

Line Art Rendering via a Coverage of Isoparametric Curves

Gershon Elber, *Member, IEEE*

Abstract—A line-art nonphotorealistic 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 of achieving an aesthetic and attractive line-art rendering that employs isoparametric curve based drawings that is suitable for printing publication.

Index Terms—Sketches, illustrations, line drawings, freeform surfaces, NURBs, gridless halftoning, printing.

I. INTRODUCTION

COMPUTER graphics have 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 [5], is a valid and common solution. Hence, the posteriori halftoning process cannot be globally optimal.

In [19], it is suggested that the two dimensional rendering process should exploit the three dimensional geometry to create a more appropriate nonphotorealistic 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 [7], pp. 197, Fig. 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 indistinguishably. In contrast, in the art of woodcarving, one can frequently find the use of the general stream lines of the object (see [6], [9], [10]).

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 [16], 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 [16] is drawn from the introduction of gridless halftoning that employs curves as the drawing primitives instead of dots or points. The work of [16] is heavily motivated by the art work of Dürer [9], [10] that exploited curves to portray the shading. The results of [16] served as a motivation for the work presented herein. In several locations, for example [18], space filling curves are used in the image plane to produce the digital halftoning. The classical space filling curves of Peano, Hilbert, and Sierpinsky are considered. This halftoning process can also be classified as traditional because its output continues to be clustered dots. In [15], a nice overview of traditional halftoning methods is presented along with new gridless halftoning approaches, all employing only the given two dimensional image. In [19], the authors investigate the possibilities of applying strokes, tones and textures, and outlines of 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 arbitrary shading models. In [2], 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 [2], 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 [2]. We exploit a *freeform surface covering scheme* of isoparametric curves that was originally introduced in [4] for tool path generation, for machining purposes. In [4], 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 Fig. 1). Herein, we tune the norm of the density of the coverage to become a function of simple lighting and shading models (see Fig. 2). Moreover, in Fig. 2, noise has been added to the scene, alleviating the artifacts that are formed in the scheme presented in [3], [4]. The curve

G. Elber is with the Department of Computer Science, Technion, Israel Institute of Technology, Haifa 32000, Israel; e-mail: gershon@cs.technion.ac.il. IEEECS Log Number V95019.

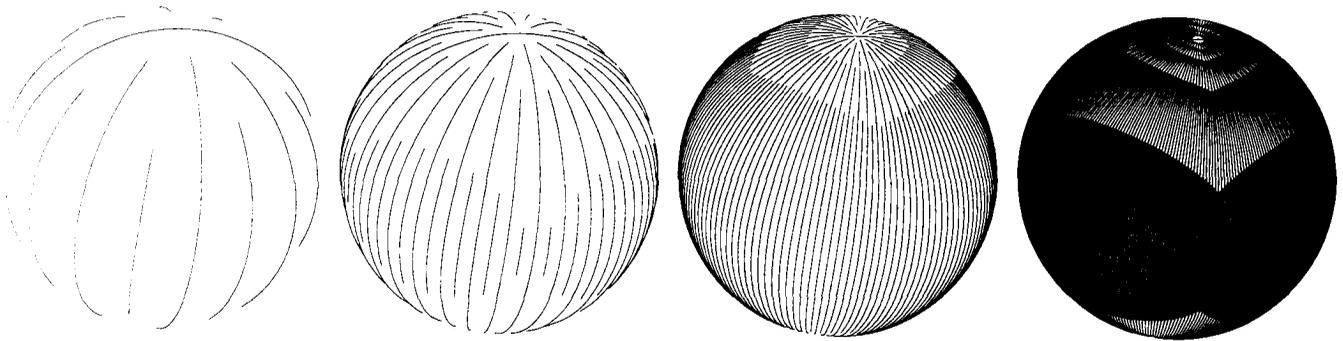


Fig. 1. Four different tolerances δ (Definition 1) resulting in four different densities of a coverage of a parametric sphere, using adaptive extraction of isoparametric curves.

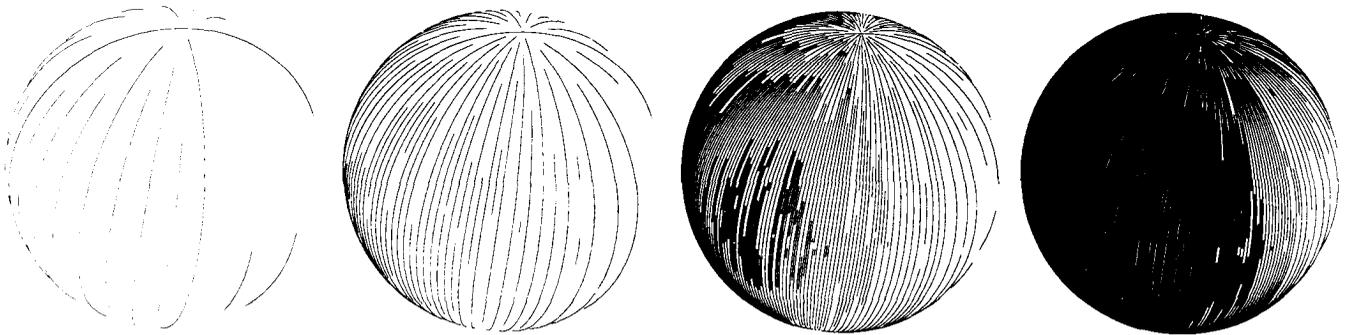


Fig. 2. Four different ξ resulting in four different densities of coverage of a parametric sphere, using tolerance δ that is a shader function.

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

This paper is organized as follows. Section II presents the basic idea and the proposed algorithm. In Section III, we present some basic examples. Several possible extensions to the shading model in this framework are considered in Section IV. Finally, in Section V, we draw some conclusions. All the figures in this paper were created using the IRIT [11] solid modeler, developed at the Technion.

II. BACKGROUND AND ALGORITHM

In [4], a method was proposed to form a coverage of free-form surfaces using isoparametric curves. From [4],

DEFINITION 1. A set of curves 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 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 $q \in C$ that is sufficiently close. Herein, as well as in [3], [4] isoparametric curves are employed in the computed coverage. While in general the distance in Definition 1 should be measured on the

surface, as $\delta \rightarrow 0$, the surface region between any two adjacent curves in C can be approximated more closely as a ruled surface and hence the Euclidean distance approaches the distance measured on the surface (See [3] for more). In [4], closeness is measured and motivated by the established bound on the scallop height that results when one drives the cutting tool along C . In [3], the tolerance δ is selected at a subpixel 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 [3], [4] and modify δ in Definition 1 to become a distance functional,

DEFINITION 2. A set of curves 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 C , such 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 [5].

In [19], 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 nonphotorealistic line illustrations. Exploiting the Phong shading model, one needs to modify the surface coverage of Definition

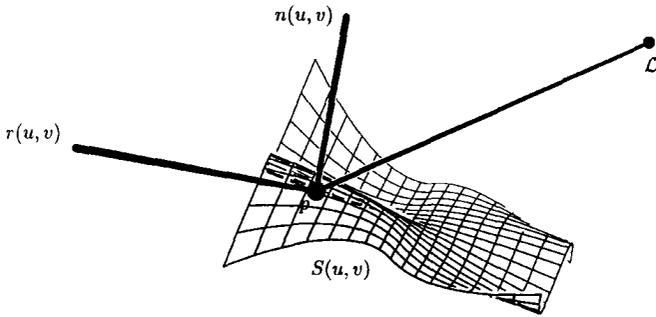


Fig. 3. The illumination of the surface $S(u, v)$ at p is derived using its normal, $n(u, v)$, the light source location or direction \mathcal{L} , and the deviation between the reflected direction, $r(u, v)$, and the viewing direction, \mathcal{V} . \mathcal{V} is orthogonal to the page.

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 Fig. 3). Given surface $S(u, v)$, let $n(u, v)$ be the unit normal of $S(u, v)$. Then,

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

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

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

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 [5] for more.

In [3], [4], 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 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,

$$\sigma(v) = \|C_0(v) - C_1(v)\|^2, \quad (3)$$

which is the square of what is denoted the *iso-distance* in [3], [4]. Denote the surface domain between $C_0(v)$ and $C_1(v)$ as $\mathcal{D}_{01}(u, v) \subset S(u, v)$, $u_0 \leq u \leq u_1$, $v_0 \leq v \leq v_1$ (See Fig. 4). Let $q = \mathcal{D}_{01}(u_2, v_2)$ be a point in the domain. Assume the curvature of isoparametric curve $\mathcal{D}_{01}(u, v_2)$, $u_0 \leq u \leq u_1$ 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 (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

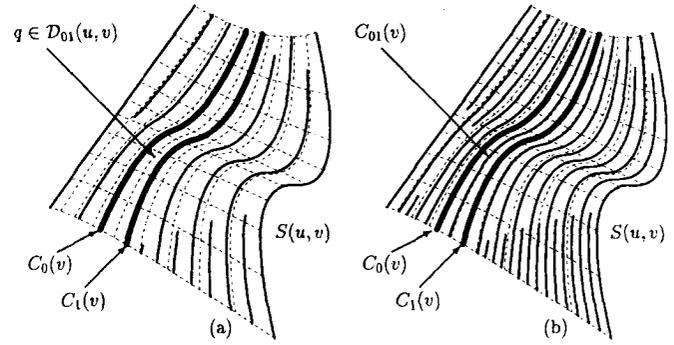


Fig. 4. The coverage near point $q \in \mathcal{D}_{01}(u, v)$ between two adjacent curves, $C_0(v)$ and $C_1(v)$ is validated in (a) by computing the iso-distance square, $\sigma(v)$ between the two curves. In (b) the necessary isocurve, $C_{01}(v)$, is introduced between $C_0(v)$ and $C_1(v)$ to form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

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 $\delta \rightarrow 0$, the iso-distance between two adjacent isocurves will be at least δ (see [3]).

It is unfortunate that both shading models ((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 (1) and (2) as NURBs, in general. Nonetheless, given an isoparametric domain of $S(u, v)$, one can approximate the lighting model of (1) and (2) within an arbitrary precision. Evaluate the shading models ((1) or (2)) at a discrete set of locations along the domain. A B-spline 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 [12], 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 inser-

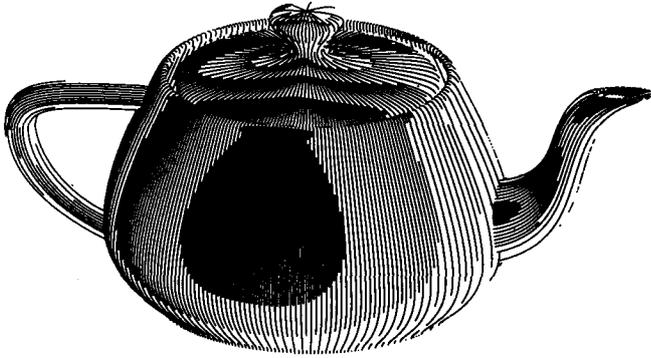


Fig. 5. Algorithm 1 yields results that are clearly synthetic. Compare with Fig. 6.

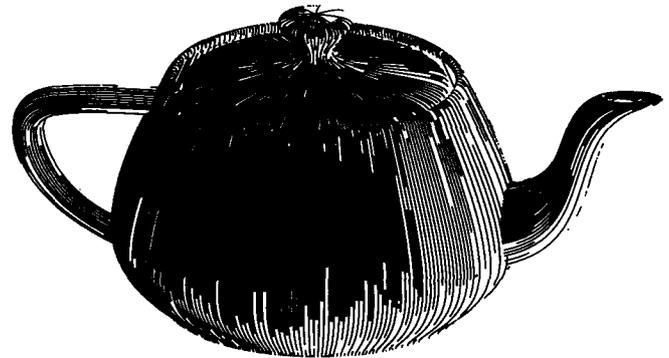


Fig. 6. Adding white noise to $\delta(v)$ just before solving for the zero set of $\Psi(v) = \sigma(v) - \delta(v)$, results in a more pleasing image. Compare with Fig. 5.

tion of the intermediate curve, $C_{01}(v)$, is enforced, regardless of the value of $\hat{\delta}(v)$. See line (2) in Algorithm 1.

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 overcome. 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.

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, Fig. 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

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

$C_0(v) \leftarrow S(u_0, v)$;

$C_1(v) \leftarrow S(u_1, v)$;

$\{C_0(v)\} \cup$

return AdapIsoShaderAux($S(u, v)$, ξ , L , V , $\{C_0(v), C_1(v)\}$, u_0, u_1) \cup
 $\{C_1(v)\}$;

end

AdapIsoShaderAux($S(u, v)$, ξ , L , V , $\{C_0(v), C_1(v)\}$, u_0, u_1)

begin

$\sigma^2(v) \leftarrow \|C_0(v) - C_1(v)\|_2$, iso-distance between $C_0(v)$ and $C_1(v)$;

(1) $\hat{\delta}(v) \leftarrow$ Bspline approximation to the illumination of S at $C_0(v)$ and $C_1(v)$;

if $(\sigma(v) < (2\xi\hat{\delta}(v))^2, \forall u)$ then
return \emptyset ;

(2) else if $(\sigma(v) > (2\xi\hat{\delta}(v))^2, \forall u)$ or $\sigma(v)$ too large then

begin

$u_{01} \leftarrow \frac{u_0 + u_1}{2}$;

$C_{01}(v) \leftarrow S(u_{01}, v)$, Middle isocurve between $C_0(v)$ and $C_1(v)$;

AdapIsoShaderAux($S(u, v)$, ξ , L , V , $\{C_0(v), C_{01}(v)\}$, u_0, u_{01}) \cup

return AdapIsoShaderAux($S(u, v)$, ξ , L , V , $\{C_{01}(v), C_1(v)\}$, u_{01}, u_1);

end

else

begin

$\{C_0^i(v), C_1^i(v)\} \leftarrow$ subdivided $\{C_0(v), C_1(v)\}$ at all u such that

$\sigma(v) = (2\xi\hat{\delta}(v))^2$;

return \cup ; AdapIsoShaderAux($S(u, v)$, ξ , L , V , $\{C_0^i(v), C_1^i(v)\}$, u_0, u_1);

end

end

$\Psi(v) = \sigma(v) - (2\xi\hat{\delta}(v))^2$, in line (1) of Algorithm 1, significantly alleviating the sensitivity to the introduced synthetic artifacts of the binary subdivision. Compare Fig. 6 with Fig. 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 [17] page description language format, ready for printing.

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

III. BASIC EXAMPLES

In Section II, 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, using the shading models of (1) or (2). One can globally control the density of the coverage, via the ξ parameter in Algorithm 1. Fig. 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 (2)). Fig. 8 shows the Utah teapot rendered with different powers, c , of the specular term.

Fig. 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.

Figs. 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 Fig. 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.

IV. 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, $n_z(u, v)$,

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

In (4), c is a control over the rate the influence of the silhouette curves decay as we move away from a silhouette area. The shader defined in (4) enhances the areas of the silhouettes of the rendered model. In Fig. 12, the silhouette model of (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 functions were em-

ployed for the generation of three dimensional rendering and texturing information [13], [14]. By making δ a trivariate density function, $\delta(x, y, z)$, one can, given a location in three-space, directly 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. Fig. 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 [10] Fig. 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 Fig. 14, the locations and the thickness of the curves were perturbed using white noise.

V. CONCLUSIONS

We have presented a different application of the isoparametric curve based coverage algorithm presented in [3], [4]. 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 Fig. 9 construct by fitting an approximation surface along several cross sections. This construction materialized in a very natural parameterization as shown in Fig. 9. It is obvious that this dependency is a major drawback 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.



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



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

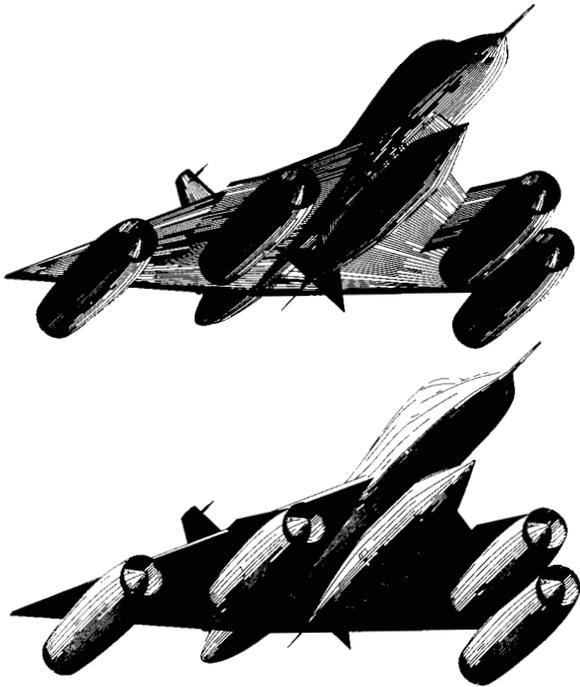


Fig. 9. Two views of a b58 airplane. Line-art using isoparametric curves, in two different specular powers and two different light source directions.



Fig. 10. A forrest of cones, rendered using isoparametric curves.

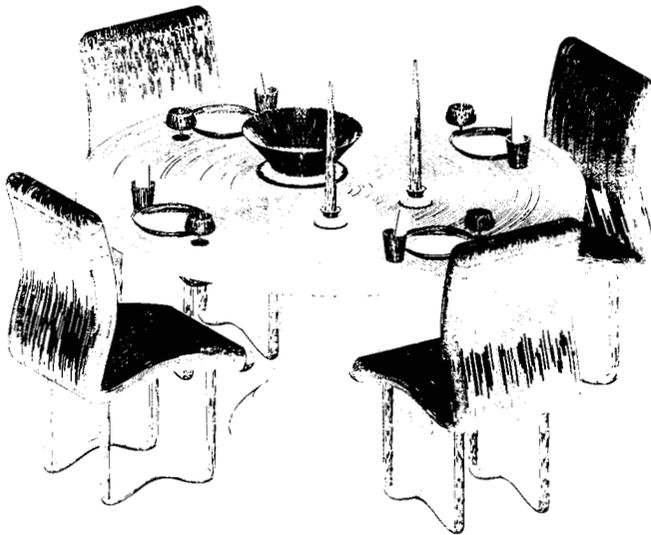


Fig. 11. A scene of a dinner table, rendered using isoparametric curves.



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

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 theirparametric domain (or thickness) to vary randomly. Fig. 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 [8] package, can im-

prove the hand drawn sensation of these illustrations and sketches.

ACKNOWLEDGMENTS

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

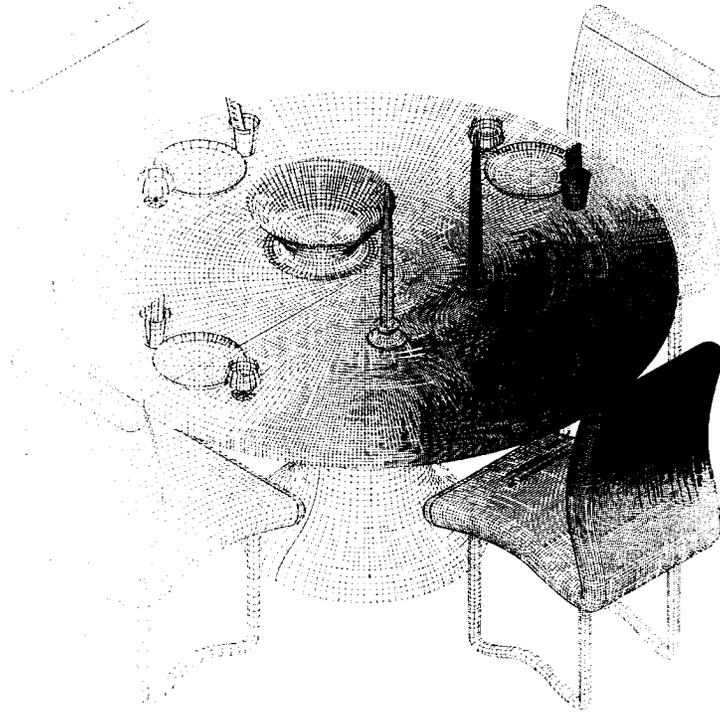


Fig. 13. A scene of a dinner table, rendered with a single point light source illuminating the scene from the upper right. Herein, dark spots represents the highlight. This image is also using cross hatching of both u and v isoparametric curves.

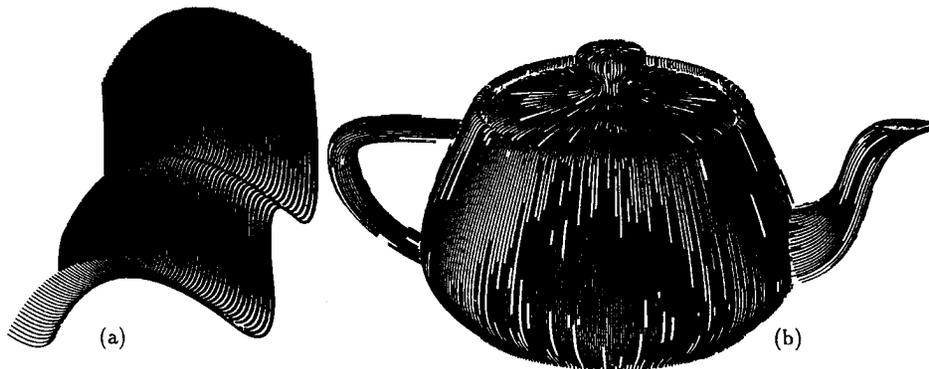


Fig. 14. One can attempt at rendering the scene using a continuous change of the width of the lines instead of their density.

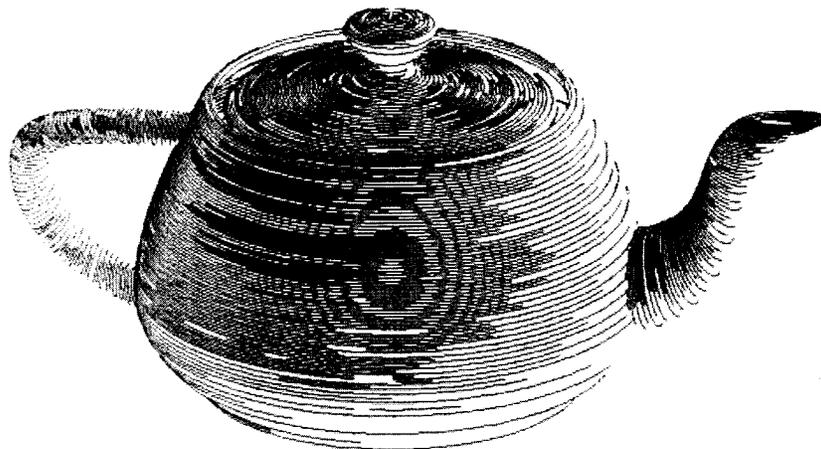


Fig. 15. A rendered image of the Utah teapot using a hardware Z-buffer. The discrete nature of the result adds high resolution noise to the image that is appealing.

REFERENCES

- [1] M.D. Carmo, "Differential geometry of curves and surfaces," Prentice Hall, 1976.
- [2] D. Dooley and M.F. Cohen, "Automatic illustration of 3D geometric models: Lines," *Computer Graphics*, vol. 24, no. 2, pp. 77-82, Mar. 1990.
- [3] G. Elber and E. Cohen, "Adaptive iso-curves based rendering for free form surfaces," Tech. Report UUCS-92-040, Dept. of Computer Science, Univ. of Utah. Submitted for publication.
- [4] G. Elber and E. Cohen, "Tool path generation for freeform surface models," *Computer Aided Design*, vol. 26, no. 6, June 1994.
- [5] J.D. Foley et al., *Computer Graphics, Principles and Practice*. Systems Programming Series, 2nd ed., Addison-Wesley, July 1990.
- [6] P.N. Hasluck, *Manual of Traditional Wood Carving*. New York: Dover Publications, 1977.
- [7] D. Hilbert and S. Cohn-Vossen, *Geometry and the Imagination*. 2nd ed., New York: Chelsea Publishing, 1990.
- [8] *Inventor 1.0. IRIS Inventor Programming Guide*. vol. I., SGI Document No. 007-1631-010, Mountain View, Calif.: Silicon Graphics, June 1992.
- [9] W.M. Ivins, *How Prints Look, Photographs with Commentary*. rev. ed., Boston: Beacon Press, 1987.
- [10] W.M. Ivins, *Prints and Visual Communication*. 8th printing, MIT Press, 1992.
- [11] IRIT, *Irit 5.0 User's Manual*. Technion, Israel, Mar. 1995.
- [12] M. Marsden and I.J. Schoenberg, "On variation diminishing spline approximation methods," *Mathematica*, vol. 8, no. 31, no. 1, pp. 61-82, 1966.
- [13] D.R. Peachey, "Solid texturing of complex surfaces," *Computer Graphics, SIGGRAPH '85*, vol. 19, no. 3, pp. 279-286, 1985.
- [14] K. Perlin, "An image synthesizer," *Computer Graphics, SIGGRAPH '85*, vol. 19, no. 3, pp. 287-296, July 1985.
- [15] Y.B. Pnueli and A.M. Bruckstein, "Gridless halftoning: A reincarnation of the old method," CIS Report No. 9323, Computer Science Dept., Technion, IIT, Oct. 1993.
- [16] 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 Dept., Technion, IIT, June 1992.
- [17] *PostScript Language Reference Manual*. Addison-Wesley, 1985.
- [18] L. Velho and J.M. Gomes, "Digital halftoning with space filling curves," *Computer Graphics, SIGGRAPH '91*, vol. 25, no. 4, pp. 81-90, July 1991.
- [19] G. Winkenbach and D.H. Salesin, "Computer-generated pen-and-ink illustration. *Computer Graphics*," *SIGGRAPH '94*, pp. 91-98, July 1994.



Gershon Elber is a lecturer in the Computer Science Department, Technion, Israel. His research interests spans computer-aided geometric designs and computer graphics. Dr. Elber received a BS degree in computer engineering and an MS in computer science from the Technion in 1986 and 1987, respectively, and a PhD in computer science from the University of Utah in 1992. He is a member of ACM and IEEE.