실시간 3축 NC 밀링 시뮬레이션을 위한 메쉬 간략화 방법

주성욱*, 이성현**, 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 다르게11,12,13,14, 생성된 NC 프로그램을 이용한 실제 가공에 앞서 하용 오차 범위를 넘는 과정작업이나 공구 간섭과 같은 NC 코드의 오류 여부를 컴퓨터상에서 시뮬레이션하는 연구들이 늘리 수행되어 왔으며, 현재 대부분의 상용 시스템들은 공작물의 실제 가공되는 상황을 실시간으로 화면상에 디스플레이에 주는 기능을 구비하고 있다11,12,13,14. 이를 위해 가공 중인 공작물의 형상을 표현하는 방법은 크게 솔리드 모델을 위한 자료 구조를 직접 이용하는 방법과 z-map이나 면의 범위 벡터들과 같이 불연속적인 값을 이용하여 표현하는 방법이 있다11,12,13,14. 솔리드 표현 방식은 수치값과 같은 형상을 포함하여 가공중인 공작물의 형상을 보다 원색하게 나타낼 수 있다는 장점이 있으나, 공구가 손상 지나간 물체와 공작물간의 불리한 작업(Boolean operation)을 수행하는데 있어 공작물의 교차 점선을 구하기 위한 계산 시간이 지나치게 많이 걸리는 단점이 있다. 또한 아니라 교차 점선 계산시 발생하는 오차로 말미암아 불리한 작업이 안정적으로 수행되기 어려운 점이 있 다. 이에 반하여 z-map이나 범위 벡터 표현 방식은 표현량 감소이나 방식에 따라 표현의 정확도는 떨어 지나 불리한 작업을 신속하고 안정하게 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
용중에, 여기서 z-map이라는 일정한 간격의 (x, y)에 대한 z 값을 2차원 배열 형태로 저장한 형태를 의미한다. 이 방법이 널리 사용되는 이유는 원래에 인접한 블록 사이로 시각화의 관리된 계산 속도가 빠르고, 알고리즘 단순하며, 자료 구조의 관리가 순연되는데다. 따라서 본 연구에서는 신속한 3채 NC 밀러 가공의 시각화선을 위하여 공작물을 표현하기 위한 자료 구조로서 z-map을 제시하였다.

Huang[21]와 온 형태 z-map 자료 구조를 사용하여 밀
링 공작을 실시간 시각화선하기 위해서 z-map의
축을 투영 방향(orientation)과 일치시키고 z-
map의 x, y 격자물 좌표값의 부분에 일치시킨 후,
그에 맞춰공작물에 대한 CAD 모델을 z-map으로
변환시킨다. 다음, 동작 지침에 따라 공구가 움직
이는 때공작물에 대한 z-map 모델을 재정하고 그 결과를
변환에 디스플레이 한다. 이 방법에서는 z-map 모델을
범위 좌표계의 중심으로 재정해야 하는데 사물상에
보는 것은 가장 가까운 것을 따르므로 볼도의
계산이 필요하지 않다. 이는 z-버퍼(z-buffer)의 원리
와 동일하고, 고속으로 디스플레이가 가능하다는 장
점이 있다. 그러나 화면에 이 방법은 다음과 같은 몇 가지 단점도 가지고 있다. 첫째, 사용자가 특히(views)
을 변경할 경우 부드럽게 그 단계까지 NC 시각화선을
반복해야 한다. 물론 z-map을 복원(voxel) 모델로
변환시켜 각 복원에 대해 좌표변환(transformation)
을 적용시키면 뒤따른 뷰에 따른 모델을 바로 얻을 수
있다. 이 방법을 사용하면 공작물 형태에 대한 시
각적인 점이 좋지 못하다. 둘째, 감기의 한계의 방향
이 뜻정방향과 일치하지 않은 경우가 발생할 수 있
는데, 이 방법에서는 점을 위한 z-map이 항상 뜻정방향
과 일치하기 때문에 이를 사용할 수 없는 문제
점이다. 실제로 3차 밀링에서는 공구의 방향이 z-
map의 z방향이 되는 경우가 가장 좋은 결과를
보이는 데 반하여, 시각적인 효과는 동축 뷰(isometric view)가 더 선호된다. 따라서 z-버퍼를 뜻정 방향과 복잡으로 가질 수 있도록 서로 분리시키는 것이 보다 바람직하다.

본 논문의 목적은 z-map을 이용한平面_game의 z-map을 제시하였다. 이 방법
은 z-map에서 일정한 x값에 대한 응용선을 선으로
그리고, 또 일정한 y값에 응용선들을 선으로 그
린다. 이 두 종류의 서로 지교환하는 응용선들을 그리
면 z-map의 간격이 종향에 따라서 각각 응용 처리
된 그래픽 결과가 나오게 된다. 또한, 응용선을 그리
면 입력된 두 선분이 허용 오차 한계에 드는 경우
이름으로 합쳐서 그려져 보였다. 실제 그려야 하는
선의 개수를 왜곡의 30%정도로 줄일 수 있다고 하
았다. 이 방법은 기존의 방법에 비하여 빠른 동작방
면의 각 인형의 결과를 제공한다는 장점이 가지고 있다.
그러나 디스플레이된 응용 장점의 정격이 가장 빠르고, 만
일 사용자가 확장해서 본다면 신인이나 신인사이의 범위
분리가 드러나게 되는 문제점이 있다. 따라서, 이 방법은
주로 연구에서 인터넷 유용성을 시각화형상에
세워 넣는 메시를 만들어 이를 네트워크의 연
구를 진행 중이라고 발표하였다.

이와 같이 큐에 육립적이면서 고화질의 결과를 얻기 위해서는 다각형 렌더링 방법(polygon rendering
method)을 사용할 필요가 있다. 이 방법은 많은 계
산 시간을 필요로 하기 때문에 실시간 시각화선에
그동안 사용되지 못하였으나 최근에 고성능의 다각
형 렌더링 이론을 갖춘 2차 그래픽 카드들이 속
화 개발됨으로써 보다 현실에 가까워 오게 되었다. 예
를 들어, 급히의 크기가 500mm×500mm이고 그
적 간격이 1mm인 경우 약 500,000개의 사각형
그려야 하며, 실시간 시각화선을 위해서는 대략
초 1초에 1000 이상의 활동 가속성이 이루어져야 하므로,
최소 5,000,000(polygons/sec)의 렌더링 속도를 가지
어야 한다. 또한 여기에 z-map 모델의 쌍방향 시간
을 고려한다면 이보다 더 빠르게 된다. 현재 3D게임
그래픽을 위한 전용 카드들은 보통 1초에 1-5액만
개의 사각형을 그릴 수 있는 수준에 끝이다. 그러나
이와 같은 그래픽 카드들은 대부분 가격이 고가와
문제점이 갖고 있으며, 또한 자동화 그래픽과 같이
같은 방법으로 1m를 넘는 대형 급히의 경우는 현재의
고성능 그래픽 카드로도 실시간 시각화선이 불가
능하게 된다.

따라서 본 논문에서의 큐에 육립적이며 고평질의
렌더링 결과를 보이는 NC 시각화선을 위해 다
각형 렌더링 방법을 사용하여, 이 방법의 속도를 향
상시키기 위한 방법을 제안하고자 한다. 그러므로써
고화질의 실시간 NC 시각화선이 보다 빠르고 그래
픽 카드에서도 구현될 수 있는 것을 도출하고자 하
한다.

2. 본 논문의 접근 방법

먼저 주어진 z-map에 대한 일반적인 기존의 큐 독
립적인 다각형 렌더링 방법을 먼저 살펴보도록 하겠다.
가중치 동작투과에 대한 응용 처리한 화상이 얻기
위해서는 일반적으로 OpenGL과 같은 그래픽 라

한국CAD/CAM학회 논문집 제5권 제3호 2000년 12월
이하의 함수를 이용하며, 이를 위해서 공작물의 가공면에 대한 삼각형 메쉬를 생성시켜주는 작업이 필요하다. 이 삼각형 메시키는 $z$-map의 각 경계에 대해 2개의 삼각형을 구성함으로써 이뤄진다. Fig. 1의 예에서 볼 수 있듯이 $z$-map에서도의 각 값이 갈수록 대단히 많기 때문에 공작물의 각 부분의 거리가 멀어질 때에 있어 상세한 형상을 빠르게 정의하기 위해 단일거리는 $z$-map 자료를 샘플링하여 네트워킹 수도 있으나, 각 부분의 복잡도를 고려하지 않은 이하의 방식은 공작물 형상을 정확히 나타내기 어렵다는 비판을 받지 않는다.

따라서 본 인 $z$-map 자료로부터 생성된 삼각형들을 다음을 계산하여 공작물의 형상의 정밀도를 유지하면서 이런 허용오차 한계 범위 내에서 시정 병역시에 삼각형의 개수를 줄일 수 있다면 반면에 속도를 자세히 세울 수 있으며, 경사 전면에 투입이 최소한의 속도 형상을 가질 수 있을 것이다. 병역의 판단 조건은 사용자와 작업의 방향 속도도 있고, 둘 이상의 변수를 고려하여 자동으로 결정되도록 할 수도 있다.

다각형의 메시키 주어질 때, 그로부터 삼각형 개수를 줄이는 방법은 주로 컴퓨터 그래픽스 분야에서 다각형의 정면 위치를 입력한 정보에 따라 이들 사용자의 상세 수준(level of detail)에 맞추어 다소의 하위 메시키하기 위한 목적으로 개발되어 왔다. 즉, 추가의 문제가 상세적으로 멀어 있을 때는 작은 수의 다각형만을 그리고, 가까이 있을 때에는 많은 수의 다각형을 그릴 수 있도록 여러 개의 다각형을 이용하여 공작물의 형상에 맞게 변형되는 방법은 Schroeder 등(19)의 연구를 시작으로 수많은 연구 결과들이 보고되었으며, 그 대표적인 방법들을 선정하여 비교한 결과가 Cignoni 등(20)의 논문에 소개되어 있다. 이들 방법은 또한 층계의 대상에 따라 크게 복잡하고 주어진 개시에서 선행자, 보수, 분할에서의 방법(edge collapse method)이다.

그러나 이러한 기존의 메시키 간략화 알고리즘들은 사용하는 모델에 비하여 본 연구에서의 렌더링 대상인 NC 밀링 머신상의 공작물들의 형상은 다음과 같은 특성을 가지고 있다.

* 기존의 메시키 간략화를 위한 모델은 원래의 형상을 변경할 필요 없으며 단지 필요에 따라 여러 상태 수준의 형상을 빠르게 표현하기 위한 기능을 반하며, 가공중인 공작물은 NC 형상이 수행될 때마다 그 형상이 계속 바뀌며,

* NC 프로그램의 각 공구 이동 명령마다 변환된 공작물의 렌더링을 하는 것이 일반적이며, 이러한 메시키 간략화 공작물의 형상 변화는 복잡한 본 연구로서는 위의 기존의 연구결과의 모델과는 다릅니다.

한편, 전체 렌더링 시간은 메시키 간략화 시간과 렌더링 시간의 합이기 때문에 이 시간이 메시키 간략화 없이 그냥 렌더링 할 때보다 항상 더 빠르게 해야 연구의 의의가 있다고 할 수 있다. 실제로 3개의 고유한 고유한 CPU의 성능이 우수한 경우 메시키 간략화를 수행시기의 결과는 그보다 그림의 메시키라는 것이 그보다 더 빠르다. 따라서 본 논문에서는 위의 가공중인 공작물 형상의 특성을 고려하여 하드웨어의 성능에 관계없이 메시키 간략화를 수행하지 않고 그만 그림의 메시키를 고려하지 않는 간략화 방법을 개발하고자 한다.

[발단 1] 전체 메시키 간략화 방법

우리는 먼저 Fig. 2의 나타난 것과 같이, 일부 공구의 이동에 대하여 변환된 공작물의 형상에 그림에서 보다 원활한 전진에 대하여 간략화 판도 작용을 수행하는 방법을 먼저 생각해 본 수 있다. 그러
나 이 방법을 적용한다면 삼각형의 길이가 높은 경우에는 어느 정도의 렌더링 속도 향상을 기대할 수 있으나, 길이가 낮은 경우에는 삼각형 길이 작업 시간이 더해지게 되어 결과적으로 전체 렌더링 시간이 오히려 증가하게 되는 결과를 초래할 수도 있다. 또한, 이 방법은 하드웨어의 성능에 따라 그 효과가 달라진다. 실제 고성능의 PC에서는 원래 메시를 그냥 그대로 그는 것이 삼각형의 개수를 줄여서 그려지는 것보다 소요 시간이 오히려 더 짧은 것으로 전달되었다.

[방안 2] 영역 분할 및 메쉬 초기 간략화 방법

이 방법은 Fig. 3에서와 같이, 초기에 공작물은 몇 개의 영역으로 분할한 다음, 각 영역에 대해 메시를 생성하여 간략화하는 방식으로, 이후 공구에 의해 형상이 변화된 영역에 대해서는 z-map을 그대로 이용하여 렌더링하는 방식이다. 즉, 이 경우에 초기에 간략화된 메시들과 이후 만들어진 간략화되지 않은 메시가 혼재하는 형태가 된다. 이 방법을 사용하면 최초 초기화 단계에서 삼각형 길이 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 확장 초기에는 삼각형 길이 효과가 있으나 일반 공작물 전체에 대하여 공구가 반복 지나가게 되면 그 이후에는 z-map을 그대로 그리는 것과 동일한 상태가 된다. 하지만 아예된 사물라이온에 소요되는 시간은 항상 z-map을 그대로 그려는 것보다는 빠르다.

[방안 3] 영역 분할 및 공구 대치 주기적 간략화 방법

이 방법은 [방안 2]의 방법과 초기는 동일하나, 주기적으로 변화된 구비 영역에 대한 메시들의 간략화 작업을 수행시키는 방법이다. 주기는 메 공구 이동 속도 있고, 여러개의 공구 이동으로 정할 수 있다. 삼각형 작업이 수행될 때에는 그 작업의 수행 시간 만큼의 일시적인 속도의 저항이 있었지만 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 보여져 있다. 본 논문에서는 [방안 3]을 채택하였으며 이를 구현하기 위한 구체적인 자료 구조와 알고리즘은 다음 장에 소개하도록 한다.

3. 전체 알고리즘 개요

본 논문에서 제안한 공구 메쉬 간략화 알고리즘은 다음과 같다.

1. 초기화 작업을 수행한다. 여기에서는 공작물의 조각번호를 일정한 크기의 작은 영역으로 분할하고, 분할된 각 영역에 대해 삼각형 길이 작업을 수행하여 그 결과를 저장한다.
2. 공작물 형상계산 모듈이 하나의 NC 공구의
동 면역을 얻어서 공작물의 z-map 자료를 채집한 후, 그 z-map 자료와 공구가 쓰고 갈 이동 범위를 센터링 모듈에 넣어준다.

3. 센터링 모듈에서는 입력받은 공구 이동 범위와 조각이기도 중동의 영역에 대해서는 z-map으로 전환시키고 z-map의 모든 객체에 대해서 삼각형을 만들어서 그리도록 하며, 그 외의 영역에 대해서는 간소화 메커니즘을 이용하여 그리도록 한다.

4. 만만 사용자에서 저장한 삼각형 감소 작업 주기에 해당하는 경우에는 z-map으로, 환원된 영역들에 해당 삼각형 감소 작업을 수행한 후 2로 돌아간다.

주기에 해당하지 않는 경우에는 감소 작업 없이 바로 2로 돌아간다.

그리면 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 메시지 저장 자료 구조이다. z-map은 일정한 간격의 격자들의 \( z \)-값을 실수의 2차원 배열 형태로 저장하였다. 또한 각 가공물을 분할한 각 영역에 대한 자료는 Fig. 5에 나타난 것과 같이, 영역의 고유 일련 번호(id). 이 영역이 z-map으로 그리 놓으면서 삼각형 감소 결과로 그리질 것을 나타내는 플래그(reduction display flag)가 있다. 이 영역이 z-map상에서 어떤 부분을 차지하는 지를 나타내는 최대 및 최소 \( x, y \)-값(min_x, min_y, max_x, max_y), 감소 작업 결과 남아있는 삼각형들의 세 봉주리점의 벡터(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 저장하고 있다. 경계에 대한 데이터는 자기에게 해당하는 자료만 저장하며, 영역이 z-map 곡면 내부에 위치해 있는 경우에는 아무런 자료를 가지고 있지 않다. 센터링 시에는 각 영역에 대해 그 영역의 reduction display flag가 true로 설정되었으나 그 영역은 그 이하로 저장된 각각의 삼각형들과 경계 라인들의 데이터를 사용하여 그려지고, false로 설정되었으면 그 영역은 z-map로부터 삼각형을 만들어서 그려하게 된다.

또한, 본 논문에서는 삼각형 감소 작업을 위한 자료 구조로서 winged-edge 자료 구조를 사용하였다. 이는 불필요한 삼각형을 제거하는 중간 단계에서는 삼각형과 같은 비삼각형 형상이 발생할 수 있으며, 또한 심숙한 삼각형 감소 작업이 이루어지기 위해서는 일정한 면, 모서리, 꼭지점들에 대한 효과적인 검색이 가능해야 한다. 이를 위하여 본 연구에서는 winged-edge 자료 구조를 선별하여 사용하였다. 그 이유는 이 자료 구조가 다양한 곡면을 표현하는 자료 구조들 가운데 최적의 자료 저장 및 검색의 효율성을 가진 것으로 알려져 있기 때문이다.

삼각형 감소 작업을 위해서는 먼저 해당 영역의 z-map 데이터로 winged-edge 자료 구조로 변환한 후, 삼각형 감소 작업을 수행하고, 그 결과를 Fig. 5의 영역 자료 구조에 저장시키게 된다. 그리고 하나의 영역에 대한 삼각형 감소 작업에 대하여 보다 세세한 설명하도록 하겠다.

5. 국부 영역에 대한 삼각형 감소 방법

제3 z-map 곡면을 일정한 크기의 영역으로 분할한 후 각 영역에 대해 삼각형 메시지를 생성시킨 후 이를 간략화시키는 과정은 다음과 같다. 이 과정은 최초에 전체 영역들에 대하여, 그리고 한 감소구간마다 그 사이 분할된 영역들에 대해서 수행되어진다.

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

2. 삼각형 메시에서 하나의 폭지점을 선택하여 제거 가능한 폭지점인지 판단한다.

3. 제거 가능한 폭지점으로 판단되면 폭지점을 제거하고 폭지점과 연결된 위상 요소들의 데이터를 수정하게 한다.

4. 모든 폭지점에 대해 검사를 이루어졌는지 판단하여 아직 검사를 받지 않은 폭지점이 있으면 위의
단계로 돌아간다. 만약 모든 검사가 이루어졌으면 감소 작업의 결과를 실수 배열로 저장하고 이 과정을 반복한다. 여기서 검사 작업의 결과는 모든 삼각형들의 세 꼭짓점들과 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 매쉬 간략화를 수행하기 위한 단계로서 주어진 z-map 데이터에서 저장된 영역에 해당되는 z-map 자료들을 취하여 winged-edge 자료 구조에 맞추어 변환 저장시킨다. 이 단계는 대량의 자료를 변환시키는 작업으로 많은 시간을 소모할 우려가 있으므로, 우리는 z-map으로부터 만들어진 메쉬의 토폴로지를 미리 정하고 이에 맞추어 해당 자료를 저장하는 방식을 사용하여 변환 시간을 최소화시킨다. 즉, 각 모서리, 꼭짓점, 면의 고유 번호를 일정한 규칙에 따라 부여한 후, 이들을 메모리 정해진 토폴로지를 만족하도록 각 주변 요소내의 포인트들을 채워 넣었다. 또한 각각 영역에 대한 감소와 작업을 할 때마다 winged-edge 자료 구조의 메쉬를 위한 메모리 영역을 새로 할당하고, 사용 후 다시 지우는 것은 반복하는 것을 많은 시간을 소비할 우려가 있으므로, 최초 한 번만 메시에 대한 메모리 영역을 할당한 후 이를 시뮬레이션을 끝날 때까지 반복해서 사용하도록 하였다.

5.2 제거 가능한 꼭짓점들의 선택
이 단계에서는 메쉬를 구성하는 꼭짓점 가운데 하나를 선택하여 이 꼭짓점이 제거 가능한 꼭짓점인지지를 판단하는 작업을 한다. 본 논문에서 사용한 제거의 가능성에 판단하는 기준은 두 가지로, 첫째, 꼭짓점을 제거할 경우 주어진 허용 오차를 넘어서항성이 간략화되는지를 먼저 판단하고, 둘째, 꼭짓점을 제거할 때 면적이 0으로 하락하거나 위치하는 면이 발생하는지의 여부를 판단적으로 한다. 그 럼에 두 가지 시험 방법의 기준에 대해서 보다 상세히 설명하도록 하겠다.

5.2.1 꼭짓점의 허용 오차 초과 판단 테스트
이 테스트는 허용한 각 꼭짓점의 각자의 오차값을 저장할 수 있도록 자료 구조 선언에서 꼭짓점에 error 필드를 지정해 준다. 이 오차 값은 초기에 0으로 설정되어 있으며, 모서리로 연결된 인접 꼭짓점이 소거되면서 소거된 오차가 누적되기 때문. 입

의 꼭짓점이 소거될 때, 인접한 꼭짓점에 부과되는 오차값은 다음의 방법에 의하여 계산되어 누적되며, 이를을 에서 제시된 방법을 원용하였다. 이를 간략히 설명하면 다음과 같다.
먼저, 각각 대상 꼭짓점을 둘러싸고 있는 면들의 가장 평균 폭면을 계산한다. 이 폭면상의 한 점은 각 면들의 중심의 가중 평균이고, 벡선 벡터는 각 면들 의 벡선 벡터의 가중 평균값을 취함으로써 평균 폭면을 구할 수 있다. Fig. 6(a)에서와 같이 각 각 면(F)의 단위 벡선 벡터(Nm)n에 대한 평균 벡선 벡터(Nw)은 Nw=\sum Nm/n, 각 면의 중심점(Pm)n에 대한 평균점 (Pw)=\sum Pm/n와 A로 구할 수 있다. 다음, Fig. 6(b)에서 나타난 것과 같이 이 평균 폭면으로부터 소각 대상 꼭짓점까지의 거리 d를 계산한다. 이 거리와 그 동안 이 꼭짓점에 누적된 오차의 합이 허용 오차를 넘지 않는다면 이 꼭짓점은 소거할 수 있는 것으로 판단한다. 꼭짓점을 소거할 때는, Fig. 6(c)에서와 같이, 그 꼭짓점에 모서리를 통하여 연결되어 있는 꼭짓점들에게 d만큼을 오차로 누적시킨다. 따라서 이와의 꼭짓점이 소거될 수 있는가 하는 것에 대한 판단은 그 꼭짓점에 누적된 오차가 허용 오차를 넘지 않는지를 살펴서 결정한다.

5.2.2 점집하기나 퇴회되는 면의 발생 여부 테스트
본 논문에서는 꼭짓점을 소거하기 위하여 모서리
봉괴(edge-collapse) 방식을 채택하였다. 즉, 소거할
모서리점에 연결되기 있는 모서리들 가운데 하나를 묶
지점과 동시에 없이며, 이를 위하여 Euler 연산자[1]
를 구현하여 사용하였다. 이때 주의해야 할 점은 한
모서리를 없애므로써 배위의 변 가운데 면적이 0으
로 되어하거나 뒤집히는 범이 발생할 수가 있다는
것이다. Fig. 7(a)의 예에서 같은 경우에는 모서리
E이 붕괴될 때 생존하는 모 F5의 면적이 0이 된다. 또
한 Fig. 7(b)의 예에서 같은 경우에는 모서리 E이 붕괴될 때 생존하는 모 F5가 뒤집히게 된다. 뒤집히
이와 같은 부적절한 토폴로지를 갖게되는 경우 이후
간단화 과정에서 위상 요소들의 인접 정보를 제대로
알 수 없을 뿐만 아니라 뒤집히는 면은 편터링시
그 부분에 구멍이 난 것과 같은 효과를 가질 수 있
다. 따라서 이와 같은 현상이 발생하지 않도록 제
제한 모서리를 선택할 때 검사해야 한다.

본 논문에서는 적지점 제거 후에도 가능하게 원래
와 가까운 형상을 유지하기 위하여 연결된 모서리들
중에서 가장 많은 모서리를 제거할 모서리로 먼저
선택해 본다. 그러나 Fig. 7에서 왼쪽의 같은 가장
많은 모서리를 없애 경우 뒤집히지 발생하지 않으나,
여기 모서리 가운데 뒤집히지 발생하지 않는 모서리를
선택하여 제거하도록 한다. 이러한 뒤집히는 둘의 발
생 여부는 적지점 제거 후의 각 면의 변형 방향을 미리 계산하여 그 방향이 (-x) 방향 성분을 가지고
있으면 뒤집힌 것으로 판단한다. 또한 면적이 0이 되
는 경우에 각 면의 벤선을 구하기 위해 왜곡을 취했
을 때 그 크기가 0이 되는 경우에 해당하므로 이를
판별하면 된다.

본 논문에서는 면을 뒤집히거나 뒤집히지 않게 발생되
는 경우 대상 모서리를 바꾸어 이 판단 기준을 통과할
때까지 점자를 반복한다. 면이 이러한 모서리도 이를
만약시키지 못하면 그 꼭짓점은 제거하지 않는다.

5.3 꼭짓점의 제거

어떤 질지점과 모서리가 결정되면 그에 따라 꼭
지점의 토폴로지를 동력화 면에 대해 있는 새로운 토폴로지
를 부여해 주어야 한다. 이들 수행하는 방법은 사전적
효익점을 둘러싼 면들을 모두 소거한 후 새로운 면을
생성시키는 전체 수정 방식과 기존의 면들 가운데 일
부만 없애고 기하학적 정보를 수정하는 부분 수정 방
식이다. 본 논문에서는 후자인 부분 수정 방법을
채택했으며 그 과정은 내부 꼭짓점과 경계 꼭짓점의
경우로 분류하여 각각에 대해 정해진 방식의 슬리
드 모델링에서 분리 사용되는 Euler 훼스를 사용하여
프로그래밍을 구현하였다. 그러면 내부와 경계 꼭짓점의
제거 방법에 대하여 각각 설명하도록 하겠다.

5.3.1 내부 꼭짓점의 제거

Fig. 8(a)에서 영역의 내부에 위치한 꼭짓점 E가
앞의 제거 조건을 만족하며 또한 여기에 연결된 가
장 많은 모서리 가운데 하나인 모서리 11을 검사한
결과 비정상적인 토폴로지의 발생은 일어날 가능성이

한국CAD/CAM학회 논문집 제5권 제4호 2000년 12월
이 없는 것으로 판정되었으나, 때때로, 이 편지침과 모서리를 없애는 과정은 다음과 같다. 먼저, 모서리 11에 점점해 있는 끝 C와 모서리 6을 오일러 적합한 끝 KEF를 사용하여 동시에 소거한다. 다음, 모서리 11에 점점해 있는 끝 H와 모서리 12를 마찬가지로 KEF를 사용하여 동시에 소거한다. 그 결과가 Fig. 8(b)에 나타나 있다. 끝으로, 소가 대상인 끝지점 e와 모서리 11을 KEV를 사용하여 소거한다.

5.3.2 격계 편지점의 제거
영역의 경계에 위치하고 있는 끝지점의 제거 작업의 기본적인 알고리즘은 내부 편지점의 제거 알고리즘과 거의 동일하지만 없어진 모서리의 선택과 KEF의 과정에서 차이가 난다. 내부 편지점의 없어진 모서리의 선택한 때는 가장 많은 모서리를 선택했지만 경계에서는 경계를 이루고 있는 두개의 모서리 중 하나만이 없어질 수 있다. Fig. 9의 경우, 없어질 끝 편지점이 끝 b가 있고 없어진 모서리로 6가 선택되었으나, 경우 외의 내부 편지점의 제거 알고리즘과 그대로 적용하면 편지점 b가 e의 위치로 뒤바뀌어 경계 선이 약간 안쪽으로 들어가다 나오는 형상이 되고 난 것이다. 이러한 현상을 방지하고 경계선을 직선을 유지하도록 하기 위해서는 제거된 모서리로는 경계선에 위치하고 있는 모서리 1, 4, 7 중 하나만이 되어야 한다. 경계선 상의 편지점을 제거할 때는 내부 편지점을 제거한 범위의 편지점과 파리 변한의 KEF와 변한의 KEV를 수

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

행한다. 즉, Fig. 9의 경우, 없어진 모서리로 모서리 4가 선택되었다면 우선 KEF과정으로 없 D와 모서리 5가 없어지지 않고 KEV과정으로 편지점 b와 모서리 4가 삭제된다.

5.3.3 편지점 제거 순서
영역에 대한 예제를 간략화 과정에서 제거할 편지점을 선택하는 방법은 몇 가지 간략화의 소요 시간과 매우 일관된 관계가 있다. (a) Fig. 10(a)에서의 값이 인접한 편지점을 계속하여 검사하고 제거하는 것이 더 정확한 결과를 가져오지 못하는 결과로, 모서리의 수가 계속 증가 하여 검사에 많은 시간을 소비하게 된다. 따라서 본 연구에서는 반도를 발생시키게 Fig. 10(b)에서의 값이 인접한 편지점을 무작위로 선택, 경계함으로써 편지점에 필요 이상으로 많은 범위는 편지점이 인접하는 것을 방지하였다.

다만 본 연구의 경우 결과가 무작위로 편지점을 제거할 경우는 순차적으로 편지점을 제거할 경우보다 결과 예제의 삼각형의 수가 다소 많은 경향을 보인다. 이는 무작위로 편지점을 선택, 제거하는 과정에서 간족 5.3.2에서 언급한 제거 조건을 명확하게 관찰하는 경우가 발생하기 때문이다. 하지만 이러한 삼각형 개수의 차이는 크지 않기 때문에 랜덤 속도에 영향을 주지 않으며 오리지널 보다 빠른 간략화를 얻게 되므로 충분히 보완될 수 있다.

6. 적용 사례 및 토의
6.1 활용 가능성의 예
먼저 Fig. 1에서와 같은 직선 경로로 공작물을 왕작 가능성을 경우에 대해 삼각형 감소 기법을 적용시키지 않고 그냥 z-map 자료를 랜덤한 경우와 본 논문의 감소 기법을 적용시켜 랜덤한 경우와 본 논문의 감소 기법을 적용시켜 랜덤한 경우와 본 논문의 감소 기법을 적용시켜 랜덤한 경우의 업계 개수와 소요 시간의 변화를 측정해 보았다. 이

![Fig. 10. Vertex removal methods (a) sequential removal method (b) random removal method.](image)
기자의 z-map 자료를 그대로 랜더링하는 것은 터보텍의 CANVAS 시스템의 프로그램을 사용하여 수행하였다. 하드웨어는 현재의 보급형 인 Pentium-II 350MHz CPU와 OpenGl 그래픽 가속 기능을 갖 추지 않은 ATI RAGE IIC 그래픽 카드를 장착한 PC를 사용하였다. 공작품은 140 mm×140 mm의 크기로서 1 mm 간격의 격자점들에 대한 z-map 데이터를 생성시켜 사용하였다. 절삭 공구로서는 블 렌더 면을 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 랜더링 시간
주어진 삼각형 감소 주기와 오차에 대하여 매시 간격이 진행되는 국부 영역의 크기를 5, 8, 10, 15, 20, 30으로 증가시키면서 얻어진 랜더링 시간의 변화를 Fig. 11에 나타내었다. 여기서는 감소 주기로서 1은 초기에 1의 감소를 수행한 후 이후에는 감 소가 수행되지 않은 것을 뜻하며, s는 해상 오차를 뜻한다. 이 그래프에 나타난 바와 같이 국부 영역의 크기에 따라 랜더링 시간과는 어떤 일반적인 상관 관계가 성립되지 않음을 알 수 있다. 단, 어떤 주어진 NC 프로그램과 공작물에 대하여 국부 영역의 크기를 최 적화시키는 것은 연구 대상이 될 수 있을 것이다.

6.1.2 해상 오차에 따른 삼각형 감소율과 랜더링 시간
국부 영역의 크기를 8x8로 고정하고, 삼각형 감소 주기를 매 20번의 NC 코드 수행으로 설정하고, 해상 오차를 0.01, 0.05, 0.1로 변화시키면서 사물레이션 과정에서 얻어진 총 삼각형 개수와 랜더링 시간의 변화가 Table 1에 나타나 있다. 여기서 s는 mm 단위의 해상 오차를 나타내며, s는 감소 주기를 NC 명령의 개수로 표시한다. s-map으로 표시된 첫 번째 열은 삼각형 감소 개수를 수행하지 않고 z-map으로부터 직접 삼각형을 생성시키고 그러한 경우를 나타낸다. 두 번째 열은 최초에 한 칸체매 삼각형 감소를 시키고 그 이후로 변경된 영역에 대해서는 z-map 자료로부터 직접 삼각형을 생성시키고 그러한 경우를 나타낸다. 나머지 세제부터 다섯째 열은 해상 오차를 0.01, 0.05, 0.1로 변화시키면서 총정한 그려진 삼각형 개수와 랜더링 시간을 나타낸다. 해상 오차를 늘리하게 될수록 랜더링에 사용되는 삼각형의 수와 시간이 감소하는 것을 볼 수 있다.

Fig. 12와 Fig. 13은 사물레이션에 진행됨에 따른 삼각형 개수와 랜더링 시간의 변화를 각 그래프로 나타낸 것이다. Fig. 13에 나타난 것과 같이 랜더링 시간은 전체적으로 계단식 형상 혼합을 보이고 있으며, 삼각형 감소주기에 해당하던 해상식 삼각형을 좀 더 시간이 경과함에 따라 점차적으로 사라지는 모습을 보이고 있다. 단, 이 해상 오차를 증가시키면 각 경우에 대한 추세선을 직선으로 할 수로 나타났다. 여기서 주의할 점은 가공 후반부로 가면서 삼각형을 감소함으로 랜더링하는 (2)나 (3)의

Table 1. Effect of mesh decimation according to error allowances

<table>
<thead>
<tr>
<th>z-map</th>
<th>e: 0.01</th>
<th>e: 0.05</th>
<th>e: 0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>s: ∞</td>
<td>s: 20</td>
<td>s: 20</td>
<td>s: 20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No. of faces (100%)</th>
<th>9996000</th>
<th>5800968</th>
<th>4968194</th>
<th>4604058</th>
<th>4020966</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display time (sec)</td>
<td>156.38</td>
<td>120.31</td>
<td>107.86</td>
<td>105.22</td>
<td>93.04</td>
</tr>
</tbody>
</table>

한국CAD/CAM학회 논문집 제5권 제4호 2000년 12월
정우 삼각형의 개수가 많거나 적은 배에도 불구하고 z-map을 그날 렌더링하는 (1)의 경우보다 시간이 더 걸리는 것을 볼 수 있다. 이는 z-map을 바로 그려지는 경우보다 좀 더 많은 스크린의 면적과 영역별로 분산된 데이터를 접근하는 데 보다 많은 시간이 소요되기 때문이다. 또한 현재의 프로그램이 다소 이해하기 어렵기 때문에 판단된다.

6.1.3 감소 주기에 따른 렌더링 시간

Table 2는 곡부 영역의 크기를 8×8로 설정하고 하중 오차를 0.05로 고정하고 감소 주기를 변화시킬 때 시뮬레이션 전체에 걸쳐 브리스플레이되는 총 면의 개수와 렌더링 시간이 어떻게 변화되는지를 관찰한 것이다. 감소 주기가 늘어날수록 브리스플레이되는 면 의 개수는 점차 증가한다. 렌더링 시간도 대체로 주가 늘어날수록 점차 증가한다. Fig. 14에 나타난 것과 같이 감소 주기가 아주 빠르게 바뀌는 시간이 크게 증가하는 환경을 보인다. 이는 삼각형 감소 작업으로 인한 시간의 증가가 그로 인해 줄어드는 삼각 형의 렌더링 시간보다 더 커지기 때문에 판단된다. 회의의 삼각형 감소 주기는 NC 프로그램의 내용. 전체 z-map을 포함하는 영역의 크기. 그리고 하드웨어 의 성능 등에 좌우된다.

Table 2. Effect of mesh decimation according to decimation periods

<table>
<thead>
<tr>
<th></th>
<th>e: 0.05</th>
<th>e: 0.05</th>
<th>e: 0.05</th>
<th>e: 0.05</th>
<th>e: 0.05</th>
<th>e: 0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>s: 1</td>
<td>s: 10</td>
<td>s: 20</td>
<td>s: 30</td>
<td>s: 40</td>
<td>s: 50</td>
</tr>
<tr>
<td>No. of faces</td>
<td>9,998,000 (100%)</td>
<td>4,217,932 (42%)</td>
<td>4,487,008 (44%)</td>
<td>4,604,058 (46%)</td>
<td>4,698,151 (47%)</td>
<td>4,820,145 (48%)</td>
</tr>
<tr>
<td>Display time (sec)</td>
<td>156.38 (100%)</td>
<td>126.04 (81%)</td>
<td>102.84 (66%)</td>
<td>105.22 (67%)</td>
<td>108.20 (69%)</td>
<td>108.81 (70%)</td>
</tr>
</tbody>
</table>

한국CAD/CAM학회 논문집 제5권 제4호 2000년 12월
Table 3. Variations of the number of faces and the display time with respect to error allowances

<table>
<thead>
<tr>
<th></th>
<th>z-map</th>
<th></th>
<th>e: 0.01</th>
<th>s: 0.01</th>
<th>e: 0.05</th>
<th>s: 0.05</th>
<th>e: 0.1</th>
<th>s: 0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of faces</td>
<td>2.88x10^6 (100%)</td>
<td>2.85x10^6 (99%)</td>
<td>2.20x10^6 (77%)</td>
<td>1.78x10^6 (62%)</td>
<td>1.54x10^6 (53%)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Display time (sec)</td>
<td>6.106 (100%)</td>
<td>7.226 (118%)</td>
<td>5.744 (94%)</td>
<td>4.926 (81%)</td>
<td>4.825 (67%)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

10.273개의 공구 이동 명령을 포함하고 있다.
6.2.2 허용 오차에 따른 삼각형 감소율과 디스플레이 시간
삼각형 감소 주기를 20으로 고정하고 허용 오차를 0.01, 0.05, 0.1로 변환시켜서 이에 따른 디스플레이 된 총 면의 개수와 소요 시간의 변화를 측정해 보았 다. 그 결과는 Table 3에 나타나 있듯이 허용 오차가 클수록 면의 개수와 디스플레이 시간이 감소하였다.

시뮬레이션의 전과정에서 실제 디스플레이된 면의 개수의 변화 추이를 살펴보면 Fig. 16에 나타난 것 과 같다. 먼저, 황색 단계를 상대로 초기 2000 블록 이에 해당된다. 이 구간 허용 오차에 대하여 삼 각형 감소율은 황색 단계 진행 중에는 거의 일정한 상태를 유지한다. 또한 Fig. 17에서와 같이 디스플레이에 소요되는 시간도 정상 단계에 비해 황색 단계 에서 많이 걸리는 것을 볼 수 있다.

그러나 정상 단계에 접어들면 가공이 진행됨에 따라 점점 디스플레이되는 면의 개수가 감소되며,
Fig. 17에서와 같이 디스플레이에 소요되는 시간도 확산 단계에 비해 정상이 진행되면서 점점 감소하는 것을 볼 수 있다. Fig. 17에서 각 그래프의 가운데 있는 작은 선은 녹색의 이동을 드러기 위해 매 207시간의 평균값을 그려 그래프의 추이를 나타낸 것이다.

7. 결론
본 논문에서는 삼각형 감소 기법을 이용하여 3차 NC 밀링 작업의 실시간 시뮬레이션의 속도를 향상 시키는 새로운 방법을 제안하였다. 이 방법은 먼저, z-map으로 표현되는 점차중인 소세할 가공면을 일정 한 크기의 소 영역으로 분할한 후 각 영역에 대해 삼각형 메쉬를 주어진 허용 오차 한계 내에서 서로 병합시켜 수볼 감소시키며 그 결과를 저장하고, 공 구가 그 영역에 들어오기 전까지는 감소된 메시로 그리고 공구가 영역에 들어오면 읽은 z-map을 바 로 그린 방법을 취하는 것이다. 그리고 앞서 가더라도 z-map상대로 그리는 영역들에 대한 삼각형 감 소 작업을 수행하도록 하였다. 이 방법은 시뮬레이션 초기에 한변의 감소 작업을 수행한 후 가공이 끝 난 때까지 언제나 z-map으로부터 직접 삼각형을 생성하여 그리는 것보다 환상 속도가 빠르며, 하드웨어의 성능에 관계없이 언제나 시뮬레이션의 속도 항상 가능할 수 있는 방법이다. 또한, 가공 도중에 일정 주기를 z-map으로 환산된 영역에 대해서 감소 작업을 수행시켜야 줄여지는 핸디파일 속 도를 더욱 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

한국CAD/CAM학회 논문집 제 5권 제 4호 2000년 12월
는 일반적으로 기계 장소와 접속 효율성이 최적에 가까운 구조나 특별한 응용 분야에 대해서는 가장 효율적이라고 할 수 있다. 따라서 효율성은 보다 높이기 위해서는 본 체계로 감소 작업에 대해 최적인 자료 구조를 고안할 필요가 있다[11].

둘째, 영역의 크기와 상작형 감소 수행 주기를 추여 진 NC 프로그램에 효율성이 되도록 시뮬레이션 수행 이전 단계에서 결정해 주는 방법의 개발이 필요하다.

셋째, 다각형 렌더링 방법은 뉴에 독립적인 방법이다. 그러나 현재와 같이 모델 공간상에서의 허용 오차를 사용자가 지정하는 것보다 현재와의 부 상대를 고려하여 사용자가 감지할 수 없는 수준의 허용 오차를 자동으로 계산하는 방법이 사용자에게 편리할 수 있으므로 이를 수용할 필요가 있을 것이다.

감사의 글

본 연구는 (주)티보텍의 지원을 받아 수행되었으며, 관심자 여러분께 감사 드립니다.

참고문헌

10. Woo, T. C., “A Combinational Analysis of Boundary

11. 양희구, 김창일, 박현호, 류현수, “CNC 효율 모델링

12. 이철수, 박광림, “Heel angle 조절에 의한 터빈 플레

14. 서석환, 최유식, 김성구, 홍희동, 조봉호, 정태혁, 김창

15. 이철수, 박광림, “기존 평면과 경계 상자를 이용한 NC


주 성 목
1998년 국민대학교 기계공학과 학사
1999년-현재 한체 현대전자 연구원
전공분야: CAD/CAM NC Simulation

이 상 현
1986년 서울대학교 기계설계학과 학사
1988년 서울대학교 기계설계학과 석사
1993년 서울대학교 기계설계학과 석사
1993년-1995년 신라조공 기술연구소 책임
연구원
1996년-현재 국민대학교 자동차공학과 조
표수
전공분야: CAD/CAM, Solid Modeling, Design, Car Body Design

박 기 현
1986년 서울대학교 전기공학과 학사
1990년 한국과학기술원 전기 및 전자공학과
공학 석사
1996년 한국과학기술원 전기 및 전자공학과
공학 박사
1997년-2003년 (주)티보텍 책임연구원
2003년-현재 (주)티보텍온 произ 표표시
관련분야: Geometric Modeling, Computer Graphics, Machine Simulation

한국CAD/CAM학회 논문집 제5권 제4호 2000년 12월