

기술문서 작성을 위한 3 차원 CAD 데이터의 도해저작 알고리즘

심현수, 양상욱, 최영, 조성욱
중앙대학교 기계공학부

Automatic Generation of Explanatory 2D Vector Drawing from 3D CAD Data for Technical Documents

H. S. Shim, S. W. Yang, Y. Choi, S. W. Cho
Department of Mechanical Engineering, Chung-Ang University

ABSTRACT

Three dimensional shaded images are standard visualization method for CAD models on the computer screen. Therefore, much of the effort in the visualization of CAD models has been focused on how conveniently and realistically CAD models can be displayed on the screen. However, shaded 3D CAD data images captured from the screen may not be suitable for some application areas. Technical document, either in the paper or electronic form, can more clearly describe the shape and annotate parts of the model by using projected 2D line drawing format viewed from a user defined view direction. This paper describes an efficient method for generating such a 2D line drawing data in the vector format. The algorithm is composed of silhouette line detection, hidden line removal and cleaning processes.

Key Words : Hidden Line Removal(은선 제거), Silhouette Line Detection(윤곽선 검출), Explanatory Drawing Algorithm(도해저작 알고리즘), Technical Document(기술문서)

1. 서론

컴퓨터 그래픽 기술이 빠르게 발전하고 이를 활용하는 CAD 소프트웨어 역시 하루가 다르게 발전하고 있다. 또한 모델을 사실적으로 보이게 하는 기술이나 세밀한 표현을 가능하게 하는 등의 그래픽 기술도 빠르게 발전하고 있다. 그러나 도면을 이해하고 다루는 사용자의 입장에서는 이러한 그래픽 요소가 도면의 이해에 부담이 되는 경우가 있다. 또한 제품 모델에 대해서 기술교범이나 사용자 매뉴얼 등의 전자 출판에 바로 사용될 수 있는 도해를 만드는 것은 제품 데이터의 활용도 측면과 제반 생산성 향상을 위해서 필요하다.

이러한 요구에 맞춰 대부분의 CAD 소프트웨어는 와이어프레임 및 은선이 제거된 선 보기 기능을 제공한다. 하지만 이러한 기능을 구현함에 있어서 빠른 결과를 얻기 위하여, 그래픽 라이브러리의 Z 버퍼를 이용하여 그리는 순서를 조절하거나 스텐실 버퍼를 이용하여 출력 이미지를 가공하는 방법을

통해서 구현하면, 결과를 벡터 형식의 데이터로 출력할 수 없다. CAD 소프트웨어에서 사용되는 외곽선 보기가 어떠한 형식으로 구현되는지는 그 소프트웨어 내부적인 부분이므로 알 수 없으나, 알고리즘의 구현 결과를 2D CAD 데이터로 출력하는 기능이 없는 것에서 단순히 보여주기만을 지원하는 것을 유추해 낼 수 있다.

이에 반하여, 본 연구에서는 화면의 출력결과나 그래픽 라이브러리를 이용하는 방법을 사용하지 않고 기하모델 및 위상 정보를 기본으로 하여 윤곽선 검출 및 은선 제거를 수행하여 결과를 2D 벡터 데이터로 저장할 수 있는 알고리즘을 개발하고자 한다.

2. 도해저작 알고리즘

도해를 위한 외곽선 데이터를 생성하기 위해서는 기본적으로 도면 데이터에 없는 선을 찾는 윤곽선 검출(Silhouette Line Detection) 알고리즘과 보이기

않는 선이나 선의 일부를 제거하는 은선 제거 (Hidden Line Removal) 알고리즘이 필요하다.

2.1 윤곽선 검출 알고리즘

윤곽선이란 화면상에서 모델을 일정 시점에서 볼 때 형상의 윤곽이 되는 선으로서, 기본 데이터 모델은 이 정보를 가지고 있지 않다. 예를 들어 Fig. 1의 맨 왼쪽 그림과 같은 원통형의 모델 정보에는 오른쪽 그림에서 굵은 선으로 표시된 선들의 데이터는 존재하지 않는다. 예로 사용한 원통과 같은 간단한 형상의 경우가 아닌 현업에서 사용하는 복잡한 자유 곡면 모델인 경우에는 원 모델 정보에 포함되지 않는 선들이 윤곽선으로 보여야 하는 경우가 많다. 이러한 계산 과정을 통칭하여 윤곽선 검출이라 한다.

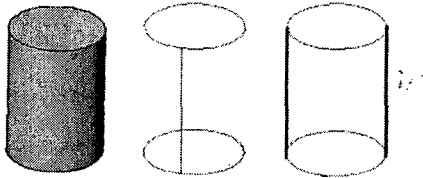


Fig. 1 Silhouette Line Detection

2.1.1 기존연구내용

윤곽선을 찾기 위해 기본적으로 사용되는 방법은 관찰자 위치를 나타내는 시점과 각 면의 법선 벡터 간의 연산을 통해서 면의 앞과 뒤를 판별한 후, 앞 뒤의 경계가 되는 선들을 통해 윤곽선을 찾아내는 것이다. 대부분의 윤곽선 검출 알고리즘은 이 방법을 기본 개념으로 사용한다.

Buchanan 과 Sousa 는 위의 기본 개념에 Edge Buffer²를 도입하여 데이터 구조를 정의하고 연산 방법을 정의하였다. 이 데이터 구조는 명확하게 계산과정 및 결과를 정리할 수 있도록 설계 되었지만 솔리드가 평면으로만 구성되어야 하고 솔리드를 구성하는 선들의 관계를 명확하게 가지고 있어야만 한다.

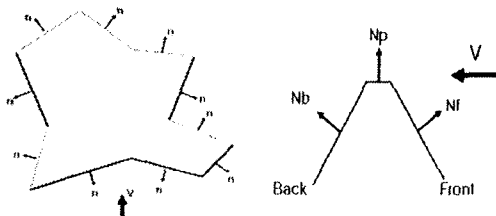


Fig. 2 Front/Back Facing Property of a Polygon^{1,2}

2.1.2 윤곽선 검출

보통 3D 모델을 화면에 표현하기 위해서는 곡면

정보를 통해 각 면을 삼각화하는 계산이 수행된다. 또한 최근 많은 연구가 진행되고 있는 웹을 기반으로 하는 여러 협력설계 및 Digital Mockup 을 지원하는 어플리케이션의 경우에는 이러한 삼각화 된 데이터 구조를 직접 가지고 있기도 하다. 본 연구에서는 이 삼각형 데이터를 이용해 윤곽선을 검출한다. 기본적으로는 2.1.1 절에서 소개한 Edge Buffer 에서 사용한 방법을 변형하여 각 곡면에서 생성된 삼각화 데이터를 이용한다. 각각의 삼각형 데이터에서 세 꼭지점의 좌표를 이용하면 Fig. 2의 왼쪽 그림처럼 면이 바라보는 법선 벡터 n 을 구할 수 있다. 이 법선 벡터와 현재 시점을 바라보고 있는 관찰자의 뷰잉 벡터 V 의 내적을 취한 값이 음수가 된다면 현재 삼각형은 Fig. 2의 오른쪽 그림의 관찰자의 입장에서 볼 수 있는 가능성을 가진 면이다. 이와 반대로 내적의 값이 양수가 된다면, 이 삼각형은 현 관찰자의 시점에서는 볼 수 없는 면이 된다. 이러한 성질을 이용하여 모든 데이터에 대해서 연산을 수행한다면, 각 삼각형 데이터의 선들의 인접정보를 통하여 계산된 내적 값이 0 또는 음수에서 양수로 바뀌는 윤곽선을 찾을 수 있다.

위의 모든 과정을 모두 수행하여 얻어지는 선들에 의해서 Fig. 3의 중간과 같은 결과를 얻는다.

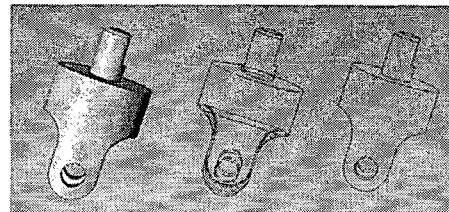


Fig. 3 Example of Silhouette Line Detection

2.2 은선 제거 알고리즘

윤곽선 검출 알고리즘이 적용된 모델 데이터는 화면상에서 모델의 일정 시점에서 보일 가능성이 있는 원 모델의 데이터와 윤곽선 데이터를 가지고 있다. 이 원 모델 데이터 및 윤곽선 데이터를 기반으로 보이지 않는 선을 지우고, 중간지점에서 삭제되어야 하는 선들을 편집하는 일련의 과정을 은선 제거(Hidden Line Removal) 알고리즘이라 한다.

2.2.1 기존연구내용

은선 제거 알고리즘은 크게 그래픽적인 효과를 위한 이미지(Image Space)의 관점과 삼차원 모델 데이터를 다루기 위한 객체(Object Space)의 관점의 두 가지 방향으로 연구가 진행 되었으며, 현재 이미지 관점의 연구는 많이 진행되어 카툰렌더링 등 게임 분야 및 이미지를 가공하는 분야에서 다양하게 활

용되고 있다. 반면 삼차원 모델 데이터를 이용하여 벡터 데이터로 만들어 내는 방법은 이미지 관점의 연구에 비해서 발전이 더딘 것이 현 실정이다.

이미지 분야에서 주로 사용하는 방법 중 하나는 Fig. 4(a) 와 같이 모델 데이터를 공간의 복잡성에 따라 분할하여 계산하는 Space Subdivision³ 알고리즘이다. 또는 Fig. 4(b) 와 같이 임의의 평면을 기준으로 순차적으로 각 픽셀을 체크하는 Z-Buffer⁴ 알고리즘이 사용되기도 한다. 이 방법들의 문제점은 기본적 데이터 구조가 각 알고리즘을 지원하여야 한다는 한계와 알고리즘을 거친 후 생성되는 데이터는 정확도 측면에서 객체 모델 데이터로 표현하기 힘들다. 따라서 본 연구의 목표인 제어가 가능한 벡터로서 표현되는 형상 데이터를 만들기 힘들다.

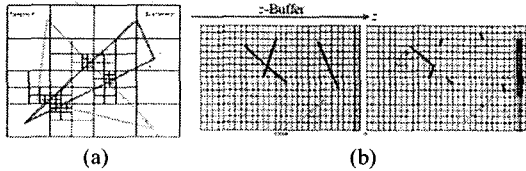


Fig. 4 Hidden Line Removal in Image Space

객체 기반 알고리즘에는 Fig. 4 (a)에 표현된 것처럼 객체의 형상 데이터를 이용하여 평면을 지나가는 선들이 가려지는 횟수를 검사하여 은선을 제거하는 Quantitative Invisibility⁵ 알고리즘이 있다. 이 알고리즘은 가장 초기에 발표되었지만, 최근에도 가장 기본적인 참고문헌으로 사용되고 있는 알고리즘이다. Fig. 4 (b)에 표현된 그림은 폴리곤의 각 선들에 해당 선과 인접하는 면들의 정보 및 Artistic Edge² 여부를 5 비트의 버퍼를 사용하여 계산하는 방법으로, Edge Buffer² 알고리즘은 이 버퍼를 사용하여 은선 제거를 시행한다. 하지만 기본적인 데이터 구조가 제한적이고, 후처리 과정에서 선들의 교차(Intersection)를 계산하는 등의 작업이 추가로 필요하다.

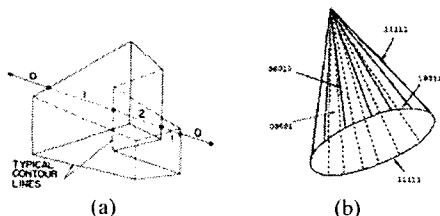


Fig. 5 Hidden Line Removal in Object Space

2.2.2 은선 제거

본 연구에서는 2.2.1 절에서 언급한 방법들 중 객체 관점의 방법만이 아니라, 효과적인 알고리즘을

위하여 이미지 관점의 접근방법을 동시에 사용한다

윤곽선 검출의 과정에서 언급한 것처럼, 3D CAD 데이터는 삼각화된 데이터를 통하여 표현된다 또한 윤곽선 검출의 결과로서 관찰자의 시점에서 표현될 가능성이 있는 선과 함께, 2.2.2 절에서 관찰자의 시점 벡터와 삼각형 데이터의 법선 벡터의 내적이 음수가 되는 삼각형 데이터만이 현재 보여질 선들을 가릴 가능성이 있으므로 연산 횟수가 줄어드는 것을 알 수 있다.

또한 이렇게 범위가 줄어드는 삼각형 데이터 및 선 데이터를 Z-Buffer 알고리즘에서 사용된 방법을 이용, 한 면에 투영을 시킨 후 생긴 2 차원 모델의 인접 및 교차 여부 등을 조사하면 각 면이 보이지 않게 할 가능성이 있는 선을 알아낼 수 있으므로 연산의 횟수를 더 줄일 수 있다.

위의 과정을 통하여 교차점(Intersection)을 구하는 연산 횟수를 줄일 수 있으며, 삼각형 데이터와 윤곽선 검출과정에서 찾아진 선들의 기하학적 데이터를 서로 연산하면 Fig. 6의 4 가지의 교차하는 경우가 생길 수 있으며, 교차되는 형태에 따라 선을 분할 또는 수정하여 최종 완성된 도해 데이터를 만들 수 있다.

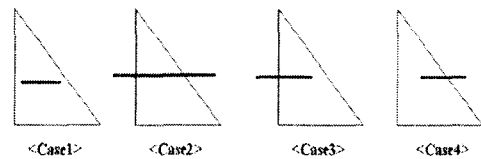


Fig. 6 Intersection Line & Triangles

3. 후처리 프로세스

2 절의 과정을 통해서 얻어진 모델 데이터는 실좌표 상에서 활용할 수 있는 벡터 데이터이다. 그러나 위의 과정을 모두 거친 데이터라 하더라도, 표현해야 할 면 데이터의 경계선이 Fig. 7의 면 1과 면 2 처럼 구분할 필요 없는 부분에서 존재할 수 있다. 본 연구에서는 이러한 필요 없는 선을 제거하기 위해 경계부근의 삼각형 데이터의 법선 벡터를 비교한다.

그리고 생성된 윤곽선이 기존의 선들과 중복될 수 있기 때문에, 각각의 면 단위로 삼각형 윤곽선을 검출하는 과정에서 면의 외곽선이 되는 부분의 데이터를 미리 검출하여, 선들이 중복되는 것을 피한다.

또한 윤곽선 검출과정에서 생성된 선 데이터들은 모두 삼각형 메시의 한 면이므로 화면상으로는 한 선으로 표현된다고 하더라도, 데이터의 구조는 각 선 세그먼트로 나누어져 표현된다. 따라서 생성

된 윤곽선들을 연결하거나 곡선 피팅을 통해 모델 데이터를 부드럽고 또한 간결하게 처리 한다.

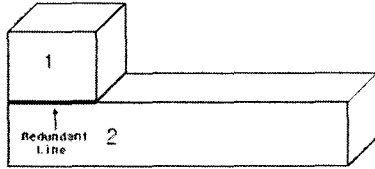


Fig. 7 An Example of Redundant Line

4. 결론

3D CAD 데이터의 전체 모델 정보가 아닌 기본적인 형상 데이터인 면과 면의 경계선을 이루는 선들, 그리고 가시화를 위한 삼각형 데이터를 이용하여 전체 모델의 외곽선을 2D 벡터 데이터 형태로 생성하는 연구를 수행하였다. 이 방법은 알고리즘에서 요구하는 모델 데이터가 복잡하거나 용량이 크지 않기 때문에, 협업설계를 지원하기 위한 가벼운 파일을 사용한 경우에도 같은 결과를 얻을 수 있다.

본 연구에서는 기존 연구의 여러 장점을 취하여 전체 및 부분적인 모델의 확대와 축소에도 정확한 표현을 할 수 있는 2D 벡터 데이터로 결과를 출력할 수 있다. 이러한 출력 데이터를 CGM 등의 컴퓨터 그래픽 메타파일로 활용하여 실질적으로 사용자 매뉴얼 등과 같은 기술 문서에 사용 할 수 있다.

후 기

본 연구는 산업자원부에서 추진하는 차세대신기술개발사업의 하나로 수행되고 있는 '글로벌 정보공유 및 지식기반의 차세대 생산시스템 개발' 과제의 지원을 받아 수행되었습니다.

참고문헌

1. S. Kumar, D. Manocha, B. Garrett, and M. Lin, 'Hierarchical back-face culling?' 7th Eurographics Workshop on Rendering, pp. 231-240, 1996.
2. J. W. Buchanan and M. C. Sousa, 'The edge buffer: A data structure for easy silhouette detection?' Proceedings of the First International Symposium on Non Photorealistic Animation and Rendering, 2000.
3. A. S. Glassner, 'Space Subdivision for Fast Ray Tracing?' IEEE CG&A Vol. 4, No. 10. pp. 15-22, Oct. 1984.
4. N. Greene, M. Kass, and G. Miller, 'Hierarchical Z-

buffer visibility?' Proceedings of the 20th annual conference on Computer graphics and interactive techniques, pp. 231-238, Sep. 1993.

5. A. Appel, 'The notion of quantitative invisibility and the machine rendering of solids?' Proceedings of ACM National Conference, pp. 387-393, 1967.
6. D. P. Dobkin and D. G. Kirkpatrick, 'Fast detection of polyhedral intersection?' Proceeding 9th International Colloq. Automata Lang. Programming, volume 140 of Lecture Notes in Computer Science, pp. 154-165, 1982.