

Java 3D를 이용한 가상 선반 시스템의 개발

이재호*, 김경철*, 이 준**, 이종태**, 박준영***

Development of Virtual Lathe System Using Java 3D

Lee, J. H.*, Kim, K. C.*, Lee, J.**, Lee, J. T.** and Park J. Y.***

ABSTRACT

We developed an on-line lathe simulator which can be used for a CAM education on the internet. Previously, VRML or static images such as JPG have been commonly used to achieve the same goal. With the 3D tool like VRML, it was possible to describe the movement of 3D object effectively. However, G-code based modeling and real-time visualization for the cutting process including parts being machined could not be accomplished. By using the simulator, we can machine a part with G-code input from a user or modeler, and parts being machined can be visualized in 2D as well as 3D. The developed system is based on EMCO lathe from Eshed and can be used through a web browser.

Key words : Virtual lathe, G-code, Java 3D

1. 서 론

CAD/CAM에서와 같이 고가의 소프트웨어가 필요한 응용 분야에서는 교육현장에서 다량의 소프트웨어를 구매하여 교육하기 어려운 실정이다. 이와 같은 문제점을 해결하기 위해서 교육현장인 대학이나 산업계 직업 훈련 기관 등에서 CAD/CAM 및 CNC에 관한 이론 및 실습을 교육하기 위한 상업용 솔루션들이 등장하였다. 예를 들면, (주)큐빅 테크의 V-CNC 등이 그것인데 이러한 제품은 비교적 저가인 CNC 밀링과 선반 등에 관한 제반 공정들을 실제 시뮬레이션을 통해서 구현해 볼 수 있다는 장점을 지니고 있다¹⁾. 그러나 PC 기반의 시스템들은 웹을 통한 교사 학습이나 온라인 에이전트와의 연계 등의 교육 효과를 배가시키기 위한 확장성과 비용 면에서 여전히 한계점을 가지고 있다.

이에 따라 현재 급속도로 확산되고 있는 인터넷을

이용한 온라인 교육 시스템의 필요성이 대두되었다. 이를 위한 대표적인 것이 웹 브라우저를 이용한 공학 교육의 시도들이다. 그러나 이러한 초창기의 시도는 대개 HTML과 이미지를 기반으로 하고 있었기 때문에 실제로 CAD/CAM 소프트웨어를 효과적으로 교육시키는데 많은 문제점을 표출하였다²⁾. 이에 VRML을 이용한 시도나 다른 웹 가시화 기술을 이용한 교육용 모델러나 시뮬레이터의 개발이 활발히 이루어지고 있다¹⁰⁾. 예를 들어, 선반 공정을 VRML로 구현한 사례는 미시간 대학의 구축사례가 있다¹¹⁾. 그러나 이는 선반의 기본 기능인 G-코드에 따른 절삭 공정을 묘사하지 못하고 있고 단지 VRML으로 구성된 피삭재와 공구의 절삭을 광원과 소리 등을 통해서 효과적으로 가시화 하는데 그치고 있다.

일반적으로 VRML을 이용한 시도는 상업적인 목적으로는 3D 가상 주택, 3D 가상 쇼핑몰 생성 등에 많이 사용되며 공학용의 경우에는 공학용 데이터의 가시화 혹은 로봇의 제어 등에 많이 사용된다¹²⁾. 이는 3차원 객체를 회전, 이동 등의 공간상의 움직임을 가시화할 때 VRML이 우수한 성능을 보이기 때문이다. 이에 반해서 선반이나 밀링과 같은 모델의 형상이 계속 변화해야 하는 절삭 공정의 경우에는

*학생회원, 동국대학교 산업공학과
**동국대학교 산업공학과
***중신회원, 동국대학교 산업공학과
- 논문투고일: 2000. 8. 7
- 심사완료일: 2000. 11. 3

활발히 이용되지는 못하고 있는데 이는 VRML이 기본적으로 Java와 같은 개발 언어가 아니라 장면을 묘사하는 모델링 언어로써 갖는 한계 때문이다. 물론 VRML의 경우에도 장면에 동적인 변화를 주기 위한 Java EAI(External Authoring Interface) 라이브러리나 외부 스크립트언어 등을 지원하고 있지만 절삭공정을 묘사하기 위해서 필요한 연산 등을 지원하기는 어려운 실정이다⁷⁾.

본 연구에서는 CAM 분야에서 중요한 공작 기계인 선반 교육을 위한 시스템을 제안하였다. 제안된 시스템을 이용하면 웹 브라우저에서 Java 애플릿으로 구현된 입력 창에 G-코드를 입력하여 이를 Java2D와 Java3D를 이용한 가시화를 통해서 선반 작업에 필요한 전반적인 교육이 가능하게 된다. 선반의 경우는 밀링에 비해서 가공될 제품의 형상이 비교적 단순하고 피삭재와 최종 완성품의 형상이 축 대칭이기 때문에 자바 애플릿을 통해서 가시화를 수행해도 별다른 무리를 주지 않는다는 장점이 있다.

본 연구에서 모델로 삼은 선반은 이스라엘 Eshed사의 EMCO이며 G-코드를 입력하면 이에 따른 2D Profile을 생성하고 이를 3D로 revolve하는 기능을 구현하고 있다. 특히 가시화를 위해서 Java3D 라이브러리를 사용하여 구현되어 있기 때문에 OpenGL을 이용한 것과 유사한 효과를 얻을 수 있었다.

가상 선반에 대한 교육 시스템 개발의 문제점들 중의 하나로는 공구가 지나가면서 생성되는 공구궤적을 실제와 동일하게 묘사하는 과정을 들 수 있다¹¹⁾. 본 연구에서 이러한 화면 갱신영역에 대한 작업은 다면체 곡면으로 가시화를 수행하였다.

2. 온라인 교육을 위한 3차원 그래픽스

2.1 온라인 교육을 위한 3차원 그래픽스의 필요성

온라인으로 교육을 하기 위해서는 먼저 피실험자들이 쉽게 접할 수 있는 인터넷 환경에서 실제로 PC 전용 선반 교육시스템에서 구현되는 것과 유사한 품질의 교육이 가능한 인터페이스를 구축하는 것이 중요하다. 일반적으로 온라인으로 가상현실기술을 이용하여 교육을 하게 되면 소프트웨어 구매비용의 절감과 교육에 소요되는 시간을 단축할 수 있다는 장점이 있다¹²⁾.

선반의 온라인 교육시 가장 중요한 것은 실제의 선반과 동일한 인터페이스를 구축해야 한다는 것이라고 할 수 있다. 이를 위해서는 3차원 피삭재와 공구 형상을 정확히 묘사할수 있는 방법이 요구된다.

기존의 HTML과 이미지 기반의 연구들의 경우에는 절삭 공정을 2D를 기반으로 도면을 가시화하는 수준에 그쳤다는 한계를 가지고 있으며 VMRL을 기반으로 한 연구의 경우에는 VRML 2.0의 등장으로 모델의 정형화된 타입은 가시화가 가능해졌지만 공구 형상과 입력된 G-코드에 따라 피삭재를 가공하는 것을 시뮬레이션하기에는 그 용도가 제한적이었다¹³⁾. 그 외에도 VRML을 이용하여 동적인 인터페이스 구축을 가능하게 하는 Java 라이브러리들이 있는데 가장 대표적인 것이 EAI(External Authoring Interface)이다.

EAI는 자바 애플릿과 VRML 장면그래프간에 직접적인 제어가 가능하게 된다는 장점이 있지만 이 경우에도 미리 정의된 객체들간의 정적인 변화는 가능하나 코드의 생성과 이를 반영한 화면의 갱신은 불가능하였다. 이 점을 개선하기 위한 X3D 컨소시엄이 조직되어 VRML의 단점을 개선하려 하지만 표준화된 규격과 이를 지원하는 개발도구들이 존재하지는 않은 것이 현실이다. 따라서 본 연구에서는 이러한 인터페이스를 최대한 실제 상업용 교육시스템과 동일한 기능을 구현하기 위해서 G-코드를 입력받고 이를 가시화할 수 있는 Java3D 가시화 함수루틴을 구현하였다.

선반의 경우에는 대개의 경우 수작업으로 직접 G-코드를 입력할 수 있어야 하며 전용 CAD 모델러에서 자동으로 생성된 G 코드를 파일형식으로 입력 받을 수도 있어야 한다. 본 가상 선반에서도 직접 G-코드를 입력할 수 있으며 큰 형상의 경우에 CAM 소프트웨어에서 형상을 모델링한 뒤 이를 G-코드로 변환하여 저장한 파일을 가시화 할 수 있다. Fig. 1은 현재 운영중인 EMCO 선반 시스템이고 Fig. 2는 Java를 이용하여 제안된 교육용 시스템의 모습이다.

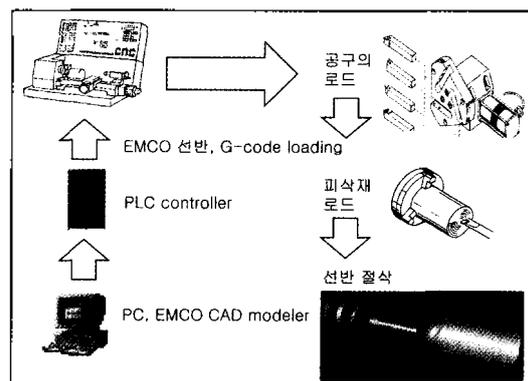


Fig. 1. Operation of EMCO lahte.

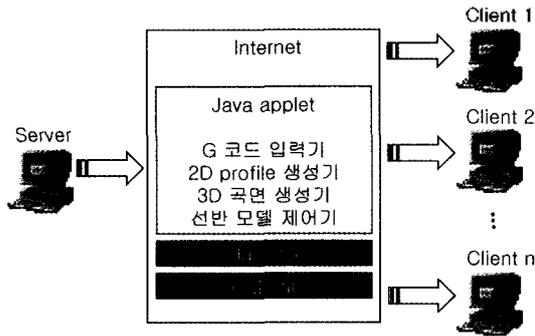


Fig. 2. Education system using java.

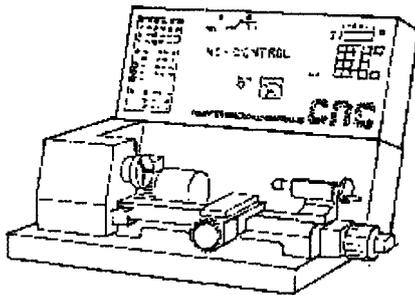


Fig. 3. Eshed EMCO lathe model.

2.2 선반의 모델링과 가시화 구현

본 연구에서 사용한 선반의 모델은 Fig. 3의 Eshed사의 EMCO의 모델을 기반으로 수행하였다.

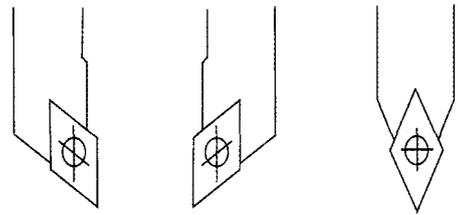
2.3 선반의 공구모델 설정

제안된 선반의 공구 모델은 오른쪽 사이드 톨, 왼쪽 사이드 톨, 가운데 사이드 톨로 구성된다. 각각의 톨은 피삭재와의 절삭이 이루어지는 점을 참조점(reference point)으로 하여 G-코드에 따라 공구를 움직이도록 명령하는 기준이 된다. Fig. 4는 본 연구에서 사용된 Eshed사의 EMCO 모델에서 사용하는 3가지 공구모델의 모습이다^[6].

2.4 G-code 해석기의 제작

본 연구에서 제작된 G-코드 해석기는 기본적으로 Eshed EMCO사에서 제공하는 G-코드에 맞추어서 제작되었다. 특히 표준 G-코드가 100개 이상의 종류로 구성되어 있지만 실제로 교육에 필요한 개수는 대략 10개 미만이라는 점에 착안하여 모든 선반 응용 프로그램을 위해서 핵심적으로 구현되어야 하는 G-코드를 추출하여 해석기를 제작하였다.

각각의 G-코드에 대응하는 부분은 자바를 통해서



Right hand side tool Left hand side tool Neutral side tool

Fig. 4. Cutting tools for virtual lathe.

Table 1. EMCO's G-codes and Embodied G-codes

코드	EMCO 선반의 가공	가상선 추후확장 반의 구현 가능성
G00	Positioning with rapid traverse	
G01	Linear Interpolation	
G02	Circular interpolation (CW)	
G03	Circular interpolation (CCW)	
G20	Hold	- ○
G21	Empty line	- ○
G22	Program end	- ×
G33	Threading	- ×
G78	Threading cycle	- ○
G84	Long turning cycle	- ○
M00	Program stop	- ○
M01	Optional stop	- ○
M02	End of program	- ○
M03	Spindle clockwise	- ○
M04	Spindle counterclockwise	- ○
M05	Spindle off	- ○
M06	Tool change	- △

클래스화했기 때문에 나머지 G-코드에 대해서도 손쉽게 확장할 수 있다는 장점을 가지고 있다. 또한 나머지 G-코드들은 모두 3가지 형상 코드의 조합으로 표현이 가능한 것도 중요한 이유이다. Tabel 1은 EMCO 선반의 G코드와 구현된 G코드이다.

3. 구축사례

3.1 시스템 구축

본 시스템은 동국대학교 CAD/CAM실험실에 장착된 가상실험실 서버를 이용하여 구축되었다. Fig. 5는 전체 시스템의 프레임이고 Fig. 6은 초기 홈페이지 화면이다. 초기화면에서 가상 선반을 선택하면 선반의 애플릿 화면이 로드된다. 이중에서 G-코드 입력 창을 통해서 G-코드를 입력하게 되면 각각의 G-코드에 해당되는 그래픽 객체들이 가시화된다. 입력

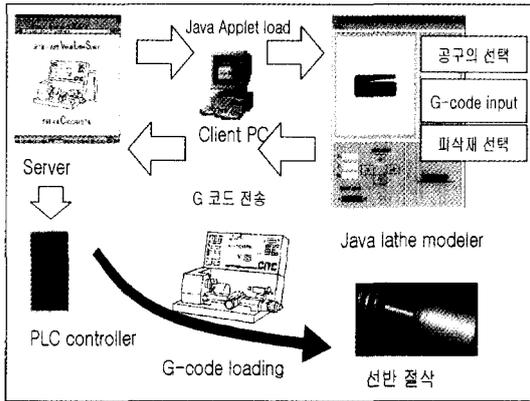


Fig. 5. Overall structure of virtual lathe.

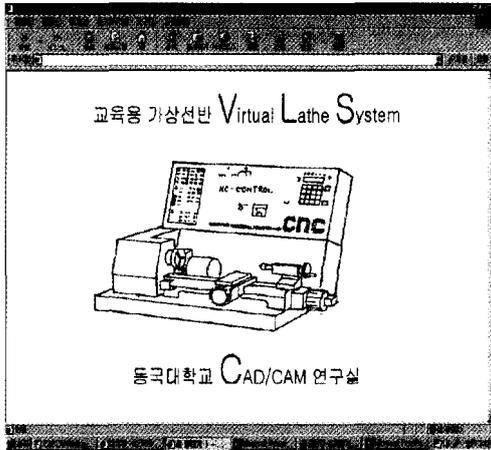


Fig. 6. Home page of virtual lab.

된 G-코드를 실행시키면 우선 2D profile이 가시화 되고 이를 3차원 sweeping시킨 곡면형태로 최종 피삭재의 형상을 가시화한다.

전체 시스템 골격은 다음과 같이 디자인되었다. 먼저 사용자가 G-코드 설명 창을 통해서 G 코드를 입력한다. 이때는 수작업으로 입력하는 방법과 Eshed사에서 제공하는 전용 선반용 CAD모델링 소프트웨어를 이용해서 만들어진 G-코드를 copy and paste해서 코드 입력창에 불러들이는 두 가지의 방법을 사용하였다. 입력된 G-코드는 피교육자가 작성한 형태 그대로 들어오기 때문에 이를 2D profile로 만들고 3D로 가시화하기 위해서는 추가적인 필터링 작업이 필요하다.

Fig. 7은 가상선반 애플릿에서 사용자의 작업공정도를 보여주고 있다. 로드된 자바 애플릿에 HTML교재를 통해서 G코드를 입력한다. 이때 입력된 G 코

본 교육용 시스템의 사용절차
• HTML을 통한 G 코드의 용례교육
• 가상 선반 코드 창, G 코드의 입력
• 경고 창에서 에러 시 경고
• 에러를 제거한 뒤 2D Profile 생성
• 3D 곡면 생성
• 공구 모델 설정
• 공구 애니메이션 실행

Fig. 7. User guidance for virtual lathe.

드는 2D profile을 생성하게 된다. 모든 G코드의 입력이 끝나게 되면 피삭재가 G코드에 의해서 절삭되고 남은 영역이 2D로 가시화된다. 이때 Run 버튼을 클릭하게 되면 삼각다면체 곡면이 생성된다.

Fig. 8은 가상선반의 초기 피삭재의 모습을 보여주고 있다. G코드가 Fig. 8의 우측 입력 창을 통해서 입력되면 우측 아래에 있는 2D 가시화 창에서 가시화되고 이 가시화된 형상을 바탕으로 3D를 생성하면 좌측 상단에 있는 java 3D scene을 통해서 가시화된다.

본 시스템의 사용자는 가상실험실의 홈페이지에 접속한 뒤 HTML을 통한 G 코드와 시스템의 사용설명을 배운 뒤 HTML과 같이 로드된 Fig. 8의 가

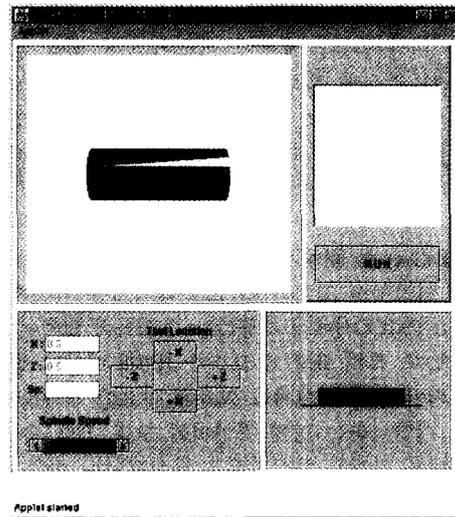


Fig. 8. Initial viewing of virtual lathe applet.

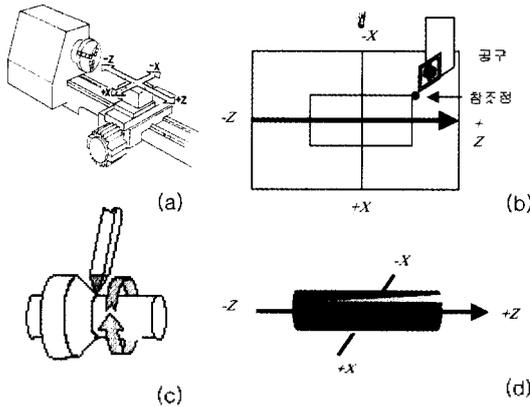


Fig. 9. Coordinates of virtual lathe (a) Coordinate of lathe (b) 2D coordinate of VR lathe (c) Tool and workpiece (d) workpiece in coordinate.

상 선반 애플릿의 G코드 입력 창에 G코드를 입력한다. 그리고 RUN 버튼을 누르면 먼저 2D profile이 생성되고 3D 곡면이 생성된다. 전체 절차는 Fig. 7 과 같다.

3.2 공구 좌표계의 설정

본 시스템은 실제 EMCO 선반과 동일한 인터페이스를 구축하기 위해서 공구 좌표계를 다음 Fig. 9와 같이 설정하였다. Fig. 9는 실제 선반과 동일하게 설정된 좌표계를 보여주고 있다. 본 연구에서는 공구가 피삭재의 우측 위 끝 점에서 진입하게끔 설정되어 있다. Fig. 9(b)에서 공구의 끝점이 바로 G 코드가 수행되는 기준이 되는 참조점으로 설정된다. 모든 G코드는 이 참조점을 기준으로 상대적인 좌표값으로 공구 진입, 이동, 직선 보간, 원호 보간 등의 가공을 실시하게 된다.

3.3 G-코드를 이용한 3D 가시화

G-코드 입력 창을 통해서 들어온 G 코드들은 공구의 이동이나 절삭과 관련된 코드와 그 외에 기타 처리를 위한 코드로 구분된다. 이 중에서 공구가 여러 번 지나가는 궤적들 중에서 피삭재의 중심에서 가장 안쪽을 지나게 되는 경로들을 저장하여 2D profile을 생성하게 된다. Fig. 10은 피삭재의 절삭 상태와 최종 profile을 얻기 까지 데이터를 가공할 수 있도록 해주는 기준이 되는 피삭재 모델이다. 이 모델은 G 코드에 의해서 수행되는 절삭작업 뒤에 남은 정확한 외곽선을 구하기 위해서 필요하다.

이 모델은 Fig. 10에서 보는 바와 같이 피삭재를

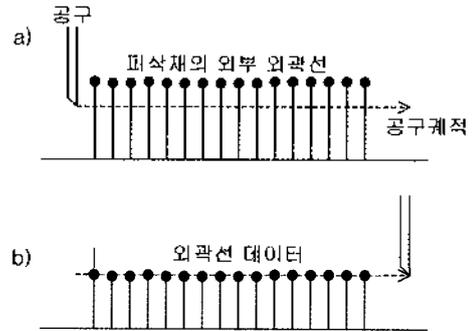


Fig. 10. Sampling of boundary data for workpiece.

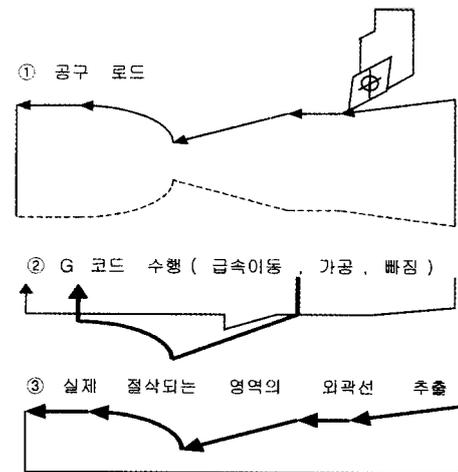


Fig. 11. Tool path generation from G-code.

중심축으로부터의 거리값을 갖는 격자화된 점들의 배열로 설정된다. 임의의 G 코드가 수행되면 G 코드에 의해서 수행된 값이 공구 시작 점에서부터 상대 좌표계로 계산되기 시작하면 이 공구가 이동시 만나게 되는 격자 점들을 차례로 저장하게 된다.

이때 선반 작업용 G 코드의 경우 한번의 공구 경로로 모든 형상을 가공할 수 있는 것이 아니라 여러 번의 공구경로를 지정하여 최종적으로 남게 되는 형상을 만들어 가기 때문에 이를 구현하기 위해서는 최종적으로 남게되는 형상을 구하는 작업이 필요하다.

Fig. 11과 같이 여러 경로들을 모두 수행한 뒤 최종적으로 피삭재의 중심축으로부터 가장 가까운 곳을 지나는 경로들만으로 외곽선을 생성시켜주어야 한다. 이를 위해서 Fig. 12와 같이 필러링 작업을 거치게 되는데 실제로 지나간 공구의 움직임은 항상 피삭재로부터의 거리가 가장 적은 격자 점들을 연결한 점 열이 되며 이를 가공후 데이터 군으로 정의한다. 이때 이 점열에서 모든 격자 점들을 사용하지는

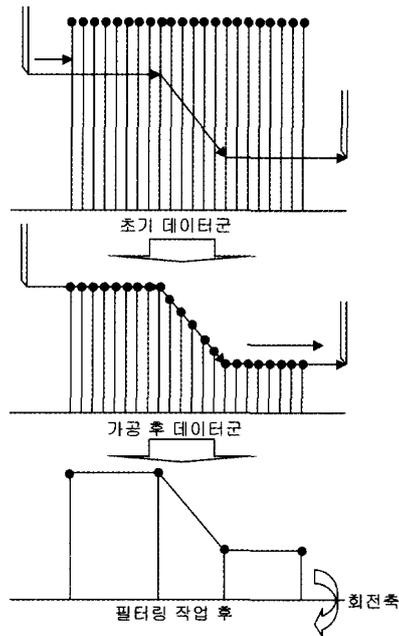


Fig. 12. Sampling of boundary data for actual cutting.

않는데 그것은 2D로 가시화된 뒤 3D로 가시화를 수행하기 위해서는 2D Profile이 최소의 점으로 표현되어야 삼각 다면체를 생성시 데이터의 양이 적고 자바에플릿에 부하가 적게 걸리기 때문이다.

Fig. 12는 초기 데이터 군을 등간격으로 설정하고 가공 후 데이터 군을 추출한 뒤 이를 필터링하여 2D profile을 생성한 모습이다. 이때 등간격이기 때문에 간격과 간격사이에 G코드로 수행된 결과가 남을수 있는데 본 연구에서는 이 경우에 가까운 격자점을 저장하는 방식을 취하였다.

Fig. 13은 필터링이 끝난 2D profile을 회전시켜

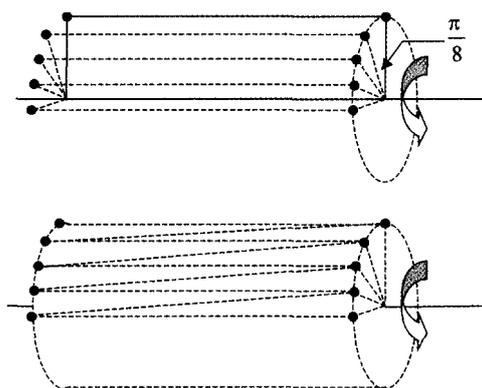


Fig. 13. Generation of 3D surfaces.

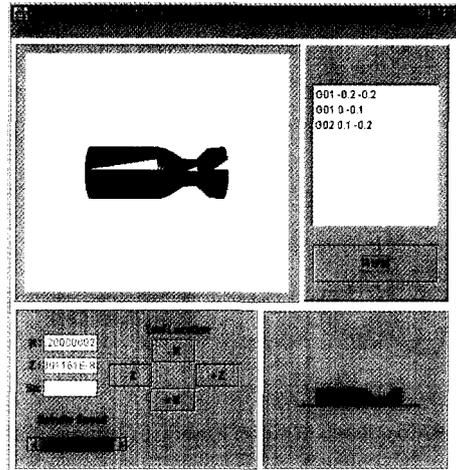


Fig. 14. 2D and 3D windows after G-code input.

3D 곡면을 생성하는 것이다. 이때 입력된 2D profile 들을 피삭재의 중심축을 기준으로 일정한 각도 만큼 회전 시켜가면서 삼각 다면체 곡면으로 연결하는 작업이 필요하다. Java3D는 입력된 임의의 곡면 데이터를 삼각다면체 형식으로 가시화 하는 정형화된 루틴을 가지고 있다. 본 가시화에는 이중에서 triangle_strip array를 통해서 가시화루틴을 작업했으며 Fig. 14는 이를 사용한 곡면의 가시화된 모습이다.

3.4 응용 사례들

본 시스템의 사용 후 일반적인 선반작업에 사용되

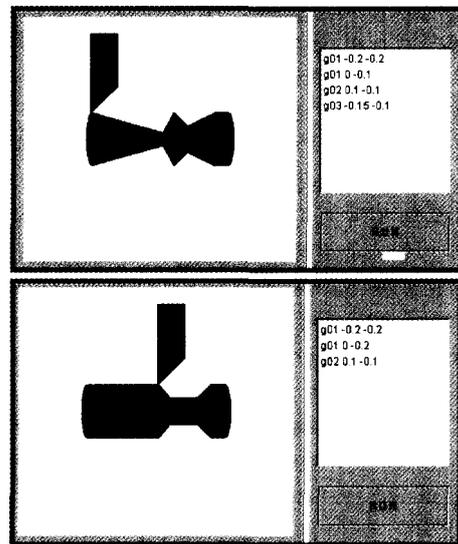


Fig. 15. 2D and 3D windows after G-code input.

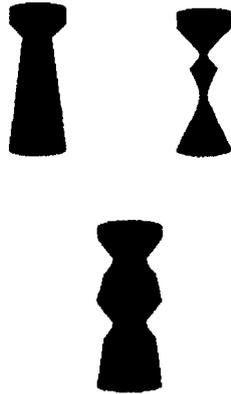


Fig. 16. Samples

는 형상을 본 시스템으로 구현해 보았다. Fig. 14는 G코드 수행 결과로 생성된 최종 형상의 모습이다.

Fig. 15는 공구를 로드한 뒤 공구 애니메이션을 수행한 결과이다. Fig. 16은 몇가지 사례들이다. 이때 작성된 G 코드들은 ASCII 형식이며 ASP를 통해서 서버로 전송이 가능하다. 서버로 전송된 파일들은 서버 관리자가 EMCO FMS 소프트웨어에서 파일을 읽어들이고 PLC를 통해서 전송하게 되면 EMCO 선반에 로드되어 실제 선반 작업이 수행된다. 이때 선반에 공구 로드와 피삭재의 로드는 수작업으로 셋업해 주게 된다.

사용자들은 웹 브라우저를 통해서 자바애플릿을 로드한 뒤 HTML상에서 G코드의 용례를 배운 뒤 실제로 G코드를 입력하여 형상을 얻게 된다. 실제 선반과 동일한 인터페이스로 구현되어 있기 때문에 피실험자들이 빨리 적응했으며 G코드에 대한 이해도가 일반 실습 강의보다 높아졌다.

4. 결 론

본 연구에서는 온라인 CAD/CAM 교육 시스템을 구축하기 위한 연구의 일환으로써 선반의 G-코드 기반 온라인 시뮬레이터를 개발하였다. 가상선반을 이용한 교육에서 G-코드를 사용자가 입력하거나 파일을 입력받은 뒤 해당 G코드를 단계 별로 읽어들이 이를 Java2D와 Java3D를 이용하여 화면에 가시화함으로써 기존의 2D 이미지를 통한 정적인 교육이나 VRML로는 묘사하기가 어려운 모델에 기하학적인 변형이 계속 수행되어야 하는 절삭공정의 가시화의

문제점을 해결하였다. 향후에는 G32나 G34에서 지원하는 나선 가공과 같은 특수한 형태의 선반 작업들을 지원하기 위한 모듈의 추가와 Java2D를 이용하여 사용자가 직접 마우스를 통해서 형상을 설계하고 이를 본 시스템과 연결하면 더욱 좋은 솔루션이 될 것으로 생각된다. 또한 현재 구현은 선반 가공에 한정되어 있으나 향후에는 밀링 공정도 추가 연구되어야 할 것이다.

감사의 글

본 연구는 학술진흥재단의 1999년 대학부설 연구소 과제(과제번호: 1999-1080-0)의 지원을 받아 수행되었으며, 이에 감사드립니다.

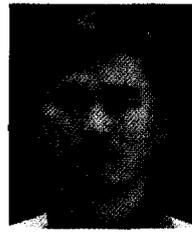
참고문헌

1. Bricken, M., "Virtual Reality Learning Environments: Potential and Challenges", *Computer Graphics*, pp 178, Vol 25, 1991.
2. Dede, C. J., "The Future of multimedia: Bringing to Virtual Worlds", *Educational Technology*, pp 54-60, Vol 32, 1992.
3. Deering, M., "High Resolution Virtual Reality", *Computer Graphics*, Vol 26, pp 195- 202, 1992.
4. "Virtual Lathe", http://www-vrl.umich.edu/sel_prj/lathe/index.html.
5. Gracanin, D., Matijasevic, M. and Nikos, C., "Virtual Reality Testbed for Mobile Robots", *Proceedings of International Symposiums on Industrial Electronics*, Vol 1, pp 293-297, 1999.
6. Nadeau, D., "What 3D API for Java should I use and why?", *SIGGRAPH 97 Conference Proceedings*, Addison Wiley, pp 452-453, 1997.
7. Nadeau, D., "Building Virtual Worlds with VR-ML", *IEEE Computer Graphics and Applications*, Vol 19, pp 18-29, 1999.
8. *Teacher's Handbook EMCO Compact 5 CNC*, pp 81-89, 1982.
9. V-CNC 매뉴얼, "CNC 가상 시뮬레이터 프로그램 Ver 3.0 따라하기 예제집", 큐빅테크 출판부, 1999.
10. 김형도, "3차원 대화형 시뮬레이션 모델 기술 언어로서의 VRML", 한국시뮬레이션학회 논문지 제6권, 제1호, 1997.
11. 박광렬, 이철수, "기준평면과 경제상자를 이용한 NC 절삭과정의 그래픽 시뮬레이션", 한국 CAD/CAM학회 논문집, Vol 2, No 3, pp 161-174, 1997.



이재호

1997년 한성대학교 산업공학과 학사
1999년 동국대학교 산업공학과 석사
1999년~현재 동국대학교 산업공학과 박사과정
관심분야: Geometric modeling, rapid prototyping, virtual reality



이종태

1981년 서울대학교 산업공학과 학사
1983년 한국과학기술원 산업공학과 석사
1990년 U. C. Berkely 산업공학과 박사
1992년~현재 동국대학교 산업공학과 부교수
관심분야: 신경망, 정보 지능 시스템



김경철

1999년 동국대학교 산업공학과 학사
1999년~현재 동국대학교 산업공학과 석사과정
관심분야: Geometric Modeling, Virtual Reality, Computer Graphics



박준영

1982년 한양대학교 기계공학과 학사
1985년 University of Minnesota 산업공학과 석사
1991년 University of Michigan 산업공학과 박사
1992년~1995년 한성대학교 산업공학과 조교수
1995년~현재 동국대학교 산업공학과 부교수
관심분야: Geometric Modeling, Computer Graphics, Mesh Generation, STEP



이준

2000년 동국대학교 산업공학과 학사
2000년~현재 동국대학교 산업공학과 석사과정
관심분야: 인공지능, 웹 기반의 전문가 시스템 설계