

실시간 3축 NC 밀링 시뮬레이션을 위한 메쉬 간략화 방법

주성욱*, 이상현**, 박기현***

Mesh Decimation for Polygon Rendering Based Real-Time 3-Axis NC Milling Simulation

Joo, S. W.*, Lee, S. H.** and Park, K. H.***

ABSTRACT

The view dependency of typical spatial-partitioning based NC simulation methods is overcome by polygon rendering technique that generates polygons to represent the workpiece, thus enabling dynamic viewing transformations without reconstruction of the entire data structure. However, the polygon rendering technique still has difficulty in realizing real-time simulation due to unsatisfactory performance of current graphics devices. Therefore, it is necessary to develop a mesh decimation method that enables rapid rendering without loss of display quality. In this paper, we proposed a new mesh decimation algorithm for a workpiece whose shape varies dynamically. In this algorithm, the z-map data for a given workpiece is divided into several regions, and a triangular mesh is constructed for each region first. Then, if any region is cut by the tool, its mesh is regenerated and decimated again. Since the range of mesh decimation is confined to a few regions, the reduced polygons for rendering can be obtained rapidly. Our method enables the polygon-rendering based NC simulation to be applied to the computers equipped with a wider range of graphics cards.

Key words : NC Simulation, 3-Axis milling, Real-time simulation, Polygon rendering, Mesh decimation, Z-map

1. 서 론

CAD 모델로부터 NC 공구 경로를 자동 생성시키는 전통적인 CAM 연구와 더불어^{1)~4)}, 생성된 NC프로그램을 이용한 실제 가공에 앞서 허용 오차 범위를 넘는 과절삭이나 공구 간섭과 같은 NC 코드의 오류 여부를 컴퓨터상에서 시뮬레이션하는 연구들이 널리 수행되어 왔으며, 현재 대부분의 상용 시스템들은 공작물이 실제 가공되는 상황을 실시간으로 화면상에 디스플레이해 주는 기능을 구비하고 있다⁵⁾. 이를 위해 가공 중인 공작물의 형상을 표현하는 방법은 크게 솔리드 모델을 위한 자료 구조를 직접 이

용하는 방법과 z-map이나 면의 법선 벡터들과 같이 불연속적인 값들을 이용하여 표현하는 방법이 있다⁶⁾. 솔리드 표현 방식은 수직벽과 같은 형상을 포함하여 가공 중인 공작물의 형상을 보다 완벽하게 나타낼 수 있다는 장점이 있으나, 공구가 끌고 지나간 볼륨과 공작물간의 불리안 작업(Boolean operation)을 수행하는데 있어 곡면간의 교차 곡선을 구하기 위한 계산 시간이 지나치게 많이 걸리는 단점이 있다. 뿐만 아니라 교차 곡선 계산시 발생하는 오차로 말미암아 불리안 작업이 안정적으로 수행되기 어려운 점이 있다. 이에 반하여 z-map이나 법선 벡터 표현 방식은 샘플링 간격이나 방식에 따라 표현의 정확도는 떨어지나 불리안 작업을 신속하고 안정되게 수행할 수 있다는 장점이 있다. 따라서 현재의 하드웨어 성능 여건상 상용 시뮬레이션 시스템의 대부분이 이 방식을 채택하고 있다. 특히, 3축 밀링 가공의 경우는 벡터 표현 방법의 하나인 z-map 표현 방법이 널리 사

*(주)현대전자
**중신회원, 국민대학교 기계·자동차공학부
*** (주)엔젠테크놀로지
- 논문투고: 2000. 8. 7
- 심사완료일: 2000. 9. 19

용되는데, 여기서 z-map이란 일정한 간격의 (x, y)에 대한 z 값을 2차원 배열 형태로 저장한 형태를 의미한다. 이 방법이 널리 사용되는 이유는 앞에서 언급한 바와 같이 시뮬레이션을 위한 계산 속도가 빠르고, 알고리즘이 단순하며, 자료 구조의 관리가 쉽다는데 있다. 따라서 본 연구에서도 신속한 3축 NC 밀링 가공의 시뮬레이션을 위하여 공작물을 표현하기 위한 자료 구조로서 z-map을 채택하였다.

Hook^[2]는 이러한 z-map 자료 구조를 사용하여 밀링 가공을 실시간 시뮬레이션하기 위해서 z-map의 z축을 뷰잉 방향(viewing direction)과 일치시키고 z-map의 x, y 격자를 화면상의 픽셀에 일치시킨 후, 그에 맞추어 공작물에 대한 CAD 모델을 z-map으로 변환시킨다. 다음, 동작 지령에 따라 공구가 움직이면 공작물에 대한 z-map 모델을 갱신하고 그 결과를 화면에 디스플레이 한다. 이 방법에서는 z-map모형을 뷰잉 좌표계에 맞추어 생성하였기 때문에 화면상에 보이는 것은 가장 가까이 있는 z값이므로 별도의 계산이 필요하지 않다. 이는 z-버퍼(z-buffer)의 원리와 동일하며, 고속으로 디스플레이가 가능하다는 장점이 있다. 그러나 동시에 이 방법은 다음과 같은 몇 가지 단점도 가지고 있다. 첫째, 사용자가 뷰(view)를 변경하면 처음부터 그 단계까지 NC 시뮬레이션을 반복해야 한다. 물론 z-map을 복셀(voxel) 모델로 변경시켜 각 복셀에 대해 좌표변환(transformation)을 적용시키면 변경된 뷰에 따른 모델을 바로 얻을 수 있으나, 이 방법을 사용하면 공작물 형상에 대한 시각적인 질이 좋지 못하다. 둘째, 검증에 적합한 방향이 뷰잉 방향과 일치하지 않는 경우가 발생할 수 있는데, 이 방법에서는 검증을 위한 z방향이 항상 뷰잉 방향과 일치하기 때문에 이를 수용할 수 없는 문제점이 있다. 실제로 3축 밀링에서는 공구의 방향이 z-map의 z방향이 되는 경우에 가장 좋은 검증 결과를 보이는 데 반하여, 시각적인 효과는 등측 뷰(isometric view)가 더 선호된다. 따라서 z 방향을 뷰잉 방향과 별개로 가질 수 있도록 서로 분리시키는 것이 보다 바람직하다.

뷰에 독립적인 렌더링을 위하여 Huang과 Oliver^[3]는 contour display method를 제안하였다. 이 방법은 z-map에서 일정한 x값에 대한 윤곽선을 선으로 그리고, 또 일정한 y값에 대한 윤곽선들을 선으로 그린다. 이 두 종류의 서로 직교하는 윤곽선들을 그리면 z-map의 간격이 촘촘하기 때문에 마치 음영 처리된 그래픽 결과가 나오게 된다. 또한, 윤곽선을 그릴 때 인접한 두 선분이 허용 오차 한계에 드는 경우

이를 하나로 합쳐서 그리게 되면 실제 그리야 하는 선의 개수를 원래의 30%정도로 줄일 수 있다고 하였다. 이 방법은 기존의 방법에 비하여 뷰에 독립적인 렌더링 결과를 제공한다라는 장점을 가지고 있다. 그러나 디스플레이된 음영 화상의 질이 거칠고, 만일 사용자가 확대해서 본다면 선과 선사이의 떨어진 부분이 드러나게 되는 문제점이 있다. 따라서^[4]에서 추후 연구로서 인접한 두 윤곽선 사이를 삼각형으로 채워 넣은 메쉬(mesh)를 만들어 이를 렌더링하는 연구를 진행 중이라고 언급하였다.

이와 같이 뷰에 독립적이면서 고화질의 결과를 얻기 위해서는 다각형 렌더링 방법(polygon rendering method)을 사용할 필요가 있다. 이 방법은 많은 계산 시간을 필요로 하기 때문에 실시간 시뮬레이션에 그동안 사용되지 못하였으나 최근에 고성능의 다각형 렌더링 엔진을 갖춘 3차원 그래픽 카드들이 속속 개발됨으로써 보다 현실에 가까이 오게 되었다. 예를 들어, 금형의 크기가 500 mm×500 mm이고 그 격자 간격이 1 mm인 경우 약 500,000개의 삼각형이 그려져야 하며, 실시간 시뮬레이션을 위해서는 대략 1초에 10번 이상의 화면 갱신이 이루어져야 하므로 최소 5,000,000(polygons/sec)의 렌더링 속도를 가져야 한다. 또한 여기에 z-map 모델의 갱신 계산 시간을 고려한다면 이보다 더 빨라야 한다. 현재 3차원 그래픽을 위한 전용 카드들은 보통 1초에 1~5백만 개의 삼각형을 그릴 수 있는 수준에 와 있다. 그러나 이와 같은 그래픽 카드들은 대부분 가격이 고가라는 문제점을 가지고 있으며, 또한 자동차 금형과 같이 길이와 폭이 1 m를 넘는 대형 금형의 경우는 현재의 고성능 그래픽 카드로도 실시간 시뮬레이션이 불가능하게 된다.

따라서 본 논문에서는 뷰에 독립적이며 고화질의 렌더링 결과를 보여주는 NC 시뮬레이션을 위해 다각형 렌더링 방법을 채용하며, 이 방법의 속도를 향상시키기 위한 방법을 제안하고자 한다. 그럼으로써 고화질의 실시간 NC 시뮬레이션이 보다 값싼 그래픽 카드상에서도 구현될 수 있는 길을 모색하고자 한다.

2. 본 논문의 접근 방법

먼저 주어진 z-map에 대한 일반적인 기존의 뷰 독립적인 다각형 렌더링 방법을 먼저 살펴보도록 하겠다. 가공중인 공작물에 대한 음영 처리된 화상을 얻기 위해서는 일반적으로 OpenGL과 같은 그래픽 라

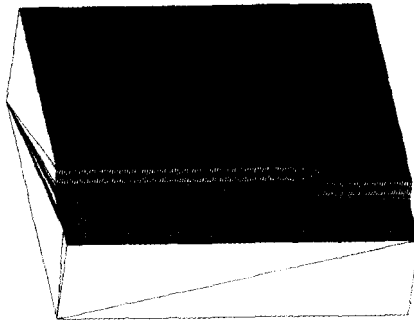


Fig. 1. A triangular mesh generated from z-map data for a workpiece.

이브러리의 함수를 이용하며¹⁶⁾, 이를 위해서 공작물의 가공면에 대한 삼각형 메쉬를 생성시켜주는 작업이 필요하다. 이 삼각형 메쉬는 z-map의 각 격자에 대해 2개의 삼각형을 구성함으로써 얻어지는데, Fig. 1의 예에서 볼 수 있듯이 z-map에서의 격자 수가 대단히 많기 때문에 고속의 그래픽 가속 장치나 CPU를 가지고 있지 않은 컴퓨터의 경우에는 이를 렌더링하는데 상당한 시간이 걸린다. 또한, 황삭의 가공 초기에 아직 절삭되지 않은 소재의 평면 부분이 많이 존재하는 경우나 사용자의 눈과 공작물과의 거리가 멀리 떨어져 있어 상세한 형상이 불필요한 경우에 대해서 전체 격자에 대한 모든 삼각형들을 생성시켜 렌더링을 하는 것은 상당히 비효율적이다. 눈과 공작물사이의 거리가 먼 경우 렌더링 속도를 증가시키기 위해 일정간격으로 z-map 자료를 샘플링하여 렌더링할 수도 있으나, 각 부위의 복잡도를 고려하지 않은 이러한 방법은 공작물 형상을 정확히 나타내기 어려우므로 바람직하지 않다.

따라서 만일 z-map 자료로부터 생성된 삼각형들을 디스플레이되는 공작물의 형상의 정밀도를 유지하면서 어떤 허용오차 한계 내에서 서로 병합시켜 삼각형의 개수를 줄일 수 있다면 렌더링 속도를 향상시킬 수 있으며, 결국 전체 시뮬레이션의 속도 향상을 기대할 수 있을 것이다. 병합의 판단 조건은 사용자가 직접 부여할 수도 있고, 뷰잉 변수들을 고려하여 자동으로 결정되도록 할 수도 있다.

삼각형 메쉬 모델이 주어졌을 때, 그로부터 삼각형 개수를 줄이는 방법은 주로 컴퓨터 그래픽스 분야에서 다량의 점군 데이터를 입력 정보로 하여 이를 사용자의 상세 수준(level of detail)에 맞추어 디스플레이 하기 위한 목적으로 개발되어 왔다. 즉, 관찰자의 눈이 상대적으로 멀리 있을 때는 적은 수의

다각형만을 그리고, 가까이 있을 때에는 많은 수의 다각형을 그릴 수 있도록 여러 레벨에 대한 메쉬 정보를 데이터 베이스에 저장하고 있다가 필요한 레벨의 정보를 추출하여 내보내도록 하였다. 이러한 목적의 모델을 다중 해상도 모델(multi-resolution model)이라고 하며, 이를 만들어 내기 위하여 주어진 점군 데이터로부터 초기의 메쉬를 만든 후 이를 각 레벨에 대하여 적절히 점들을 삭제하여 삼각형을 줄이는 방법을 개발하였다. 이와 같이 메쉬를 간략화시키는 방법은 Schroeder 등¹⁴⁾의 연구를 시작으로 수많은 연구 결과들이 발표되었으며, 그 대표적인 방법들을 선정하여 비교한 결과가 Cignoni 등¹⁵⁾의 논문에서 소개되어 있다. 이들 방법은 또한 제거되는 대상에 따라 크게 꼭지점을 줄이는 방법¹⁴⁾, 모서리를 없애는 방법¹⁶⁾, 면을 제거하는 방법¹⁷⁾의 세 가지로 나누어 볼 수 있으며, 이 가운데 가장 많이 사용되는 방법은 모서리를 없애는 방법(edge collapse method)이다.

그러나 이러한 기존의 메쉬 간략화 알고리즘들이 사용하는 모델에 비하여 본 연구에서의 렌더링 대상인 NC 밀링 머신상의 공작물의 형상은 다음과 같은 특징을 가지고 있다.

- 기존의 메쉬 간략화를 위한 모델은 원래의 형상에는 변함이 없으며 단지 필요에 따라 여러 상세 수준의 형상을 뽑아내기 위한 것인데 반하여, 가공 중인 공작물은 NC 명령이 수행될 때마다 그 형상이 계속 바뀐다.

- NC 프로그램의 각 공구 이동 명령마다 변화된 공작물을 렌더링을 하는 것이 일반적이며, 이때 매 프레임사이의 공작물의 형상 변화는 극부적이다.

한편, 전체 렌더링 시간은 메쉬 간략화 시간과 렌더링 시간의 합이기 때문에 이 시간이 메쉬 간략화 없이 그냥 렌더링 할 때보다 항상 더 빠를 때에만 연구의 의의가 있다고 할 수 있다. 실제로 3차원 그래픽 가속기와 CPU의 성능이 우수할 경우 메쉬 간략화를 수행시켜 그리는 것보다 그냥 원래 메쉬를 그리는 것이 더 빠르다. 따라서 본 논문에서는 위의 가공 중인 공작물 형상의 특징을 고려하여 하드웨어의 성능에 관계없이 메쉬 간략화를 수행하지 않고 그냥 그릴 때보다 항상 빠른 알고리즘을 개발하고자 한다.

[방안 1] 전체 메쉬 간략화 방법

우리는 먼저 Fig. 2에 나타난 것과 같이, 만일 공구의 이동에 대하여 변경된 공작물의 형상을 그릴 때마다 매번 가공면 전체에 대하여 삼각형 감소 작업을 수행하는 방법을 먼저 생각해 볼 수 있다. 그러

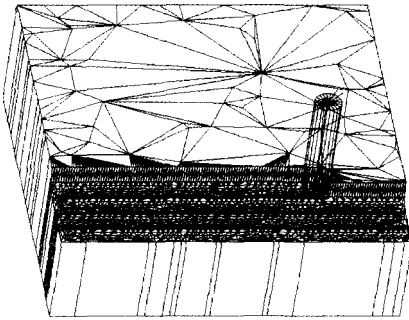


Fig. 2. Global mesh decimation for each tool movement.

나 이 방법을 적용한다면 삼각형의 감소율이 높은 경우에는 어느 정도의 렌더링 속도 향상을 기대할 수 있으나, 감소율이 낮은 경우에는 삼각형 감소 작업 시간이 더해지게 되어 결과적으로 전체 렌더링 시간이 오히려 증가하게 되는 결과를 초래할 수도 있다. 또한, 배 렌더링 사이의 공작물의 형상 변화는 그리 크지 않음에도 불구하고 전체에 대한 삼각형 감소 작업을 수행하는 것은 매우 비효율적이라고 할 수 있다. 또한, 이 방법은 하드웨어의 성능에 따라 그 효과가 달라진다. 실제 고성능의 PC에서는 원래 메쉬를 그냥 그리는 것이 삼각형의 개수를 줄여서 그리는 것보다 소요 시간이 오히려 더 짧은 것으로 관찰되었다.

[방안 2] 영역 분할 및 메쉬 초기 간략화 방법

이 방법은 Fig. 3에서와 같이, 초기에 공작물을 몇 개의 영역으로 분할하여 각 영역에 대해 메쉬를 생성하여 간략화시킨 다음, 이후에 공구에 의해 형상이 변화된 영역에 대해서는 z-map을 그대로 이용하여 렌더링하는 방법이다. 즉, 이 경우에는 초기에 간략화된 메쉬들과 이후 만들어진 간략화되지 않은 메쉬가 혼재하는 형태가 된다. 이 방법을 사용하면 최

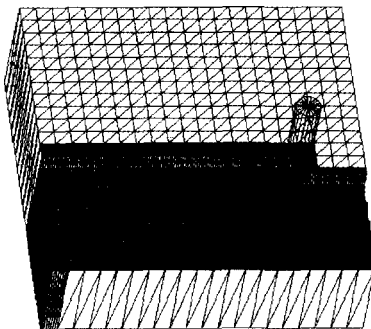


Fig. 3. One-time mesh subdivision and decimation.

초 초기화 단계에서 삼각형 감소 작업에 대한 시간이 소요되기는 하나, 그 이후부터 가공이 종료될 때까지 z-map을 그대로 그리는 방법에 비해 항상 적거나 같은 수의 삼각형을 그리게 된다. 이 방법은 항상 초기에는 삼각형 감소 효과가 있으나 일단 공작물 전체에 대하여 공구가 한번 지나가게 되면 그 이후에는 z-map을 그대로 그리는 것과 동일한 상태가 된다. 하지만 어찌되든 시뮬레이션에 소요되는 시간은 항상 z-map을 그대로 그리는 것보다는 빠르다.

[방안 3] 영역 분할 및 국부 메쉬 주기적 간략화 방법

이 방법은 [방안 2]의 방법과 초기는 동일하나, 주기적으로 변화된 국부 영역에 대한 메쉬들의 간략화 작업을 수행시켜 렌더링하는 방법이다. 주기는 매 공구 이동일 수도 있고, 몇 개의 공구 이동으로 정할 수 있다. 감소 작업이 수행될 때에는 그 작업의 수행 시간 만큼의 일시적인 속도의 지연이 있겠지만 그 이후에는 다시 렌더링 속도를 증가시킬 수 있으므로 [방안 2]의 문제점인 가공의 진행에 따른 렌더링 속도의 저하를 해소시킬 수 있을 것이다. 이 방법을 적용한 예가 Fig. 4에 소개되어 있다. 본 논문에서는 [방안 3]를 채택하였으며 이를 구현하기 위한 구체적인 자료 구조와 알고리즘은 다음 장에 소개하도록 하겠다.

3. 전체 알고리즘 개요

본 논문에서 채택한 국부 메쉬 간략화 알고리즘은 다음과 같다.

- ① 초기화 작업을 수행한다. 여기에서는 공작물의 가공면을 일정한 크기의 작은 영역들로 분할하고, 분할된 각 영역에 대해 삼각형 감소 작업을 수행하여 그 결과를 저장한다.
- ② 공작물 형상 계산 모듈이 하나의 NC 공구 이

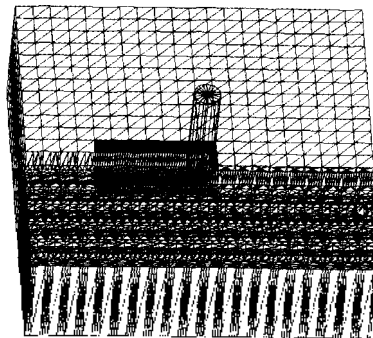


Fig. 4. Periodical local mesh decimation method.

동 명령을 읽어서 공작물의 z-map 자료를 갱신한 후, 그 z-map 자료와 공구가 끌고 간 이동 범위를 렌더링 모듈에 넘겨준다.

③: 렌더링 모듈에서는 입력받은 공구 이동 범위와 조금이라도 충돌하는 영역에 대해서는 z-map으로 전환시키고 z-map의 모든 격자에 대해서 삼각형을 만들어서 그리도록 하며, 그 외의 영역에 대해서는 감소된 메쉬를 이용하여 그리도록 한다.

④: 만일 사용자가 지정한 삼각형 감소 작업 주기에 해당하는 경우에는 z-map으로 환원된 영역들에 대해 삼각형 감소 작업을 수행한 후 ②로 돌아간다. 주기에 해당하지 않는 경우에는 감소 작업 없이 바로 ②로 돌아간다.

그러면 이제부터 본 시스템을 만들기 위해 필요한 자료 구조와 삼각형 감소 방법에 대하여 자세히 설명하도록 하겠다.

4. 자료 구조

본 논문에서 위의 작업을 구현하는 데 필요한 자료 구조는 z-map, 영역(region), 그리고 삼각형 감소 작업을 수행하기 위한 메쉬 저장 자료 구조이다. Z-map은 일정한 간격의 격자들의 z값을 실수의 2차원 배열 형태로 저장하였다. 또한 가공면을 분할한 각 영역에 대한 자료는 Fig. 5에 나타난 것과 같이, 영역의 고유 일련 번호(id), 이 영역이 z-map으로 그려질 것인지 삼각형 감소 결과로 그려질 것인지를 나타내는 플래그(reduction_display_flag), 이 영역이 z-map상에서 어떤 부분을 차지하는지를 나타내는 최대 및 최소 x,y값(min_x, min_y, max_x, max_y), 감소 작업 결과 남아있는 삼각형들의 세 꼭지점의 배열(triangle), 그리고 영역이 전체 z-map 곡면의 경계에 위치해 있는 경우 삼각형 감소 작업 후 경계를 구성할 라인들에 대한 데이터(top, bottom, right, left)를

```
struct Region {
    int id ;
    bool reduction_display_flag;
    int min_x, min_y ;
    int max_x, max_y ;
    int triangle_count ;
    float *triangle[3] ;
    int top_count, bottom_count;
    int right_count, left_count;
    float *top[3], *bottom[3], *right[3], *left[3] ;
};
```

Fig. 5. Data structure for regions.

저장하고 있다. 경계에 대한 데이터는 자기에게 해당하는 자료만 저장하며, 영역이 z-map 곡면 내부에 위치해 있는 경우에는 아무런 자료를 가지고 있지 않다. 렌더링시에는 각 영역에 대해 그 영역의 reduction_display_flag가 true로 세팅되었으면 그 영역은 그 이하에 저장된 감소된 삼각형들과 경계 라인들의 데이터를 사용하여 그리게 되고, false로 세팅되었으면 그 영역은 z-map으로부터 삼각형을 만들어서 그리게 된다.

또한, 본 논문에서는 삼각형 감소 작업을 위한 자료 구조로서 winged-edge 자료 구조^[10]를 사용하였다. 이는 불필요한 삼각형을 제거하는 중간 단계에서는 사각형과 같은 비삼각형 형상이 발생할 수 있으며, 또한 신속한 삼각형 감소 작업이 이루어지기 위해서는 인접한 면, 모서리, 꼭지점등에 대한 효율적인 검색이 가능해야 한다. 이를 위하여 본 연구에서는 winged-edge 자료 구조를 선택하여 사용하였다. 그 이유는 이 자료 구조가 다면체 곡면을 표현하는 자료 구조들 가운데 최적에 가까운 저장 장소 및 검색의 효율성을 가진 것으로 알려져 있기 때문이다^[10]. 삼각형 감소 작업을 위해서는 먼저 해당 영역의 z-map 데이터를 이 winged-edge 자료 구조로 변환시킨 후, 삼각형 감소 작업을 수행하고, 그 결과를 Fig. 5의 영역 자료 구조에 저장시키게 된다. 그러면 하나의 영역에 대한 삼각형 감소 작업에 대하여 보다 상세히 설명하도록 하겠다.

5. 국부 영역에 대한 삼각형 감소 방법

전체 z-map 곡면을 일정한 크기의 영역으로 분할한 후 각 영역에 대해 삼각형 메쉬를 생성시킨 후 이를 간략화시키는 과정은 다음과 같다. 이 과정은 최초에 전체 영역들에 대하여, 그리고 매 감소주기마다 그 사이 변화된 영역들에 대해서 수행되게 된다.

①: Z-map 데이터에서 주어진 영역에 속하는 부분의 자료를 취하여 그로부터 삼각형 메쉬의 위상 및 기하 정보를 winged-edge 자료 구조에 맞추어 생성시킨다.

②: 삼각형 메쉬에서 하나의 꼭지점을 선택하여 제거 가능한 꼭지점인지 판명한다.

③: 제거 가능한 꼭지점으로 판단되면 꼭지점을 제거하고 꼭지점과 연결된 위상 요소들의 데이터를 수정하게 된다.

④: 모든 꼭지점에 대해 검사가 이루어졌는지 판단하여 아직 검사를 받지 않은 꼭지점이 있으면 위의

②단계로 돌아간다. 만일 모든 검사가 이루어졌으면 감소 작업의 결과를 실수 배열로 저장하고 이 과정을 끝낸다. 여기서 감소 작업의 결과는 모든 삼각형들의 세 꼭지점들과 영역이 가공면의 경계에 위치한 경우 경계의 점들이 된다.

그러면 이들 각 단계에 대하여 보다 상세히 설명하도록 하겠다.

5.1 Z-map에서 winged-edge 자료 구조로의 변환

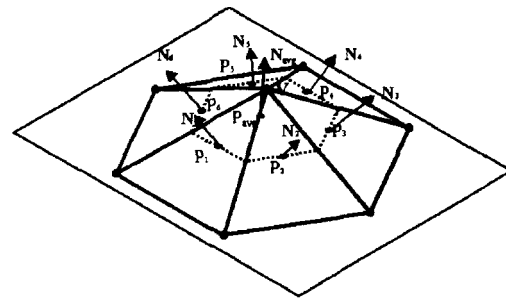
이 단계는 메쉬 간략화를 수행하기 위한 준비 단계로서 주어진 z-map 데이터에서 지정된 영역에 해당되는 z-map 자료들을 취하여 winged-edge 자료 구조에 맞추어 변환 저장시킨다. 이 단계는 대량의 자료를 변환시키는 작업으로서 많은 시간을 소모할 우려가 있으므로 우리는 z-map으로부터 만들어질 메쉬의 토폴로지를 미리 정하고 이에 맞추어 해당 자료를 채워 넣는 방식을 사용하여 변환 시간을 최소화시켰다. 즉, 미리 각 모서리, 꼭지점, 면의 고유 번호를 일정한 규칙에 따라 부여한 후, 이들을 미리 정해진 토폴로지를 만족하도록 각 위상 요소내의 포인트들을 채워 넣었다. 또한 매번 영역에 대한 감소화 작업을 할 때마다 winged-edge 자료 구조의 메쉬를 위한 메모리 영역을 새로 할당하고, 사용 후 다시 지우는 것은 반복하는 것은 많은 시간을 소비할 우려가 있으므로, 최초 한번만 메쉬에 대한 메모리 영역을 할당한 후 이를 시뮬레이션이 끝날 때까지 반복해서 사용하도록 하였다.

5.2 제거 가능한 꼭지점들의 선택

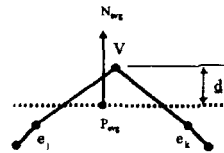
이 단계에서는 메쉬를 구성하는 꼭지점 가운데 하나를 선택하여 이 꼭지점이 제거 가능한 꼭지점인지를 판단하는 작업을 한다. 본 논문에서 사용한 제거의 가능성을 판단하는 기준은 두 가지로서, 첫째, 꼭지점을 제거했을 경우 주어진 허용 오차를 넘어서 형상이 간략화되는 지를 먼저 판단하고, 둘째, 꼭지점을 제거했을 때 면적이 0으로 퇴화하거나 뒤집히는 면이 발생하는지의 여부를 판단하도록 한다. 그러면 이 두 가지 시험 방법의 기준에 대해서 보다 상세히 설명하도록 하겠다.

5.2.1 꼭지점의 허용 오차 초과 판단 테스트

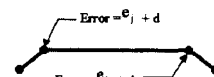
이 테스트를 위해서 먼저 각 꼭지점이 자신의 오차값을 저장할 수 있도록 자료 구조 선언시 꼭지점에 error 필드를 지정해 준다. 이 오차 값은 초기에는 0으로 설정되어 있으나, 모서리로 연결된 인접 꼭지점이 소거되면 소정의 오차가 누적되게 된다. 임



(a)



(b)



(c)

Fig. 6. Vertex error calculation (a) an average plane (b) distance between a vertex and the average plane (c) error distribution to adjacent vertices.

의 꼭지점이 소거될 때, 인접한 꼭지점에 부여되는 오차값은 다음의 방법에 의하여 계산되어 할당되며, 이는¹⁴⁾에서 제시된 방법을 원용하였다. 이를 간략히 설명하면 다음과 같다.

먼저, 소거 대상 꼭지점을 둘러싸고 있는 면들의 가중 평균 평면을 계산한다. 이 평면상의 한 점은 각 면들의 중심의 가중 평균이고, 법선 벡터는 각 면들의 법선 벡터의 가중 평균값을 취함으로써 평균 평면을 구할 수 있다. Fig. 6(a)에서와 같이 각 면(F_i)의 단위 법선 벡터(N_i)들에 대한 평균 법선 벡터(N_{avg})는 $N_{avg} = \sum A_i N_i / A_{\Sigma}$, 각 면의 중심점(P_i)들에 대한 평균점(P_{avg})은 $P_{avg} = \sum A_i P_i / A_{\Sigma}$ 로 구할 수 있다. 다음, Fig. 6(b)에 나타난 것과 같이 이 평균 평면으로부터 소거 대상 꼭지점까지의 거리 d 를 계산한다. 이 거리와 그동안 이 꼭지점에 누적된 오차의 합이 허용 오차를 넘지 않는다면 이 꼭지점을 소거하는 것으로 판명한다. 꼭지점을 소거할 때는, Fig. 6(c)에서와 같이, 그 꼭지점에 모서리를 통하여 연결되어 있는 꼭지점들에게 d 만큼을 오차로 누적시킨다. 따라서 임의의 꼭지점이 소거될 수 있는가 하는 것에 대한 판단은 그 꼭지점에 누적된 오차가 허용 오차를 넘지 않는지를 살펴서 결정한다.

5.2.2 뒤집히거나 퇴화되는 면의 발생 여부 테스트
본 논문에서는 꼭지점을 소거하기 위하여 모서리

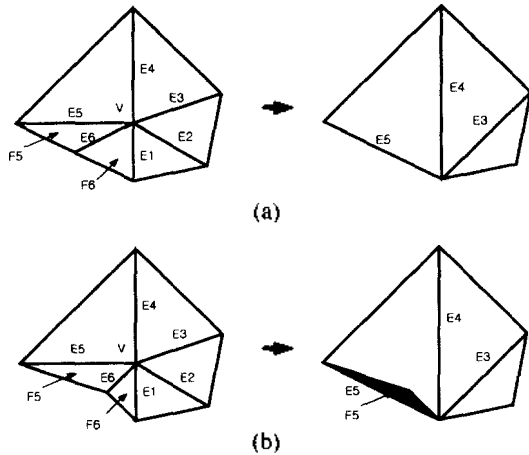


Fig. 7. Illegal topology change caused by edge collapse (a) generation of a degenerate face (b) generation of a reversed face.

붕괴(edge-collapse) 방식을 채택하였다. 즉, 소거할 꼭지점에 연결되어 있는 모서리들 가운데 하나를 꼭지점과 동시에 없애며, 이를 위하여 Euler 연산자^[8]를 구현하여 사용하였다. 이때 주의해야 할 점은 한 모서리를 없애므로 메쉬의 면 가운데 면적이 0으로 퇴화하거나 뒤집히는 면이 발생할 수가 있다는 것이다. Fig. 7(a)의 예에서 같은 경우에는 모서리 E1이 붕괴될 때 생존하는 면 F5의 면적이 0가 된다. 또한 Fig. 7(b)의 예에서와 같은 경우에는 모서리 E1이 붕괴될 때 생존하는 면 F5가 뒤집히게 된다. 메쉬가 이와 같은 부적절한 토폴로지를 갖게되는 경우 이후 간략화 과정에서 위상 요소들의 인접 정보를 제대로 얻어낼 수 없을 뿐만 아니라 뒤집힌 면은 렌더링시 그 부분에 구멍이 난 것과 같은 효과를 가질 수 있다. 따라서 이와 같은 현상이 발생하는지 여부를 삭제할 모서리를 선택할 때 검사해야 한다.

본 논문에서는 꼭지점 제거 후에도 가능하면 원래와 가까운 형상을 유지하기 위하여 인접한 모서리들 가운데 가장 짧은 모서리를 제거할 모서리로 먼저 선택해 본다. 그러나 Fig. 7에서와 같이 가장 짧은 1번 모서리를 없애면 경우 뒤집힘이 발생하므로, 나머지 모서리 가운데 뒤집힘이 발생하지 않는 모서리를 선택하여 제거하도록 한다. 이러한 뒤집히는 면의 발생 여부는 꼭지점 제거 후의 각 면의 법선 방향을 미리 계산하여 그 방향이 (-)z 방향 성분을 가지고 있으면 뒤집힌 것으로 판명한다. 또한 면적이 0이 되는 경우에 각 면의 법선을 구하기 위해 외적을 취했을 때 그 크기가 0이 되는 경우에 해당하므로 이를

판별하면 된다.

본 논문에서는 만일 뒤집힘이나 퇴화된 면이 발생되는 경우 대상 모서리를 바꾸어 이 판단 기준을 통과할 때까지 검사를 반복한다. 만일 어떠한 모서리도 이를 만족시키지 못하면 그 꼭지점은 제거하지 않는다.

5.3 꼭지점의 제거

없어질 꼭지점과 모서리가 결정되면 그에 따라 꼭지점을 둘러싼 면들에게 그에 맞는 새로운 토폴로지를 부여해 주어야 한다. 이를 수행하는 방법은 삭제될 꼭지점을 둘러싼 면들을 모두 소거한 후 새로운 면을 생성시키는 전체 수정 방법과 기존의 면들 가운데 일부만 없애고 기하학적 정보를 수정하는 부분 수정 방법이 있다. 본 논문에서는 후자인 부분 수정 방식을 채택했으며 그 과정을 내부 꼭지점과 경계 꼭지점인 경우로 분류하여 각각에 대해 경계 표현 방식의 솔리드 모델러에서 흔히 사용되는 Euler 작업을 사용하여 프로그램을 구현하였다. 그러면 내부와 경계 꼭지점의 제거 방법에 대하여 각각 설명하도록 하겠다.

5.3.1 내부 꼭지점의 제거

Fig. 8(a)에서 영역의 내부에 위치한 꼭지점 *e*가 앞의 제거 조건을 만족하며 또한 여기에 연결된 가장 짧은 모서리 가운데 하나인 모서리 11을 검사한 결과 비정상적인 토폴로지의 발생은 일어날 가능성

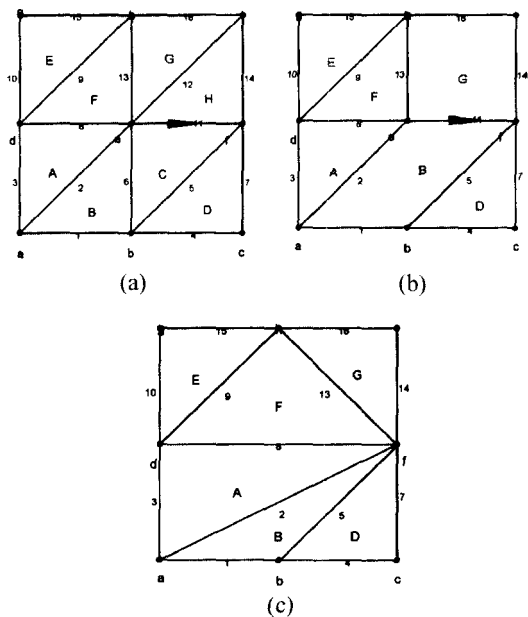


Fig. 8. Edge collapse procedure (a) initial shape (b) result of 2*KEF (c) result of KEV.

이 없는 것으로 판명되었다고 할 때, 이 꼭지점과 모서리를 없애는 과정은 다음과 같다. 먼저, 모서리 11에 인접해 있는 면 C와 모서리 6을 오일러 작업가운데 KEF를 사용하여 동시에 소거한다. 다음, 모서리 11에 인접해 있는 면H와 모서리 12를 마찬가지로 KEF를 사용하여 동시에 소거한다. 그 결과가 Fig. 8(b)에 나타나 있다. 끝으로 소거 대상인 꼭지점 e와 모서리 11을 KEV를 사용하여 소거한다.

5.3.2 경계 꼭지점의 제거

영역의 경계에 위치하고 있는 꼭지점의 제거 작업의 기본적인 알고리즘은 내부 꼭지점의 제거 알고리즘과 거의 동일하지만 없어질 모서리의 선택과 KEF 과정에서 차이가 난다. 내부 꼭지점의 없어질 모서리를 선택할 때는 가장 짧은 모서리를 선택했지만 경계에서는 경계를 이루고 있는 두개의 모서리 중 하나만이 없어질 수 있다. Fig. 9의 경우, 만일 없어질 꼭지점이 b이고 없어질 모서리로 6이 선택되었을 경우 앞의 내부 꼭지점의 제거 알고리즘을 그대로 적용하면 꼭지점 b가 e의 위치로 팔려 들어가 경계선이 영역 안쪽으로 들어갔다 나오는 형상이 되고 말 것이다. 이러한 현상을 방지하고 경계선이 직선을 유지하도록 하기 위해서는 제거될 모서리로는 경계선에 위치하고 있는 모서리 1,4 둘 중의 하나가 되어야 한다. 경계선 상의 꼭지점을 제거할 때는 내부 꼭지점과는 달리 한번의 KEF와 한번의 KEV를 수

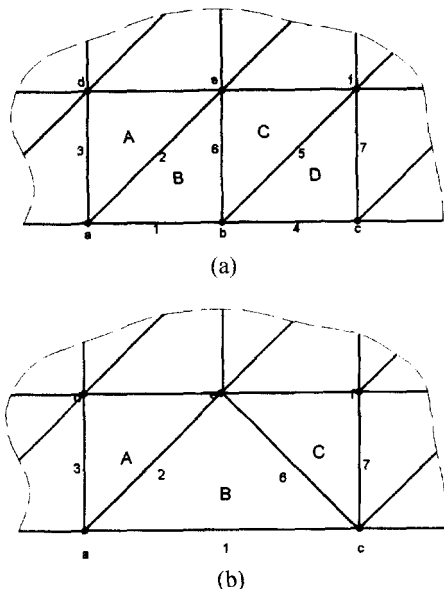


Fig. 9. Boundary edge collapse (a) initial shape (b) result of KEF and KEV.

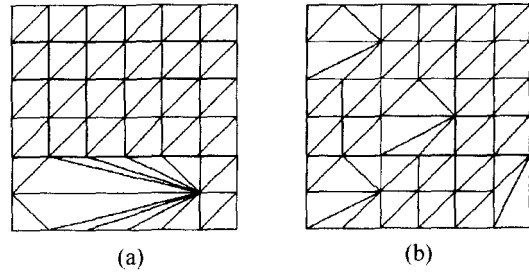


Fig. 10. Vertex removal methods (a) sequential removal method (b) random removal method.

행한다. 즉, Fig. 9의 경우, 없어질 모서리로 모서리 4가 선택되었다면 우선 KEF과정으로 면 D와 모서리 5가 없어지고 다음으로 KEV과정으로 꼭지점 b와 모서리 4가 삭제된다.

5.3.3 꼭지점 제거 순서

영역에 대한 배쉬를 간략화 과정에서 제거할 꼭지점을 선택하는 방법은 배쉬 간략화의 소요 시간과 매우 밀접한 관계가 있다. 만일 Fig. 10(a)에서와 같이 인접한 꼭지점을 계속하여 검사하고 제거해 나간다면 삼각형 감소 작업이 진행되어 나감에 따라 꼭지점에 인접해 있는 면과 모서리의 수가 계속 증가하게 되어 검사에 많은 시간을 소비하게 된다. 따라서 본 연구에서는 난수를 발생시켜 Fig. 10(b)에서와 같이 꼭지점들을 무작위로 선택, 검사함으로써 꼭지점에 필요이상으로 많은 면이나 꼭지점이 인접하는 것을 방지하였다.

다만 본 연구의 실험 결과 무작위로 꼭지점을 제거할 경우에는 순차적으로 꼭지점을 제거할 경우보다 결과 배쉬의 삼각형의 수가 다소 많은 경향을 보인다. 이는 무작위로 꼭지점을 선택, 제거하는 과정에서 간혹 5.2.2에서 언급한 제거 조건을 만족하지 못하는 경우가 발생하기 때문이다. 하지만 이러한 삼각형 개수의 차이는 크지 않기 때문에 렌더링 속도에 영향을 주지 않으며 오히려 보다 빠른 간략화를 얻게 되므로 충분히 보완될 수 있다.

6. 적용 사례 및 토의

6.1 환삭 가공의 예

먼저 Fig. 1에서와 같은 직선 경로로 공작물을 환삭 가공하는 경우에 대하여 삼각형 감소 기법을 적용시키지 않고 그냥 z-map 자료를 렌더링한 경우와 본 논문의 감소 기법을 적용시켜 렌더링한 경우의 면의 개수와 소요 시간의 변화를 측정해 보았다. 여

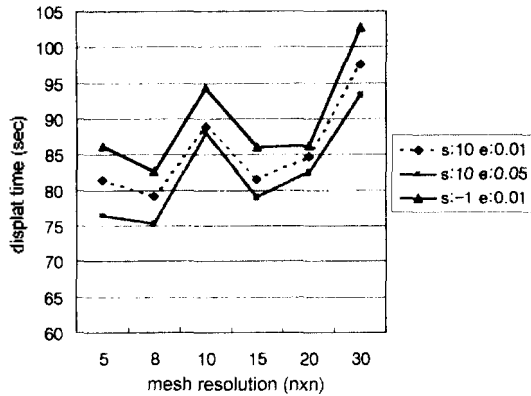


Fig. 11. Relation between display times and region sizes.

기서 z-map 자료를 그대로 렌더링하는 것은 터보텍크의 CANVAS 시스템의 프로그램을 사용하여 수행하였다. 하드웨어는 현재의 보급형인 Pentium-II 350MHz CPU와 OpenGL 그래픽 가속 기능을 갖추지 않은 ATI RAGE IIC 그래픽 카드를 장착한 PC를 사용하였다. 공작물은 140 mm×140 mm의 크기로서 1 mm 간격의 격자점들에 대한 z-map 데이터를 생성시켜 사용하였다. 절삭 공구로서는 볼 앤드 밀을 사용하였으며, 공구의 직경은 11 mm, 길이는 50 mm이다. 여기에 사용된 NC 프로그램은 255개의 공구 이동 명령을 포함하고 있다.

6.1.1 국부 영역의 크기에 따른 렌더링 시간

주어진 삼각형 감소 주기와 오차에 대하여 메쉬 간략화가 진행되는 국부 영역의 크기를 5, 8, 10, 15, 20, 30으로 증가시키면서 얻어진 렌더링 시간의 변화를 Fig. 11에 나타내었다. 여기서 s는 감소 주기로서 -1은 초기에 1회 감소를 수행한 후 이후에는 감소가 수행되지 않은 것을 뜻하며, e는 허용 오차를 뜻한다. 이 그래프에 나타난 바와 같이 국부 영역의 크기와 렌더링 시간과는 어떤 일반적인 상관 관계가 성립되지 않음을 볼 수 있다. 단, 어떤 주어진 NC 프로그램과 공작물에 대하여 국부 영역의 크기를 최적화시키는 것은 연구 대상이 될 수 있을 것이다.

6.1.2 허용 오차에 따른 삼각형 감소율과 렌더링 시간

국부 영역의 크기를 8x8로 고정하고, 삼각형 감소

주기를 매 20번의 NC 코드 수행으로 설정하고, 허용 오차를 0.01, 0.05, 0.1로 변화시키면서 시뮬레이션 과정에서 그려진 총 삼각형의 개수와 렌더링 시간의 변화가 Table 1에 나타나 있다. 여기서 e는 mm 단위의 허용 오차를 나타내며, s는 감소 주기를 NC 명령의 개수로 표시한 것이다. z-map으로 표시된 첫 번째 열은 삼각형 감소 작업을 수행하지 않고 z-map으로부터 생성된 모든 삼각형을 그린 경우를 나타낸다. 두 번째 열은 최초에 한 차례만 삼각형 감소를 시키고 그 이후에 변경된 영역들에 대해서는 z-map 자료로부터 직접 삼각형을 생성시켜 그린 경우를 나타낸다. 나머지 셋째부터 다섯째 열은 허용 오차를 0.01, 0.05, 0.1로 변화시키면서 측정된 그려진 삼각형의 개수와 렌더링 시간을 나타낸다. 허용 오차를 느슨하게 줄수록 렌더링에 사용되는 삼각형의 수와 시간이 감소하는 것을 볼 수 있다.

Fig. 12과 Fig. 13은 시뮬레이션이 진행됨에 따른 삼각형 개수와 렌더링 시간의 변화를 각각 그래프로 나타낸 것이다. Fig. 13에 나타난 것과 같이 렌더링 시간은 전체적으로 계단식 증가 형태를 보이고 있으며, 삼각형 감소주기에 해당되는 때에는 삼각형을 줄이기 위한 시간이 추가됨으로 말미암아 소요 시간이 순간적으로 증가하는 모습을 보이고 있다. 독자의 이해를 돕기위해 각 경우에 대한 추세를 직선으로 함께 나타내었다. 여기서 주의할 점은 가공 후반부로 가면 삼각형을 감소하여 렌더링하는 (2)나 (3)의

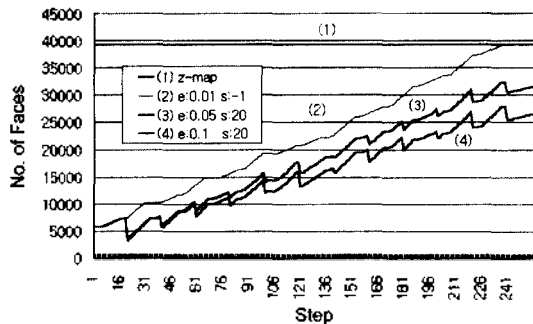


Fig. 12. Numbers of faces displayed for each frame during the rough cutting simulation.

Table 1. Effect of mesh decimation according to error allowances

	z-map	e: 0.01 s: ∞	e: 0.01 s: 20	e: 0.05 s: 20	e: 0.1 s: 20
No. of faces	9996000 (100%)	5800968 (58%)	4968194 (50%)	4604058 (46%)	4020966 (40%)
Display time(sec)	156.38 (100%)	120.31 (77%)	107.86 (69%)	105.22 (67%)	93.04 (59%)

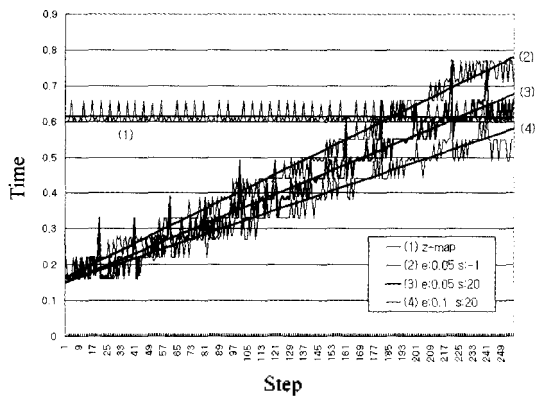


Fig. 13. Display times for each frame during the rough cutting simulation.

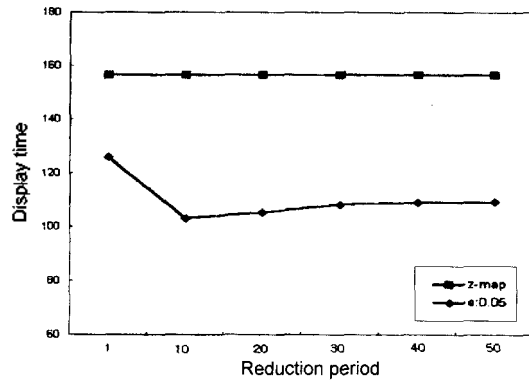


Fig. 14. Display times for the rough cutting simulation along with the reduction period.

경우 삼각형의 개수가 같거나 적은 데에도 불구하고 z-map을 그냥 렌더링하는 (1)의 경우보다 시간이 더 걸리는 것을 볼 수 있다. 이는 z-map을 바로 그리는 경우보다 좀더 많은 스텝의 명령과 영역별로 분산된 데이터를 접근하는데 보다 많은 시간이 소요되기 때문이며, 또한 현재의 프로그램이 다소 비효율적으로 짜여졌기 때문으로 판단된다.

6.1.3 감소 주기에 따른 렌더링 시간

Table 2는 국부 영역의 크기를 8×8로 설정하고 허용 오차를 0.05로 고정하고 감소 주기를 변화시킬 때 시뮬레이션 전체에 걸쳐 디스플레이되는 총 면의 개수와 렌더링 시간이 어떻게 변화되는지를 관찰한 것이다. 감소 주기가 늘어남에 따라 렌더링되는 면의 개수는 점차 증가한다. 렌더링 시간도 대체로 주기가 늘어남에 따라 점차 증가하나, Fig. 14에 나타난 것과 같이 감소 주기가 아주 짧을 때는 시간이 크게 증가하는 현상을 보인다. 이는 삼각형 감소 작업으로 인한 시간의 증가가 그로 인해 줄어든 삼각형의 렌더링 시간보다 더 커지기 때문으로 판단된다. 최적의 삼각형 감소 주기는 NC 프로그램의 내용, 전체 z-map을 분할하는 영역의 크기, 그리고 하드웨어의 성능 등에 좌우된다.

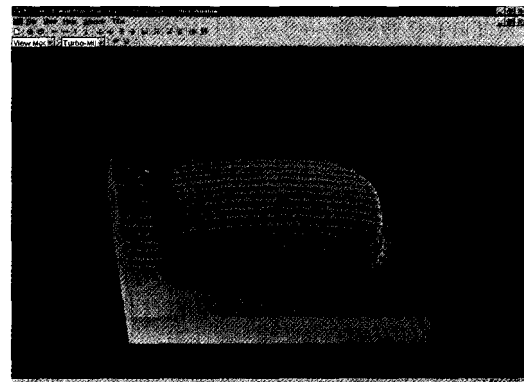


Fig. 15. Rough and finish cutting simulation for a mouse mold.

6.2 사출 금형의 황삭 및 정삭 가공의 예

Fig. 15에서와 같이 마우스 케이스의 사출 금형을 황삭에서부터 정삭에 이르기까지 전 과정에 걸쳐 시뮬레이션을 수행하여 보았다. 컴퓨터와 절삭 공구의 사양은 앞의 황삭의 경우와 동일하다. 공작물은 140 mm×100 mm의 크기로서 1 mm 간격의 격자점들에 대한 z-map 데이터를 생성시켜 사용하였으며, 메쉬 간략화가 적용될 각 국부 영역은 8×8의 크기를 사용하였다. 여기에 사용된 NC 프로그램은 총

Table 2. Effect of mesh decimation according to decimation periods

	z-map	e: 0.05 s: 1	e: 0.05 s: 10	e: 0.05 s: 20	e: 0.05 s: 30	e: 0.05 s: 40	e: 0.05 s:
No. of faces	9,996,000 (100%)	4,217,932 (42%)	4,474,038 (44%)	4,604,058 (46%)	4,698,151 (47%)	4,820,145 (48%)	5,800,968 (58%)
Display time (sec)	156.38 (100%)	126.04 (81%)	102.84 (66%)	105.22 (67%)	108.20 (69%)	108.81 (70%)	120.31 (77%)

Table 3. Variations of the number of faces and the display time with respect to error allowances

	z-map	e: 0.01 s: ∞	e: 0.01 s: 20	e: 0.05 s: 20	e: 0.1 s: 20
No. of faces	2.88×10^8 (100%)	2.85×10^8 (99%)	2.20×10^8 (77%)	1.78×10^8 (62%)	1.54×10^8 (53%)
Display time (sec)	6,106 (100%)	7,226 (118%)	5,744 (94%)	4,926 (81%)	4,825 (67%)

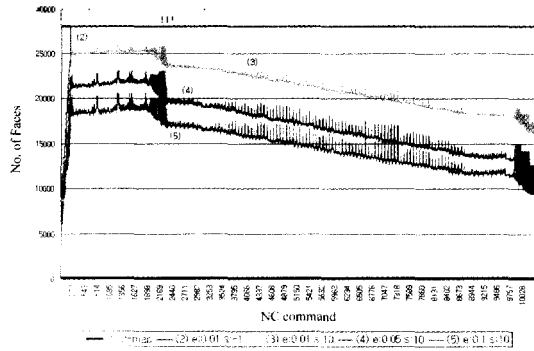


Fig. 16. Numbers of faces displayed for each frame during the mouse mold cutting simulation.

10,273개의 공구 이동 명령을 포함하고 있다.

6.2.1 허용 오차에 따른 삼각형 감소율과 디스플레이 이 시간

삼각형 감소 주기를 20으로 고정하고 허용 오차를 0.01, 0.05, 0.1로 변화시키면서 이에 따른 디스플레이 된 총 면의 개수와 소요 시간의 변화를 측정해 보았다. 그 결과는 Table 3에 나타나 있듯이 허용 오차가 클수록 면의 개수와 디스플레이 시간이 감소하였다.

시뮬레이션의 전과정에 걸쳐 디스플레이된 면의 개수의 변화 추이를 살펴보면 Fig. 16에 나타난 것과 같다. 먼저, 황삭 단계를 살펴보면 초기 2000 블록이 이에 해당된다. 주어진 허용 오차에 대하여 삼각형 감소율은 황삭 단계 진행 중에는 거의 일정한

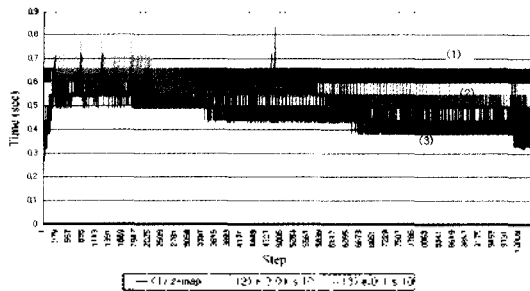


Fig. 17. Display times of each frame during the mouse mold cutting simulation.

상태를 유지한다. 또한 Fig. 17에서와 같이 디스플레이에 소요되는 시간도 정상 단계에 비해 황삭 단계에서 많이 걸리는 것을 볼 수 있다.

그러나 정상 단계에 접어들면 가공이 진행됨에 따라 점점 디스플레이되는 면의 개수가 감소되며,

Fig. 17에서와 같이 디스플레이에 소요되는 시간도 황삭 단계에 비해 정상이 진행되면서 점점 감소하는 것을 볼 수 있다. Fig. 17에서 각 그래프의 가운데 있는 검은 선은 독자의 이해를 돕기 위해 매 20구간의 평균값을 그려 그래프의 추이를 나타낸 것이다.

7. 결 론

본 논문에서는 삼각형 감소 기법을 이용하여 3축 NC 밀링 작업의 실시간 시뮬레이션의 속도를 향상시키는 새로운 방법을 제안하였다. 이 방법은 먼저, z-map으로 표현되는 절삭중인 소재의 가공면을 일정한 크기의 소 영역으로 분할한 후 각 영역에 대해 삼각형 메쉬를 주어진 허용 오차 한계 내에서 서로 병합시켜 수를 감소시킨 후 그 결과를 저장하고, 공구가 그 영역에 들어오기 전까지는 감소된 메쉬로 그리고 공구가 영역에 들어오면 원래의 z-map을 바로 그리는 방법을 취하는 것이다. 그리고 일정 주기로 z-map상태로 그리는 영역들에 대한 삼각형 감소 작업을 수행하도록 하였다. 이 방법은 시뮬레이션 초기에 한번의 감소 작업을 수행한 후 가공이 끝날 때까지 언제나 z-map으로부터 직접 삼각형을 생성하여 그리는 것보다 항상 속도가 빠르며, 하드웨어의 성능에 관계없이 언제나 시뮬레이션의 속도 향상을 가져올 수 있는 방법이다. 또한, 가공 도중에 일정 주기로 z-map으로 환원된 영역에 대해서 감소 작업을 수행시켜 줌으로써 저하되어가는 렌더링 속도를 다시 가속시키도록 되어있다.

그러나 현재의 프로그램을 더욱 효율적으로 하기 위해서는 다음과 같은 작업이 향후 이루어져야 할 것으로 본다.

첫째, 현재 사용하고 있는 winged-edge 자료 구조

는 일반적으로 기억 장소와 검색 효율성이 최적에 가까운 구조이나 특정한 응용 분야에 대해서는 가장 효율적이라고 할 수 없다. 따라서 효율성을 보다 높이기 위해서는 본 메쉬 감소 작업에 대해서 최적인 자료 구조를 고안할 필요가 있다^[10].

둘째, 영역의 크기와 삼각형 감소 수행 주기를 주어진 NC 프로그램에 효율적이 되도록 시뮬레이션 수행 이전 단계에서 결정해 주는 방법의 개발이 필요하다.

셋째, 다각형 렌더링 방법은 뷰에 독립적인 방법이다. 그러나 현재와 같이 모델 공간상에서의 허용 오차를 사용자가 지정하는 것보다 현재의 뷰 상태를 고려하여 사용자가 감지할 수 없는 수준의 허용 오차를 자동으로 계산하는 방법이 사용자에게 편리할 수 있으므로 이를 추가할 필요가 있을 것이다.

감사의 글

본 연구는 (주)터보테크의 지원을 받아 수행되었으며 관계자 여러분에게 감사 드립니다.

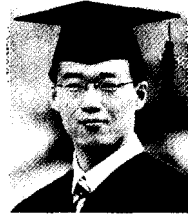
참고문헌

1. Choi, B.K., Robert, B. and Jerard, R.B., *Sculptured Surface Machining*, Kluwer Academic Publishers, 1998.
2. Van, H.T., "Real-Time Shaded NC Milling Display", *Proceedings of SIGGRAPH '86 in Computer Graphics*, Vol. 20, No. 4, pp. 15-20, 1986.
3. Huang, Y. and Oliver, J. H., "Integrated Simulation, Error Assessment, and Tool Path Correction for Five-Axis NC Milling", *Journal of Manufacturing Systems*, Vol. 14, No. 5, pp. 331-344, 1995.
4. Schroeder, W. J., Zerge, J. A. and Lorensen, W. E., "Decimation of Triangle Meshes", *Proceedings of SIGGRAPH '92 in Computer Graphics*, Vol. 26, No. 2, pp. 65-70, July 1992.
5. Cignoni, P., Montani, C. and Scopigno, R., "A Comparison of Mesh Simplification Algorithms", *Computers & Graphics*, Vol. 22, No. 1, pp. 37-54, 1998.
6. Hoppe, H., "Progressive Meshes", *Proceedings of SIGGRAPH '96 in Computer Graphics*, Vol. 30, pp. 99-108, 1996.
7. Hamann, B., "A data reduction scheme for triangulated surfaces", *Computer Aided Geometric Design*, Vol. 11, No. 2, pp. 197-214, 1994.
8. Lee, K., *Principles of CAD/CAM/CAE Systems*, Addison Wesley, 1999.
9. Wright, R. S. and Sweet, M., *OpenGI Superbible*, Waite Group Press, 1996.
10. Woo, T. C., "A Combinational Analysis of Boundary

Data Structure Schemata", *IEEE Computer Graphics and Applications*, pp. 19-27, March 1985.

11. 양희구, 김석일, 박천홍, 류근수, "CNC 호브 릴리빙 선반의 CAM 시스템 개발", 한국 CAD/CAM학회 논문집, Vol. 1, No. 2, pp. 150-157, 1996.
12. 이철수, 박광렬, "Heel angle 조정에 의한 터빈 블레이드의 5축 NC 가공 데이터 생성", 한국 CAD/CAM학회 논문집, Vol. 4, No. 2, pp. 110-120, 1999.
13. 조정훈, 서석환, "STEP-NC의 피쳐 기반 공구경로 생성 및 갱신", 한국 CAD/CAM학회 논문집, Vol. 4, No. 4, pp. 295-311, 1999.
14. 서석환, 지우석, 김성구, 홍희동, 조정훈, 정대혁, 김창남, "CNC Shop Floor 조업 지원용 다능형 CAM 시스템(PosCAM) 개발", 한국 CAD/CAM학회 논문집, Vol. 4, No. 4, pp. 339-349, 1999.
15. 이철수, 박광렬, "기준 평면과 경제 상자를 이용한 NC 절삭 과정의 그래픽 시뮬레이션", 한국 CAD/CAM학회 논문집, Vol. 2, No. 3, pp. 161-174, 1997.

주 성 욱



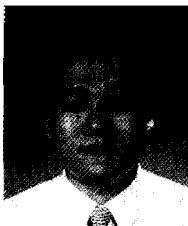
1998년 국민대학교 기계공학과 학사
1999년-현재 현대전자 연구원
관심분야: CAD/CAM NC Simulation

이 상 현



1986년 서울대학교 기계설계학과 학사
1988년 서울대학교 기계설계학과 석사
1993년 서울대학교 기계설계학과 박사
1993년-1995년 신도리코 기술연구소 책임 연구원
1996년-현재 국민대학교 자동차공학과 조교수
관심분야: CAD/CAM, Solid Modeling, Design, Car Body Design

박 기 현



1986년 서울대학교 전기공학과 학사
1989년 한국과학기술원 전기 및 전자공학과 공학 석사
1996년 한국과학기술원 전기 및 전자공학과 공학 박사
1997년-1999년 (주)터보테크 책임연구원
2000년-현재 (주)엔젠테크놀로지 대표이사
관심분야: Geometric Modeling, Computer Graphics, Machine Simulation