社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2007－CG－127　(4)
2007／7／9

形状の理解を容易にするラインレンダリング

アンドレ アレクシ　　齋藤 豪　　張 英夏　　中嶋 正之

東京工業大学 情報理工学研究科
{alexis, suguru, chang, nakajima}@img.cs.titech.ac.jp

形状把握はテクスチャや陰影といった様々な要因に影響されるため、正しい形状の理解を促すことのできるレンダリング手法は重要である。本稿では、我々は形状を単純な線で表現するレンダリング手法を新たに提案する。レンダリングされた物体上においてヒトが認識した法線と実際の法線を比較することにより、本手法の有効性を示す。本手法は特に、曲率の主方向を用いた線描画レンダリング法と比較し、正しい形状認識を促すことができる。

A Simple Line Generation Algorithm for Shape Recognition

Alexis Andre　Suguru Saito　Youngha Chang　Masayuki Nakajima

Graduate School of Information Science and Engineering, Tokyo Institute of Technology.
{alexis, suguru, chang, nakajima}@img.cs.titech.ac.jp

Shape perception depends on a lot of factors, for example texture or shade. When rendering a surface, the question whether the viewer correctly understands the shape of the rendered object is open. The use of different rendering methods can improve or harm the perception of the shape. This paper presents a new line rendering style and evaluates its success at conveying the shape of the underlying object with respect to other styles of rendering. An experiment, comparing the perceived normal vector with the true normal vector at some points chosen on the object, shows that the style we propose performs better than a style based on principal directions of curvature for some cases.

1 Introduction

Non-Photorealistic Rendering (NPR) methods can be separated into two main approaches. One is the simulation of traditional art techniques, and the other is new rendering techniques that offer more insight on the depicted object. Since Saito and Takahashi [14], various ways to represent three dimensional (3D) models as a two dimensional (2D) image have been proposed that perform better than simple projection when it comes to the problem of shape recognition. We propose here a really simple method to trace lines on any surface that convey the underlying shape in an efficient way.

2 Related Works

Efficient visualization methods are used in various scientific fields. From Saito and Takahashi [14], who set the basis for non-photorealistic rendering research, that is, computer generated technical illustrations, to the visualization of volumetric data with suggestive contours [1], various methods have been proposed to put emphasis on a feature of the underlying data.

A large part of previous visualization research focused on the representation of vector fields with lines. Such visualization techniques often used streamlines, to convey the direction of the vector fields. For example, Turk and Banks [16] proposed a technique to generate streamline images with controlled density. This paper deals with the evaluation study of efficient vector fields, or directly lines, to convey three-dimensional shape information.

For three dimensional data, for example mesh models, various methods to generate textures that convey information have been proposed in the past, for example the line integral convolution method [2], or also the works by Interrante et colleagues [8][5][9]. The latter works carried out a psychophysical evaluation of their texture patterns. Our evaluation method borrows from theirs.

Vision research literature also studied the perception of line drawings. Perkins [12] was among the first to study the perception of orthogonal lines, when they form the corner of a cube, a case that often occurs in CAD. Stevens [15] also noted that two crossing lines appear perpendicular in the absence of other visual clues. He also noted that parallel lines in two dimensions are viewed as parallel lines in three dimensions as well. Hoffman [7] extended this rule to a whole set of laws that the human visual system seems to follow.

The closest work to this paper that we are aware of comes from Girshick et al. [4], where the authors argued that principal directions of curvature were a good vector field to convey the shape in line drawings.

As Stevens [15] and Mamassian and Landy [11] showed, principal directions of curvature are strong hints on the underlying shape. These directions are often used in non-photorealistic rendering systems, for example suggestive contours [3], pen-and-ink illustration [17], and many others [6].

However, if the principal directions of curvature are efficient to convey the shape, we found that at least one other form of rendering may perform better at the same task, for particular surfaces.

3 Proposed Rendering Style

3.1 Inspiration

One common way of drawing a convex part of a surface is shown on the Figure 1, with lines that go away from each other and gather on the other side of the convex part. The rate at which the strokes differ from their immediate neighbour represents the slope. We tried to simulate such strokes with our method.

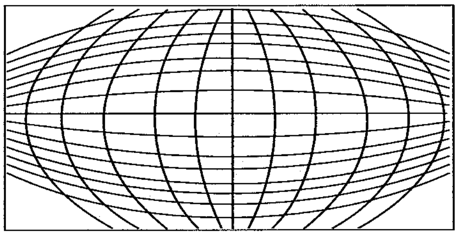

Figure 1: Common way to depict a convex surface, viewed from above.

3.2 Algorithm

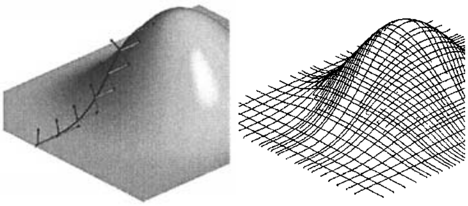

Figure 2: The axis direction (light gray) is projected on the surface along the normal direction at this point (dark gray). Resulting line drawing on the right.

We present here our rendering style. This line drawing consists of horizontal and vertical lines projected along the normal vector to the surface at this point. Figures 2 presents an example of this rendering method. Resulting lines seem to follow the shape, taking turns where the shape is round, but are still keeping a global direction. Perception in this case might come from the difference between neighbor lines. From now on, we will use the term "follow the shape" to describe this rendering method.

We start from seed points uniformly taken from the whole surface, and we draw strokes in the direction of the projection of axis vectors ($u = (1, 0, 0)$ and $v = (0, 1, 0)$ for example) along the normal vector to the surface at the current point. If P_i represents the i-th point of the stroke, and $n(P_i)$ the normal at this point, and a the axis vector, we have:

$$P_{i+1} = P_i + a - \alpha n(P_i), \qquad (1)$$

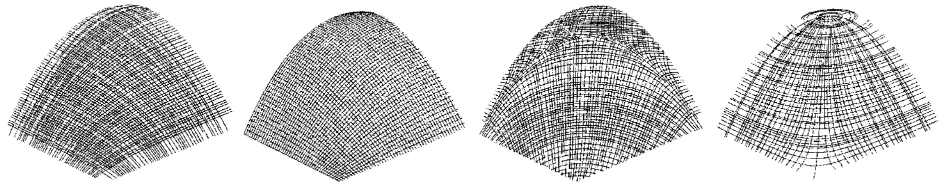

Figure 3: The four styles of rendering we compare in this paper. From the left: Follow-shape, Grid, Lines of curvature and Iso-level lines.

and α is chosen such that P_{i+1} lies on the surface. The stroke continues on the surface while there is no other stroke in the same direction nearby: we allow strokes to cross, but not to join. This is a simple density control that prevents the creation of overloaded drawings, when the shape presents strong curvature, where lines tend to gather.

3.3 Remarks

This rendering style is also easy to implement on various kinds of input data, as only the normal vector is required. For triangular meshes, the literature provides lots of various algorithms to estimate the normal vector from the face information, for example [13].

This style presents a strong grid structure, however, as the lines go *along* the shape, when viewed from above, there is still some information about the surface shape available to the viewer. Moreover, such lines can be drawn by people with no mathematical background. This is not directly linked to the topic of this paper, however, this may play a role in the recognition process if the viewer is used to draw such lines.

4 Other Styles

We present here the styles we use for the evaluation of our method.

4.1 Description

Grid: This simple style is achieved by projecting a square grid (of axis u and v, here the x and y axis) onto the surface along the z-direction. See Figure 3 (b).

Lines of curvature: We calculate the curvature at every point on the surface and the directions for which the change of curvature is extremum are called the principal directions of curvature. Each line follows that direction on the surface. See Figure 3 (c).

Iso-level lines: The contour lines rendering style is the combination of iso-level lines, where the height z is fixed, with lines of greater descent, where the variation along the z direction is maximum. See Figure 3 (d).

4.2 Remarks

We chose the previously described styles as it seemed to us that they represent common grounds for comparison, each style being heavily used for visualization. We tried to choose styles with strong perpendicularity in order to keep as close as possible the various styles. One might argue that the spacing of the lines has a strong influence on the perception of the shape. While it is possible to control the spacing for the grid and follow-shape style, the nature of the remaining two styles makes it impossible. We tried, however, to put all the styles at the same level, by making all factors that we could control the same.

5 Psychophysical Experiment

5.1 Description and Stimuli Generation

In order to evaluate how well the previously described rendering systems convey the shape, we performed an experiment in which users had to orientate a probe put on the surface in the direction of the perceived normal vector to the surface at this point.

We generated two smooth surfaces (the one on the Figure 3 and a concave one) using B-spline surfaces, with a grid of 3x3 control points uniformly distributed on the xy plane. Each control point on the inside of the surface was given a z value, to generate the two shapes. We chose to use one concave and one convex shapes, as the domain of the orientation of the normal vector for these shapes covers a wide range with respect to the viewpoint (from close to the viewpoint to more than a right angle for the concave shape). However, these shapes present different profiles when rendered with the principal directions of curvature.

Each surface is then fed to the four rendering algorithms. As we know the parametrization of the surface (from the definition of B-spline surfaces), both normal vector and curvature can be calculated at any point on the surface. This allows us to generate the lines with great precision.

Each algorithm includes a simple density control mechanism. As all the algorithms generate two sets of strokes that are mostly perpendicular to each other, we did not allow two strokes from the same set of lines to cross. For example, if two lines following the first principal direction of curvature are pointing in one direction, one line stops near the intersection. This simple method yields visually pleasant results. This was carried out in order to produce drawings with a similar number of lines, and with similar information. For example, the number of non-white pixels in the renderings of the shape of the Figure 3 are $n_{follow} = 26.4\%, n_{grid} = 26.9\%, n_{curvature} = 24.8\%, n_{level} = 20.9\%$, respectively.

One important point of this study is the absence of any shading. We focus on the influence of the line direction only. We believe that the shading information, as seen in Figure 4, cannot help understand the shape to the level of precision we aim at. Previous research confirms that visual perception of shape from shading is limited in absence of strong visual clues such as contours [10].

All the surfaces were viewed from the same point of view, slightly from above, in an oblique direction, as the Figure 3 shows. The viewpoint was chosen to allow good recognition in the four types of rendering.

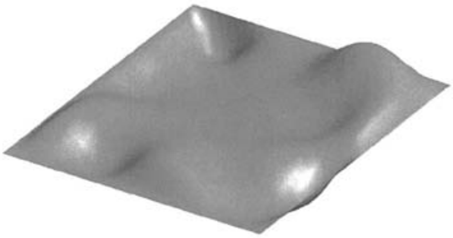

Figure 4: The surface used for training.

5.2 Experimental Setup

We describe here the experiment we carried out to evaluate the various rendering methods, inspired from [9]. The evaluation study is performed on a standard computer. The visual stimuli are displayed on the screen and the participant rotates a probe, made of two concentric discs and a short line to represent the normal, by dragging the mouse on the screen until the user believes the oriented probe matches the true normal vector. See Figure 5.

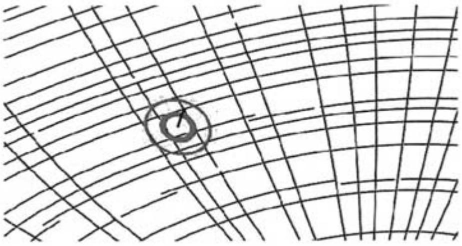

Figure 5: The probe used in the experiment.

Instructions were written, and read aloud in front of the training system to the volunteer. Each participant had to pass a training test before taking the experiment, on a shaded surface, in order to get used to the system. Figure 4 shows the surface used during the training process. We asked the user to reach a mean of twenty-five degrees of precision for twenty-five probe locations to pass.

Once the subject has passed the test, each participant is presented with the two surfaces, in the four styles successively, in a random order, but we made sure that the same surface did not appear twice in a row. Each participant has to orient nine probes, uniformly distributed on the surface. As a result, each subject orientates 342 probes (9 probes times 2 surfaces times 4 styles times 6 trials).

5.3 Results

The subjects understood the task correctly, in a short time. On five participants, all went through the training in one trial. The participants spent from 46 minutes to 102 minutes to orient the whole set of 342 probes, and small breaks were allowed. All participants had standard, or corrected to standard visual acuity.

The comments from the participants were encouraging, all of them showed interest in the study. They thought in majority that the task was difficult, and that the process was tiring. Most of them, despite the fact the instructions clearly stated that they will have to deal with four different types of renderings, missed the difference between the grid and follow-shape types. The different natures of the rendering algorithms were not specified in the instruction. When asked about it, four out of the five participants believed there was two types, one with level lines, and one with the grid. The same subject as the previous comment guessed that the principal directions of curvature were used in one type. All subjects thought that the grid and the follow-shape (according to the previous remark) types was the "easiest to deal with". One subject went even further, saying that for the grid type, local clues were enough to understand the shape, while for the curvature-driven method or the level lines, he needed to take a moment to grasp the whole surface before looking at the current probe. From a statistical viewpoint, considering only the mean, the style we propose performs best, followed by the isolevel and grid lines, and finally the directions of curvature. See Table 1 for detailed values and Figure 6.

(degrees)	Grid	Follow	Curv.	Level
Mean	15.63	13.18	22.65	14.97
Median	14.06	11.54	22.58	14.22
Std. Dev.	9.77	7.77	13.52	8.66

Table 1: Statistics of the error for the four tested styles.

We performed a simple ANOVA (analysis of variance) to evaluate the significance of the results. Among all variables, the p-value tells us that the styles were the most important factor, followed by the user and the probe location. Using Tukey's HSD (Honestly Significant Difference) method for more detailed analysis, we only found that the *follow-shape* style performs better than the *curvature* in 95% significance level. Other styles were not separated enough to draw conclusions, but we discuss this point on the following section.

variable	dF	F	p-value
Style	3	281	1.05×10^{-149}
User	4	105	1.01×10^{-80}
Probe	8	22	2.22×10^{-32}

Table 2: ANOVA Results.

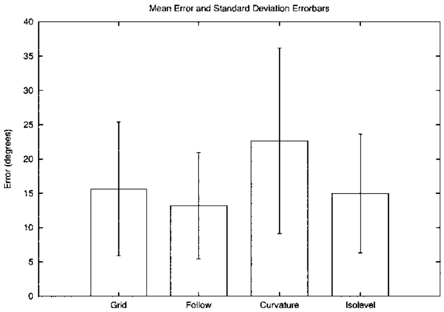

Figure 6: Mean values of each style's error and corresponding error bars.

6 Discussion

While we cannot extract from the data that our style performs better than the grid style, we still believe that our style performs better as the variation of the lines is not viewpoint dependant. The grid style, viewed from above, presents a perfect grid, from which no information about the shape can be extracted. The same can be stated about the contour lines rendering style, viewed from the side.

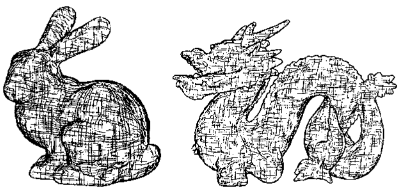

Figure 7: Models rendered with the proposed method. The Stanford Bunny and the left, and the Dragon model on the right.

Principal directions of curvature have been used as the vector field for many applications, notably non-photorealistic rendering. One example is for cross-hatching [6], but for that purpose, the raw directions were not used, as they present patterns with unneeded complexity. A smoother vector field, a mixture of geodesics and principal directions of curvature was used instead. We believe that our method could generate nice hatching patterns for cross-hatching, while being computationally efficient.

Since our method is able to convey the shape efficiently, we present in Figure 7 models depicted with this line rendering technique, with other NPR effects, such as alpha fading and flaring, and silhouette.

7 Conclusion

We presented a new line generation algorithm on a 3D surface that only requires the normal vector information, and that conveys the shape better than the previously established principal directions of curvature, at least in some cases. We hope to investigate in the next future various methods to determine the two axis that give the most information about the shape, for example by letting the user specify the directions on parts on the model.

References

[1] Michael Burns, Janek Klawe, Szymon Rusinkiewicz, Adam Finkelstein, and Doug DeCarlo. Line drawings from volume data. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 512–518, New York, NY, USA, 2005. ACM Press.

[2] Brian Cabral and Leith Casey Leedom. Imaging vector fields using line integral convolution. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 263–270, New York, NY, USA, 1993. ACM Press.

[3] Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. Suggestive contours for conveying shape. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 848–855, New York, NY, USA, 2003. ACM Press.

[4] A. Girshick, V. Interrante, S. Haker, and T. Lemoine. Line direction matters: an argument for the use of principal directions in 3d line drawings. *First International Symposium on Non Photorealistic Animation and Rendering (NPAR)*, 2000.

[5] Gabriele Gorla, Victoria Interrante, and Guillermo Sapiro. Texture synthesis for 3d shape representation. *IEEE Transactions on Visualization and Computer Graphics*, 9(4):512–524, 2003.

[6] Aaron Hertzmann and Denis Zorin. Illustrating smooth surfaces. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 517–526, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[7] Donald D. Hoffman. *Visual Intelligence: How We Create What We See.* W. W. Norton & Company, February 2000.

[8] V. Interrante and S. Kim. Investigating the effect of texture orientation on the perception of 3D shape. In Bernice Rogowitz and Thrasyvoulos Pappas, editors, *Proc. SPIE Human Vision and Electronic Imaging VI*, volume 4299, pages 330–339, June 2001.

[9] Sunghee Kim, Haleh Hagh-Shenas, and Victoria Interrante. Conveying three-dimensional shape with texture. In *APGV '04: Proceedings of the 1st Symposium on Applied perception in graphics and visualization*, pages 119–122, New York, NY, USA, 2004. ACM Press.

[10] Pascal Mamassian and D. Kersten. Illumination, shading and perception of local orientation. *Vision Research*, 36:2351–2367, 1996.

[11] Pascal Mamassian and Michael S. Landy. Observer biases in the 3d interpretation of line drawings. *Vision Research*, 38:2817–2832, 1998.

[12] David Perkins. Cubic corners. *Quarterly Progress Report*, (89):MIT Research Laboratory of Electronics, 207–214, 1968.

[13] Szymon Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. *Symposium on 3D Data Processing, Visualization, and Transmission*, 2004.

[14] Takafumi Saito and Tokiichiro Takahashi. Comprehensible rendering of 3-d shapes. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 197–206. ACM Press, 1990.

[15] Kent Stevens. Implementation of a theory for inferring surface shape from contours. *A.I. Memo*, (676):MIT Artificial Intelligence Laboratory, 1982.

[16] Greg Turk and David Banks. Image-guided streamline placement. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 453–460, New York, NY, USA, 1996. ACM Press.

[17] Georges Winkenbach and David H. Salesin. Computer-generated pen-and-ink illustration. *Computer Graphics*, 28(Annual Conference Series):91–100, 1994.