3차원 폴리곤 모델로 부터의 라인-드로윙 알고리즘 연구

김수균*, 최재연**, 이찬섭***

요약

컴퓨터 그래픽스 분야 이외에 라인 드로윙 기법은 패턴 인식 (의학적 진단, 문장 처리, 신약 발견과 언어 인식), 머신 러닝, 데이터 압축 등에 널리 쓰인다. 일반적으로 라인 드로윙 알고리즘은 단순화된 형태로 입력 데이터를 변환하는 처리 과정이며, 가능한 상대적으로 많은 정보를 유지해야 된다. 컴퓨터 그래픽스에서의 라인 드로윙은 비사실적인 랜더링 분야에서 넓게 쓰일 뿐만 아니라, 모델 재설계에도 쓰이고 있다. 본 논문에서는 선형적인 특성을 보여주는 다음과 같은 기법 - 윤곽선, 리지 및 밸리 선, 분명한 리지와 하이라이트 윤곽선-을 소개한다.

The Research of Line-Drawing Algorithm from 3D Polygonal Meshes

Soo-Kyun Kim*, Jae-Yeon Choi**, Chan-Seob Lee***

ABSTRACT

Outside of computer graphics field the line drawing is widely used in pattern recognition (medical diagnosis, text processing, drug discovery, and speech recognition), machine learning, data compression and so on. Generally line drawing is a process of transformation of an input data into a reduced representation, but keeping as much relevant information as possible. line drawing in computer graphics has wide application in non-photorealistic rendering to create artistic approach, but also can be used for model reconstruction. We will provide slight description of such linear feature, like contours, ridge and valley lines, suggestive contours, apparent ridges and highlight contours.

Key Words : line-drawing, contour, image space, mesh processing, 3D polygonal meshes

* 배재대학교 게임공학과
** 남서울대학교 정보통신공학과
*** 혜천대학교 물류유통정보과
· 제1저자(First Author) : 김수균 · 교신저자(Correspondent Author) : 김수균
· 접수일(2010년 3월 17일), 수정일(1차 : 2010년 4월 19일), 게재확정일(2010년 4월 26일)

Ⅰ. Introduction

Generally, feature extraction is a process of transformation of an input data into a reduced representation, but keeping as much relevant information as possible. Feature extraction in computer graphics has wide application in non-photorealistic rendering to create artistic like line drawing, but also can be used for model reconstruction [1].

Feature extraction is special form of dimensionality reduction. Outside of computer graphics field the feature extraction is widely used in pattern recognition (medical diagnosis, text processing, drug discovery, and speech recognition), machine learning, data compression and so on.

Recently line drawing approaches are aimed to convey geometric information from 3D polygonal meshes [2], while others want to extract lines, which will imitate artistic drawing (Figure 1).

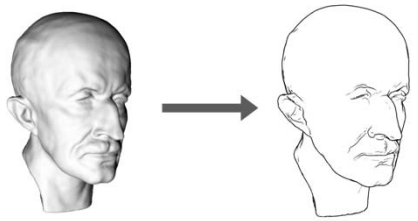

그림 1. 막스 플랑크 모델. (a) 쉐이드, (b) 라인 드로윙 [3]
Fig. 1. Max Plank model. (a) Shaded, (b) Line Drawing [3]

In this survey we provide rough classification of existing methods into two categories: object-space and image space approaches. The former approaches directly process the polygonal mesh data, while the latter process image, produced by projection of the mesh onto a view plane. First we will concentrate on method descriptions, and in the end we will provide with comparison of the methods.

Ⅱ. Direct Mesh Processing Methods

In this section we present methods that directly process discrete 3D polygonal mesh data without any fitting. Since mesh data has only C^0 continuity, it is impossible to directly estimate curvature. For this purpose standard technique introduced in[4] is used. This section contains description of such linear features: suggestive contours, apparent ridges, high light lines.

그림 2. 윤곽선과 서제스티브 윤곽의 혼합 [5]
Fig. 2. Combination of contour lines and suggestive contours [5]

2.1 Suggestive Contours

DeCarlo et al. in [5] introduced another type of features - suggestive contours (Figure 3).

Suggestive contours are lines drawn on clearly visible parts of the surface, where a true contour would first appear with a minimal change in viewpoint.

Intuitively, suggestive contour is a set of points, where contour would appear in nearby viewpoint (Figure 3). More formally, suggestive contour is an

inflection point between convex and concave regions on the model, point p on the figure 3.

From an image-space perspective suggestive contours are defined as valleys in intensity of shaded image. Authors also provide image-space algorithm that exploits this fact. They first produce smoothly shaded image by placing diffuse light source at the camera origin. Then image space algorithms are used to estimate valley from produced image.

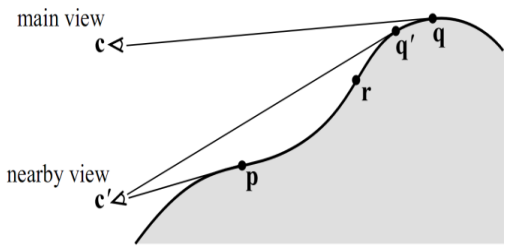

그림 3. 점 p에서의 서제스티브 윤곽 [5]
Fig. 3. Suggestive contour at point p. As the view point moves to c',a contour suddenly appears at p, while the contour at q'slides along the surface from q [5]

2.2 Apparent Ridges

Judd et al [3] introduced new type of linear features, called apparent ridges (Figure 1), which are, in contrast to ridge and valley lines, are view-dependent and provide more artistically pleasant results. The definition of apparent ridges is based on view-dependent curvature, which is defined in this paper as the variation of the surface normal with respect to a viewing screen plane. Apparent ridges are loci of points that maximize a view-dependent curvature.

Intuitively, view-dependent curvature is how much the surface is seen to bend from the viewpoint. It takes into account both the curvature of the object and the foreshortening due to surface orientation (Figure 4). At front facing parts of the object, the values of curvature and view-dependent curvature are similar. As the object normal turns away from the viewer, view-dependent curvature becomes much larger due to projection (Figure 4). Consequently view-dependent curvature approaches a maximum of infinity at the contours and so contours are extracted as apparent ridges.

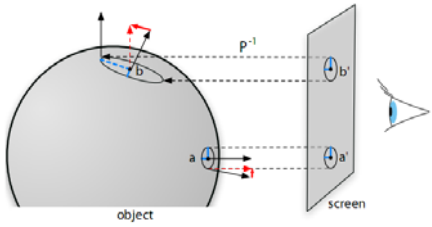

그림 4. 시점-종속적인 곡률 [7]
Fig. 4. View-dependent curvature. The maximal view-dependent curvature at b' is much larger than at a' because of projection [7]

2.3 Highlint Lines

Sometimes dark lines are not sufficient to convey geometry of shape. For example, when dark lines are conveying both convex and concave parts of the model, both lines are mixed and it is becoming hard to distinguish them. It would be more natural to draw convexity with lighter lines. DeCarlo et.al in [6] and, independently, Lee et al. [7] proposed algorithms to draw both dark and white lines against non-white background (Figure 11 and Figure 7).

The serious limitation of the suggestive contours is an inability to convey concave parts of the model.

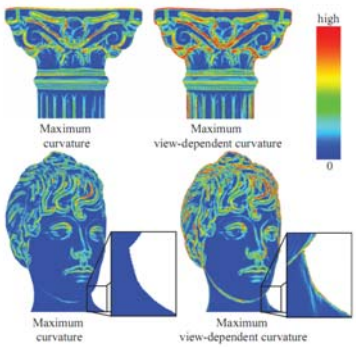

그림 5. 시점-종속 곡률과 곡률의 비교 [7]
Fig. 5. Comparison of curvature and view-dependent curvature [7].

그림 6. 데이빗 모델에서의 하이라이트 선들 [3]
Fig. 6. Highlight lines on David model [3]

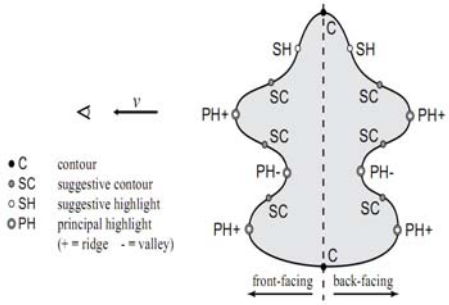

그림 7. 주어진 시점 방향과 다른 형태의 선들 [3]
Fig. 7. Different types of lines given the viewing direction v(for clarity this figure uses orthographic projection)[3]

Highlight lines introduced in [3] effectively improve suggestive contours and eliminate the limitation. Highlight lines are described by two types of lines: suggestive contours and principal highlights (Figure 8). Suggestive highlights appear at view-dependent inflection points, and can be seen as opposite of suggestive contours introduced in [2]. Principal highlights appear when the surface is viewed along a principal curvature direction.

III. Image Space Based Methods

Most of the papers written on linear feature extraction are based on object-space processing. We suppose that most of research underestimate capabilities of image-space based processing. Methods, described below, showed that such methods can yield comparable results and, moreover, can be processed using graphics processing unit (GPU) more efficiently then object-based approaches, since they can greatly utilize modern GPU features.

3.1 G-Buffers

In 1990 Saito and Takahashi [8] introduced method for comprehensible rendering of 3-D models using image processing algorithms. The main contribution of the paper is an idea of extracting geometry information of a 3-D model to special textures, called Geometric Buffers (G-Buffers). When every required data from a 3-D model is stored in these G-Buffers all further processing (edge enhancement, pattern recognition etc) is made on these G-Buffers. More over we can apply different image processing algorithms to process geometrical information of the scene. For example, in [8] edge enhancement was made by finding first and second order differentials from a depth image (Figure 10).

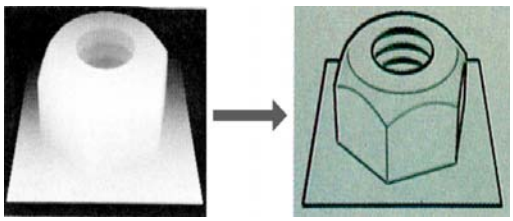

그림 8. 깊이 이미지와 에지 이미지의 결과 [8]
Fig. 8. Depth image (left) and resulting edge image (right) [8]

G-Buffer is not just a projection of the model into the view plane, as can be seen at first sight. We can pack into G-Buffers such data like world position, normal vectors, depth information, texture coordinate, etc. This ability generalizes G-Buffers to solve different kind of problems in computer graphics.

G-Buffers application is very popular in modern computer graphics. Computation of lighting that uses G-Buffers is called deferred shading, and widely used in modern computer games. Classic example is S.T.A.L.K.E.R created by GSC Game World [9] and Tabula Rasa by NCSoft [10]. Botsch et al. showed successful application of deferred shading in point-based rendering [11].

3.2 Abstract Shading

While highlight contours is just an improvement over suggestive contours to catch convex parts of the model, [7] introduces employment of GPU for feature line extraction. Usage of GPU is not very popular in the field of feature line extraction, but [7] showed that GPU can be effectively used for this task.

Lee et al. [7] were motivated by an observation that a line drawing can be understood as an abstraction of the shaded image. This makes sense: if some wants to convey shape effectively, but can only afford to draw a sparse set of lines, a good strategy might be to choose lines that convey the essential feature of the shading.

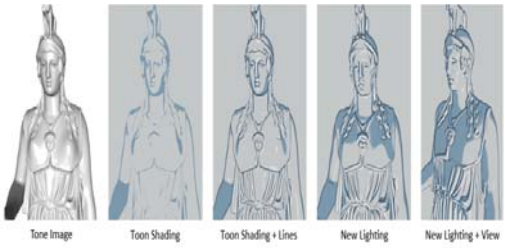

그림 9. 쉐이딩을 통한 라인 드로윙 [9]
Fig. 9. Line drawing via abstract shading [9]

They proposed following candidates for "essential features" are boundaries between dark and light regions, and thin areas of shading that are well approximated by lines. These lines can also be defined as ridges and valleys of illumination image.

Proposed algorithm consists of two steps or shading passes [12]: in the first pass, render a grayscale tone image describing how the scene is illuminated, blur it and save to texture memory; in the second pass, fragment shader on the GPU is used to render dark lines in thin areas of dark tone. White lines are produced by processing a specular component [13] of the lighting.

Ⅳ. Conclusion

Extracted linear features can give us a lot of information about the shape of the model. Ridge and valley lines [1] are view-independent lines that can be successfully used for shape reconstruction.

Because of view-independent nature, these lines look too blocky, and cannot be applied for artistic lines extraction.

In contrast to ridge and valley lines, suggestive contours [5] and apparent ridges [3] depend on the view position and were introduced to provide artistic feel of extracted lines. Apparent ridge extend contours along edges of the object, while suggestive contours extends onto the face of the object [3].

Because suggestive contours appear on inflection of model's convex and concave parts, they have problems when conveying model that does not contain concavity. Highlight lines [6] (suggestive highlights and principal highlights) successfully eliminate this problem and provide new types of features: suggestive highlights and principal highlights.

White lines that represent convexity combined with black features can give more information about the geometry of a shape [6, 7]. Methods based on GPU processing can successfully solve problems of feature extraction [7].

It is possible to apply image-space algorithms to process 3D models information, stored in G-Buffers [8]. G-Buffers can be efficiently processed on modern GPU technique and therefore they have wide application in modern computer graphics.

Reference

[1] Ohtake Y., Belyaev A. G., Seidel H.-P., Ridge-valley lines on meshes via implicit surface fitting. Proc. Of ACM SIGGRAPH (2004), 609 - 612.

[2] Ohtake Y., Belyaev A. G., Seidel H.-P., A Multi-scale Approach to 3D Scattered Data Interpolation with Compactly Supported Basis Functions, Proceedings of the Shape Modeling International 2003, p.153, May 12-15.

[3] Judd, T., Durand, F., and Adelson, E., Apparent Ridges for Line Drawing. ACM Transaction on Graphics (Proc. SIGGRAPH) 26, 3, Article 19. 2007

[4] Rusinkiewicz, S., Estimating Curvatures and Their Derivatives on Triangle Meshes. In Symposium on 3D Data Processing, Visualization, and Transmission. 2004

[5] DeCarlo, D., Finkelstein, A., Rusinkiewicz, S., and Santella, A., "*Suggestive contours for Conveying Shape*". ACM Transactions on Graphics 22, 3, 848-855.2003

[6] DeCarlo, D., Rusinkiewicz, S., Highlight lines for conveying shape, Proceedings of the 5th international symposium on Non-photorealistic animation and rendering, August 04-05, San Diego, California, 2007

[7] Lee, Y., Markosian, L., Lee, S., and Hughes J.F., Line drawing via abstract shading. ACM Transaction on Graphics 26, 3. 2007

[8] Saito, T., and Takahashi, T., Comprehensible Rendering of 3-D Shapes. In Computer Graphics (Proceedings of SIGGRAPH 90), vol. 24, 197-206. 1990

[9] Shishkovtsov, O., Deferred Shading in S.T.A.L.K.E.R., GPU Gems 2, Chapter 2. 2005

[11] Koonce, R., Deferred Shading in Tabula Rasa, GPU Gems 3, Chapter 19. 2007

[12] Fernando R., Kilgard M.J., The Cg Tutorial. The definitive guide to programmable real-time graphics. Addison-Wesley. 2005

[13] Woo, M., Neider, J., Davis T., OpenGL Architecture Review Board (Corporate Author), 1997, OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.1, 2nd edition.

[14] 송성도 외, "CUDA를 이용한 포인트 렌더링 알고리즘", *한국지식정보기술학회 논문지*, 제5권 제1호, pp.17-26, 2010.

Acknowledgement

"본 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No.2010-0015563)"

김수균(Soo-Kyun Kim)

2006년 고려대학교 컴퓨터학과
(이학박사)
2006.3 ~2008.2 삼성전자 통신연구소
책임연구원

2008년~현재 배재대학교 게임공학과 조교수
※ 관심분야 : 기하모델링, 게임그래픽, 실감미디어

이찬섭(Chan-Seob Lee)

2000년 한남대학교 컴퓨터공학과
(공학석사)
2003년 한남대학교 컴퓨터공학과
(공학박사)

2003년~현재 혜천대학 물류유통정보과 조교수
※ 관심분야 : 웹 DB, 유비쿼터스, 캐쉬

최재연(Jae-Yeon Choi)

1987년 한양대학교 전자통신공학과
(공학석사)
1998년 한양대학교 전자통신공학과
(공학박사)
삼성종합기술원, LG정보통신연구소
1996년~현재 남서울대학교 정보통신공학과 교수
남서울대학교 정보통신연구소
※ 관심분야 : MMIC, 안테나공학