembled. For this, an arbitrary crease edge is chosen, and edges sharing one vertex of the edge are connected, until a corner is found or the chain is closed. A vertex is a corner if at least three sharp edges meet [HDD$^+$94]. Then the same chaining happens for smooth silhouette edges, stopping at corners, and, additionally, at crease vertices (vertices on a crease). Sharp

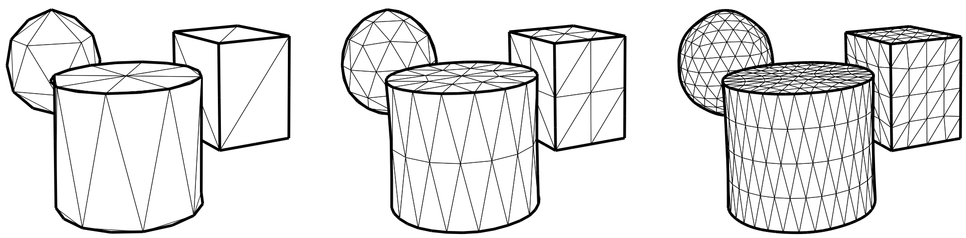

Figure 5: Drawing by subdivision: Curves were subdivided once.

silhouettes could be handled as well if a distinguished drawing style for silhouettes was desired.

Then the actual curve subdivision is performed on the chains. The 4-point subdivision scheme is used (see Figure 6). For closed chains, the regular stencil can be applied everywhere. For open chains, the first and last vertex are weighted with 1/2.

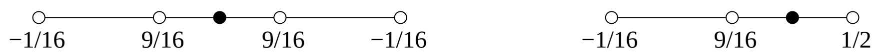

Figure 6: Stencil of 4-point curve subdivision scheme: Regular (left) and end of chain (right).

Note that we can drop the depth component of the curve points altogether and use 2D curve subdivision as a drawing primitive. The curve-subdivision only considers the vertex positions of the curve itself and thus is completely unrelated to the surface tessellation. So except for perspective foreshortening, the interpolation of projected 2D vertices produces the same drawings as the projection of the three-dimensionally subdivided curve.

One level of curve subdivision is of course not quite sufficient. But with three levels, the curves become smooth, as shown in Figure 7. In fact, they are even smoother now than in the last image of Figure 2, which has one more level of 3D surface subdivision. This is only noticeable in magnification, though.

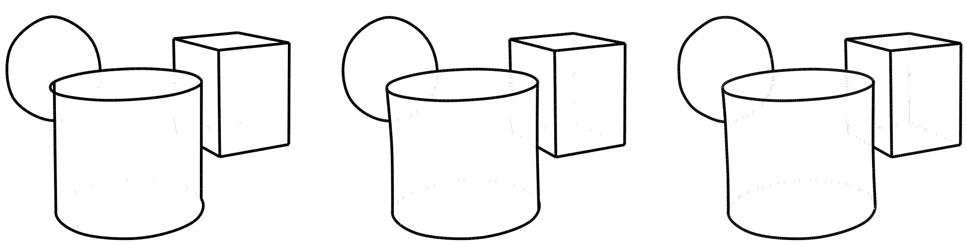

Figure 7: Drawing by subdivision: Curves were subdivided three times.

5 Results

The presented methods were experimentally implemented using the Squeak Smalltalk environment [Squ01]. Being an purely interpreted system, no absolute timings comparable with, for example, [MKT^{+}97] can be given. However, using curve subdivision refinement rather than a full surface subdivision step gave a relative speed-up by the factor 10.

A problem with using the Modified Butterfly subdivision scheme is that it is not widely supported yet in modeling programs. Our approach was to model the polygon mesh while imagining the resulting surface, and then to evaluate the result of subdivision in our test application. A model created this way is shown in Figures 8 and 9.

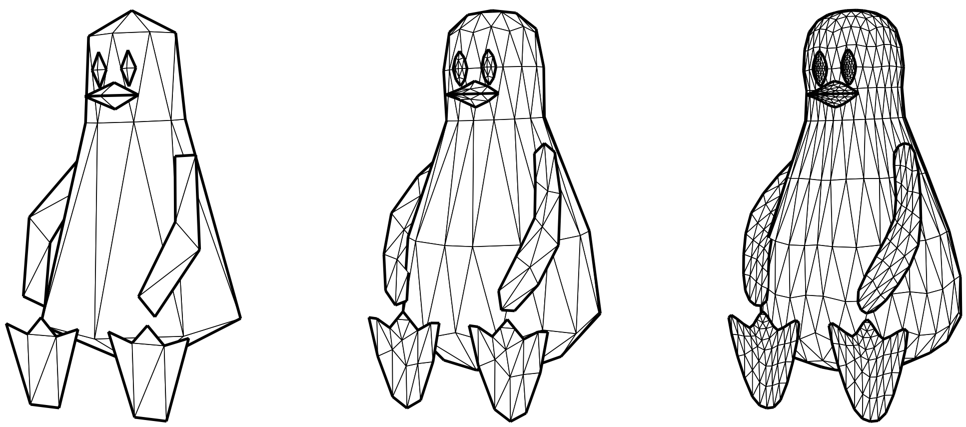

Figure 8: Sample object without curve subdivision in three refinement levels.

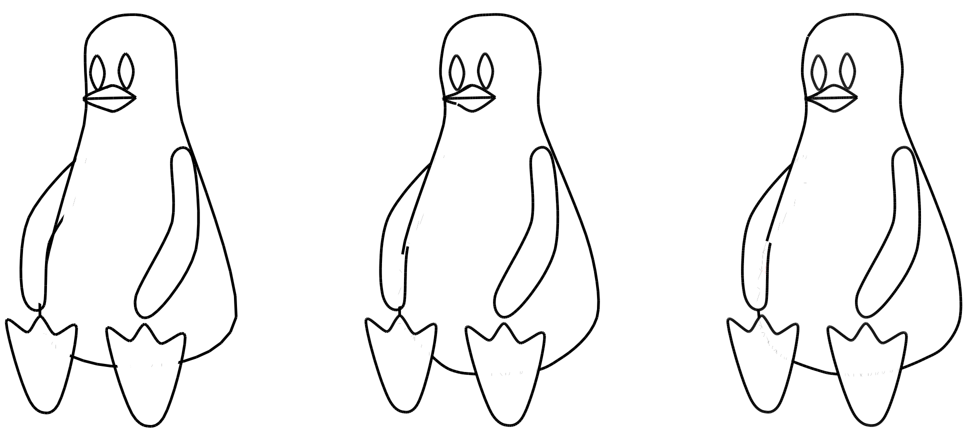

Figure 9: Sample object with curve subdivision, same three refinement levels.

6 Conclusion

In this work, the use of subdivision techniques for non-photorealistic rendering was explored. In particular, novel approaches for rendering line drawings that explicitly exploit subdivision techniques, both for surfaces and curves, and both in two and three dimensions, were introduced.

The silhouette propagation technique is best employed in one-frame (off-line) rendering, or for dynamic subdivision. In an interactive environment with a static control mesh the subdivision itself would be not be performed for each frame. Still, silhouette propagation could be applied if the subdivision levels are retained.

An interesting extension to this method would be to integrate it with hidden line removal. This "visibility propagation" would determine visibility for the control mesh and only update visibility in silhouette regions when subdividing.

Considering that the subdivision of curves is much cheaper than subdividing surfaces in 3D, we now have a tool that enables a user to make a speed vs. quality tradeoff. Even in the roughest surface subdivision levels, the smoothing of creases and silhouettes by curve subdivision yields results undisturbed by polygonal outlines.

The quality of the drawing could be improved by using a curve subdivision scheme that is sensitive to the chordal length of each segment. This would suppress the over-shooting of the interpolating curve when long and short segments are chained.

Hopefully, this work is encouraging for others to investigate subdivision techniques in non-photorealistic rendering and not to treat subdivision meshes as just another surface representation.

References

[DLG87] N. Dyn, D. Levin, and J. Gregory. A 4-point interpolatory subdivision scheme for curve design. *Computer Aided Geometric Design*, 4(4):257–268, 1987.

[HDD$^+$94] Hugues Hoppe, Tony DeRose, Tom Duchamp, Mark Halstead, Hubert Jin, John McDonald, Jean Schweitzer, and Werner Stuetzle. Piecewise smooth surface reconstruction. *Proceedings of SIGGRAPH 94*, pages 295–302, July 1994.

[HZ00] Aaron Hertzmann and Denis Zorin. Illustrating smooth surfaces. *Proceedings of SIGGRAPH 2000*, pages 517–526, July 2000.

[KGC00] Matthew Kaplan, Bruce Gooch, and Elaine Cohen. Interactive artistic rendering. *NPAR 2000: First International Symposium on Non Photorealistic Animation and Rendering*, pages 67–74, June 2000.

[MKT+97] Lee Markosian, Michael A. Kowalski, Samuel J. Trychin, Lubomir D. Bourdev, Daniel Goldstein, and John F. Hughes. Real-time nonphotorealistic rendering. *Proceedings of SIGGRAPH 97*, pages 415–420, August 1997.

[Squ01] Squeak 3.0. `http://squeak.org`, 2001.

[ZSS96] Denis Zorin, Peter Schröder, and Wim Sweldens. Interpolating subdivision for meshes with arbitrary topology. *Proceedings of SIGGRAPH 96*, pages 189–192, August 1996.