大韓機械學會論文集 第 14 卷 第 2 號, pp. 367~375, 1990.

〈論 文〉

개인용 컴퓨터를 이용한 전개도 작성에 관한 연구

채 희 창* · 정 인 성**

(1989년 9월 22일 접수)

Constructing the Development of Solids by Personal Computer

Hee-Chang Chae and In-Sung Chung

Key Words: Development(전개), Homogeneous Transformation(변환), Personal Computer (개인용 컴퓨터), Dual-Screen(2개의 화면), Plate Work(판금작업)

Abstract

A data structure representing 3-D objects was designed for the personal computer. It is very simple to be used in the personal computers which have small memory and low speed. A homogeneous transformation for developing 3-D objects was derived. Each developing procedure consists of five transformations : one translational-three rotational-one translational. Developing a solid is a creative work. So the results of developing vary with the order of surfaces to be developed. One method to reduce the length of seam was considered. The programs used in this study were written in Pascal and Assembly and a modeller that generates 3-D primitives was included. This program is an interactive dual-screen system. While all the menus in Korean are displayed at the monochrome monitor, the development figures with projective views are drawn at the color monitor. The program has wide applications for plate works.

1. 서 론

물체의 전개도를 작성하는 일은 제관이나 판금작업에 있어서 가장 기본이 되는 작업으로서 나날이 그 중요성이 높아가고 있다. 물체의 전개도는 대개 도학적인 방법[1]을 이용하여 작성하여 왔으나, 도형이 복잡해지거나 상관선이 존재할 경우에는 상당한 어려움이 뒤따른다. 도학적인 방법을 컴퓨터를 이용하여 처리할 수도 있으나, 여러가지 형태에 대하여 각각 프로그램을 작성해야 하는 단점이 있다. 따라서 본 연구에서는 널리 보급되어 있고 가격도 비교적 저렴한 개인용 컴퓨터를 이용하여 입체를 다면체로 모델링한 후 이를 전개하는 시스템을 개발하고저 한다.

*정회원, 전북대학교 대학원
**정회원, 전북대학교 공과대학 기계공학과

2. 사용언어와 컴퓨터

레코드(record), 포인터(pointer), 동적변수(dynamic variable)의 관리기능을 갖고 있는 언어로서 Pascal, C 등이 있으나, 본 연구에서는 Pascal을 사용하였으며 경우에 따라서는 어셈블리(assembly)를 사용하여 작성한 루우틴(routine)을 Pascal과 결합시켜서 사용하였다.

사용 컴퓨터는 IBM PC-AT 호환기종이며 주변기기는 640×480 해상도를 갖는 칼러모니터(VGA)와 단색모니터(Hercules), 8색 플로터(HP 9872C)를 사용하였다.

IBM PC에서는 그래픽(graphic) 화면과 텍스트(text) 화면을 동시에 사용하기 어렵다. 영문텍스트의 경우 그래픽 모우드에서 화면에 출력하는 것이 가능하기 때문에, 대부분의 경우 메뉴를 영문으로 나타낸다. 한글을 그래픽모우드에서 출력하기 위해서는 별도의 한글자형이 필요하며, 이때 한글의 글자수가 너무 많은 것이 문제이다. 그러나 칼러모니터와 비교하여 훨씬 저가인 단색모니터와 그에 해당하는 한글 보드에는 이미 한글자형을 내장하고 있기 때문에 텍스트모우드 상태에서는 이를 쉽게 사용할 수 있다. 2개의 모니터(dual screen)를 사용하고 텍스트와 그래픽을 서로 다른 모니터에 동시에 나타냄으로써, 그래픽 화면의 일부를 할애하여 메뉴를 나타내거나 화면을 교대로 나타낼(swap) 필요가 없다. 한글출력이 가능한 단색모니터(monochrome monitor)는 한글로 작성된 메뉴를 나타내고, 칼러모니터는 전개도 및 투상도를 나타내었다.

Fig. 1은 IBM PC에서 VGA(Video Graphic Array)와 Hercules 카드(Monochrome Graphic Adapter)가 장착되었을 경우의 비데오 메모리 구조를 나타낸 것이다. 일단 그래픽 모우드(VGA)로

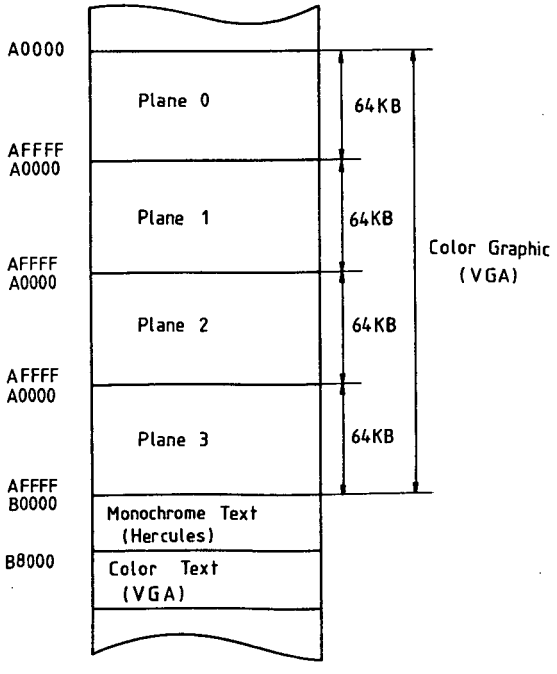

Fig. 1 Video memory map of graphic adapters

들어가면, BIOS(Basic Input Output System)[2] 또는 언어 자체가 제공하는 문자출력 프로그램에 의하여 단색모니터에 문자를 출력하는 일이 불가능해지기 때문에, B 0000(16진법)번지에서 시작하는 단색모니터용 텍스트 비데오 메모리를 직접 엑세스(access)하는 문자출력 프로그램을 어셈블리로 작성하였다. BIOS를 통하지 않고, 직접 비데오 메모리에 문자와 그 속성(attribute)을 출력(move)함으로써 문자를 화면에 나타내었기 때문에 하드웨어(hardware)적으로 한글을 처리하는 기종에서만 한글출력을 할 수 있다. 많은 한글카드가 이 방식을 택하고 있으며, 한글자형(font)이 이미 ROM에 내장되어 있기 때문에 별도의 한글자형이 불필요 하다. 또한 한글의 출력은 흑백모니터화면에 국한하며 플로터에 출력할 수 있는 것은 아니다.

3. 데이터 구조(Data Structure)

데이터 구조에는 배열(array) 형태와 링크리스트(linked list) 방식이 있는데[3] 이들사이에는 속도와 메모리 관리면에 있어서 서로 대치관계가 있다. 그러므로 이 양자를 조화시켜 개인용 컴퓨터에서의 메모리, 속도의 한계등을 타결시킬 수 있는 데이터 구조를 설계하였으며 그의 대략도를 Fig. 2에 도시하였다. 데이터베이스[4]를 사용하면 프로그램은 간단하지만 키(index key)를 관리해야 하므로 이에 대한 부수적인 메모리가 소요되고, 검색시간이 너무 커지는 것이 단점이다. 따라서, 본 연구에서는 꼭지점과 다각형에 대한 자료는 일정크기의 배열을 한 페이지(page)로 하여 필요시 페이지 단위로 동적영역(heap)에 확보하여 사용하고, 정적메모리에는 그 주소만 기록하였다. 크기의 변화폭이 큰 다각형 내부의 정보 즉, 삼각형에 대해서는 링크리스트 방식을 사용하였다. Fig. 2는 입체 1개에 대한 데이터 구조를 나타낸 것이고, Fig. 3은 정육면체에 대하여 페이지크기를 6으로 하였을 때의 예이다.

(1) 꼭지점의 표시

꼭지점은 꼭지점의 좌표를 꼭지점 페이지에 있는 배열에 저장한다. 아울러 검색, 추가, 삭제의 편의를 위하여 페이지내의 꼭지점수(Nv), empty stack, flag fv를 둔다.

(2) 다각형의 표시

모든 다각형은 삼각형으로 분할한후 평면의 방정식($Nx \cdot X + Ny \cdot Y + Nz \cdot Z + D = 0$)과 더불어 삼

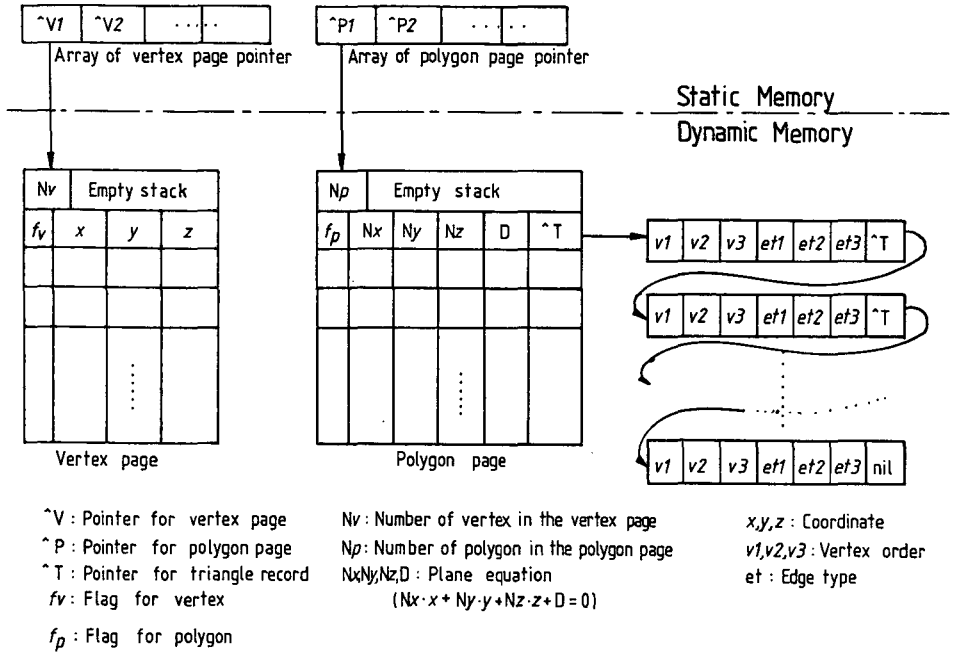

Fig. 2 Data structure for an object

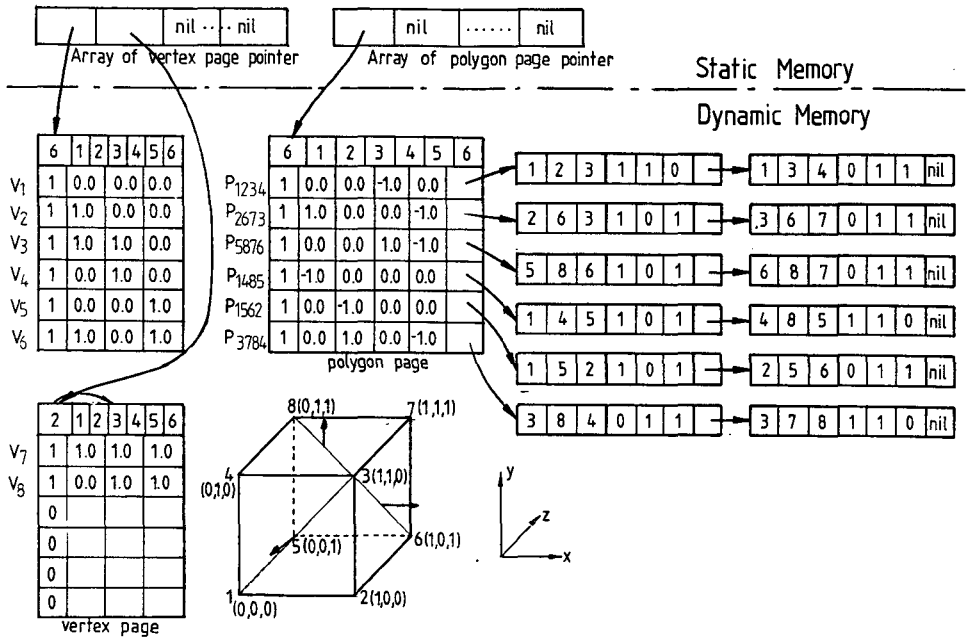

Fig. 3 Example data structure of cube

각형의 링크리스트로 표시하였다. 추후 모든 조작은 삼각형을 기준으로 하고 삼각형의 링크리스트를 변화시킨다. *Np*, empty stack, flag *fp*등은 꼭지점의 경우와 같지만, *fp*의 경우 1 byte를 bit 단위로 구분하여 다양한 목적으로 사용될 수 있다(Fig 4).

(3) 삼각형의 표시

다각형을 모서리로서 표시하지 않고, 삼각형의 링크리스트로 표시하였다. *V*1, *V*2, *V*3는 꼭지

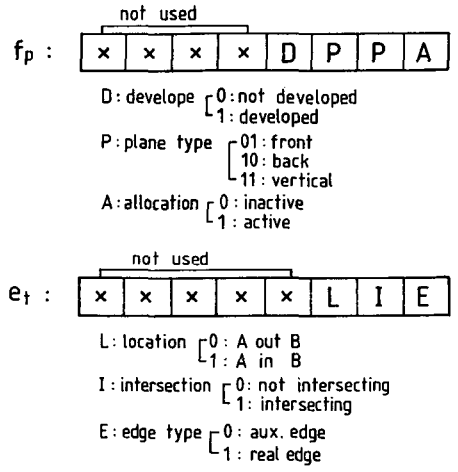

Fig. 4 Allocation flag f_p and edge type e_t

점번호이며 이 순서가 면의 방향을 나타낸다[4]. *et*1, *et*2, *et*3는 각각 $V1$-$V2$, $V2$-$V3$, $V3$-$V1$ 모서리 형태를 표시하였다. 결국 반모서리(half edge)[5]에 대한 자료가 삼각형에 포함된 형태이다. *et*1, *et*2, *et*3도 *Np*와 마찬 가지로 bit 단위로 구분하여 상관선, 보조모서리, 실모서리, A in B, A out B[6]등을 나타내는데 사용한다(Fig. 4).

(4) Empty stack

데이터의 추가삭제를 돕기 위하여 해당 페이지가 처음 만들어 질때 1부터 시작하여 1 씩 증가시켜 넣어 놓는다. 컴퓨터의 stack 구조와 같이, 배열중 특정 항이 삭제되면 *Nv*, *Np*를 감소시킨후 이들이 가리키는 곳에 삭제된 항목번호를 넣는다. 새로운 항을 추가할때는 *Nv*, *Np*가 가르키는 곳에 나타난 배열번호를 찾아 그에 해당하는 위치에, 추가되는 값을 넣고 *Nv*, *Np* 값을 증가시킨다.

(5) 복수입체의 표현

복수의 입체는 Fig. 2의 데이터가 여러개 존재한다. 고정메모리상에 포인터배열이 다시 배열을 이루고 있다.

4. 입체의 생성

입체를 나타내는 가장 기초적 모델은 다면체로서 이는 면수를 증가시키면 실입체에 근접하는 특징을 가지고 있어[7] 입체의 모델에 많이 쓰이고 있다. 또한 다면체는 다각형으로 이루어지며 다각형을 표시하는 형태에는 여러가지가 있으나, 본 연구에서는 기하학적으로 가장 간단한 형태인 삼각형의 리스트로 표시하였으며, 삼각형 꼭지점의 순서가 다각형의 방향(outward normal)이 되도록 표시[4]함으로써 위상적 일관성(topological consistency)을 만족하도록 하였다.

4.1 기본 입체의 생성

기본 입체의 생성은 주 메뉴(main menu)에서 대화식으로 입체를 생성(generation)시키거나 외부파일로 부터 읽어들일 수 있도록 하였으며, 다음과 같은 기본입체 발생기를 가지고 있다.

(1) 직사각형
(2) 원 및 타원
(3) 직육면체
(4) 직원주 및 사원주(타원주 포함)
(5) 중공원주 및 중공사원주
(6) 원추 및 사원추(타원추 포함)
(7) 구(달걀형 포함)
(8) 반구

4.2 복잡한 입체의 생성

복잡한 입체는 4.1에서 생성시킨 위상적 일관성을 만족하는 기본입체를 결합(joining), 절단(splitting), boolean operation등의 위상적 일관성을 유지시키는 조작을 행함으로서 구성할 수 있다. 이때 다각형을 직접 다루기 어려우므로 기하학적으로 가장 간단한 형태인 삼각형의 합으로 표시하여 삼각형의 변화를 다루면 된다.

5. 입체의 전개

5.1 변환(Homogeneous Transformation)

입체를 다면체의 합으로 표시하여 구성 하였으므로 전개를 하기 위해서는 이들 각각의 면들을 평행이동 또는 회전 시켜서 전개를 한다. 면의 법선벡터가 (0,0,1)이 되고 접합모서리의 방향벡터가 서로 반대가 되어야 한다. Fig. 5는 상관선에 의하여 모서리가 절단되는 경우까지 포함한 가장 일반적인 경우에 대한 전개과정을 나타낸 것이다. 다각형 P_{4589}과 P_{5678}이 이미 전개되어 P'_{4589}, P'_{5678}가 되었을때 P_{123}을 전개할 경우, 이미 전개한 다각형들의 모든 반모서리를 조사하여 결합할 수 있는 공통모서리를 구한다. 즉, 다각형 P_{4589}의 반모서리 9-4의 방향벡터 e_{94}와 다각형 P_{123}의 반모서리 1-2

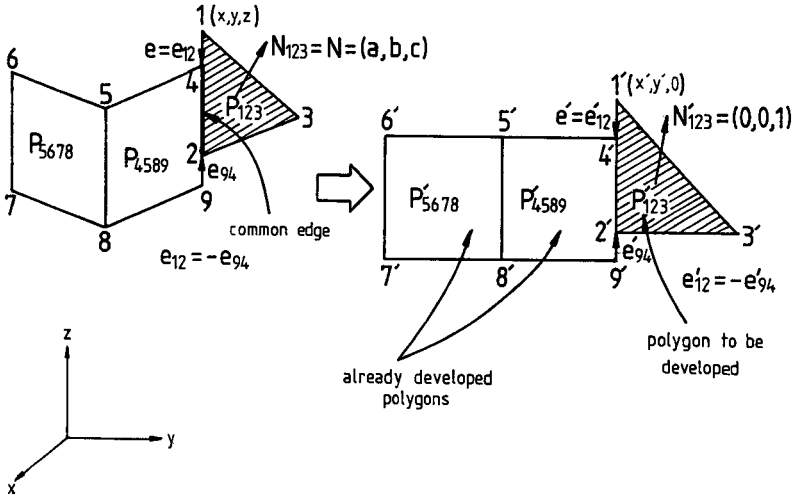

Fig. 5 Developing procedure

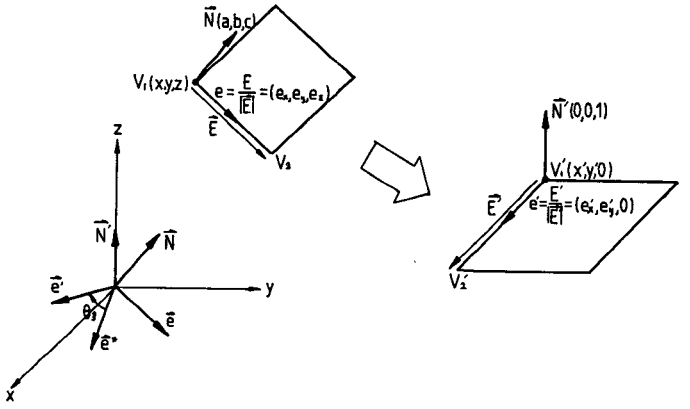

Fig. 6 Rotational transformation

의 방향벡터 e_{12}가 서로 반대방향이고 공통모서리가 존재하므로 전개가 가능하다. 일직선상에 있는 꼭지점 1, 2, 4, 9의 상대적인 거리를 이용하여 꼭지점 1이 변환될 1'의 좌표$(x', y', 0)$를 4', 9'의 좌표로 부터 구할수 있으며, 변환될 모서리의 방향벡터 e'_{12}는 e'_{94}와 반대방향이다.

꼭지점 1의 좌표(x, y, z)가 $(x', y', 0)$, 모서리 방향벡터 $e_{12}(e)$가 $e'_{12}(e')$, 전개할 면의 법선벡터 $N_{123}(N)$이 $(0,0,1)$이 되도록 변환시킨다(Fig. 6). 이 과정은 총 5개의 연속된 변환과정이라 볼 수 있다. 다음은 그 과정을 나타낸 것이다[8,9].

(1) 좌표(z, y, z)를 $(0,0,0)$로 평행이동

평행이동시의 변환행렬을 $T_{-x,-y,-z}$이라 하면

$$T_{-x,-y-z}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x & -y & -z & 1 \end{bmatrix}$$ 이 된다.

(2) 법선벡터 $N(a, b, c)$을 $(0, 0, 1)$로 회전이동

Fig. 7에서 보듯이 x축을 중심으로 θ_1회전 $(R_{\theta_1 i})$, y축을 중심으로 $-\theta_2$회전 $(R_{-\theta_2 j})$으로 이루어 진다.

$\lambda=\sqrt{b^2+c^2}$이라 할때

$\lambda \neq 0$일 경우

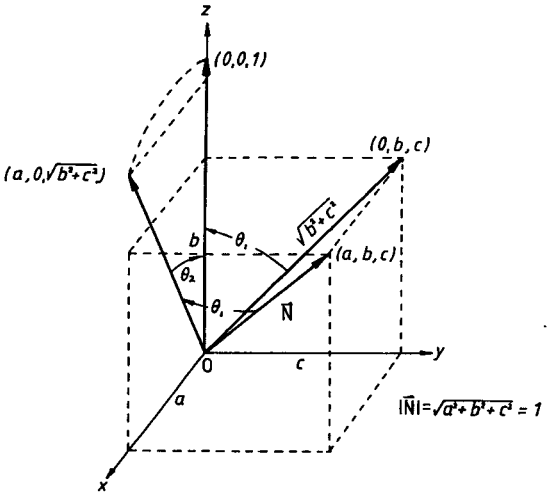

Fig. 7 Alignment of normat vector with k(0,0,1)

$$R_1=\begin{bmatrix} \lambda & 0 & a & 0 \\ \frac{-ac}{\lambda} & \frac{c}{\lambda} & b & 0 \\ \frac{-ac}{\lambda} & \frac{-b}{\lambda} & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\lambda=0$일 경우

$$R_1=\begin{bmatrix} 0 & 0 & \frac{a}{|a|} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{-a}{|a|} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

R_1에 의하여 모서리의 방향벡터 e는 회전을 하여 e^*가 된다(Fig. 6). 이때 e^*의 z좌표는 0이 된다.

$$e^*=[e_x^*,\ e_y^*,\ 0, 0]$$
$$=[e_x, e_y, e_z, 0]\,[R_1]$$

(3) 변환된 방향벡터 e^*를 e'와 일치시키기 위한 휜전이동

e^*와 e'이 이루는 각도를 θ_3라 하면

$\cos\theta_3=e^*\cdot e'=e_x^*e_x'+e_y^*e_y'$

$\sin\theta_3=(e^*\ x\ e')\cdot k=e_x^*e_y'-e_y^*e_x'$

z축을 기준으로 θ_3 만큼 회전을 하면($R_{\theta_3}k$), e^*와 e'이 일치된다.

$$R_{\theta_3}k=\begin{bmatrix} e_x^*e_x'+e_y^*e_y' \\ -(e_x^*e_y'-e_y^*e_x') \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} e_x^*e_y'-e_y^*e_x' & 0 & 0 \\ e_x^*e_x'+e_y^*e_y' & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

그러므로 전체적인 회전이동의 변환 행렬은 다음과 같다.

$$R=R_{\theta_1}i\cdot R_{-\theta_2}j\cdot R\gamma_3 k$$

(4) (0, 0, 0)에서 $(x', y', 0)$로 평행이동

$$T_{x',y',0}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x' & y' & 0 & 1 \end{bmatrix}$$

전체적인 변환행렬은 다음으로 표시된다.

$$T=T_{-x,-y,-z}\cdot R\cdot T_{x',y',0}$$

5.2 전개 알고리즘

입체를 전개하는 방법은 동일입체라 하더라도 전개하는 면의 순서및 위치등이 다를 수 있다. 실용적으로는

(1) 재료의 손실이 적어야 하고

(2) 접합방법상으로 부터 야기되는 강성 또는 강도적인 측면과 접합길이 등을 포함한 경제성을 고려 해야한다.

또한 개개의 면의 접합순서와 접합위치등을 지정하는 수동적인 방법과 자동적인 방법으로 구별된다. 자동적인 방법의 한 예로서 본 논문에서는 접합길이(seam length)를 최소화 하는 방법을 연구하였다. 접합길이를 최소화하는 방법은 다음과 같다.

1-1 단계 : 입체의 모든 반모서리(half edge)중 실모서리만 Fig. 8과 같은 모서리 버퍼에 저장한다. 이때 $Vd1$은 0으로 설정한다.

1-2 단계 : 길이가 큰 순서로 quick sorting 한다[10]. (동일모서리에 대하여 2개의 모서리가 나타 날수도 있다.)

2-1 단계 : 가장 큰 길이의 모서리에 대하여, 시작점(Vst), 끝점($Vend$)을 구하고 시작점의 좌표 $P(x,y,z)$, 모서리 방향벡터 e, 모서리가 속한 다각형의 법석벡터 N을 구한다.

2-2 단계 : 5.1에서와 같이 e는 (0, 1, 0), N는 (0, 0, 1), $P(x,y,z)$는 $P(0,0,0)$이 되도록 하는 변환행열을 구한다.

2-3 단계 : 이 모서리가 속한 다각형을 찾아서 다각형에 있는 모든 모서리를 2-2에서 구한

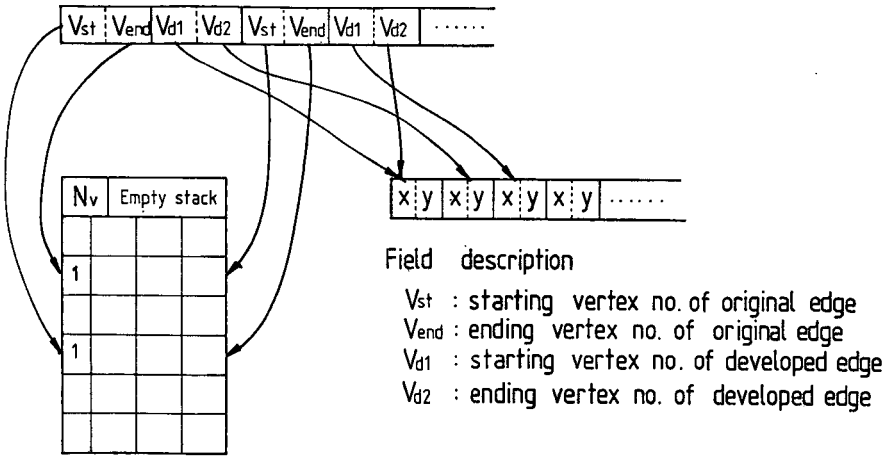

Fig. 8 Half edge buffer for developing

변환행열을 이용하여 전개시킨다(seeding). 이때 전개된 모서리의 꼭지점번호 $Vd1$, $Vd2$는 0이 아닌 값을 갖는다.

3-1 단계 : 모서리 버퍼를 앞에서 부터 검색하여(모서리 길이순으로), 전개되지 않은 모서리($Vd1=0$)를 선택하여, 시작점(Vst), 끝점($Vend$)을 구하고 시작점의 좌표 $P(x, y, z)$, 모서리 방향벡터 e, 모서리가 속한 다각형의 법선벡터 N을 구한다.

3-2 단계 : 3-1단계에서의 모서리와 서로 방향이 반대이며, 접해있는 전개된 모서리($Vd1 \neq 0$)를 찾아 e'과 시작점 $P(x, y, z)$가 전개될 위치 $P(x', y', 0)$를 계산한다.

3-3 단계 : 앞에서 구한 두개의 모서리를 접합시키는 변환행열을 구한다.

3-4 단계 : 전개시킬 모서리가 속한 다각형에 있는 모든 모서리를 3-3 단계에서 구한 변환행열로서 전개시켜 $Vd1$과 $Vd2$를 구한다.

모든 모서리가 전개될때 까지 3-1에서 부터 3-4 단계를 반복한다.

6. 결 과

본 연구에서 개발한 알고리즘을 이용하여 입체를 전개한 전개도면이 Fig. 9~15에 나타나 있다. 화면을 둘 이상으로 분할(view port)하여 화면 1에서는 투상도를 그리고 화면 2 또는 3에서는 전개도를 그리게 하였다. 일률적으로 scaling 하기 어려우므

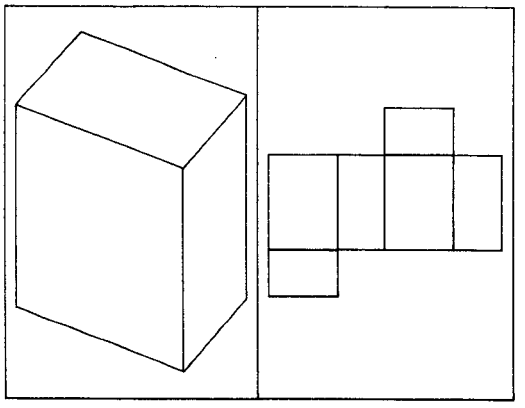

Fig. 9 Development of hexahedron

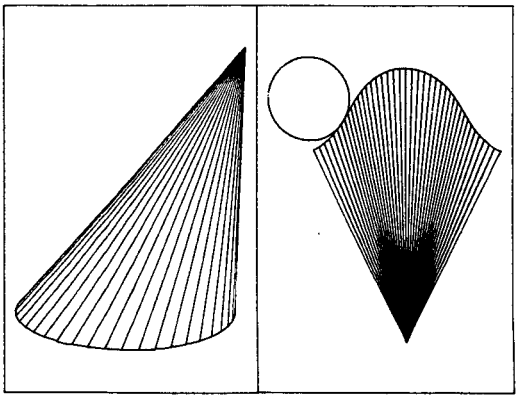

Fig. 10 Development of cone(oblique)

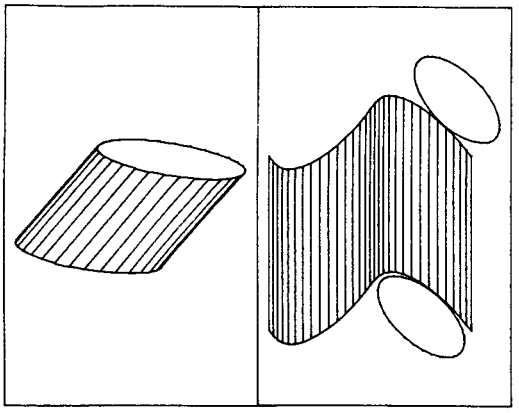

Fig. 11 Development of cylinder(oblique, elliptic)

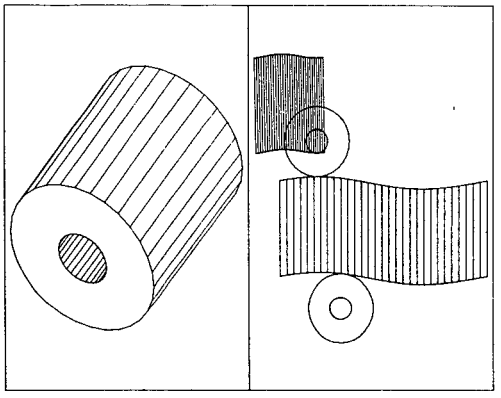

Fig. 12 Development of hollow cylinder

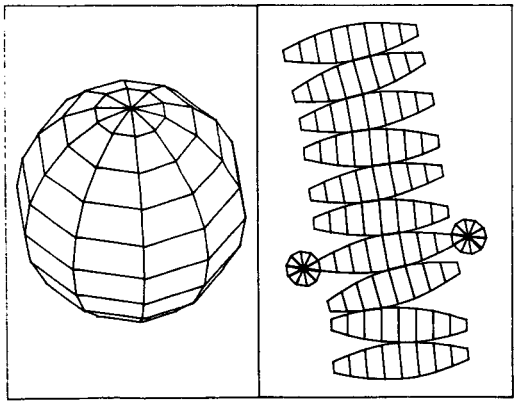

Fig. 13 Development of sphere(Ⅰ)

로 자동으로 화면 가득히 그리도록 하였다.

자동으로 접합길이를 최소화 한다면, 중공원주등의 속이 빈 물체의 경우는 전개도가 겹치게 나타나며(Fig. 12), 비슷한 형태의 입체에 대해서 외형치

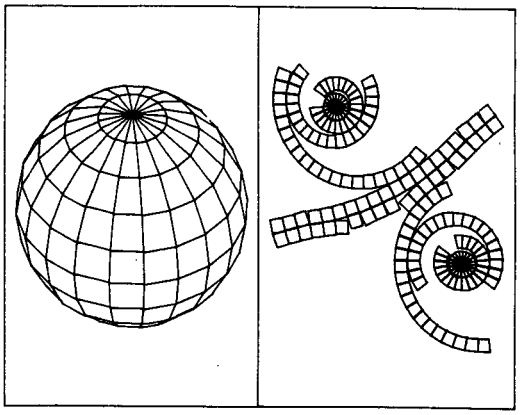

Fig. 14 Development of sphere(Ⅱ)

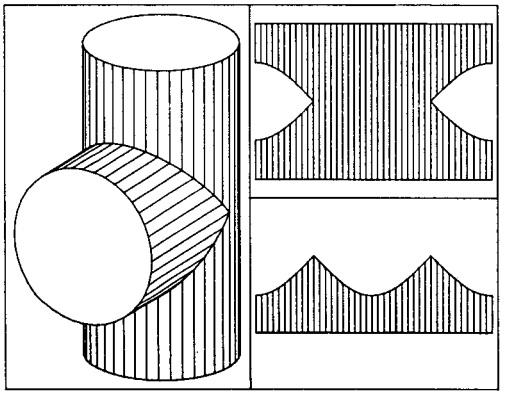

Fig. 15 Development intersecting cylinders

수가 달라짐에 따라 전개도의 모양이 달라질 수도 있다(Fig. 13, 14). 또한 같은 크기의 모서리가 많은 경우 반올림 오차에 따라서도 전개되는 형태가 달라질 수 있다. 따라서 일부분을 삭제하여 부분적으로 적용하거나, 어느 한 방향의 모서리 길이가 크다면 이방향과 수직방향으로 주로 전개되는 성질을 이용할 수 있다. 그러나 제관작업에 있어서 어려운 곳은 곡면부분이며, 이를 다수의 평면으로 근사시킴으로써 곡선부 보다 큰 모서리가 많기 때문에 사용상 큰 문제점은 없다.

Fig. 15는 동일 직경의 원기둥(원을 40 등분)을 직교시킨 것으로 상대편의 내부에 속한 부분(A in B, B in A)을 제거한후 각각 전개한 것이다.

7. 결 론

개인용 컴퓨터에서 사용할 수 있는 간단한 데이

터 구조를 설계하고, 기본적인 입체를 다면체 모델로 생성시키는 모델러(modeller)를 개발하였다. 또한 이를 통하여 발생시킨 입체에 대하여 접합길이를 최소로 하여 전개하는 알고리즘과 프로그램이 개발되었다. 이 프로그램은 모든 메뉴가 단색모니터에 한글로 표시되고 칼러모니터에는 투시도 및 전개도가 그려지는 dual-screen 방식의 대화형이므로, 사용자가 키보드를 통해서 메뉴를 선택하고 데이터를 입력함으로써 판금 및 제관작업등에 필요한 전개도를 쉽게 얻을 수 있다. 그러나 접합길이를 최소로 하는 방법은 외형치수의 영향을 받으며, 일반적인 모든 입체에 대하여 적용하기는 어렵지만, 대부분의 제관작업에 필요한 밑그림을 제공해 줄 수 있다.

참 고 문 헌

(1) 정인성, 1986, "공업제도", 동명사, pp. 117~128.

(2) Norton, Peter, 1985, "The Peter Norton Pogramer's guide to the IBM PC", Microsoft Press, pp. 171~186, 225~240.

(3) Giloi, W.K., 1978, "Interctive Computer Graphics", Prentice-Hall, pp. 26~74.

(4) 채희창, 1988, "개인용 컴퓨터를 이용한 상관선의 도시", 대한기계학회논문집, 제12권, 제1호, pp. 173~182.

(5) Mäntylä, M., 1983, "Topoligical Analysis of Polygon Meshes", Computer-Aided Design, Vol. 15, No.4. pp. 228~234.

(6) Mäntylä, M., 1988, "An Introduction to Solid Modeling", Computer Science press, pp. 263~300.

(7) Newman, W.M. and Sproull, R.F., 1982, "Principles of Interactive Computer Graphics", Addison-Wesley, pp. 505~511.

(8) Plastok, R.A. and Kalley, G., 1986, "Theory and Problems of Computer Graphics", McGraw-Hill, pp. 115~126.

(9) Park, Chan S., 1985, "Interactive Microcomputer Graphics", Addison-Wesley, pp. 133~150.

(10) Wirth, N., 1976, "Algorithms+Data Structures =Programs", Prentice-Hall, pp. 56~82.