

# 컴퓨터에 의한 원피스드레스 原形의 自動製圖에 관한 研究

— 프린세스라인 원피스드레스 —

## A Study on the Automatic Drafting of Onepiece Dress Pattern by Computer — Princess Line Onepiece Dress —

嶺南大學校 家政大學 衣類學科  
시간강사 權 美 靜

*Dept. of Clothing and Textiles, College of Home Economics, Yeungnam Univ.*

A Part Time Lecturer; **Mee Jeoung Kwon**

### < 목 차 >

I. 序 論	IV. 結 論
II. 研究方法 및 節次	參考文獻
III. 結 果	

### < Abstract >

The purpose of this study was to develop a computer program to draft princess line onepiece dress pattern.

To this study, the Turbo Pascal Compiler and the Pascal language were used.

The procedures of the study were as follows:

1. A basic onepiece dress pattern was drafted by the hand operation.
2. The co-ordinate points were instituted for indication of relative position of all necessary points in drafting. Total sixty-nine co-ordinate points (front: thirty-five, back: thirt-four) were instituted.
3. The two subprograms for main program were developed to solve geometric problems and to draft straight lines and curved lines of the onepiece dress pattern.
4. The program was accomplished by putting individual body measurement
5. Grading fo five size was given.

### I. 序 論

컴퓨터는 인간이 해온 복잡하고 많은 작업 과정을

능률적이고 신속하게 처리 해준다. 일부 분야에서는 이미 정보처리, 자료분석 등에 폭넓게 사용되고 있으며 최근에는 그래픽 (graphic)을 이용하는 디자인 (design)에도 도입되어 능률화와 신속화를 기하고

있다. 生産部門과는 달리 設計(design)부문의 컴퓨터화는 設計라는 작업이 手作業이라는 점에서 디지털(digital) 취급을 중심으로 하는 컴퓨터화의 이행이 곤란하다고 생각되어져 온 것도 사실이다.

自動設計(automatic design)와 초기의 CAD (Computer Aided Design)는 1960년대에 들어서면서 발표가 되어 1970년대에는 컴퓨터 산업이 비약적으로 성장하고 그 규모와 성능이 발전함으로써 CAD

와 CAM(Computer Aided Manufacture)의 효과적인 결합이 이루어졌다. 1980년대에는 CAD/CAM System을 대기업에서는 물론 중소기업에서도 도입하고 있는 실정이다.

衣類産業體는 歐美와 日本에서는 1970년대부터, 國內에서는 1980년대 초부터 컴퓨터를 도입하여 패턴 메이킹(Pattern making), 그레이딩(Grading), 마킹(Marking), 재단(Cutting), 봉제(Sewing) 작

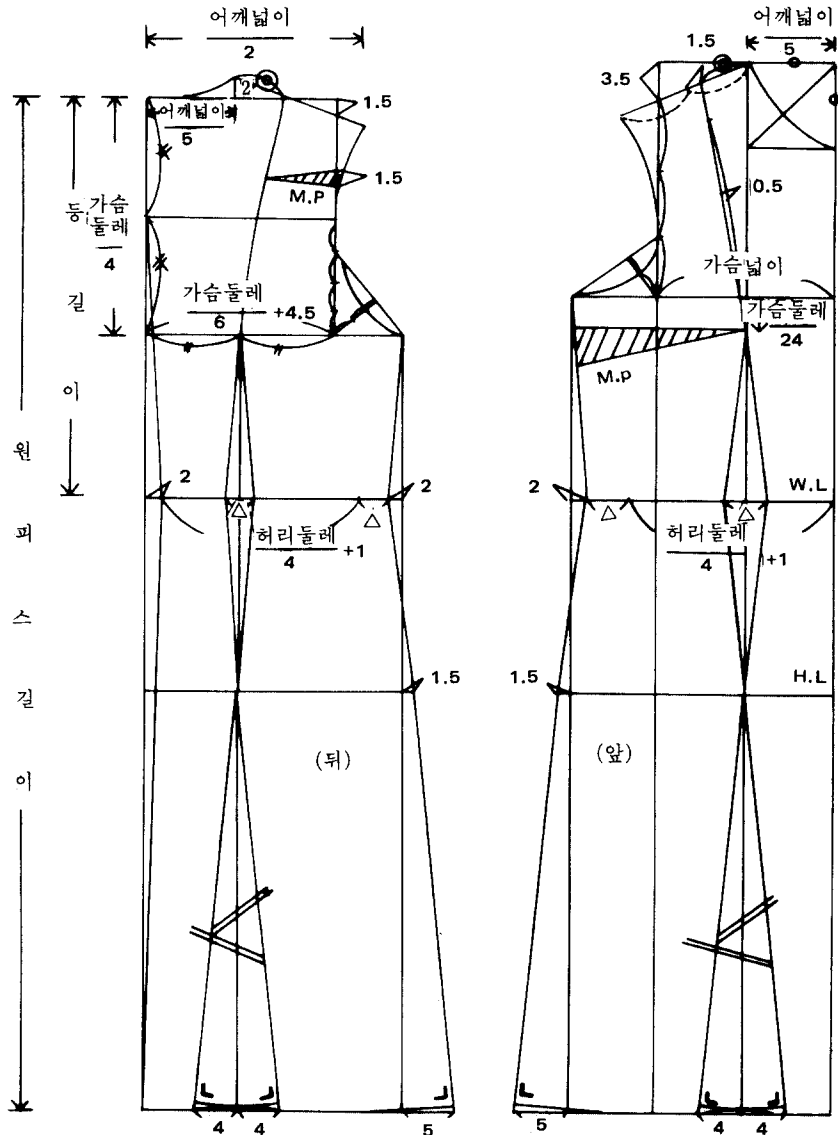


Fig. 1. 프린세스라인 원피스트레스 원형의 제도법 (1/7축도)

업에서 CAD/CAM화를 통한 自動화를 이루려는 경향이다.

衣類學分野에서는 컴퓨터를 이용하여 여러 衣服原形을 自動製圖할 수 있는 프로그램을 개발하여 正確性和 效率성을 높일 수 있음을 증명하였다.

本 研究는 파스칼(Pascal)언어를 사용하여 원피스드레스 原形을 自動製圖할 수 있는 프로그램을 開發하고자 한다.

많은 선행 연구들은 身體計測值 項目을 최소로 하면서 몸에 잘 맞는 원형을 제도하는데 초점을 맞추어 왔으나 入力項目을 제한하면 여러 기하학적(Geometric)인 문제들의 모듈(Module)화가 이루어지지 않을 뿐 아니라 이들 프로그램의 일부를 재활용하기가 쉽지 않음을 고려해 볼 수 있다.

본 연구에서는 신체계측치 입력 과정을 능률화하여 필요에 따라 한 항목에서 일곱 항목의 신체계측치를 입력하여 프로그램을 실행할 수 있게 하고 原形製圖에서 문제시 되는 곡선(curve) 부분을 해결하기 위하여 부프로그램(subprogram)들을 자체 개발하여 사용하였다.

## II. 研究方法 및 節次

### 1. 使用機種

프로그램은 Pascal語로 만들어 IBM PC/AT용 Turbo Pascal Compiler (Ver, 4.0)로써 실행시켰다.

### 2. 원피스드레스原形의 製圖法

本 研究에서 사용한 원피스드레스原形은 李順媛, 林元子の 프린세스라인 드레스의 기본형을 기초로 하여 실무자의 의견을 참고하여 선정하였다. 이는 (Fig. 1)과 같다.

원피스드레스를 手動으로 製圖하기 위해 필요로 하는 身體計測值는 가슴둘레, 등길이, 어깨넓이, 가슴넓이, 허리둘레, 엉덩이둘레, 원피스길이의 전체 일곱 項目이다. 이 외에도 프로그램상에 정해놓은 등넓이, 다이트길이, 밑위길이, 밑단넓이 등의 치수를 임의의 다른 치수로 바꾸어 입력 할 수 있게 하여 상세한 부분까지 수정 가능하도록 하였다. 이는 정성적인 표준체형이 아닌 경우에도 손쉽게 보정할 수 있는

장점을 가진다.

## 3. 프로그램의 構成

### 3-1. 座標點 設定

自動製圖를 위해서 필요한 모든 基準點들의 상대적 위치를 나타내는 좌표점을 설정하였다. 총 69개 (앞 : 35개, 뒤 : 34개)의 좌표점으로 (Fig. 2)에 나타나 있다.

### 3-2. 부프로그램 (Subprogram)

주프로그램에 사용되는 Unit에는 자체개발한 GEOMET과 GGRAPH\*가 있다.

原形製圖는 직선부분과 곡선부분으로 구성되는 원형제도 (Pattern Making)의 주요점은 곡선부분을 어떻게 처리하느냐에 달려있다. 이를 위해 GEOME-

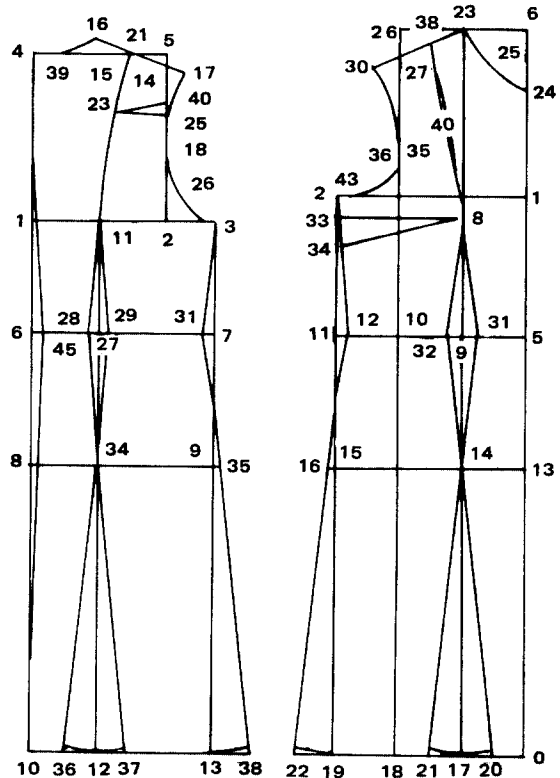


Fig. 2. 좌표점 설정 (1/10축도)

\*Unit GGRAPH는 그래프를 출력하는 프로그램으로서 Hercules Graphic Device 와 Qnix 社의 QLBP 2000 Laser printer로 그래픽이 출력되도록 프로그램 되어 있다.

T와 GGRAPH에 다음과 같은 부프로그램(Subprogram)을 설정하였으며 각각의 기능은 다음과 같다.

Unit GEOMET는 여러 기하학적인 문제 해석 프로그램을 제공한다(Table 1, Table 2)는 GGRAPH의 부프로그램들이다. 이들 부프로그램은 의복원형의 직선부와 곡선부를 그려 주는데 특히 원 퍼스트레스原形 製圖時의 곡선부(목둘레, 진동둘레, 프린세스라인, 옆선, 밑단선)에는 GARC3P(Fig. 3) GRACR2P(Fig. 4), TRIANGLE3(Fig. 5), BÉZER(Fig. 7)가 있다.

또한 이들 부프로그램에서 곡선을 그리는 방법에는, 호(Arc)의 방법, 삼각함수를 이용하는 방법, 그리고 베지에(BÉZIER) Curve 등이 있다.

**Table 1. Unit GEOMET**

---

TYPE  
POINT=RECORD  
X, Y: REAL;  
END;  
FUNCTION DIST (A,B: POINT): REAL;  
{두점 A, B사이의 거리를 구한다.}  
PROCEDURE PDIST (A, B: POINT; D: REAL; VAR P: POINT);  
{두점 A, B를 잇는 직선을 따라 점 B로부터 거리 D만큼 떨어진 점 P를 구한다.}  
PROCEDURE OTHPOS (A, B, C:POINT; VAR D: POINT);  
{두점 A, B를 잇는 직선에 연직인 선분이 점 C를 지날때 이 직선과 선분(AB)이 만나는 점 D를 구한다.}  
PROCEDURE MIDPOS (A, B:POINT; VAR C: POINT);  
{두점 A, B사이의 중점 C를 구한다.}  
PROCEDURE TWO PER (A, B, C, D, R:POINT; VAR Q:POINT);  
{직선 (AB)에 놓은 한점 R에서 직선(AB)에 연직인 선분이 다른 직선(CD)와 만나는 점 A를 구한다.}  
FUNCTION ANGLE (A, B: POINT): REAL;  
{두점 A, B를 잇는 직선이 x축과 이루는 각을 구한다. (0..pi)}

---

\*Bézier Curve는 프랑스 공학자 Pierre Bézier가 1960년경 Computer Aided Design을 하기위해 개발했다. Bézier Curve는 Bézier Cubics라고도 하는데 직선에 가까운 곡선을 그릴때 또는 호와 직선만으로 표현하기 곤란한 부분을 그리는 데 쓰인다.

FUNCTION PANGLE (A: POINT): REAL;  
{한점 a가 원점에 대하여 x축과 이루는 각을 구한다. (0..2\*pi)}  
PROCEDURE SUPERPOS (A, B, C, D: POINT; VAR Q: POINT);  
{두 직선(AB, CD)가 만나는 교점 Q를 구한다.}  
PROCEDURE ORTHO (AR, BR, CR: POINT; VAR R: POINT; VAR RAD: REAL);  
{한직선상에 있지 않은 세점을 지나는 원의 반경(RAD)과 원점의 좌표 R를 돌려준다.}  
FUNCTION THRANGLE (A,B,C,:POINT):REAL  
{세점 ABC가 이루는 각 ABC의 값을 돌려준다(0-180)}  
PROCEDURE CIRINTER (AR, BR, CR: POINT; RAD: REAL; VAR P1, P2, R: POINT);  
{한 직선상에 있지 않은 세점 A,B,C가 이루는 각도 내에 반경 Rad인 원을 둘 때 두직선 AB, BC와 만나는 점 P1, P2와 원의 중심좌표 R을 구한다.}  
PROCEDURE PARALLEL (AP, BP: POINT; DIST:: REAL; AVR CP, DP: POINT);  
{두점 AP, BP가 만드는 선분에 직각으로 거리 dist만큼 평행이동}

---

**Table 2. Unit GGRAPH**

---

TYPE BEZIER·POINT = ARRAY[0..20] OF POINT;  
PROCEDURE GLINE (P1, P2: POINT);  
{두점 P1, P2 사이에 직선을 긋는다}  
PROCEDURE GINIT (ID: INTEGER);  
{Graphic evic을 Open한다}  
PROCEDURE GARC3P (P1, P2, P3: POINT);  
{세점 P1, P2, P3가 주어질 때 P1, P2, P3의 삼각형의 외접원에 해당하는 호를 P1에서 시작하여 P3까지 그린다.}  
PROCEDURE GARCR2P (CNT, P1, P2: POINT; RADIUS: REAL);  
{중점(CNT)에서 반경(Radius)에 해당하는 호를 각(P1, CNT, P2) 만큼 P1에서 시작하는 호를 그린다.}  
PROCEDURE TRIANGLE3 (P1, P2, A, B: POINT; S: INTEGER);  
{두점 P1, P2사이의 호를 그리기 위하여 직선 A, B라는 보조선을 이용한다. S에 따라 호를 발전된다.}  
PROCEDURE BEZIER (N, M: INTEGER; CONTROL: BEZIER POINT);  
{Bézier curve를 그린다. N(0..20)은 control point의 개수 M(0..50)은 곡선을 몇 등분하여 그릴지를 결정하며 CONTROL은 bécier의 control point를 지칭

한다.

PROCEDURE GTRIANG-DATA (P<sub>1</sub>, P<sub>2</sub>, A, B:  
POINT; S: INTEGER; CNT: POINT; RAD:  
REAL);

{Triangle 3로써 그리고자 하는 원의 반경(Rad) 및  
중심좌표(CNT)를 구한다.}

PROCEDURE GWINDOW (SHX, SHY, RATE:  
REAL; DIRF, DIRY: INTEGER);

{출력기기의 기준점에서 좌표를 SHX, SHY만큼 이  
동시키고 실측과 출력할 그림의 비율을 RATE로써  
정하며(X, Y)좌표의 양의 방향을(DIRX, DIRY)로  
정한다.}

PROCEDURE CSTRPOS (CNT, A, B:POINT; RAD:  
REAL; P: POINT);

{중심좌표 CNT와 반경 RAD인 원과 직선 A, B가  
만나는 교점중 점 A와 가장 가까운 점 P를 구한다.}

PROCEDURE GTERM;

{Graphic evicel을 Close한다.}

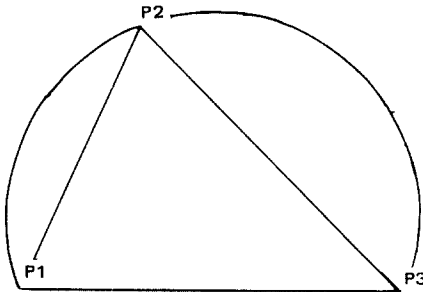


Fig. 3. GARC3P

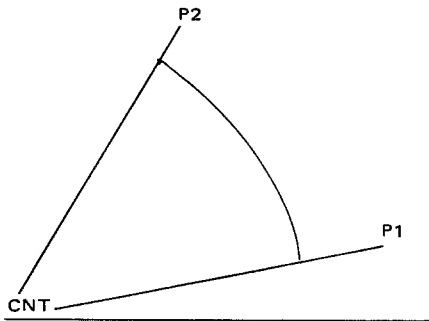


Fig. 4. GARCR2P

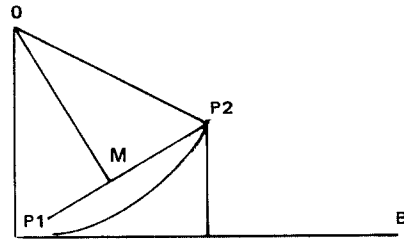


Fig. 5. TRIANGLE3

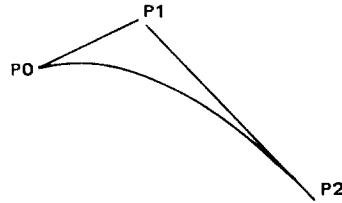


Fig. 6. BEZIER

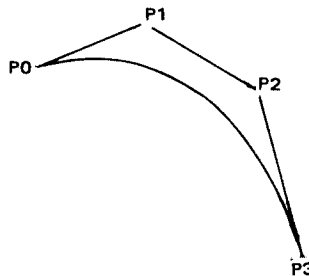


Fig. 7. BEZIER

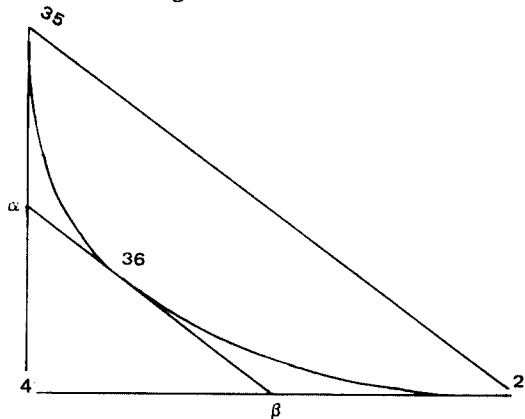


Fig. 8.

本 研究에서는 베지에 Curve를 이용하여 곡선을  
그렸으며, 방법으로는 Bézier Curve의 점을 매개변  
수 U로 표현하였다.

방정식은 다음과 같다.

$$P(U) = \sum_{k=0}^n P_k B_{k,n}(U)$$

$$B_{k,n}(U) = C(n,k) U^k (1-U)^{n-k} (0 \leq U \leq 1)$$

$$C(n, k) = \frac{n!}{K!(n-k)!}$$

$P_k$  : 조정점 (control point)

Fig. 6은 조정점이 3개, Fig. 7은 4개로 이루어진 경우이다.

**3-3. 원피스드레스의 曲線部 形成**

Table 3은 GGRAPH의 부프로그램중에서 원피스드레스의 곡선부 製圖에 적용된 좌표점을 나타낸 것이다.

GARC3P, GAPCR2P, TRIANGLE3의 곡선형성은 앞에서 설명한 바와 같고 Bézier curve가 적용된 곡선은(Fig. 8, 9, 10, 11)에 나타나 있다. 이 곡선의 조정점들은 다음과 같이 ;

① F35-36-2

35에서 2에 평행하고 36을 지나는 직선 중에서 35, A에 만나는 점을  $\alpha$ , 4, 2에 만나는 점을  $\beta$ 라 둔다.

35와 36을 그리는 곡선의 조정점은  $P_0=35$   $P_1=\alpha$   $P_2=36$ 이며 36과 그를 그리는 곡선의 조정점은  $P_0=36$   $P_1=\beta$   $P_2=2$ 이다.

② F 12-16-22

12와 16의 중점을 지나며 12와 16에 연직인 직선과 16과 22에 연직인 직선이 만나는 점을 조정점중의 하나로 둔다.

③ B 18-26-3

26을 지나면서 기울이가 1인 직선이 2와 18에 만나는 점을  $\alpha'$ 라 두고 2와 3이 만나는 점을  $\beta'$ 라 둔다. 18에서 26사이의 곡선을 그리기 위한 조정점은  $P_0=18$ ,  $P_1=2'$ ,  $P_2=26$ , 3에서 26사이의 곡선의 조정점은  $P_0=3$   $P_1=\beta'$   $P_2=26$ 이다.

④ B 4-39-16

16, 15사이의 거리와  $\alpha$ , 15사이의 거리가 같도록

$\alpha$ 를 구한다. 조정점은  $P_0=4$ ,  $P_1=39$ ,  $P_2=\alpha$ ,  $P_3=16$ 이다.

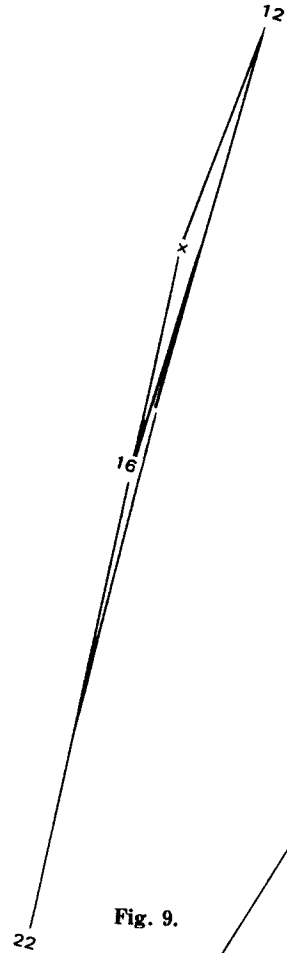


Fig. 9.

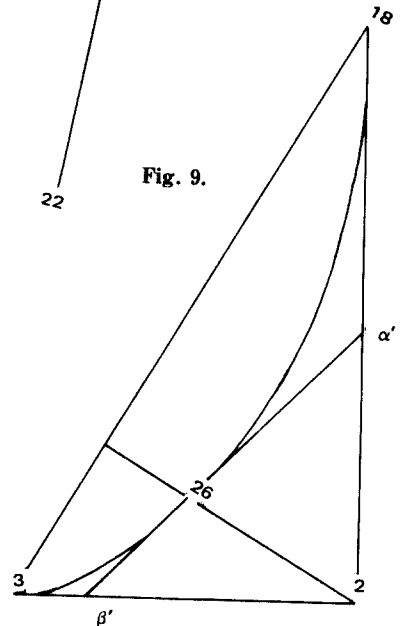


Fig. 10.

Table 3. 원피스드레스의 곡선부 형성표

	GAR3P		GAPCR2P		TRINGLE3		BEZIER	
	3점		2점		곡선 보조선		3점 이상	
F	23-25-24	F	0-21	F	30-35	35-26	F	35-36-2
F	8-40-38	F	20-22	F	2-12	12-43	F	12-16-22
		B	38-36	B	17-18	18-5	B	18-26-3
		B	37-10	B	21-11	11-29	B	4-39-16
				B	31-3	3-7		
				B	31-35	35-38		

F : 앞판 B : 뒷판

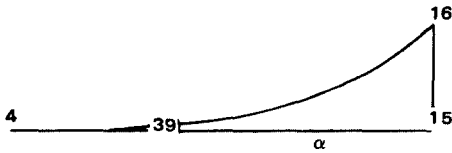


Fig. 11.

Table 4.

```

*****
***** AUTO ONEPIECE DESIGN *****
*****
***** PROCEDURE READVALUE,CALCULATE,GRADING,OTHERVALUE *****
***** STANDINPUT *****
*****

USES GGRAPH,GEOMET,CRT:

LABEL 100:

TYPE
POSITION-ARRAY(0..50) OF POINT:
CONST
ONERAD=0.017453293:
VAR
BASEFLAG:CHAR: OTHERFLAG,SGRADING,I:INTEGER:
FRONT,BACK:POSITION:
STATURE,BUST,WAIST,HIP,BUSTLINE,BACKLG,HIPLINE,OPL,NP,BP:REAL:
FBETA,FKAPA,FRONTDART,DM10,BACKW,ARMPITW,BREASTW,SHOULDW,AG:REAL:
FTBETA,FTNBETA,BRAD41,BRAD42,BRAD44,BRAD45,TEMPDIST:REAL:
BALPHA,BOMEGA,HIPHEIGHT,BPOINT,FRONTLG,FRONTWIDT,NOBSP:REAL:
FSKY1,FUARM,FBARM:BEZIER_POINT: { FRONT BEZIER CONTROL POINTS }
BNECK,BUARM,BBARM:BEZIER_POINT: { BACK BEZIER CONTROL POINTS }
TNP,TNQ:POINT: { TEMPORARY VALUE }

PROCEDURE READVALUE(X,T:INTEGER; VAR VALUE:REAL):
BEGIN
GOTOXY(X,Y): WRITE(' ':20):
GOTOXY(X,Y): READLN(VALUE):
END: ( ***** END READVALUE ***** )

PROCEDURE CALCULATE:
( ***** DEFINE BASE FUNCTION ***** )
BEGIN
NOBSP:=4:
BPOINT:=3:
BALPHA:=-1.5:
BOMEGA:=-1.0: { Space on Back Princess line }
FBETA:=-1.5:
FKAPA:=-1.5: { Space on Front princess line }
HIPHEIGHT:=18:
FRONTDART:=BUST/24:
( ***** LENGTH ***** )
BUSTLINE:=BUST/4:
BP:=BUSTLINE+BPOINT:
FRONTLG:=BACKLG+FRONTDART:
HIPLINE:=HIPHEIGHT+BACKLG:
( ***** WIDTH ***** )
NP:=SHOULDW/5:
BACKW:=BREASTW*2:
FRONTWIDT:=BUST/2+NOBSP:
END: ( *** END CALCULATE *** )

PROCEDURE GRADING(N:INTEGER):
CONST
BUSTS :ARRAY [1..5] OF REAL=(82,85,88,91,94):
BACKLGS:ARRAY [1..5] OF REAL=(35,36,37,38,40):
SHOULDS:ARRAY [1..5] OF REAL=(36,39,40,41,42):
BREASTWS:ARRAY [1..5] OF REAL=(31,32,33,34,35):
OPLS :ARRAY [1..5] OF REAL=(82,83,84,85,88):
HIPS :ARRAY [1..5] OF REAL=(87,90,92,94,96):
WAISTS :ARRAY [1..5] OF REAL=(66,68,70,73,78):
BEGIN
BUST :=BUSTS(N):
WAIST :=WAISTS(N):
HIP :=HIPS(N):
BACKLG:=BACKLGS(N):
OPL :=OPLS(N):
SHOULDW:=SHOULDS(N):
BREASTW:=BREASTWS(N):
CALCULATE:
END: ( ***** END GRADING ***** )

PROCEDURE OTHER_VALUE:
LABEL 10:
VAR
CH:CHAR:
BEGIN
IF OTHERFLAG=0 THEN CALCULATE:
OTHERFLAG:=1:
10:
GOTOXY(10,12): WRITE('-----'):
GOTOXY(10,13): WRITE('A. BACK WIDTH :',BACKW:10:3):
GOTOXY(10,14): WRITE('B. FRONT DART SIZE :',FRONTDART:10:3):
GOTOXY(10,15): WRITE('C. NECK BASE GIRTH :',NP:10:3):
GOTOXY(10,16): WRITE('D. HORIZONTAL SPARE :',NOBSP:10:3):
GOTOXY(10,17): WRITE('E. CROTCH LENGTH :',HIPHEIGHT:10:3):
GOTOXY(10,18): WRITE('F. AUTO CALCULATION '):
GOTOXY(10,19): WRITE('G. EXIT '):
GOTOXY(10,20): WRITE(' SELECT NUMBER (A..G) ? '):
CH:=READKEY:
CASE UPCASE(CH) OF
'A':READVALUE(35,13,BACKW):

```

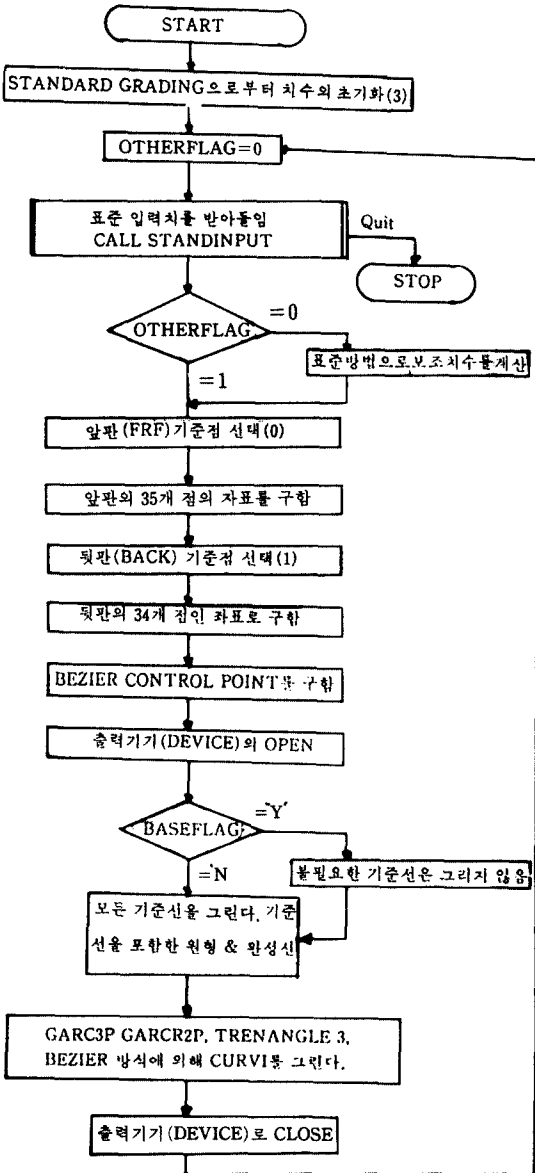


Fig. 12. 흐름도 A

```

'B':READVALUE(35,14,FRONTDART);
'C':READVALUE(35,15,NP);
'E':READVALUE(35,16,HORSP);
'F':READVALUE(35,17,HIPRIGHT);
'G':CALCULATE;
'H':EXIT;
END;
GOTO 10;
END: { ***** END OTHERVALUE ***** }

PROCEDURE STANDINPUT;
LABEL 9,10;
VAR
  CH:CHAR;
BEGIN
  10:CLRSR;
  GOTXY(10, 4); WRITE('0. GIVEN STANDARD GRADING : ',SGRADING);
  GOTXY(10, 5); WRITE('1. BUST GIRTH : ',BUST:10:3);
  GOTXY(10, 6); WRITE('2. BACK LENGTH : ',BACKLG:10:3);
  GOTXY(10, 7); WRITE('3. WAIST GIRTH : ',WAIST:10:3);
  GOTXY(10, 8); WRITE('4. SHOULDER WIDTH : ',SHOULDW:10:3);
  GOTXY(10, 9); WRITE('5. CHEST WIDTH : ',BREASTW:10:3);
  GOTXY(10,10); WRITE('6. HIP GIRTH : ',HIP:10:3);
  GOTXY(10,11); WRITE('7. ONEPIECE DRESS LENGTH : ',OPL:10:3);
  GOTXY(10,12); WRITE('8. OTHER VALUE ');
  GOTXY(10,13); WRITE('9. ERASE BASE LINE (Y,N) : ',BASEFLAG:3);
  GOTXY(10,14); WRITE('A. <<<< DRAW >>>> ');
  GOTXY(10,15); WRITE('Q. <<<< QUIT >>>> ');
  GOTXY(10,17); WRITE(' SELECTION NUMBER (0..A,Q) : ');
  CH:=UPCASE(READKEY);
  CASE CH OF
    '0':BEGIN
      GOTXY(10,3);
      WRITE(' ENTER GRADING NUMBER (1..5) : ');
      CH:=READKEY;
      IF (CH<'1') OR (CH>'5') THEN GOTO 10;
      SGRADING:=ORD(CH)-48;
      GRADING(SGRADING);
      GOTO 10;
    END;
    '1':READVALUE(39, 5,BUST);
    '2':READVALUE(39, 6,BACKLG);
    '3':READVALUE(39, 7,WAIST);
    '4':READVALUE(39, 8,SHOULDW);
    '5':READVALUE(39, 9,BREASTW);
    '6':READVALUE(39,10,HIP);
    '7':READVALUE(39,11,OPL);
    '8':OTHER_VALUE;
    '9':BEGIN
      9:GOTXY(38,13); WRITE(' ':20);
      GOTXY(38,13); READLN(BASEFLAG);
      BASEFLAG:=UPCASE(BASEFLAG);
      IF (BASEFLAG<>'Y') AND (BASEFLAG<>'N') THEN GOTO 9;
    END;
    'A':EXIT;
    'Q':HALT(1);
  END;
  GOTO 10;
END: { ***** END STANDINPUT ***** }

{ ***** MAIN PROGRAM FOR AUTO_ONEPIECE DESIGN ***** }

BEGIN
  { MAIN VALUE }
  SGRADING:=3;
  GRADING(SGRADING);
  BASEFLAG:='Y';
  100:OTHERFLAG:=0;
  STANDINPUT;
  IF OTHERFLAG=0 THEN CALCULATE;
  END GIVEN VALUE ----- }
  { for i:=0 to 49 do begin
    front[i].x:=300; front[i].y:=300; }
    back[i].x:=front[i].x; end; }
  FRONT[0].X:=0.0; { ORIGIN }
  FRONT[0].Y:=0.0;
  FRONT[1].X:=FRONT[0].X;
  { DEFINE FRONT BASELINE }
  FRONT[1].Y:=FRONT[0].Y+OPL-BUSTLINE+FRONTDART;
  FRONT[5].X:=FRONT[1].X;
  FRONT[5].Y:=FRONT[0].Y+OPL-BACKLG;
  FRONT[6].X:=FRONT[0].X;
  FRONT[6].Y:=FRONT[0].Y+OPL+FRONTDART;
  FRONT[8].Y:=FRONT[6].Y-HP;
  FRONT[13].X:=FRONT[1].X;
  FRONT[13].Y:=FRONT[0].Y+OPL-HIPLINE;
  { END DEFINE FRONT BASELINE }
  FRONT[4].X:=FRONT[1].X+BREASTW/2;
  FRONT[4].Y:=FRONT[1].Y;
  FRONT[2].X:=FRONT[0].X+(FRONTWIDTH/2+0.5);
  FRONT[2].Y:=FRONT[1].Y;
  FRONT[3].X:=FRONT[1].X+(FRONT[1].X+FRONT[4].X)/2;
  FRONT[3].Y:=FRONT[1].Y;
  FRONT[7].X:=FRONT[3].X;
  FRONT[7].Y:=FRONT[6].Y;
  FRONT[8].X:=FRONT[3].X;
  FRONT[9].X:=FRONT[3].X;
  FRONT[9].Y:=FRONT[5].Y;
  FRONT[10].X:=FRONT[4].X;
  FRONT[10].Y:=FRONT[5].Y;
  FRONT[11].X:=FRONT[2].X;
  FRONT[11].Y:=FRONT[5].Y;
  FRONT[12].X:=FRONT[11].X-FBETA;
  FRONT[12].Y:=FRONT[5].Y;
  FRONT[14].X:=FRONT[3].X;
  FRONT[14].Y:=FRONT[13].Y;
  FRONT[15].X:=FRONT[2].X;
  FRONT[15].Y:=FRONT[13].Y;
  FRONT[16].X:=FRONT[15].X+1;
  FRONT[16].Y:=FRONT[15].Y;
  FRONT[17].Y:=FRONT[0].Y;
  FRONT[17].X:=FRONT[3].X;
  FRONT[18].X:=FRONT[4].X;
  FRONT[18].Y:=FRONT[0].Y;
  FRONT[19].X:=FRONT[2].X;
  FRONT[19].Y:=FRONT[0].Y;
  FRONT[20].X:=FRONT[17].X-4;
  FRONT[20].Y:=FRONT[0].Y;
  FRONT[21].X:=FRONT[17].X+4;
  FRONT[21].Y:=FRONT[0].Y;
  FRONT[22].X:=FRONT[18].X+5;
  FRONT[22].Y:=FRONT[0].Y;
  FRONT[23].X:=FRONT[6].X+NP;
  FRONT[23].Y:=FRONT[6].Y;
  FRONT[24].X:=FRONT[6].X;
  FRONT[24].Y:=FRONT[6].Y+NP;
  (25) PDIST(FRONT[4],FRONT[6],-7,FRONT[25]);
  FRONT[26].X:=FRONT[4].X;
  FRONT[26].Y:=FRONT[6].Y;
  FRONT[27].X:=FRONT[4].X;
  FRONT[27].Y:=FRONT[26].Y-FRONTDART;
  FRONT[28].X:=FRONT[1].X+SHOULDW/2;
  FRONT[28].Y:=FRONT[6].Y;
  FRONT[29].X:=FRONT[28].X;
  FRONT[29].Y:=FRONT[1].Y;
  (30) SUPERPOS(FRONT[23],FRONT[27],FRONT[28],FRONT[29],FRONT[30]);
  { FRONT WAIST DART }
  (31) TEMPDIST:=ABS(FRONT[12].X-FRONT[5].X)-(WAIST/4+FKAPA);
  IF TEMPDIST<0 THEN HALT(1);
  FRONT[31].X:=FRONT[9].X-TEMPDIST/2;
  FRONT[31].Y:=FRONT[5].Y;
  FRONT[32].Y:=FRONT[5].Y;
  FRONT[43].X:=(FRONT[2].X+FRONT[12].X)/2;
  FRONT[43].Y:=FRONT[2].Y;
  (33) FRONT[33].X:=FRONT[2].X;
  FRONT[33].Y:=FRONT[8].Y;
  CTRIANG DATA(FRONT[2],FRONT[12],FRONT[43],-1,TMP,TEMPD);
  CSTRPOS(TMP,FRONT[8],FRONT[33],TEMPDIST,FRONT[33]);
  (34) FRONT[34].X:=FRONT[2].X;
  FRONT[34].Y:=FRONT[33].Y-FRONTDART;
  CSTRPOS(TMP,FRONT[8],FRONT[34],TEMPDIST,FRONT[34]);
  FRONT[35].X:=FRONT[4].X;
  FRONT[35].Y:=FRONT[4].Y+ABS(FRONT[27].Y-FRONT[4].Y)/3;
  (36) OTHPOS(FRONT[35],FRONT[2],FRONT[4],FRONT[36]);
  PDIST(FRONT[4],FRONT[36],-DIST(FRONT[36],FRONT[4])/2,FRONT[36]);
  (37) FRONT[37].X:=(FRONT[23].X+FRONT[30].X)/2;
  FRONT[37].Y:=(FRONT[23].Y+FRONT[30].Y)/2;
  PDIST(FRONT[23],FRONT[37],1.5,FRONT[37]);
  (38) FRONT[38]:=FRONT[37];
  { FRONT ARC }
  (39) MIDPOS(FRONT[30],FRONT[35],FRONT[39]);
  PARALLEL(FRONT[30],FRONT[35],1,THP,FRONT[39]);
  (40) MIDPOS(FRONT[38],FRONT[8],FRONT[40]);
  PARALLEL(FRONT[8],FRONT[40],0.5,THP,FRONT[40]);
  (44) SUPERPOS(FRONT[0],FRONT[13],FRONT[21],FRONT[14],FRONT[41]);
  OTHPOS(FRONT[0],FRONT[22],FRONT[41],TMP);
  FRAD44:=DIST(TMP,FRONT[41]);
  (45) SUPERPOS(FRONT[20],FRONT[14],FRONT[22],FRONT[16],FRONT[42]);
  OTHPOS(FRONT[0],FRONT[22],FRONT[42],TMP);
  FRAD45:=DIST(TMP,FRONT[42]);
  ----- BACK TYPE ----- }
  BACK[1].X:=FRONT[1].X+(BUST/2+20);
  { ----- BACK DEFINE BASELINE }
  BACK[1].Y:=FRONT[0].Y+(OPL-BUSTLINE);
  BACK[6].Y:=FRONT[0].Y+(OPL-BACKLG);
  BACK[8].Y:=FRONT[0].Y+(OPL-HIPLINE);
  BACK[10].Y:=FRONT[0].Y;
  { END DEFINE BASELINE }
  BACK[2].X:=BACK[1].X+BACKW/2;
  BACK[2].Y:=BACK[1].Y;
  BACK[3].X:=BACK[1].X+(FRONTWIDTH/2);
  BACK[3].Y:=BACK[1].Y;
  BACK[4].X:=BACK[1].X;
  BACK[4].Y:=BACK[1].Y+BUSTLINE;
  BACK[5].X:=BACK[2].X;
  BACK[5].Y:=BACK[4].Y;
  BACK[6].X:=BACK[1].X;
  BACK[7].X:=BACK[3].X;
  BACK[7].Y:=BACK[6].Y;
  BACK[8].X:=BACK[1].X;

```



```

BACK[ 9].X:=-BACK[ 3].X;
BACK[ 9].Y:=-BACK[ 8].Y;
BACK[10].X:=-BACK[ 1].X;
BACK[11].X:=(BACK[ 1].X+BACK[ 2].X)/2;
BACK[11].Y:=-BACK[ 1].Y;
BACK[12].X:=-BACK[11].X;
BACK[12].Y:=-BACK[10].Y;
BACK[13].X:=-BACK[ 3].X;
BACK[13].Y:=-BACK[10].Y;
BACK[14].X:=-BACK[ 2].X;
BACK[14].Y:=-BACK[ 5].Y-1.5;
BACK[15].X:=-BACK[ 4].X-NP;
BACK[15].Y:=-BACK[ 4].Y;
BACK[16].X:=-BACK[15].X;
BACK[16].Y:=-BACK[ 4].Y+2;
(17) PDIST(BACK[14],BACK[16],-DIST(FRONT[23],FRONT[30]),BACK[17]);
BACK[18].X:=-BACK[ 2].X;
BACK[18].Y:=(BACK[ 4].Y+BACK[ 2].Y)/2;
BACK[19].X:=-BACK[ 2].X;
BACK[19].Y:=-ABS(BACK[ 4].Y-BACK[ 1].Y)+2/3*BACK[ 1].Y;
BACK[20].X:=-BACK[ 1].X;
BACK[20].Y:=-BACK[18].Y;
(21) BACK[21].X:=(BACK[17].X+BACK[16].X)/2;
BACK[21].Y:=(BACK[17].Y+BACK[16].Y)/2;
PDIST(BACK[17],BACK[21],1.5,BACK[21]);
BACK[22].X:=-BACK[ 1].X;
BACK[22].Y:=-BACK[19].Y;
BACK[27].X:=-BACK[11].X;
BACK[27].Y:=-BACK[ 6].Y;
(28) BACK[45].X:=-BACK[16].X-BALPHA;
BACK[45].Y:=-BACK[16].Y;
BACK[31].X:=-BACK[17].X+BAI*PDI;
TEMPDIST:=-ABS(BACK[31].X-BACK[45].X)-(WAIST/4+BOMEGA);
IF TEMPDIST<0 THEN HALT(1);
BACK[28].X:=-BACK[27].X-TEMPDIST/2;
BACK[28].Y:=-BACK[ 6].Y;
BACK[29].X:=-BACK[27].X+TEMPDIST/2;
(23) CTRIANG_DATA(BACK[11],BACK[21],BACK[11],BACK[29],-1,TMP,TEMPDIST);
CSTRPOS(TMP,BACK[22],BACK[19],TEMPDIST,BACK[23]);
(24) BACK[24].X:=-BACK[ 2].X;
BACK[24].Y:=-BACK[23].Y-1.5/2;
GTRIANG_DATA(BACK[18],BACK[17],BACK[18],BACK[ 5],-1,TMP,TEMPDIST);
CSTRPOS(TMP,BACK[23],BACK[24],TEMPDIST,BACK[24]);
(25) BACK[25].X:=-BACK[ 2].X;
BACK[25].Y:=-BACK[24].Y+1.5;
CSTRPOS(TMP,BACK[23],BACK[25],TEMPDIST,BACK[25]);
(26) OTHPOS(BACK[ 3],BACK[18],BACK[2],BACK[26]);
PDIST(BACK[2],BACK[26],-DIST(BACK[26],BACK[2])/3,BACK[26]);
(30) BACK[29]:=-BACK[29];
BACK[31].Y:=-BACK[ 6].Y;
BACK[34].X:=-BACK[11].X;
BACK[34].Y:=-BACK[ 8].Y;
BACK[35].X:=-BACK[ 8].Y-1;
BACK[35].Y:=-BACK[ 8].Y;
BACK[36].X:=-BACK[18].X+4;
BACK[36].Y:=-BACK[10].Y;
BACK[37].X:=-BACK[12].X-4;
BACK[37].Y:=-BACK[10].Y;
BACK[38].X:=-BACK[13].X-5;
BACK[38].Y:=-BACK[10].Y;
(39) BACK[39].Y:=-BACK[15].X+ABS(BACK[15].X-BACK[ 4].X)+2/3;
BACK[39].Y:=-BACK[ 4].Y;
(40) BACK[40].X:=(BACK[17].X+BACK[18].X)/2;
BACK[40].Y:=(BACK[17].Y+BACK[18].Y)/2;
PARALLEL(BACK[17],BACK[40],1,TMP,BACK[40]);
(41 AND RADIUS )
SUPERPOS(BACK[38],BACK[35],BACK[36],BACK[34],BACK[41]);
OTHPOS(FRONT[0],FRONT[22],BACK[41],TMP);
BRAD41:=DIST(TMP,BACK[41]);
(42 AND RADIUS )
SUPERPOS(BACK[37],BACK[34],BACK[10],BACK[ 8],BACK[42]);
OTHPOS(FRONT[0],FRONT[22],BACK[42],TMP);
BRAD42:=DIST(TMP,BACK[42]);
(43) BACK[43]:=-BACK[11];
(44) BACK[44].Y:=(BACK[ 4].Y+BACK[ 1].Y)/2.0;
BACK[44].X:=-BACK[ 1].X;
(46) SUPERPOS(BACK[44],BACK[45],BACK[ 3],BACK[ 1],BACK[46]);
(47) SUPERPOS(BACK[45],BACK[10],BACK[ 9],BACK[ 8],BACK[47]);
( INCLINE ANGLE )
THETA:=-THRANGLE(FRONT[33],FRONT[8],FRONT[34]);
BTHTA:=-THRANGLE(BACK[24],BACK[23],BACK[25]);
( FRONT BEZIER CONTROL POINTS )
FUARM[0]:=-FRONT[35];
OTHPOS(FRONT[35],FRONT[2],FRONT[36],TMP);
TEMPDIST:=DIST(TMP,FRONT[36]);
TMP:=-FRONT[35];THQ:=FRONT[2];
PARALLEL(TMP,THQ,-TEMPDIST,TMP,THQ);
SUPERPOS(TMP,THQ,FRONT[ 4],FUARM[1]);
SUPERPOS(TMP,THQ,FRONT[ 2],FRONT[ 4],FBARM[1]);
FUARM[2]:=-FRONT[36];
FBARM[0]:=-FRONT[36];
FBARM[2]:=-FRONT[ 2];
FSKTI[0]:=-FRONT[16];
WIDPOS(FRONT[16],FRONT[12],FSKTI[1]);
TWO_PER(FRONT[12],FRONT[16],FRONT[16],FRONT[22],FSKTI[1],FSKTI[1]);
FSKTI[2]:=-FRONT[12];
( BACK BEZIER CONTROL POINTS )
BNECK[0]:=-BACK[16];
BNECK[1].X:=-BACK[15].X+ABS(BACK[15].Y-BACK[16].Y);
BNECK[1].Y:=-BACK[15].Y;
BNECK[2]:=-BACK[39];
BNECK[3]:=-BACK[ 4];
BUARM[0]:=-BACK[18];
BBARM[0]:=-BACK[ 3];
BUARM[2]:=-BACK[26];
BBARM[2]:=-BACK[26];
BUARM[1].X:=-BACK[ 2].X;
BUARM[1].Y:=(BACK[2].X-BACK[26].X)+BACK[26].Y;
BBARM[1].Y:=-BACK[2].Y;
BBARM[1].X:=(BBARM[1].Y-BACK[26].Y)+BACK[26].X;
( ***** DRAW ***** )
GINIT(1);
GWINDOW(10.5,1,-1,-1);
( 1. BACK FRAME LINE )
IF BASEFLAG='N' THEN GLINE(FRONT[ 6],FRONT[26]); ( HORIZONTAL LINE )
GLINE(FRONT[ 1],FRONT[ 2]);
IF BASEFLAG='N' THEN GLINE(FRONT[ 5],FRONT[11])
ELSE GLINE(FRONT[5],FRONT[12]);
GLINE(FRONT[13],FRONT[16]);
GLINE(FRONT[ 0],FRONT[22]);
IF BASEFLAG='N' THEN GLINE(FRONT[ 6],FRONT[ 0])
ELSE GLINE(FRONT[24],FRONT[ 0]); ( VERTICAL LINE )
IF BASEFLAG='N' THEN BEGIN
GLINE(FRONT[ 7],FRONT[17]);
GLINE(FRONT[26],FRONT[18]);
GLINE(FRONT[ 2],FRONT[19]);
END;
( 2. BACK FRAME LINE )
IF BASEFLAG='N' THEN GLINE(BACK[ 5],BACK[ 4]);
IF BASEFLAG='N' THEN GLINE(BACK[ 3],BACK[ 1]);
ELSE GLINE(BACK[ 3],BACK[46]);
IF BASEFLAG='N' THEN GLINE(BACK[ 7],BACK[ 6])
ELSE GLINE(BACK[31],BACK[45]);
IF BASEFLAG='N' THEN GLINE(BACK[35],BACK[ 8])
ELSE GLINE(BACK[35],BACK[47]);
GLINE(BACK[38],BACK[10]); ( HORIZONTAL LINE )
IF BASEFLAG='N' THEN BEGIN
GLINE(BACK[ 3],BACK[13]);
GLINE(BACK[ 5],BACK[ 2]);
GLINE(BACK[11],BACK[12]);
END;
IF BASEFLAG='N' THEN GLINE(BACK[ 4],BACK[10])
ELSE GLINE(BACK[ 4],BACK[44]); ( VERTICAL LINE )
( ***** FRONT DART ***** )
GLINE(FRONT[33],FRONT[ 8]);
GLINE(FRONT[ 8],FRONT[34]);
( ***** BACK DART ***** )
GLINE(BACK[25],BACK[23]);
GLINE(BACK[23],BACK[24]);
( 3. FRONT OTHER FRAME LINE )
GLINE(FRONT[23],FRONT[37]);
GLINE(FRONT[37],FRONT[ 8]);
IF BASEFLAG='N' THEN BEGIN
GLINE(FRONT[37],FRONT[ 8]);
GLINE(FRONT[ 8],FRONT[31]);
GLINE(FRONT[ 8],FRONT[32]);
END;
GLINE(FRONT[31],FRONT[14]);
GLINE(FRONT[32],FRONT[14]);
GLINE(FRONT[14],FRONT[20]);
GLINE(FRONT[14],FRONT[21]);
IF BASEFLAG='N' THEN GLINE(FRONT[ 8],FRONT[38]);
GLINE(FRONT[38],FRONT[30]);
IF BASEFLAG='N' THEN BEGIN
GLINE(FRONT[ 2],FRONT[12]);
GLINE(FRONT[12],FRONT[16]);
END;
GLINE(FRONT[16],FRONT[22]);
GLINE(FRONT[ 8],FRONT[32]);
GLINE(FRONT[ 8],FRONT[31]);
( 3. BACK OTHER FRAME LINE )
GLINE(BACK[16],BACK[21]);
IF BASEFLAG='N' THEN BEGIN
GLINE(BACK[21],BACK[11]);
GLINE(BACK[11],BACK[29]);
GLINE(BACK[11],BACK[28]);
END;
GLINE(BACK[29],BACK[34]);
GLINE(BACK[28],BACK[34]);
GLINE(BACK[34],BACK[37]);
GLINE(BACK[34],BACK[36]);
GLINE(BACK[21],BACK[17]);
IF BASEFLAG='N' THEN BEGIN
GLINE(BACK[ 3],BACK[31]);
GLINE(BACK[31],BACK[35]);
END;
GLINE(BACK[35],BACK[38]);
GLINE(BACK[44],BACK[45]);
GLINE(BACK[45],BACK[10]);
GLINE(BACK[11],BACK[29]);

```

```

GLINE(BACK[11],BACK[28]);
{ 4. FRONT NECK CURVE }
GARC3P(FRONT[24],FRONT[25],FRONT[23]);
{ 4. BACK NECK CURVE }
BEZIER(3,15,BNECK);
{ 5. FRONT ARMPIT CURVE }
TRIANGLES(FRONT[30],FRONT[35],FRONT[35],FRONT[26],-1);
BEZIER(2,15,FUARM);
BEZIER(2,15,FBARM);
{ 5. BACK ARMPIT CURVE }
TRIANGLES(BACK[18],BACK[17],BACK[18],BACK[5],-1);
BEZIER(2,15,BUARM);
BEZIER(2,15,BBARM);
{ 6. FRONT BOTTOM ARC }
GARC2P(FRONT[41],FRONT[ 0],FRONT[21],FRAD44);
GARC2P(FRONT[42],FRONT[20],FRONT[22],FRAD45);
{ 6. BACK BOTTOM ARC }
GARC2P(BACK[41],BACK[38],BACK[36],BRAD41);
GARC2P(BACK[42],BACK[37],BACK[10],BRAD42);
{ 7. FRONT SKIRT CURVE }
TRIANGLES(FRONT[ 2],FRONT[12],FRONT[12],FRONT[43],-1);
BEZIER(2,15,FSKT1);
{ 7. BACK SKIRT CURVE }
TRIANGLES(BACK[31],BACK[35],BACK[35],BACK[38],-1);
TRIANGLES(BACK[31],BACK[ 3],BACK[ 3],BACK[ 7],-1);
{ 8. FRONT MIDDLE CURVE }
GARC3P(FRONT[38],FRONT[40],FRONT[ 8]);
{ 8. BACK MIDDLE CURVE }
TRIANGLES(BACK[11],BACK[21],BACK[11],BACK[29],-1);
{ ***** END DRAW ***** }
CTERM;
GOTO 100;
END.
{ ***** END PROGRAM ***** }

```

III. 結 果

원피스드레스의 基本原形을 선정하여 座標點을 설



Fig. 14. 출력상태 A

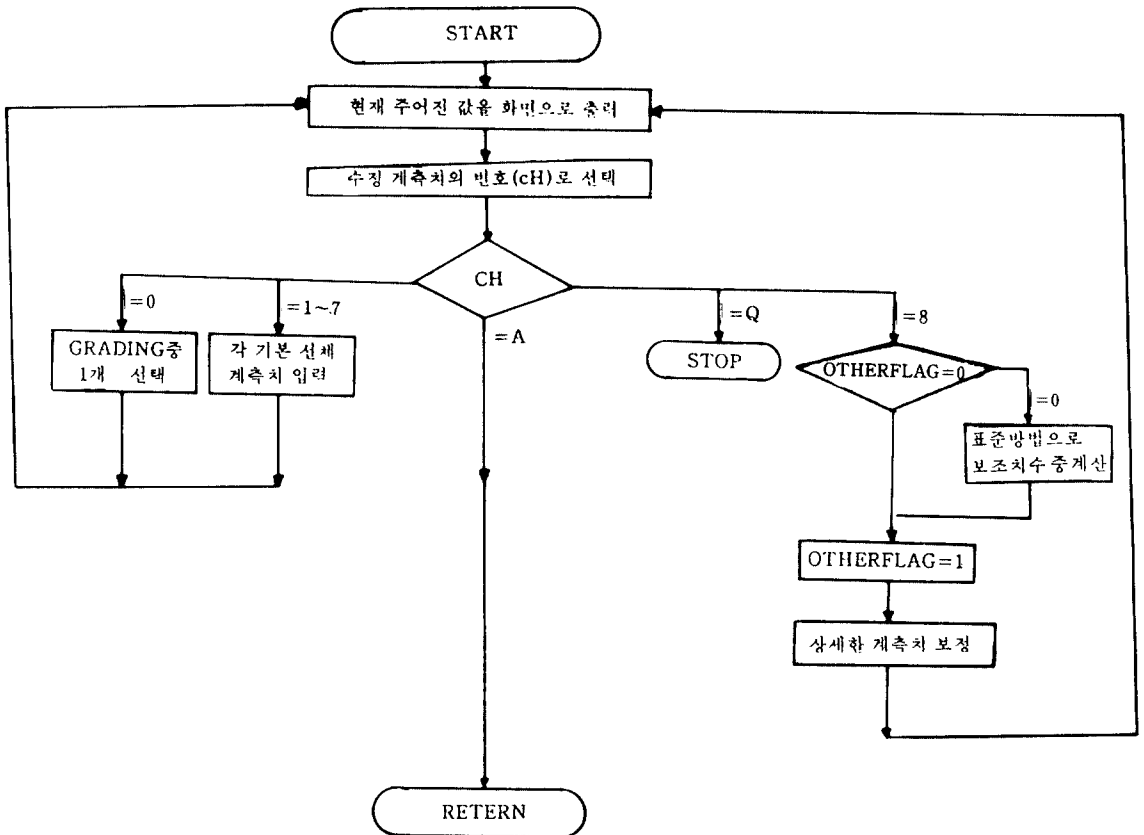


Fig. 13. 흐름도 B(부프로그램 STANDINPUT의 흐름도)

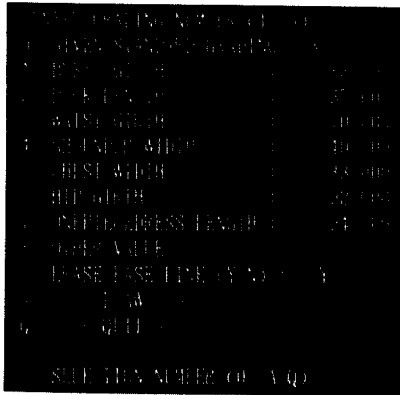


Fig. 15. 출력상태 B

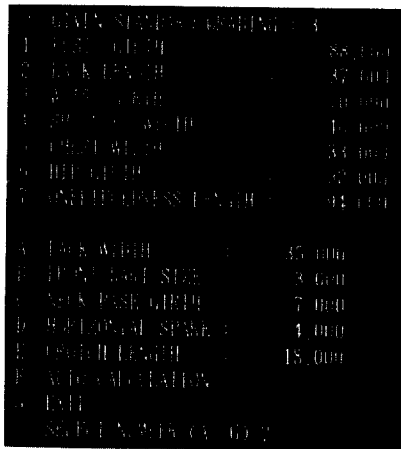


Fig. 16. 출력상태 C

Table 5. 성인여자의 사이즈별 신체치수 (단위 : cm)

계측항목 \ 사이즈	1	2	3	4	5
가슴둘레	82	85	88	91	94
등길이	35	36	37	38	40
어깨넓이	38	39	40	41	42
가슴넓이	31	32	33	34	35
엉덩이둘레	87	90	92	94	96
허리둘레	66	68	70	73	76
원피스길이	92	93	94	95	96

(자료 : 산업의 표준치 설정을 위한 국민 체위 조사연구, 한국과학기술 연구소, 1980)

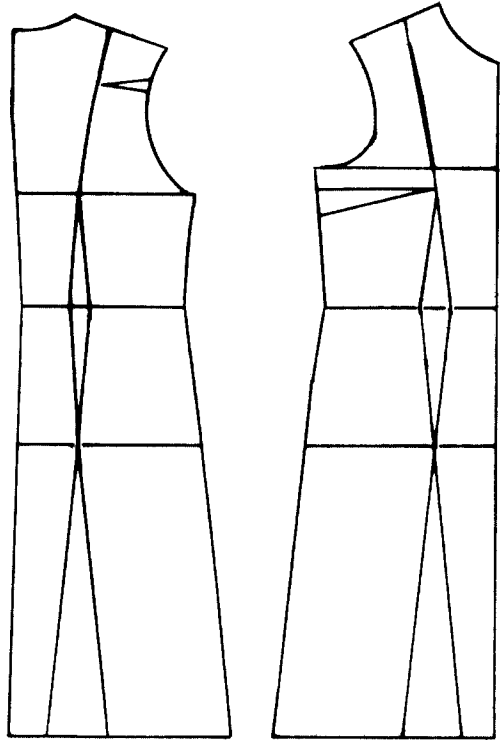


Fig. 17. 출력된 원피스드레스 원형 (1/10축도)

정한 후 製圖에 필요한 모든 기준점과 이들을 연결하는 직선 및 곡선을 수식으로 나타내었다.

프로그램을 개발하기 쉽도록 부프로그램을 따로 개발하여 사용하였으며 프로그램의 구성은(Fig. 12, Fig. 13)의 흐름도와 같다(Table 4)는 개발한 주프로그램의 List이다.

이 프로그램을 실행하면(Fig. 4)와 같이 컴퓨터 화면상에 출력된다. SELECTION NUMBER(O.A.Q)라는 메시지(message)에서 0~9, A, Q의 문자중 1개를 입력받게 된다. 여기서 0번을 입력시키면(Fig. 15)와 같이 ENTER GRADING NUBER(1,,5)라는 메시지가 나타나 1~5사이의 신체치수가 入力되기를 기다린다.

성인여자의 사이즈별 신체치수는 Table 5와 같다. 이 치수를 입력시키면 다시 Fig. 14로 돌아간다. 여기서 8번(OTHER VALUE)을 선택하면 보조치수를 입력할 수 있는 상태(Fig. 16)가 된다. 원하는 신체치수를 수정한 뒤 G를 선택하면 다시(Fig. 14)로

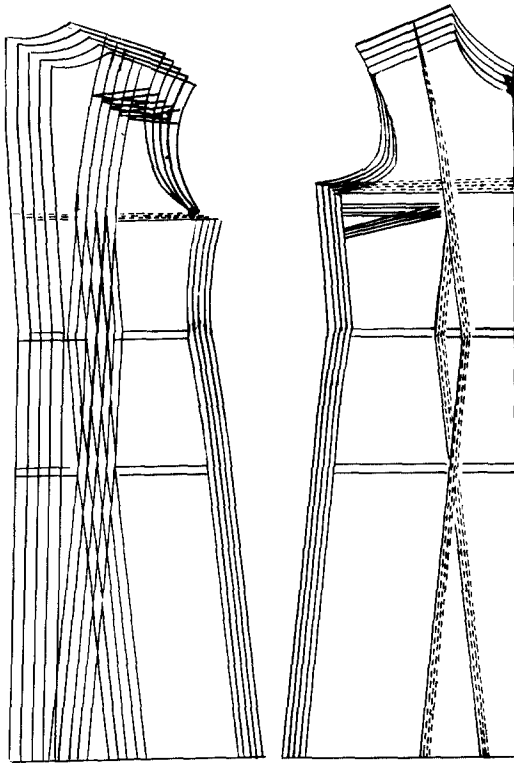


Fig. 18. 출력된 원피스드레스 원형의 그레이딩  
(1/10축도)

돌아오며 A를 선택하면 화면 또는 다른 출력기기로 출력하게 된다. 이와 같은 과정으로써 출력하면 완성된 원형이 나타난다(Fig. 17)은 출력비 1/10의 원피스드레스의 원형이다(Fig. 18)은(Table 5)의 사이즈별 신체치수를 입력하여 1/10 축도로 그레이딩한 것이다.

#### IV. 結 論

본 연구에서는 컴퓨터를 이용하여 프린세스라인 원피스드레스를 자동제도 할 수 있는 프로그램을 개발하였다.

프로그램 개발을 위해 다음과 같이 행하였다.

1. 주프로그램에는 필요한 부프로그램을 두개의

Unit로 묶어 여러 기하학적인 문제와 직선과 곡선부 형성 문제를 해결하였다.

2. 필요한 신체계측치(가슴둘레, 등길이, 어깨넓이, 가슴넓이, 허리둘레, 엉덩이둘레, 원피스길이)를 입력하면 개인의 체형에 맞는 원형이 빠르고 정확하게 제도된다.

3. 7항목의 신체계측치 외에 보조계측치를 따로 두어 개인에게 알맞은 치수를 입력할 수 있게 만들어져 세밀한 보정을 원하는 경우에 부분적인 변경을 쉽게 할 수 있어서 시간의 단축화, 효율화 그리고 보다 정확한 패턴을 얻을 수 있게 하였다.

4. 곡선제도에는 Arc(호)의 방법, 삼각함수를 적용하여 호를 그리는 방법, 베지에 곡선(Bézier curve)을 이용하였다.

5. 프로그램상에는 5사이즈로 그레이딩하였다.

6. 프로그램 전체를 볼 때는 연속적인 디자인 작업이 가능하고 처음 작성한 패턴을 기본으로 하여 조금 활용하면 많은 종류의 드레스를 용이하게 만들 수 있어 효율성을 높일 수 있다.

#### 參 考 文 獻

1. 鄭善謨, CAP/CAM概論, 대한교과서주식회사, 1987.
2. 孔錫鵬, 朴信雄, 縫製科學, 교문사, 1986.
3. 李順媛, 林元子, 衣服構成原理, 한국방송통신대학, 1983.
4. 維新文化社 편집부 譯, 서양피복구성학 I, 유신문화사, 1983.
5. 丁明淑, 컴퓨터에 의한 아동복원형의 제도연구, 서울대 대학원 석사학위논문, 1985.
6. 朴愛蘭, 퍼스널 컴퓨터에 의한 여성복원형의 자동제도에 관한 연구, 1986.
7. 盧熙淑, 컴퓨터에 의한 부인복원형의 제도연구, 1987.
8. 李順媛의 2人, 컴퓨터에 의한 의복원형제도의 기초연구, 한국의류학회지 Vol. 9, No. 1, 1985.
9. Donald Hearn & M. Panline Baker, colmputer Graphics, prentic-Hall, Inc. U.S.A.
10. Roy E. Myers, Microcomputer Graphics, Addison-Wesley Pub. Co., 1982.