PUBLICATIONS OF THE KOREAN ASTRONOMICAL SOCIETY
5:95~103, 1990

천문계산용 그래픽 라이브러리

김 갑 성·민 영 기
경희대학교 우주과학과
(1990. 10. 17 접수)

요 약

일반인에게 널리 보급된 IBM 계열의 16비트 PC(XT, AT)로도 손쉽게 도형출력이 가능한 천문 계산용 포트란 그래픽 라이브러리를 CALCOMP계의 문법에 맞추어 개발하였다. 본 라이브러리를 사용하면 그래픽 어댑터(HGC, CGA, EGA, VGA)에 임의로 대응되는 화면출력과 XY플롯터는 물론이거니와 EPSON계의 9핀 혹은 24핀 도트프린터에 의한 고품질(9핀 프린터; 0.21mm/dot, 24핀 프린터; 0.14mm/dot)의 도형출력이 가능하다. 프린터에 의해 출력 가능한 도형의 크기는 하드디스크의 용량이 허용하고 처리 속도를 인내할 수 있는 한 무한대로 확장할 수 있는데 비교적 빠른 시간 내에 처리할 수 있는 표준 도형 크기로서 최대 작도범위를 80 컬럼용 프린터 용지 한 페이지 크기인 25cm×20cm를 지정하였다.

I. 서 론

도형의 처리 및 출력을 위해 국내 천문학자들이 주로 이용하고 있는 그래픽소프트웨어로는 Sigma Plot, Harvard Graphics 등이 있는데, 여러가지 장점에도 불구하고 고급언어에 의한 프로그램으로 도형을 처리할 수 있도록 고안되지 않아 수치계산과 동시에 작도가 가능하지 않다. 이에 비해 BASIC언어의 화면처리 개념처럼 작도에 필요한 기본명령어 혹은 라이브러리 루틴을 사용자가 작성한 프로그램 내에서 호출하는 방법으로 수치를 계산한다든지 아울러 자유롭게 도형처리를 수행할 수 있는 소프트웨어로는 CALCOMP의 그래픽 루틴, TEKTRONIX의 PLOT10 그리고 DIN의 GKS 등이 널리 알려져 있다. 필자는 대형계산기에 장착된 XY플로터용 CALCOMP 925/1036과 그래픽프린터용 VERSATEC-1200 그리고 개인용 계산기 NEC9801의 화면출력용으로 개발된ProFortran의 그래픽 소프트웨어를 사용한 경험이 있다. 그러나 이들 팩케이지는 그래픽 단말기, 플롯터 및 프린터 등의 출력장치에 크게 의존하는데, 현재 상당수 보급되어 있는 IBM PC에서는 그래픽 하드웨어가 서로 달라 그것들을 그대로 사용할 수 없기 때문에 계산결과의 도형처리에 상당한 불편을 느껴왔던 바이다. 물론 도형처리 소프트웨어는 상기의 팩케이지 이외도 여러가지 있는 것으로 알고 있으나 대부분 포트란언어를 주로 사용하는 천문계산에 이용하기 불편하고 또 그 출력도형의 크기도 크게 제한되어 있다. 개인용 PC를 이용한 그래픽 처리를 위해 많은 시도가 있었는데 그간에 등장한 PC용 그래픽 소프트웨어는 다듬어지지 않은 것이 많고 주로 ASSEMBLY나 C언어 등으로 짜여져 있기 때문에 비록 소스프로그램이 공개

되어 있더라도 준저급 언어에 익숙치 않은 사용자들이 사용목적에 맞추어 프로그램을 개량하는 데에는 많은 어려움이 있어 왔다.

본 연구에서 필자는 고가의 특수한 그래픽 주변기기를 요하지 않는 일반의 개인용 PC 및 출력장치로도 천문학자 그 누구를 막론하고 손쉽게 포트란 프로그램에 의해 도형처리가 가능한 그래픽 라이브러리를 작성하여 소개한다. 라이브러리 내의 서브루틴은 플롯터 루틴으로 널리 알려진 CALCOMP계열의 문법을 이용하였다. 여기에서 개발된 그래픽 소프트웨어는 그래픽 카드에 임의로 대응하는 화면출력 뿐 아니라 24핀의 고밀도 프린터에 의한 대형의 도형출력도 가능하다. 그리고 각국의 전산소에서는 거의 CALCOMP계열의 그래픽 팩케이지를 지원하고 있으므로 본 라이브러리를 이용해 작성한 프로그램은 그대로 대형계산기에서 사용할 수 있는 장점을 지니고 있다.

Ⅱ. CALCOMP 그래픽 라이브러리

본 그래픽 라이브러리는 크게 기본루틴과 기능별 루틴으로 나누어진다. 기본루틴에는 작도에 필요한 1차적인 기초기능이 포함되어 있고 이 기본루틴을 조합 활용하여 사무용, 제도용 혹은 과학계산용으로 만든 2차 루틴이 기능루틴이다. 따라서 기본루틴은 각 출력장치의 하드웨어를 제어하기 위한 부분이 담겨 있는데 이 루틴은 FORTRAN 언어의 CALL 문으로 호출 이용할 수 있다.

1. 기본루틴의 종류

PLOTS ; 각종 파라미터를 지정하고 도형 출력용 데이터파일을 열거나 그래픽장치를 초기화시킨다.

PLOT ; 펜을 내리거나 올린 상태에서 펜을 이동하고 혹은 원점을 이동시킨다. 도형출력용 데이터파일을 닫아 도형처리를 끝마친다.

SYMBOL ; 영자, 숫자 혹은 특수기호 등의 문자를 그린다.

NUMBER ; 수치를 지정된 위치에 표시한다.

SCALE ; 주어진 데이터 수치에 배율을 부여한다.

AXIS ; 좌표축을 그린다.

LINE ; 연속한 데이터좌표점 사이를 직선으로 잇는다.

FACTOR ; 도형의 배율을 지정한다.

WHERE ; 현재의 펜위치와 척도계수를 구한다.

NEWPEN ; 펜의 색 혹은 굵기를 선택한다.

2. 기능루틴의 종류

1) 일반용

CIRCL ; 원을 그린다.

DASHP ; 점선을 그린다.

DASHL ; 주어진 데이터 사이를 점선으로 이어간다.

ELIPS ; 타원을 그린다.

GRID ; 모눈금을 그린다.

RECT ; 사각형을 그린다.

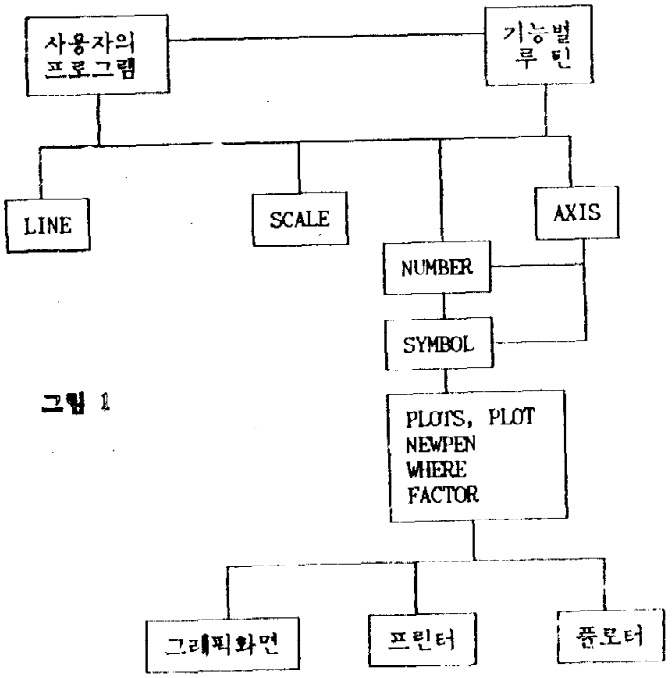

그림 1

2) 제도용

AROHD ; 여러 형태의 화살표를 그린다.

ARROW ; 주어진 데이터 사이를 연결하는 끝에 화살표를 그린다.

LABEL ; 지정된 위치에 주석을 단다.

기본루틴과 기능 루틴사이의 관계는 그림 1에 나타나 있다. 그림에서 보는 바와 같이 PLOT 루틴은 반드시 호출하게 되어 있다. 프로그램 실행과 작도의 효율을 향상시키기 위해서는 기본루틴, 그중에서도 PLOT루틴을 많이 활용하는 것이 추천된다.

CALCOMP 그래픽 루틴의 사용방법에 대한 상세한 내용은 '그래픽 라이브러리 사용자 설명서(1990, 김갑성)'를 참조하기 바란다.

Ⅲ. 화면 출력

이상의 그래픽 루틴으로 작성한 도형을 디스플레이 화면으로 출력하는 방법에 대해 설명하고자 한다. 대형계산기에 부착된 그래픽 단말화면의 프레비유(Preview)용으로는 TEKTRONIX의 PLOT 10이 사용되고 있었는데 그것과 동일한 이용방법으로 Fortran 프로그램에 의해 개인용 PC의 모니터 화면에 도형처리를 처음 시도한 시기는 MS Fortran Ver. 3.31이 한창 성행하였던 때 이었다. 그 뒤 ANSI Fortran 77의 문법이 전면 지원되었던 4.0판이 등장하여 대형 계산기에서나 구사할 수 있었던 포트란 언어의 모든 기능이 개인용 계산기로도 거의 활용되게 되었다. 또한 비슷한 시기에 나온 MS-C Ver. 5.0은 Turbo계열의 언어에서 구사되었던 다양한 그래픽처리 기능을 내장하고 있어 C언어 자체의 도형처리가 가능하였다. 그래픽처리를 위해 특별히 Micro-Software사의 고급언어를 선택한 이유는 천문학자들이 주로 사용하는 Fortran을 위시하여 Assembly, C 및 Pascal 등의 모든 언어에 대한 컴파일러를 제공하고 있어 상호간의 언어를 혼용하여 호환성 있는 프로그램을 작성할

```
mode(ext)
struct {long xmax, ymax}*ext;
{ struct videoconfig vc ;
  int tmode, vmode ;
  _getvideoconfig(&vc) ;
        /* Select best text and graphics modes and adjust menus */

        switch (vc.adapter) {
                case _MDPA :
                        puts("No graphics mode available.\n");
                        puts("Execute the MSHERC or SIMCGA !\n");
                        while (!kbhit());
                        vmode = _HRESBW;
                        break;
                case _CGA :
                        vmode = _HRESBW;
                        break;
                case _EGA :
                        if (vc.memory > 64)
                                vmode = _ERESCOLOR;
                        else
                                vmode = _HRES16COLOR;
                        break;
                case _HGC :
                        vmode = _HERCMONO;
                        break;
                case _VGA :
                        if (vc.memory > 128)
                                vmode = _VRES16COLOR;
                        else
                                vmode = _VRES2COLOR;
                       break;
                case _MCGA :
                        vmode = _MRES256COLOR;
                        break;
        }
        switch (vc.mode) {
                case _TEXTBW80 :
                case _TEXTBW40 :
                        _setvideomode(_TEXTBW80);
                        break;
                case _TEXTC40 :
                        tmode = _TEXTC80;
                        break;
                case _TEXTMONO :
                case _ERESNOCOLOR :
                        tmode = _TEXTMONO;
                        break;
                default :
                        tmode = _TEXTC80;
```

```
        }
 _clearscreen(_GCLEARSCREEN);
 _setvideomode(vmode);
ext->xmax = vc.numxpixels;
ext->ymax = vc.numypixels;
}
```

FORTRAN Program

```
INTERFACE TO SUBROUTINE mode[C](A)
INTEGER*4 A
END

COMMON /  EXT   / XMAX,YMAX
INTEGER*4 XMAX, YMAX
CALL MODE(LOC(vmode))
CALL WINDOW(0.,0.,YMAX*1.,XMAX*1.)
 :
 :
```

Characters available with the SYMBOL routine

0		16	경희대 우주과	32		48	0	64	@	80	P	96	`	112	p
1		17		33	!	49	1	65	A	81	Q	97	a	113	q
2		18		34	"	50	2	66	B	82	R	98	b	114	r
3		19		35	#	51	3	67	C	83	S	99	c	115	s
4		20		36	$	52	4	68	D	84	T	100	d	116	t
5		21		37	%	53	5	69	E	85	U	101	e	117	u
6		22		38	&	54	6	70	F	86	V	102	f	118	v
7		23		39	'	55	7	71	G	87	W	103	g	119	w
8		24		40	(	56	8	72	H	88	X	104	h	120	x
9		25		41	)	57	9	73	I	89	Y	105	i	121	y
10		26		42	*	58	:	74	J	90	Z	106	j	122	z
11		27		43	+	59	;	75	K	91	[	107	k	123	
12		28		44	,	60	<	76	L	92	\	108	l	124	
13		29		45	-	61	=	77	M	93	]	109	m	125	
14		30		46	.	62	>	78	N	94		110	n	126	
15		31		47	/	63	?	79	O	95		111	o	127	

표 1.; 사용문자 및 심벌. 본라이브러리에서 사용하는 문자와 심벌이 표시되어 있는데 0~13번까지는 센터심벌이 등록되어 있다.

수 있기 때문이다. Assembler를 이용한 그래픽 루틴도 시중에 통용되고 있었으나 C언어에 포함되어 있는 다양한 그래픽 명령어를 활용하고자 Fortran언어와 C언어를 혼용하여 천문계산에 필요한 화면처리용 라이브러리를 작성하였다. 그런데 최근에 등장한 MS Fortran

5.0판은 별도로 MS-C와 똑 같은 그래픽 라이브러리(GRAPHICS. LIB)를 탑재하고 있어 포트란언어 단독으로도 화면상의 도형처리가 가능하게 되었다.

임의의 그래픽 어댑터에 대응하는 화면출력은 Micro Software사의 GRAPHICS. LIB에 내장되어 있는 _getvideoconfig와 _setvideoconfig의 명령어를 사용하였다.

다음은 포트란 프로그램에서 C언어로 작성한 부 프로그램을 불러 도형처리하는 예를 소개한다.

칼러화면 출력에 있어서 NEWPEN문은 펜의 색상을 지정하는데 선의 굵기는 도트 하나의 크기로 결정되어 조절 가능하지 않다. 그런데 본 라이브러리는 칼러화면상의 색상은 물론 선의 굵기도 동시에 조절할 수 있는 비표준형 루틴 NPEN을 첨가하였다. 그리고 그래픽 문자와 숫자를 이용할 수 있도록 ASCII코드의 0에서 125번까지의 모든 문자 숫자에 대한 FONT를 표 1과 같이 제작하였다. 그다지 모양이 좋지 않다고 생각되거나 새롭게 등록하고자 하는 심볼에 대해서는 다음과 같은 방법으로 FONT심볼을 제작하고 라이브러리에서 BLOCK DATA루틴을 교체하면 된다. 표 1을 출력하기 위해 작성한 소스프로그램은 다음과 같다.

```
C  +-----------------------------------------+
C  |  Alphanumeric Characters and Symbols    |
C  +-----------------------------------------+
      CALL PLOTS
      CALL WINDOW(0.,0.,640.,400.)
      DX = 75
      DY = 20
      XO = 20
      YO = 310
      CALL NEWPEN(10)
      CALL SYMBOL(45.,350.,15.,'Characters available with the SYMBOL rou
     *tine',0.,44)
      CALL GRID(10.,5.,75.,325.,8,1)
      CALL NEWPEN(11)
      DO 10 I=0,127
      IF(I.NE.0.AND.MOD(I,16).EQ.0) N=N+1
      X = XO + DX*N
      Y = YO - DY*MOD(I,16)
      CALL NUMBER(X,Y,10.,I*1.,0.,-1)
      CALL SYMBOL(X+40.,Y,15.,CHAR(I),0.,1)
10    CONTINUE
      CALL PLOT(0.,0.,999)
      END
```

Ⅳ. 프린터 출력

프린터는 플로터처럼 펜을 자유롭게 이동시킬 수 없기 때문에 프로그램에 의해 처리가 완료된 도형전체를 기억해 두었다가 일시에 출력하는 방법을 사용하여야 한다. 이를 위해서 전 도형데이터를 기억시킬 장소가 요구되는데, 현재 OS로서 MS-DOS를 이용한 계산기는 실제 계산기의 CPU공간이 많다 하더라도 총 기억용량은 640K를 초과할 수 없다. 따라서 1 바이트 정수혹은 문자를 선언한다 하더라도 배열의 크기는 산술적으로 800×800을 초

과할 수 없다. 그러나 실제로는 OS로 할당된 영역과 프로그램이 차지하는 공간 모두를 고려해야 하기 때문에 기껏해야 720×720 정도의 배열 크기가 최대인 것 같다. 그런데 720×720의 화소로 구성되는 화판의 그림은 180 dots/inch(=0.14mm/dot) 분해능의 24핀 프린터로 약 10cm×10cm 크기의 도형이 출력되는데 자료로 활용하기엔 너무 크기가 작은 형편이다. 20cm×20cm 크기의 도형을 얻기 위해서는 1430×1430의 배열이 요구되는데 이는 2메가 바이트에 상당하기 때문에 이 정도의 데이터를 CPU의 기억내에서 한꺼번에 처리할 수 없다. 물론 램이나 하드디스크상에 기록해 두었다가 조금씩 읽어서 처리하는 방법이 고려될 수 있으나 이는 처리속도에 있어서 엄청난 손실을 가져오기 때문에 크게 추천되지 않는다.

이 문제를 해결하기 위해 다음과 같은 비트이미지 방법을 사용하여 도형 데이터를 압축하였다. 즉 단색으로 도형을 그릴 경우에는 한 픽셀이 여러 색상(4비트, 16색)이 아닌 흑백의 1비트 정보만이 요구되는데 화판의 내용을 전부 1바이트의 문자로 바꾸어 버리면 한 문자가 8비트로 구성되어 있기 때문에 픽셀 8점을 한 문자로 표시할 수 있게 되어 앞에서 요구된 배열 크기가 1/8로 줄어들어 상당한 양의 메모리 절약이 실현된다.

25cm×20cm의 도형출력을 위해 필요한 화면의 픽셀 수는 1772×1417 정도인데 대략 315K 바이트 정도의 용량을 필요로 한다. 이 정도 규모의 도형은 CPU에서 한번에 처리가능한 양으로 작도시간이 짧다. 그런데 이 보다 큰 도형 이를테면 VERSATEC처럼 작도범위가

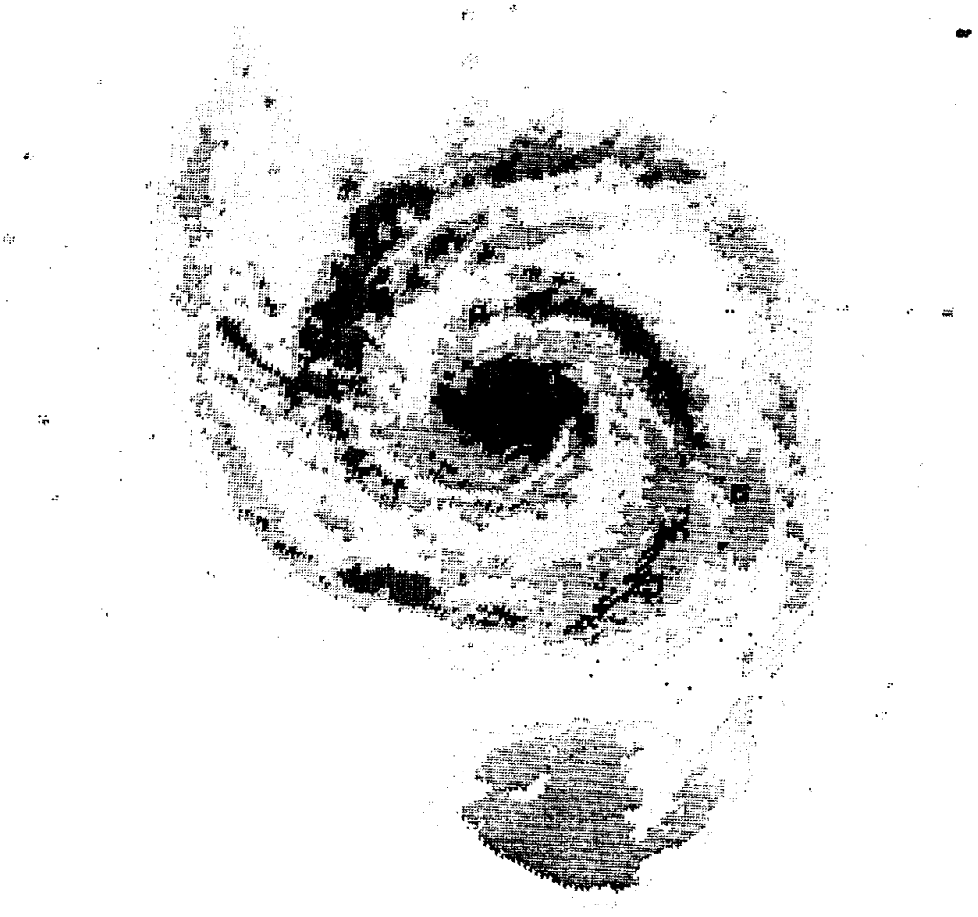

그림 1. M51 은하

그림 2. Hα 로 본 태양

26cm×2m 정도되는 큰 도형을 그리기 위해서는 램 혹은 하드디스크에 기록 저장해 두었다가 일정량을 읽어 처리해야 하는데 계산기와 주변기기 사이에 데이터를 주고 받는 액세스 타임이 길어 도형처리에 많은 시간이 요구는 되지만 그래도 하드디스크의 용량이 허용하고 처리속도를 사용자가 인내할 수만 있다면 작도범위는 얼마든지 확장 가능하다.

선을 긋는데 있어서 화면과 플로터는 커다란 차이가 있다. 전자의 경우는 두점을 잇는 직선상에 놓여있는 도트를 하나하나 순서대로 임의의 색상을 넣어 가는 방법을 이용하나 후자의 경우에는 펜을 내린 상태에서 임의의 점으로 스텝모터에 의해 구동되는 펜을 이동시켜 선을 긋는다. 따라서 사선을 그릴 경우 도트를 차례로 세트해 가는 방법으로는 플로터처럼 선이 매끄럽지 못하다. 그런데 플로터를 이용하여 화면처리와 같은 방법으로 점을 찍어가는 방법으로 선을 그려가면 기계적으로 상당한 시간이 요구된다. 이를테면 칼리 그래픽 화면 복사를 플로터를 이용하여 수행시킬 경우 펜의 색을 갈아끼워 가며 수많은 점을 찍어 표현하는 데에는 큰 무리가 있다. 그러나 도트프린터에 의한 화면출력은 상당히 용이하여 색프린터기가 준비되어 있다면 물론 그대로 복사 가능하나 단색 프린터기라도 그레이 레벨을 임의로 넣어 명암 혹은 색의 차이를 표시할 수 있다. 화면의 픽셀 간격은 도트피치로 표시되는데 그 해상도의 경우 0.28~0.33mm 정도인데 반하여 프린터의 고밀도 모드는

0.14mm/dot이다. 화면의 픽셀 하나에 대응하는 프린터 화판의 픽셀을 2×2로 구성하면 5단계의 그레이 레벨을 구성할 수 있고 3×3의 배열로 구성하면 총 10개조의 그레이 레벨을 만들 수 있다. 표시할 수 있는 명암의 단계수를 증가시킬수록 기본 픽셀의 크기가 커져 도형의 크기는 증가하는 반면에 해상도가 떨어지는 결점이 있으나 단색으로도 밝기의 차이를 나타낼 수 있다는 데에 큰 잇점이 있다. 그림 1과 2에 M51 은하와 Hα 태양상이 각각 나타나 있는데 특별한 화상처리 없이 3×3 배열로 구성된 픽셀에 명암을 넣어 단순하게 출력한 것이다.

V. 토의 및 결론

그간 천문학자들 사이에 많이 활용되어 왔던 PC용 MS FORTRAN (Ver. 4.0 이하)으로는 생각할 수 없었던 도형처리가 이 그래픽 라이브러리로 인해 BASIC언어에서처럼 가능하게 되었고 각국의 대형계산기 센터에서 표준적으로 지원하고 있는 CALCOMP의 루틴을 라이브러리화시켰기 때문에 계산기 기종에 구애받지 않는 프로그램을 작성할 수 있다.

앞으로 보다 많은 기능별 루틴이 작성되길 희망하며 기본루틴과 함께 첨가된 이 기능루틴을 활용하면 콤퓨터에 의해 설계도 가능한 CAD시스템을 만들 수 있다.

하나의 작도용 소스프로그램을 가지고 단지 출력장치에 대응하는 라이브러리를 결합시켜주는 것으로 모니터 화면, 그래픽프린터 혹은 플로터에 도형출력이 가능하다는 장점을 지니고 있다. 아직 레이저 프린터를 위한 라이브러리는 준비하지 못하였으나 곧 빠른 시일내에 완성될 것으로 기대된다.

문자 및 숫자에 대한 폰트가 한종류밖에 되지 않아 미흡하게 생각하고 있다. 보다 다양한 폰트가 제작되고 더우기 화면출력용 비트 맵의 폰트가 아닌 스케일 벡터로 작성되어 확대, 축소가 가능한 한글 폰트가 등장하여 그래픽 한글이 사용되길 바라는 바이다.

참 고 문 헌

김갑성, 1990, 그래픽라이브러리 사용자설명서, 경희대 우주과학과.