Line Art Rendering via a Coverage of Isoparametric Curves

Gershon Elber
Department of Computer Science
Technion, Israel Institute of Technology
Haifa 32000
Israel
Email: gershon@cs.technion.ac.il

Abstract

A line-art non-photorealistic rendering scheme of scenes composed of freeform surfaces is presented. A freeform surface coverage is constructed using a set of isoparametric curves. The density of the isoparametric curves is set to be a function of the illumination of the surface determined using a simple shading model, or of regions of special importance such as silhouettes. The outcome is one way at achieving an aesthetic and attractive line-art rendering that employs isoparametric curve based drawings that is suitable for printing publication.

Key Words: Sketches, Illustrations, Line Drawings, Freeform Surfaces, NURBs, Gridless Halftoning, Printing

1 Introduction

Computer graphics has been extraordinarily successful at producing photorealistic synthetic images. The ray tracing and Radiosity rendering techniques are typically employed to produce the highest possible image quality. Significant amount of information is frequently devoted for each individual pixel in a photorealistic image rendering. It is common to find many bits of information assigned for each pixel of the high quality produced image. In contrast, the typical printing process allows each individual pixel to assume two values only, corresponding to black or white. An a posteriori halftoning process is frequently employed to transform the image to the color space of the production or printing quality. Local evaluation of the introduced error due to color quantization and halftoning, with or without error diffusion [Foley 90], is a valid and common solution. Hence the posteriori halftoning process cannot be globally optimal.

In [Winkenbach 94], it is suggested that the two dimensional rendering process should exploit the three dimensional geometry to create a more appropriate non-photorealistic image. Add to the three dimensional geometry the a-priori knowledge on the target display and one is provided with important cues into the making of a better rendering of the scene. It is typical in art work to introduce features into the art-piece in order to convey more information from the three dimensional model than just the shading. Felix Klein explored the aesthetic use of parabolic curves and had all the parabolic curves of Apollo Belvidere's statue computed and drawn (See [Hilbert 90], pp 197, Figure 204), with unsatisfactory results. Consider a scene

with a hand and a shoulder that both are equally illuminated. Traditional halftoning process will paint both surfaces indistinguishbly. In contrast, in the art of wood carving, one can frequently find the use of the general stream lines of the object (See [Hasluck 77, Ivins 87, Ivins 92]). The objects are rendered with curves that are drawn along the general shape and features of the object, along the hands in our recent example. It is plausible that one should pursue and exploit similar techniques in synthetic images generated by computers with the expectancy of more pleasing results. In [Pnueli 94], a system has been developed to compute halftoned images employing some of these techniques from two dimensional image information. The significance of the work in [Pnueli 94] is drawn from the introduction of gridless halftoning that employs curves as the drawing primitives instead of dots or points. The work of [Pnueli 94] is heavily motivated by the art work of Dürer [Ivins 87, Ivins 92] that exploited curves to portray the shading. The results of [Pnueli 94] served as a motivation for the work presented herein. In several locations, for example [Velho 91], space filling curves are used in the image plane to produce the digital halftoning. The classical space filling curves of Peano, Hilbert, and Sierpinky are considered. This halftoning process can also be classified as traditional because its output continues to be clustered dots. In [Pnueli 93], a nice overview of traditional halftoning methods is presented along with new gridless halftoning approaches, all employing only the given two dimensional image. In [Winkenbach 94], the authors investigate the possibilities of applying strokes, tones and textures, and outlines or silhouette curves for polygonal computer models, employing three dimensional information.

In this work, we present an approach that computes halftoned images using a coverage of curves for scenes formed of freeform NURBs surface models that can employ arbitrarty shading models. In [Dooley 90], freeform NURBs surface models are processed into a line drawing by extracting feature curves from the surfaces. Boundary curves, silhouette curves, discontinuity curves and isoparametric curves are the four types that are displayed. The different curves are drawn with different width and dotted attributes computed as function of the level of visibility of the specific curve. However, in [Dooley 90], no shading considerations are being made. Herein, the three dimensional geometry is utilized in order to create the coverage along the general stream lines of the model. The method presented in this work can be completely combined with the approach of [Dooley 90]. We exploit a *freeform surface covering scheme* of isoparametric curves that was originally introduced in [Elber 94] for tool path generation, for machining purposes. In [Elber 94], the coverage is guaranteed to cover the entire surface to within some prescribed tolerance. That is, the density of the curves is bounded from above (See Figure 1). Herein, we tune the norm of the density of the coverage to become a function of simple lighting and shading models (See Figure 2). Moreover, in Figure 2, noise has been added to the scene, alleviating the artifacts that are formed in the scheme presented in [Elber 92, Elber 94]. The curve based coverage employed herein retains its isoparametric origin and hence, it typically follows the general shape of the object. As a result, we obtain a halftoned coverage that is aesthetic and pleasing as

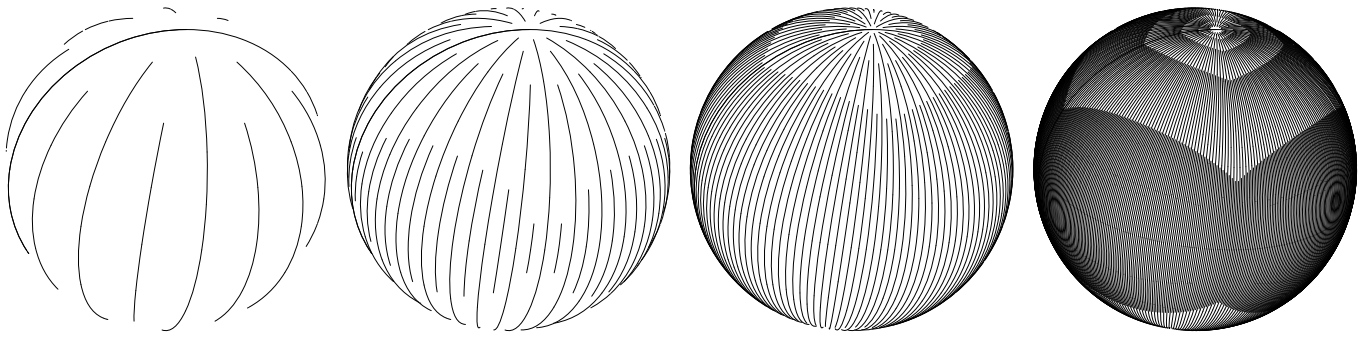

Figure 1: Four different tolerances δ (Definition 1) resulting in four different densities of a coverage of a parametric sphere, using adaptive extraction of isoparametric curves.

it follows the general parameterization and the streamlines of the original model, yet it is parametrization dependent.

This paper is organized as follows. Section 2 presents the basic idea and the proposed algorithm. In Section 3, we present some basic examples. Several possible extensions to the shading model in this framework are considered in Section 4. Finally, in Section 5, we draw some conclusions. All the figures in this paper were created using the IRIT [IRIT 95] solid modeller, developed at the Technion.

2 Background and Algorithm

In [Elber 94], a method was proposed to form a *coverage* of freeform surfaces using isoparametric curves. From [Elber 94],

Definition 1 *A set of curves $\mathcal{C}$ in a given surface $S(u,v)$ is called a* valid coverage for S with respect to some constant δ *if, for any point $p = S(u_p, v_p)$, there is a point q on one of the curves in $\mathcal{C}$, such that $\|p - q\|_2 < \delta$, where $\|\cdot\|_2$ denotes the Euclidean distance.*

Following Definition 1, the resulting coverage guarantees that for every point $p \in S$, there is a different point $q \in \mathcal{C}$ that is sufficiently close. Herein, as well as in [Elber 92, Elber 94] isoparametric curves are employed in the computed coverage. While in general the distance in Definition 1 should be measured on the surface, as $\delta \to 0$, the surface region between any two adjacent curves in $\mathcal{C}$ can be approximated more closely as a ruled surface and hence the Euclidean distance approaches the distance measured on the surface (See [Elber 92] for more). In [Elber 94], closeness is measured and motivated by the established bound on the scallop height that results when one drives the cutting tool along $\mathcal{C}$. In [Elber 92], the tolerance δ is selected at a sub-pixel level to ensure a complete coverage, for the purpose of direct freeform surface rendering. In this work, we pursue a different direction to the one presented in [Elber 92, Elber 94] and modify δ in Definition 1 to become a distance functional,

Definition 2 *A set of curves $\mathcal{C}$ in a given surface $S(u,v)$ is called a* valid coverage for $S(u,v)$ with respect to some functional $\delta(u,v)$ *if, for any point $p = S(u_p, v_p)$, there is a point q on one of the curves in $\mathcal{C}$, such*

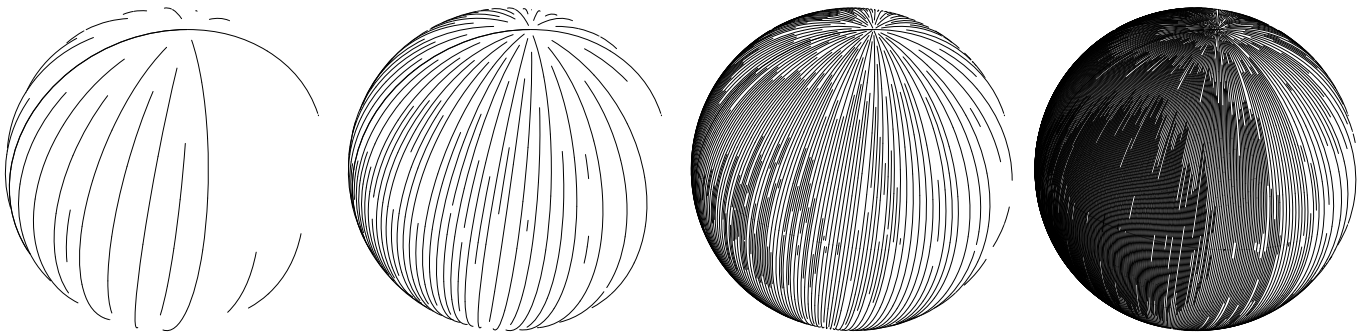

Figure 2: Four different ξ resulting in four different densities of coverage of a parametric sphere, using tolerance δ that is a shader function.

that $\|p - q\|_2 < \delta(u_p, v_p)$, *where* $\|\cdot\|_2$ *denotes the Euclidean distance.*

By considering δ as a functional, we are able to locally control and manipulate the quality and density of the coverage. $\delta(u, v)$ can denote the amount of light illuminating surface $S(u, v)$, $\forall u,\ v$. In particular, δ can be a function of the viewing direction and light source orientation, employing a simple shading model that incorporates diffuse and ambient light [Foley 90].

In [Winkenbach 94], the Phong model has been employed for a similar application. Herein, we have also found the Phong cosine lighting model to be sufficient for the generation of non-photorealistic line illustrations. Exploiting the Phong shading model, one needs to modify the surface coverage of Definion 1 so δ can be employed as a simple shader. Denote by $\mathcal{L}$ and $\mathcal{V}$ the unit direction of the light source and the unit view direction, respectively (see Figure 3). Given surface $S(u, v)$, let $n(u, v)$ be the unit normal of $S(u, v)$. Then,

$$\delta(u, v) = \alpha \langle \mathcal{L}, n(u, v) \rangle , \tag{1}$$

where α is a handle on the intensity control. $\delta(u, v)$ in Equation (1) prescribes a simple shader with only a diffuse lighting term. Adding a specular term yields,

$$\delta(u, v) = \alpha \langle \mathcal{L}, n(u, v) \rangle + \beta \langle \mathcal{V}, r(u, v) \rangle^c , \tag{2}$$

for some handles on the intensity control, α and β, where $r(u, v)$ is the reflection direction of light from the light source off $S(u, v)$,

$$r(u, v) = 2 \langle n(u, v), \mathcal{L} \rangle n(u, v) - \mathcal{L},$$

and c is the power of the specular term. See [Foley 90] for more.

In [Elber 92, Elber 94], a divide and conquer approach provides a solution to the coverage problem, minimizing the amount of redundancy in the coverage. This approach starts with two boundary curves of

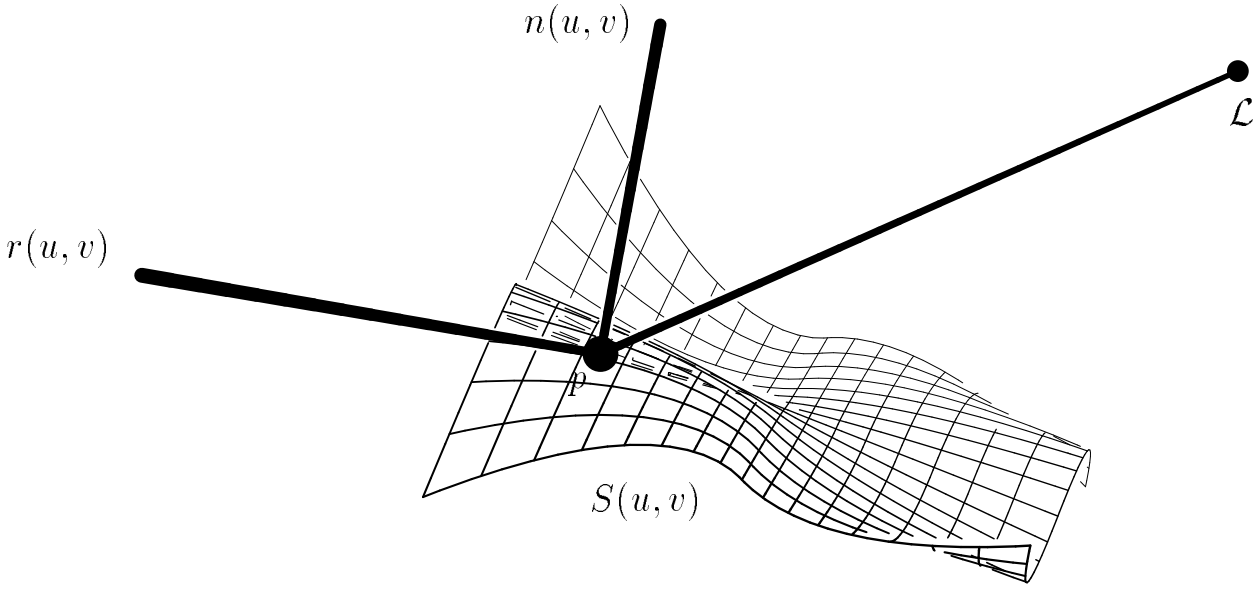

Figure 3: The illumination of the surface $S(u,v)$ at p is derived using its normal, $n(u,v)$, the light source location or direction $\mathcal{L}$, and the deviation between the reflected direction, $r(u,v)$, and the viewing direction, $\mathcal{V}$. $\mathcal{V}$ is orthogonal to the page.

a NURBs surface $S(u,v)$, $C_0(v) = S(u_0,v)$ and $C_1(v) = S(u_1,v)$. The algorithm proceeds by defining and computing the functional,

$$\sigma(v) = \|C_0(v) - C_1(v)\|^2, \tag{3}$$

which is the square of what is denoted the *iso-distance* in [Elber 92, Elber 94]. Denote the surface domain between $C_0(v)$ and $C_1(v)$ as $\mathcal{D}_{01}(u,v) \subset S(u,v)$, $u_0 \le u \le u_1$, $v_0 \le v \le v_1$ (See Figure 4). Let $q = \mathcal{D}_{01}(u_2,v_2)$ be a point in the domain. Assume the curvature of isoparametric curve $\mathcal{D}_{01}(u,v_2)$, $u_0 \le u \le u_1$ is "small". Hence, the distance between point $\mathcal{D}_{01}(u_0,v_2) \in C_0(v)$ and point $\mathcal{D}_{01}(u_1,v_2) \in C_1(v)$ provides an upper bound on the maximal distance between $q \in \mathcal{D}_{01}(u,v)$ and the closest point to q on either $C_0(v)$ or $C_1(v)$. For such a point q, and with no other curve in $\mathcal{D}_{01}(u,v)$, $C_0(v)$ and $C_1(v)$ are the closest two curves that can provide the coverage for q. Moreover, at the extreme case, q is equidistant from the two curves. Hence, $\sigma(v)$ in Equation (3) can be used as a bound on $\delta(u,v)$ of the coverage as square of twice δ. The zero set of $\Psi(v) = \sigma(v) - (2\delta)^2$ is then numerically approximated. If $\Psi(v) < 0$, $\forall v$, the algorithm terminates because the two curves are sufficiently close to each other. If $\Psi(v) > 0$, $\forall v$, an intermediate isoparametric curve $C_{01}(v) = S(\frac{u_0+u_1}{2}, v)$ is introduced and the two distance functionals of $\sigma_1(v) = \|C_0(v) - C_{01}(v)\|^2$ and $\sigma_2(v) = \|C_{01}(v) - C_1(v)\|^2$ are recursively evaluated. Alternatively, $\sigma(v)$ may be less than $(2\delta)^2$ for only a fraction of the parametric domain. Then, the two adjacent isoparametric curves are subdivided at the zero set locations of $\Psi(v)$ and the algorithm recurses only into the regions along the v parameter for which $\Psi(v) > 0$. This surface coverage algorithm generated a set of curves that are neighboring isocurves are at most 2δ iso-distance apart. Furthermore, on the limit, as $\delta \to 0$, the iso-distance between two adjacent isocurves will be at least δ (see [Elber 92]).

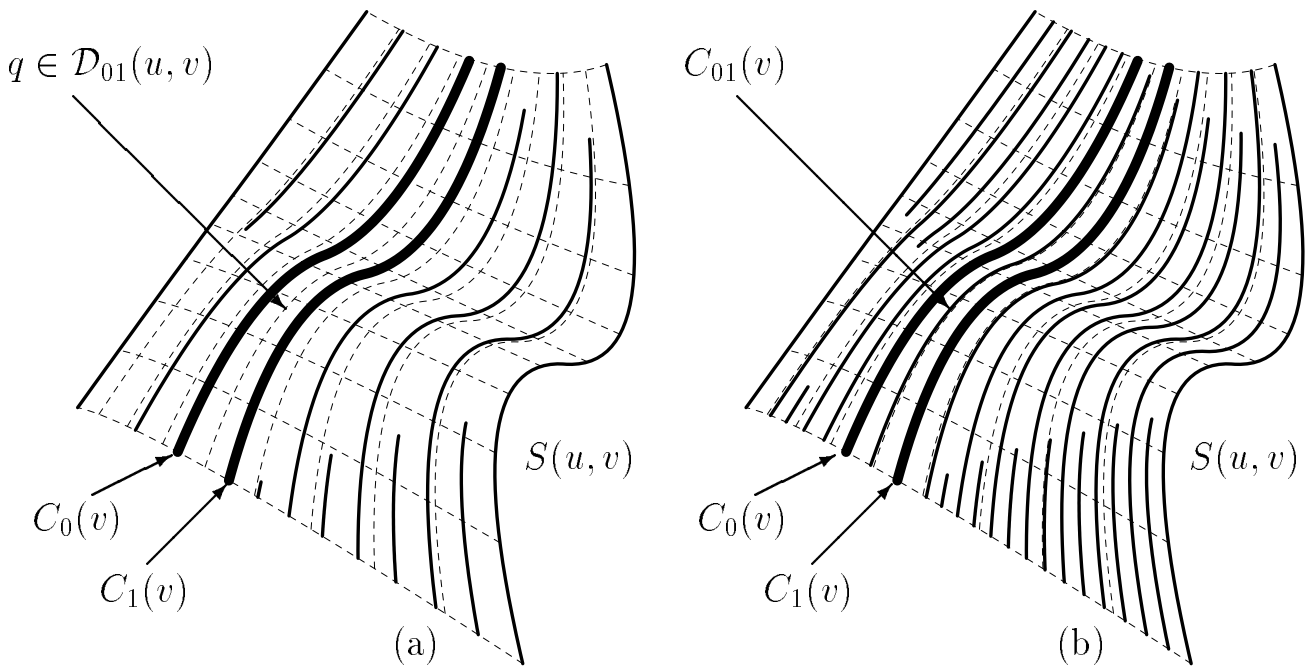

Figure 4: The coverage near point $q \in \mathcal{D}_{01}(u,v)$ between two adjacent curves, $C_0(v)$ and $C_1(v)$ is validated in (a) by computing the iso-distance square, $\sigma(v)$ between the two curves. In (b) the necessary isocurve, $C_{01}(v)$, is introduced between $C_0(v)$ and $C_1(v)$ to form the valid coverage for δ. The original surface is shown in dotted lines.

It is unfortunate that both shading models (Equation (1) and (2)) are not (piecewise) rational due to the square root normalization factors of the unit vectors. Therefore, we are unable to exactly represent Equation (1) and (2) as NURBs, in general. Nonetheless, given an isoparametric domain of $S(u,v)$, one can approximate the lighting model of Equations (1) and (2) within an arbitrary precision. Evaluate the shading models (Equations (1) or (2)) at a discrete set of locations along the domain. A Bspline approximation to the evaluated sampled set is expected to quadratically converge to the original shading model due to the Schoenberg variation diminishing property of the representation [Marsden 66], as the number of samples is increased. Let $\delta(v) = \delta(u_i, v)$ be the iso-distance functional along the $S(u_i, v)$ isoparametric surface of S. Denote by $\hat{\delta}(v)$ the piecewise polynomial approximation of $\delta(v)$. Then, $\Psi(v)$ is computable and representable in the space of piecewise rationals as $\Psi(v) \approx \sigma(v) - (2\hat{\delta}(v))^2$. Algorithm 1 summarizes the suggested approach. In line (1) of Algorithm 1, we approximate the shading model as a functional. In line (1), $\delta(v)$ is approximated as the average of the two isocurves of the iso-distance functional $\delta(u,v)$ at $C_0(v)$ and $C_1(v)$. An approximation to the shading model for both $C_0(v)$ and $C_1(v)$ is computed and an average of the two is employed as an approximation for the final $\hat{\delta}(v)$ for the entire domain of S between $C_0(v)$ and $C_1(v)$. If, however, the two curves are too far apart by the iso-distance norm, the insertion of the intermediate curve, $C_{01}(v)$, is enforced, regardless of the value of $\hat{\delta}(v)$. See line (2) in Algorithm 1.

Algorithm 1: Isoparametric curve based shader. Iso-u curves are assumed.

```
Input:
  S(u,v), surface to render;
  L, unit size light source direction;
  V, unit size viewing direction;
  ξ, a tolerance control over the rendering;
Output:
  A coverage of S(u,v) using isoparametric curves;
Algorithm:
  AdapIsoShader( S(u,v), L, V, ξ )
  begin
    C_0(v) ⇐ S(u_0,v);
    C_1(v) ⇐ S(u_1,v);
            { C_0(v) } ∪
    return  AdapIsoShaderAux( S(u,v), ξ, L, V, {C_0(v),C_1(v)}, u_0, u_1 ) ∪
            { C_1(v) };
  end

  AdapIsoShaderAux( S(u,v), ξ, L, V, {C_0(v),C_1(v)}, u_0, u_1 )
  begin
    σ^2(v) ⇐ ||C_0(v) − C_1(v)||_2, iso-distance between C_0(v) and C_1(v);
(1) δ̂(v) ⇐ Bspline approximation to the illumination of S at C_0(v) and C_1(v);
    if (σ(v) < (2ξδ̂(v))^2, ∀u) then
      return ∅;
(2) else if (σ(v) > (2ξδ̂(v))^2, ∀u) or σ(v) too large then
      begin
        u_01 ⇐ (u_0+u_1)/2;
        C_01(v) ⇐ S(u_01,v), Middle isocurve between C_0(v) and C_1(v);
        return  AdapIsoShaderAux( S(u,v), ξ, L, V, {C_0(v),C_01(v)}, u_0, u_01 ) ∪
                AdapIsoShaderAux( S(u,v), ξ, L, V, {C_01(v),C_1(v)}, u_01, u_1 );
      end
    else
      begin
        {C_0^i(v),C_1^i(v)} ⇐ subdivided {C_0(v), C_1(v)} at all u such that
                                                          σ(v) = (2ξδ̂(v))^2;
        return ∪_i AdapIsoShaderAux( S(u,v), ξ, L, V, {C_0^i(v),C_1^i(v)}, u_0, u_1 );
      end
  end
```

Consider a closed surface, S, where the first and last isoparametric curves, $C_0(v)$ and $C_1(v)$, of S are identical. Then, Algorithm 1 would terminate immediately on S. Clearly, by enforcing a single subdivision, this problem can be overcomed. It can occur, however, that two different regions of the surface are close to each other suggesting a similar deficiency as in the closed surface case just considered. In practice, and mainly since δ is far less than the surface size, it is highly unlikely that such a failure will ever occur.

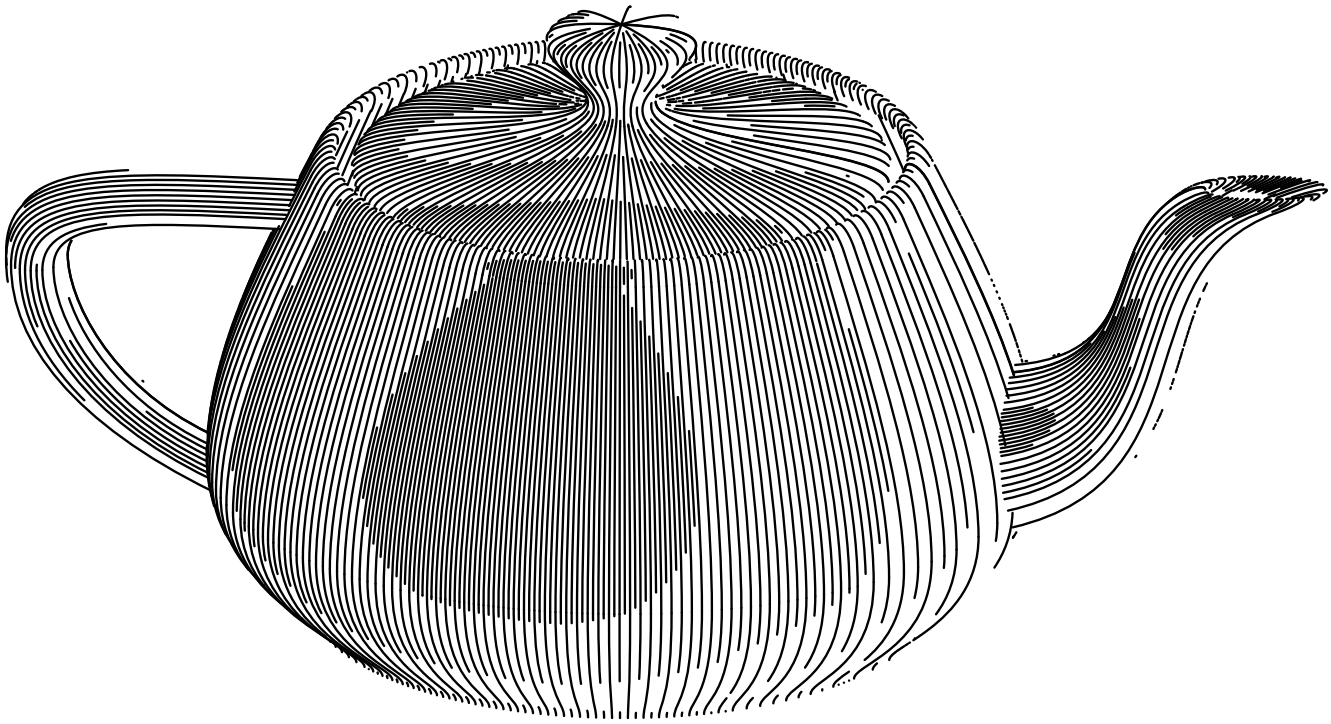

Figure 5: Algorithm 1 yields results that are clearly synthetic. Compare with Figure 6.

All the distance functionals in Algorithm 1 are computed in the *image plane* between the projected two dimensional curves, while the shading model is evaluated in the three dimensional object space. The result is a coverage that is approaching uniformity in the drawn image. Unfortunately, the results that Algorithm 1 yields also appear synthetic due to the binary subdivision nature of the algorithm. See for example Figure 5. Nevertheless, there exists a simple remedy. One can introduce white noise into the shader $\hat{\delta}(v)$ before solving for the zero set of $\Psi(v) = \sigma(v) - (2\xi\hat{\delta}(v))^2$, in line (1) of Algorithm 1, significantly alleviating the sensitivity to the introduced synthetic artifacts of the binary subdivision. Compare Figure 6 with Figure 5.

In order to present the results of Algorithm 1, one needs to eliminate the invisible portion of the coverage. In this work, we have exploited a Z-buffer to render the freeform surface model and save a Z-depth map of the rendered scene. Then, the computed isoparametric coverage is translated a small fraction in Z toward the viewer and its visibility is verified against the Z-depth map to isolate the visible portion of the isoparametric curve's based coverage. The visible portion of the coverage was then converted into cubic Bézier curves in the Postscript [Postscript 85] page description language format, ready for printing.

In the next section, we demonstrate the use of the algorithm on different scenes of freeform models.

3 Basic Examples

In Section 2, a method was presented to construct a pleasing line-art rendering using isoparametric coverage of freeform surfaces. We have several degrees of freedom when the isoparametric based illustration is created,

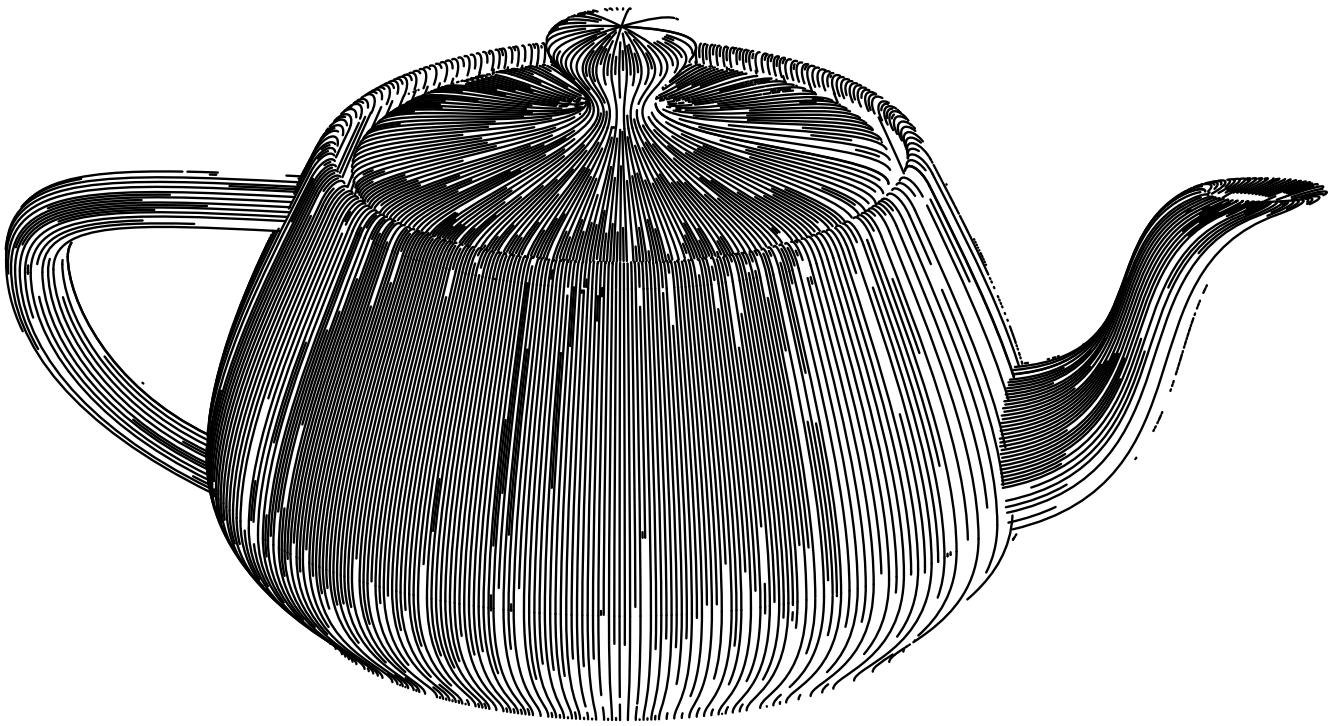

Figure 6: Adding white noise to $\hat{\delta}(v)$ just before solving for the zero set of $\Psi(v) = \sigma(v) - \hat{\delta}^2(v)$, results in a more pleasing image. Compare with Figure 5.

using the shading models of Equations (1) or (2). One can globally control the density of the coverage, via the ξ parameter in Algorithm 1. Figure 7 shows the Utah teapot with several different ξ density values. Alternatively and since a Phong cosine shader is employed, one can modify the light source direction as well as the power of the specular term (c, in Equation (2)). Figure 8 shows the Utah teapot rendered with different powers, c, of the specular term.

Figure 9 shows a more complex model, of a b58 airplane, rendered with the introduced technique, using two different light sources, and two different specular power terms.

Figures 10 and 11 present two more complex scenes that were rendered using the proposed method. An additional controllable degree of freedom employed in the scene of Figure 11 is the reversed direction of the isoparametric curves of the rounded table. Furthermore, the table in this scene is rendered with a reduced coverage density.

4 Extending the Shading model

The set of silhouette curves of a freeform object is also considered important in conveying the general shape of the object. One can set δ to be a function that is sensitive to the surface normal, or more specifically to the z coefficient of the unit normal of the surface, $n_z(u, v)$,

$$\delta(u, v) = 1 - n_z(u, v)^c. \tag{4}$$

Figure 7: The Utah teapot line-art using isoparametric curves, with different densities, controlled via ξ.

Figure 8: The Utah teapot line-art using isoparametric curves, with different specular powers, c, of the lighting model of Equation (2).

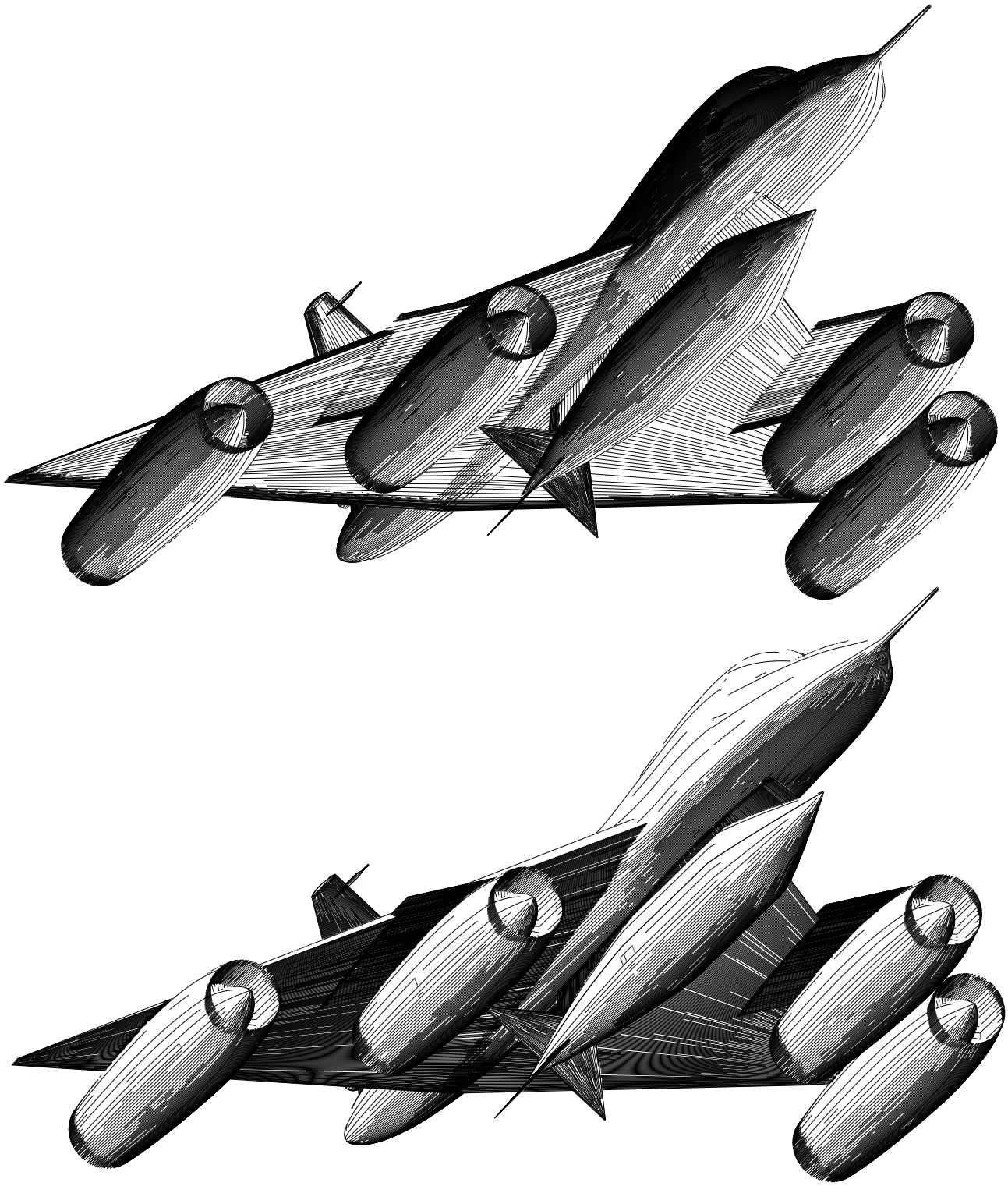

Figure 9: Two views of a b58 airplane. Line-art using isoparametric curves, in two different specular powers and two different light source directions.

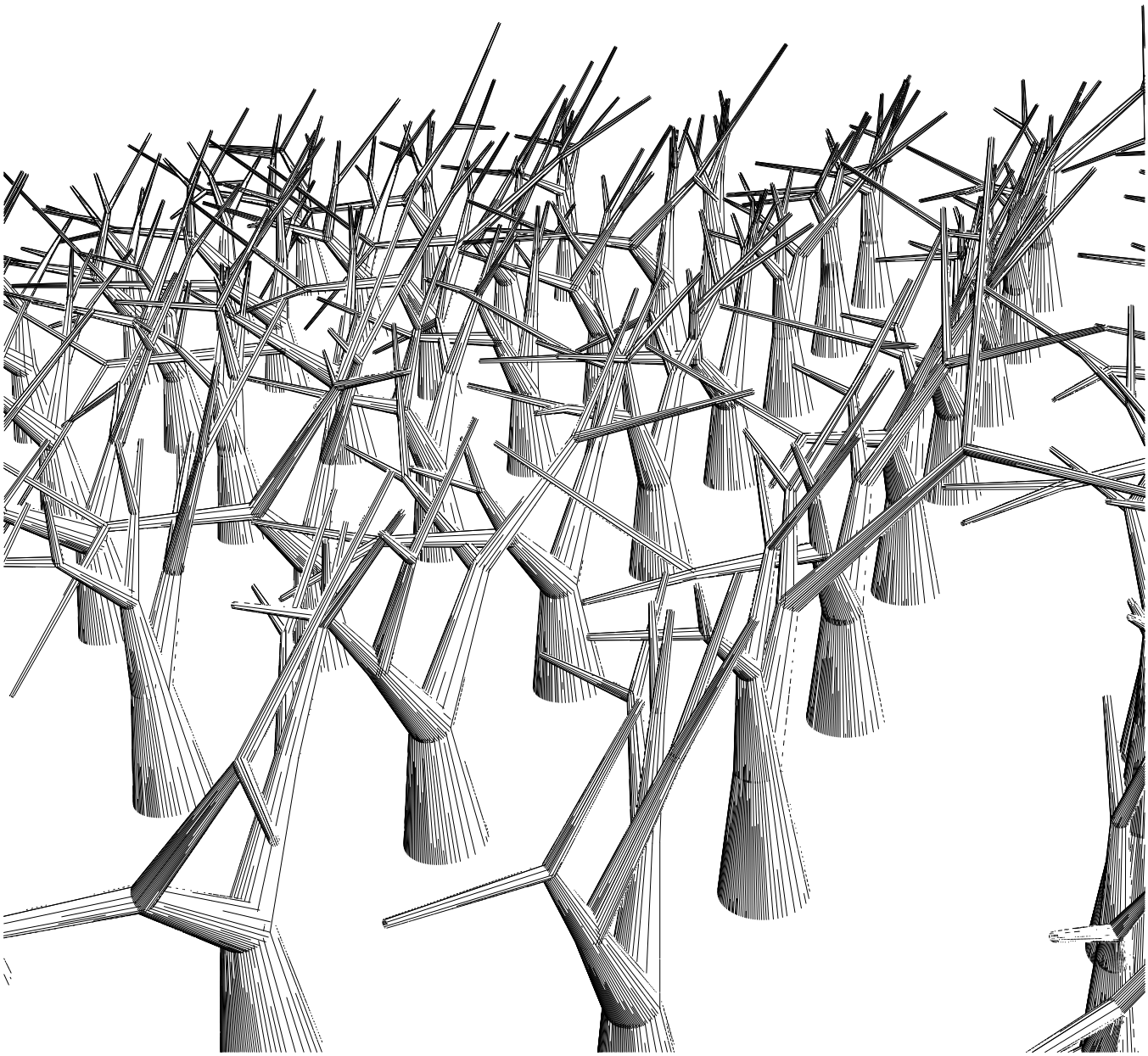

Figure 10: A forrest of cones, rendered using isoparametric curves.

In Equation (4), c is a control over the rate the influence of the silhouette curves decays as we move away from a silhouette area. The shader defined in Equation (4) enhances the areas of the silhouettes of the rendered model. In Figure 12, the silhouette model of Equation (4) has been exploited. In this figure, isoparametric curves in both the u and the v directions are extracted, creating cross hatching of the rendered surfaces.

One can set $\delta(u, v)$ to become a function of the three dimensional location in space. Trivariate function were employed for the generation of three dimensional rendering and texturing information [Peachey 85, Perlin 85]. By making δ a trivariate density function, $\delta(x, y, z)$, one can, given a location in 3-space, directly

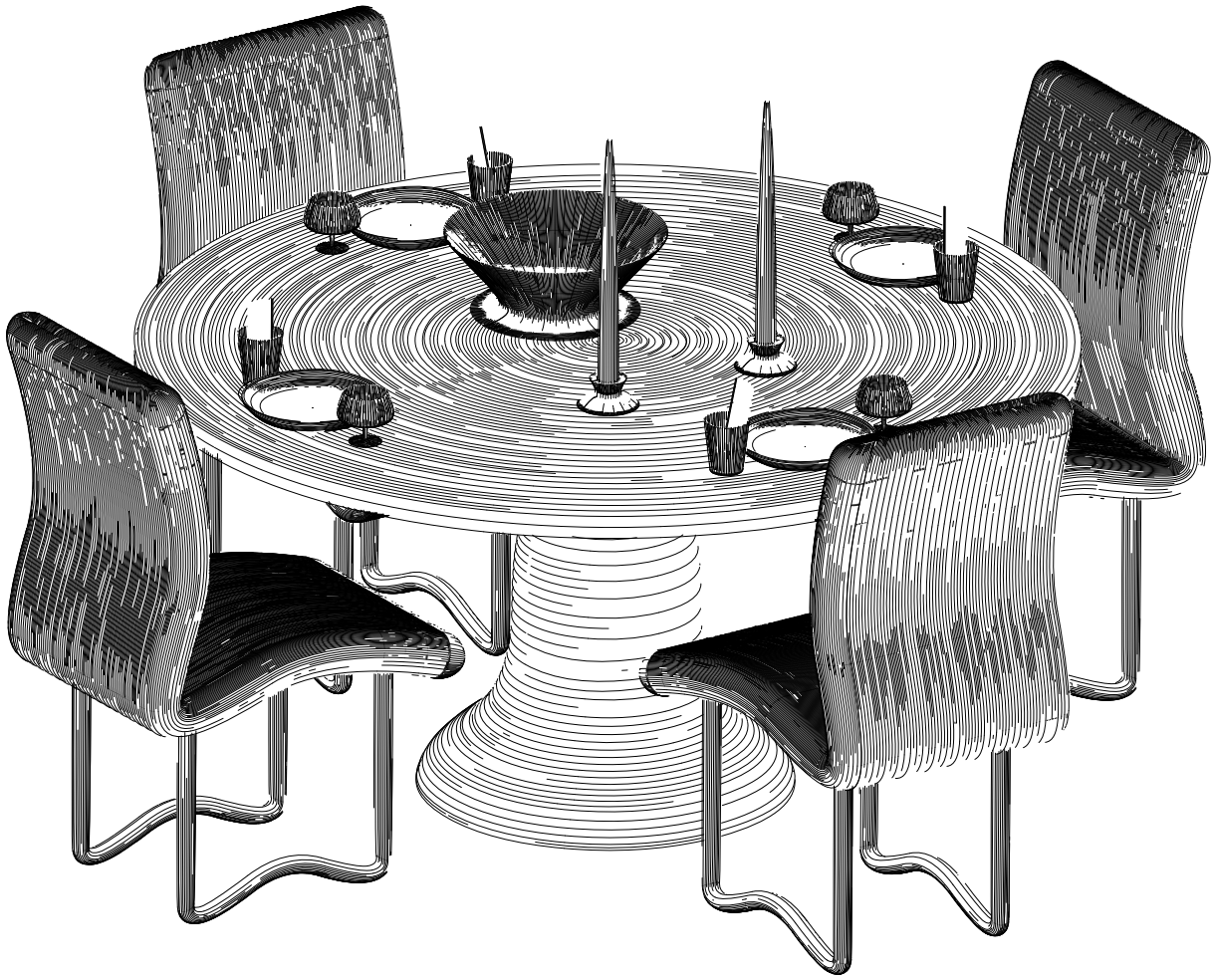

Figure 11: A scene of a dinner table, rendered using isoparametric curves.

prescribe the density of coverage. $\hat{\delta}(v)$ in line (1) in Algorithm 1 is derived from $\delta(x, y, z)$ by sampling the later along the processed isocurve. For example, a point (or a set of) light sources can accurately represent the quadratic distance attenuation of the light in this model, in a simple manner. Figure 13 shows the scene of the dinner table with $\delta(x, y, z)$ which is the distance to the fourth for a single point light source in space.

In [Ivins 92] Figure 38, ("The Spinello" by Baudet) the shading effect is achieved not by modifying the density of the lines, but by modifying their thickness. In order to nicely emulate this idea, one needs to compute a coverage of the surface so that all curves are equidistant from their neighboring curves, in screen space. Unfortunately, the coverage employed in this work does not guarantee that. The iso-distance between adjacent isoparametric curves can abruptly vary by up to a factor of two, in the limit. Again, one can alleviate the eye's sensitivity to this deficiency by randomizing the locations of these discontinuities in the coverage. In Figure 14, both the locations and the thickness of the curves were perturbed using white noise.

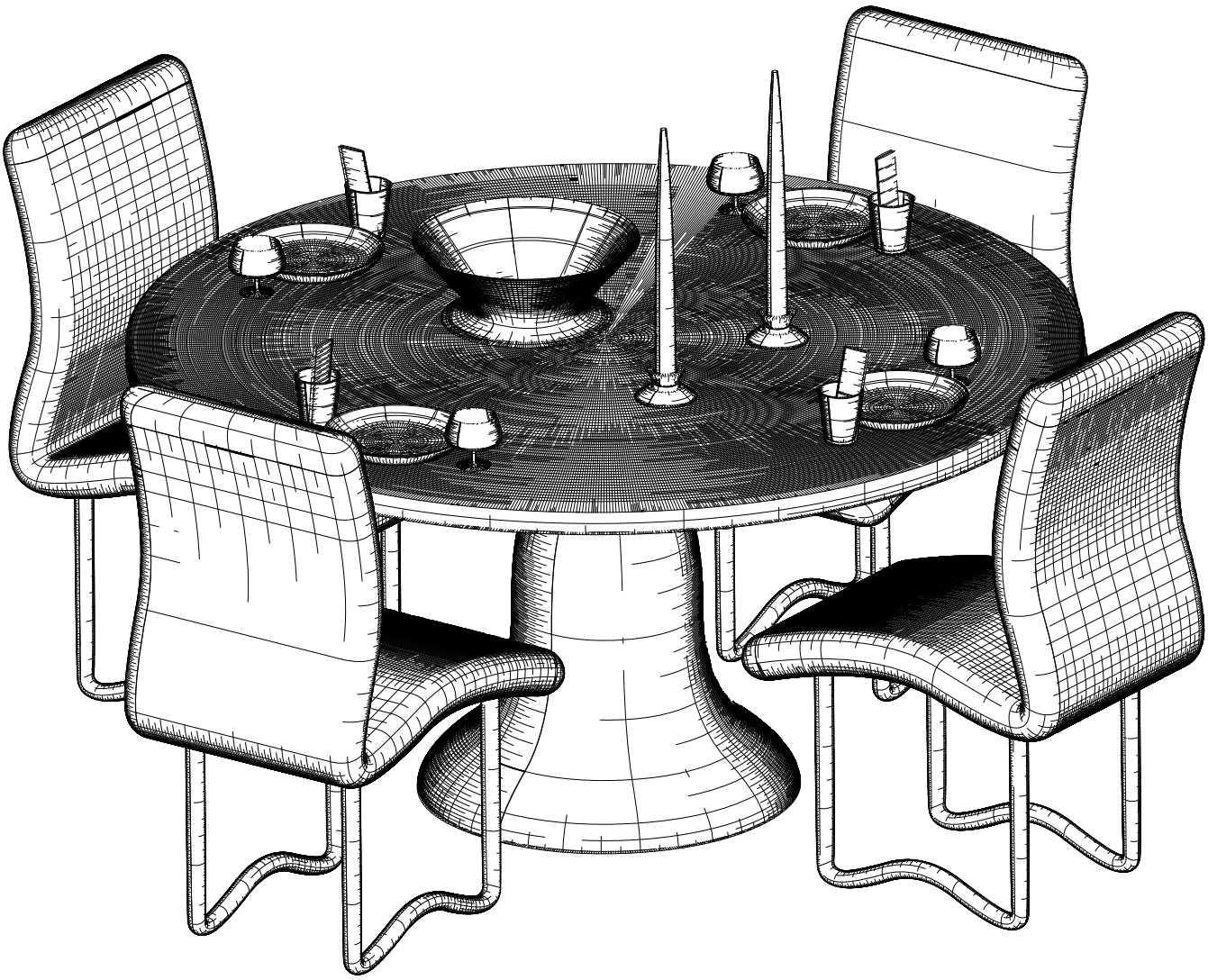

Figure 12: A scene of a dinner table, enhancing the silhouette areas, with cross hatching of both u and v isoparametric curves.

5 Conclusions

We have presented a different application of the isoparametric curve based coverage algorithm presented in [Elber 92, Elber 94]. The model presented in this work exploits a simple lighting model to control the density of the coverage. It is probable that a more advanced lighting model can improve the quality of the result. Most noticeably, shadows can greatly enhance the illustrative power of the result. The proposed method is only one step toward gridless halftoning and new methods must be sought to achieve highly aesthetic line-art renderings of synthetic scenes.

In this work, an isoparametric based coverage has been selected, mainly because of the simplicity in computing the coverage. Clearly, one pays for this simplicity by becoming parametrization dependent. Fortunately, in many instances the parametrization of the surfaces looks natural due to the way the surfaces are constructed. The fuselage of the b58 in Figure 9 was construct by fitting an approximation surface

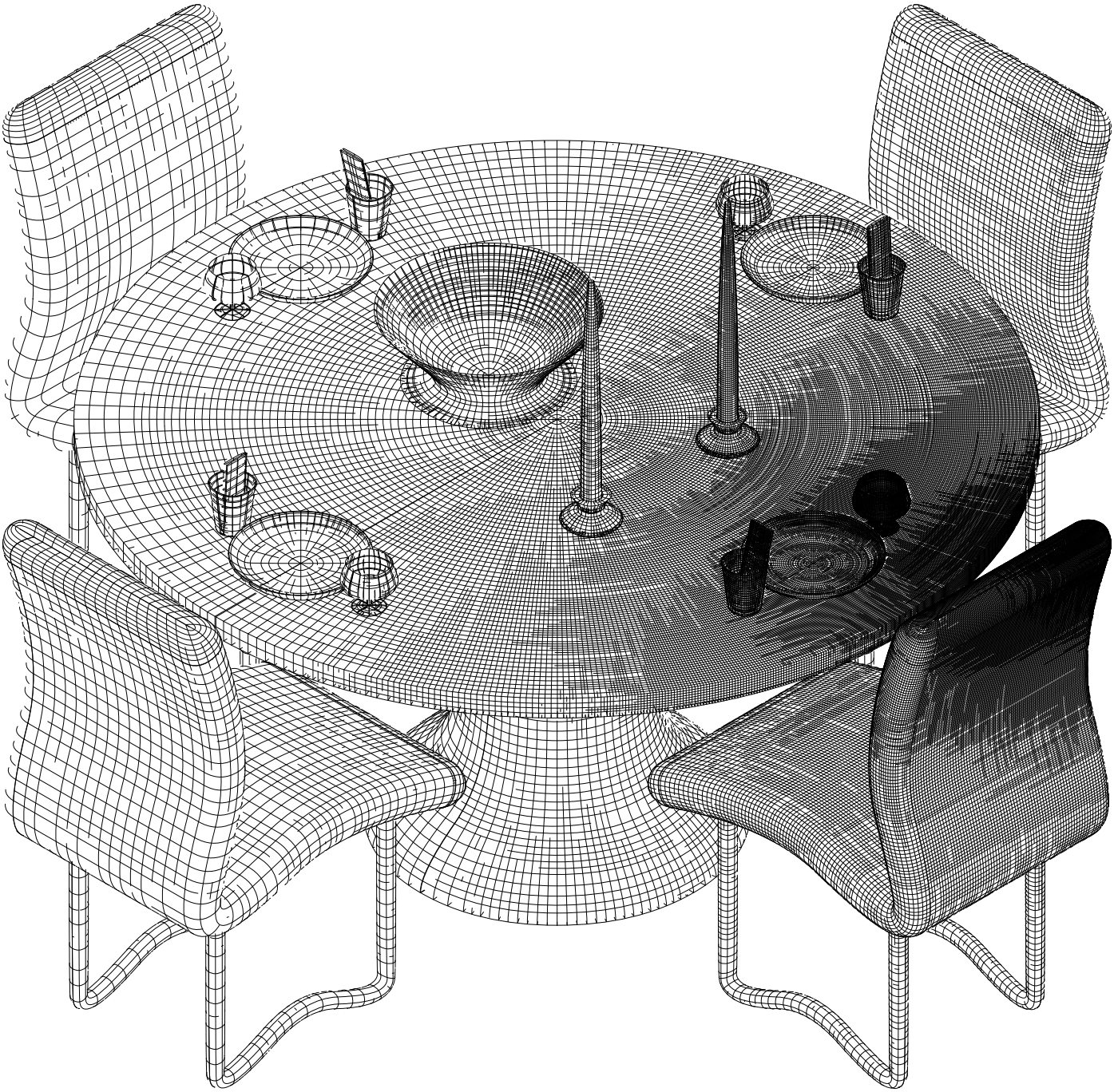

Figure 13: A scene of a dinner table, rendered with a single point light source illuminating the scene from the upper right. Herein, dark spots represents the highlight. This image is also using cross hatching of both u and v isoparametric curves.

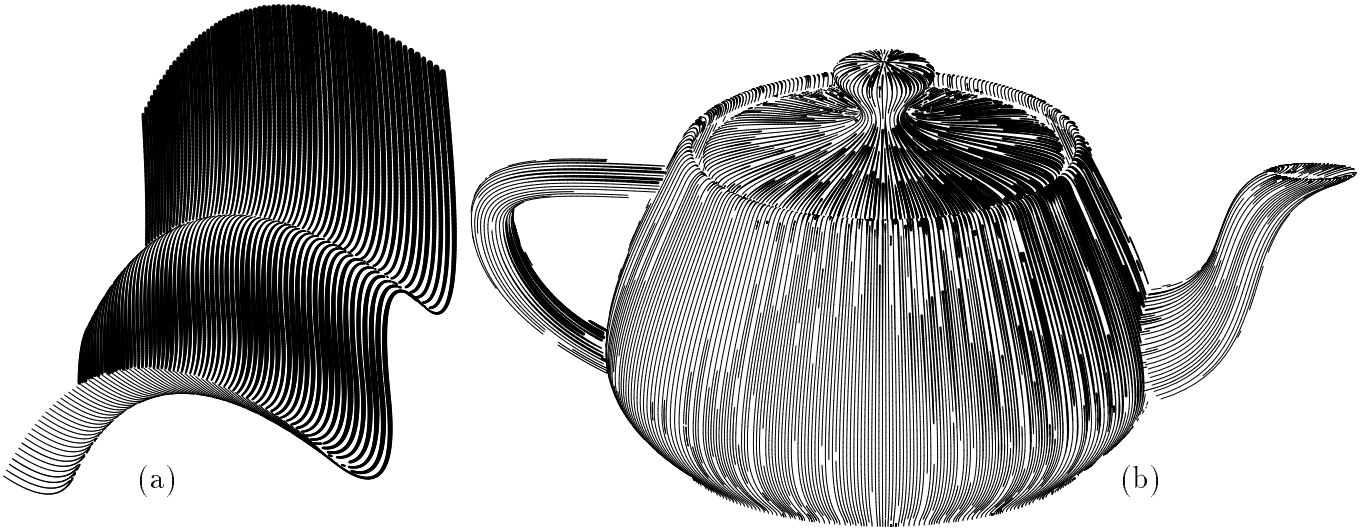

Figure 14: One can attempt at rendering the scene using a continuous change of the width of the lines instead of their density.

along several cross sections. This construction materialized in a very natural parameterization as shown in Figure 9. It is obvious that this dependency is a major draw back of the proposed approach and while there exist a rich set of surfaces that this algorithm can be applied for with natural and appealing results, efficient ways to remove this dependency and create a parametrization independent coverage must be pursuit.

Since the aesthetic value of the resulting image is highly subjective, some viewers might find that the added noise is insufficient in alleviating the artifacts introduced due to the discontinuities in the computed coverage. One can employ the variable curve thickness approach to further reduce the influence of these artifacts, by properly setting the width at discontinuous locations, where intermediate isocurves terminates.

During preliminary development of this illustration tool, the results were displayed on a raster display device using Z-buffer hardware. The discrete nature of the raster display added high resolution noise to the image which was found to be quite appealing as an artwork. The results presented in this work draw the exact isoparametric curves as they were extracted from the surfaces of the model. We allowed only their parametric domain (or thickness) to vary randomly. Figure 15 shows one extract from the Z-buffer hardware of such image. It is plausible to assume then that random perturbation of the curves themselves using high resolution noise, as done with the wobble and warp line types in the Inventor [Inventor 92] package, can improve the hand drawn sensation of these illustrations and sketches.

6 Acknowledgment

This work was supported in part by the Fund for Promotion of Research at the Technion, Haifa, Israel.

The author is grateful to Gennady Agranov for his help in making the Z buffer usage for depth queries a feasible solution, using the SGI's hardware. Also, the author is grateful to the anonymous reviewers of this paper for their valuable remarks on the original draft of this paper.

Figure 15: A rendered image of the Utah teapot using a hardware Z-buffer. The discrete nature of the result adds high resolution noise to the image that is appealing.

References

[DoCarmo 76] M. D. Carmo. Differential Geometry of Curves and Surfaces. Prentice-Hall 1976.

[Dooley 90] D. Dooley and M. F. Cohen. Automatic illustration of 3D geometric models: Lines. Computer Graphics, pp 77-82, Vol 24, No. 2, March 1990.

[Elber 92] G. Elber and E. Cohen. Adaptive Iso-Curves Based Rendering for Free Form Surfaces. Submitted for Publication. Also Technical Report UUCS-92-040, Department of Computer Science, Univeristy of Utah.

[Elber 94] G. Elber and E. Cohen. Tool Path Generation for Freeform Surface Models. Computer Aided Design, Vol 26, No. 6, June 1994.

[Foley 90] J. D. Foley et al. Computer Graphics, Principles and Practice, Second Edition. Addison-Wesley Systems Programming Series, Jul. 1990.

[Hasluck 77] P. N. Hasluck Manual of Traditional Wood Carving. Dover Publications, Inc., NewYork, 1977.

[Hilbert 90] D. Hilbert and S. Cohn-Vossen. Geometry and the Imagination. Chelsea Publishing Company, New York, Second edition 1990.

[Inventor 92] Inventor 1.0. IRIS Inventor Programming Guide, Volume I. SGI document number 007-1631-010, June 1992. Silican Graphics, Inc., Mountain View, California.

[Ivins 87] W. M. Ivins. How Prints Look, Photographs with Commentary. Beacon Press, Boston, Revisied edition 1987

[Ivins 92] W. M. Ivins. Prints and Visual Communication. The MIT Press, Eighth printing, 1992.

[IRIT 95] IRIT. Irit 5.0 User's Manual, Technion, March 1995.

[Marsden 66] M. Marsden and I. J. Schoenberg. On Variation Diminishing Spline Approximation Methods. *Mathematica*, Vol. 8(31), No. 1, pp 61-82, 1966.

[Peachey 85] D. R. Peachey. Solid texturing of Complex Surfaces. Computer Graphics, Vol. 19, No. 3, pp. 279-286, Siggraph Jul. 1985.

[Perlin 85] K. Perlin. An Image Synthesizer. Computer Graphics, Vol. 19, No. 3, pp. 287-296, Siggraph Jul. 1985.

[Pnueli 93] Y. B. Pnueli and A. M. Bruckstein. Gridless Halftoning: A reincarnation of the Old Method. CIS report No. 9323, Computer Science Department, Technion, IIT, October 1993.

[Pnueli 94] Y. B. Pnueli and A. M. Bruckstein. DigiDurer - A Digital Engraving System. The Visual Computer, Vol 10, pp 277-292, 1994. Also CIS report No. 9210, Computer Science Department, Technion, IIT, June 1992.

[Postscript 85] PostScript Language Reference Manual. Addison Wesley Publishing Company, 1985.

[Velho 91] L. Velho and J. M. Gomes. Digital Halftoning with Space Filling Curves. Computer Graphics, Vol 25, No. 4, pp. 81-90, Siggraph Jul. 1991.

[Winkenbach 94] G. Winkenbach and D. H. Salesin. Computer-Generated Pen-and-Ink Illustration. Computer Graphics, pp. 91-98, Siggraph Jul. 1994.