

Cuboid가 형성하는 공간의 표상

이 재 순 (경산초등학교)

이 청 주 (충북대학교)

김 가 연 (충북대학교)

한 재 영 (충북대학교)

수식에 의한 컴퓨터 그래픽의 입력과 출력에 관한 프로그램의 개발은 수학의 역동화, 인간화, 보편화에 기여하고 있다. 현실적으로 해결해야 할 문제와 수식에 의한 해답이 전부인 현재의 수학을 소프트웨어를 활용하여 그래픽 기능을 첨부하면 움직이는 수학을 가시화 할 수 있다. 컴퓨터 프로그램에 의한 수학의 실현은 수학자들 모두의 염원으로 전세계적으로 활발한 연구가 진행되고 이는 것이다. 수학의 원리와 응용성을 가미한 수학적 그래픽의 발전은 새로운 천년을 장식할 새로운 학문분야로 등장하고 있다. 기초과학의 여러 자료를 분석, 검토하여 그래픽으로 조립하는 작업은 수학적 그래픽의 힘으로 가능하며, 실험과 실습의 양상을 바꾸어 놓고 있다. 건축이나 토목 또는 전기전자 학과의 응용수학은 새로운 소프트웨어의 출현으로 컴퓨터 강의로 전환되고 있으며 수학 그 자체로 전산기능을 강화하는 방향으로 개편되고 있다. 수학 교과 내용의 전산 프로그램화와 컴퓨터 활용 수학 학습은 거부할 수 없는 시대적 요구이다. 이 연구에서는 새로운 천년의 시작은 컴퓨터 프로그램에 의한 완성된 그래픽의 연출이라는 시각에서 수식에 의한 컴퓨터 그래픽의 기본 방향을 제시하고 있다. 2차원 평면이나 3차원 공간에 이와 같은 다변수함수의 역할을 구현함으로써 다양한 그래픽을 영상화할 수 있다. 다중화면의 연출, 다단계화면의 조합, 다단계다중화면의 영상화 등은 수학에 의한 애니메이션의 기초가 된다. 평면도형의 기본동작을 화면에 구체화시키는 Table 기능을 실제로 구현한다. 연습과 실행 그리고 재구성을 반복하여 조형미를 갖춘 수학적 그래픽을 실현한다. 수학의 학습에 적용할 가치가 있는 학습조형물을 개발하고, 프로그램의 단순화에 노력한다. 미분기하학의 여러 공식을 이용하여 숨어 있는 그림을 표출하며 미분방정식의 해가 갖는 그래픽의 묘미를 형상화한다. 수식에 의하여 출력된 그래픽의 여러 효과를 응용수학에 활용할 수 있도록 재조립하는 과정을 걸쳐 완성하는데 이 연구의 참된 의미가 있다.

I. 서론

이 논문은 한 변의 길이가 1인 정육면체가 형성하는 공간 그래픽의 다양한 구조를 영상으로 나타낸 것이다. 직사각형이나 정사각형이 평면도형의 면적을 구하는 미적분의 요소인 것과 같이 직육면체나 정육면체는 공간도형의 부피를 얻는데 유용하게 쓰인다. 점과 원, 정사각형 등이 평면 형성의 같은 성질의 모체인 것처럼 공간 위의 점이나 구, 정육면체 등은 공간전체의 기본 요소이다. 우리 주변의 형상들은 대부분이 직육면체나 정육면체들의 교묘한 합성으로 형성되어 있다고 볼 수 있다.

여기에 언급된 컴퓨터 그래픽은 소프트웨어 Mathematica의 Kernel의 기능을 활용한 것으로 수학의 기본 원리에 충실한 것들로 구성되어 있다. 기호대수의 형태인 수식으로 되어 있는 수학적 표현들을 그림의 형태로 연출하여 수학의 역동성을 제고하려는 의도에서 엮어진 프로그램 하나 하나는 컴퓨터 언어의 기본 구조에 따라 편성된 것이다. 난해한 현대수학을 컴퓨터 그래픽으로 재구성하여 화상으로 직접 볼 수 있게 만드는 작업은 수학의 인간화에 도움을 주리라 본다.

II. 본문

1. 정육면체가 형성하는 공간 그래픽

1.1. Graphics 구성의 기본 프로그램

1.1.1. 한 변의 길이가 1 인 정사각형의 출력은 Cuboid[a,b,c]이다(Stephen, 1988). 이 논문에서 사용될 정육면체의 기본구도는 다음과 같은 프로그램이다.

```
In[1] = Graphics3D[ {{EdgeForm[RGBColor[1,1,0], Thickness[0.01]],
    FaceForm[RGBColor[0.5,0.5,0.5]], Cuboid[{a,b,c]}},
    {EdgeForm[RGBColor[1,1,1], Thickness[0.01]],
    FaceForm[RGBColor[0.5,0.5,1]], Cuboid[{c,d,f},{p,q,r}]}}
```

1.1.2. 기존의 그림에서 어떤 조건을 변경하여 새로운 그림을 출력하는 프로그램은 다음과 같다.

```
In[2] = Show[ Graphics/(Axes->Automatic)->(Axes->None), PlotRange->{-1,2},
    GraphicsSpacing->{0.5,0.3}]
```

1.1.3. 여러 개의 그림을 한 화면에 모아서 출력하면 서로를 비교하기 편리하며 그림의 표출형태를 관찰하는데 유익하다. 다음과 같은 프로그램은 여러 그림을 적당한 간격으로 배열하는 기본 구도이다.

```
In[3] = Show[ GraphicsArray[{ }], Boxed->False, GraphicsSpacing->{0.5,0.5}]
```

1.2. Cuboid의 형태를 형성하는 기초요소의 프로그램

1.2.1. 다음은 구간의 시전과 종점, 구간의 등분, 색의 조정에 이용되는 다변수함수의 정의를 구현하는 기본 프로그램이다. 하나의 Cuboid에 의하여 그려지는 연계된 그림을 출력하는 그래픽 함수의 정의이다.

```
In[4] = gp[{{a,b,c},{d,e,f},{g,h,i}}]:=
    Show[ Graphics3D[Table[{{Hue[Sin[t]], Table[{Cuboid[{s,t,u}},{u,0,a,d]}},
    {s,0,b,e},{t,0,c,f}]],Lighting->False, ViewPoint->{g,h,i},
    AspectRatio->Automatic]
```

1.2.2. 두 개의 Cuboid에 의하여 그려지는 연계된 그림을 출력하는 그래픽 함수의 정의이다(한재영, 1988).

```
In[5] = gpp[{{a_,b_,c_},{d_,e_,f_},{g_,h_,i_}}]:=
  Show[ Graphics3D[Table[{{Hue[Sin[t]], Table[{Cuboid[{s,t,u},
  Cuboid[{s,u,t}]}],{u,0,a,d}]}],{s,0,b,e},{t,0,c,f}], Lighting->False,
  ViewPoint->{g,h,i}, AspectRatio->Automatic]
```

1.2.3. 세 개의 Cuboid에 의하여 그려지는 연계된 그림을 출력하는 그래픽 함수의 정의이다(한재영, 1999).

```
In[6] = gppp[{{a_,b_,c_},{d_,e_,f_},{g_,h_,i_}}]:=
  Show[ Graphics3D[Table[{{Hue[Sin[t]],
  Table[{Cuboid[{s,t,u}], Cuboid[{u,s,t}], Cuboid[{s,u,t}]}],{u,0,a,d}]}],
  {s,0,b,e},{t,0,c,f}], Lighting->False, ViewPoint->{g,h,i}, AspectRatio->Automatic]
```

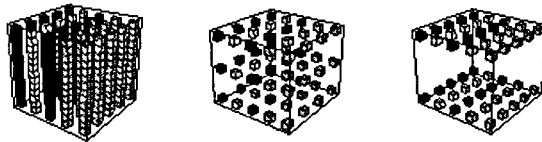
1.2.4. 프로그래밍의 편리를 위하여 등분의 크기만을 함수로 한 프로그램이다.

```
In[7] = gpdef = Show[ gp[{{12,12,12}, {d,e,f},{2.794,2.090,1.551}}]]
In[8] = gppdef = Show[ gpp[{{12,12,12}, {d,e,f},{2.794,2.090,1.551}}]];
In[9] = gpppdef = Show[ gppp[{{12,12,12}, {d,e,f},{2.794,2.090,1.551}}]]];
```

1.3. 구간의 크기의 조정으로 구현되는 그림의 실제

1.3.1. d, e, f를 모두 변경하여 출력되는 그림을 비교한 것이다.

```
In[10] = arraygpdef =
  how[ GraphicsArray[{{gpdef/.{d->1,e->3,f->3}, gpdef/.{d->5,e->4,f->3},
  gpdef/.{d->10,e->3,f->3}}], Boxed->False, GraphicsSpacing->{0.5,0.5}]
```



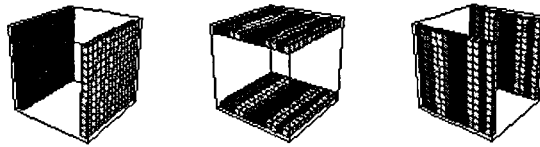
1.3.2. 높이 d만을 조정하여 얻은 그래픽의 출력이다.

```
In[11] = arraygpdef = Show[ GraphicsArray[{{
  gpdef/.{d->6,e->1,f->1}, (Lighting->False)->(Lighting->True)},
  gpdef/.{d->6,e->1,f->1}, Boxed->False},
  gpdef/.{d->6,e->1,f->1}, Hue[Sin[t]]->Hue[1/Sin[t]]}], Boxed->False,
  GraphicsSpacing->{0.5,0.3}]
```



1.3.3. { d, e, f}=(1, 1, 12), {1, 12, 1}, {12, 1, 1}로 표출되는 마주보는 두 면의 그림이다.

```
In[12] = arraygpdef =
Show[ GraphicsArray[{gpdef/.{d->1,e->1,f->12}, gpdef/.{d->1,e->12,f->1},
gpdef/.{d->12,e->1,f->1}], Boxed->False, GraphicsSpacing->{0.5,0.3}]
```



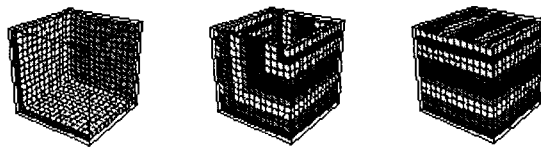
1.3.4. 두 쌍의 면이 조합하여 구현하는 그림들을 출력하는 프로그램이다.

```
In[13] = arraygppdef =
Show[ GraphicsArray[{gppdef/.{d->6,e->3,f->1}, gpdef/.{d->12,e->1,f->1},
gpdef/.{d->13,e->1,f->1}], Boxed->False, GraphicsSpacing->{0.5,0.3}]
```



1.3.5. 세 쌍의 면을 연계하여 얻을 수 있는 아름다운 그림들이다.

```
In[14] = arraygpppdef = Show[ GraphicsArray[{
gpppdef/.{d->13,e->1,f->1, (Lighting->False)->(Lighting->True)},
gpppdef/.{d->1,e->12,f->1}, gpppdef/.{d->12,e->1,f->1}], Boxed->True,
GraphicsSpacing->{0.5,0.3}]
```



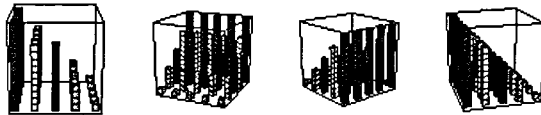
1.4. 관찰하는 지점의 조정으로 표출되는 그래픽의 효능

1.4.1. ViewPoint의 조정으로 나타나는 그림의 역동성을 보여주는 프로그램이다(Stephen, 1988).

```
In[15] = arraygpvdef =
  Show[ GraphicsArray[{{gp[{{(12-s,12,12)},{1,3,3},{2.794,2.090,1.551}}]},
    gp[{{(12-t,12,12)},{1,3,3},{2.794,2.090,1.551}}]},
    gp[{{(24-t-s,12,12)},{1,3,3},{2.794,2.090,1.551}}]}], Boxed->False,
  GraphicsSpacing->{0.5,0.3}]
```

1.4.2. ViewPoint을 함수와 연계하여 연출되는 애니메이션 효과를 구현한 프로그램이다.

```
In[16] = Show[ GraphicsArray[
  Table[gp[{{(12-t,12,12)},{1,3,3},{5Cos[n],5Sin[n], 1.5}}]/(
  AspectRatio->Automatic)->(AspectRatio->1/2)],{n,0,2Pi,2}],
  Boxed->False, GraphicsSpacing->{0.5,0.3}]
```



2. 직육면체가 형성하는 공간 그래픽의 표상

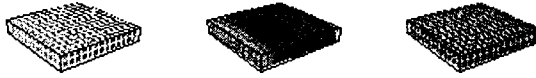
2.1. Graphics 구성의 기본 프로그램

2.1.1. 직육면체의 형성요소를 조정하는 프로그램이다. 직육면체의 모서리를 이루는 선의 형태와 면의 표상을 적절히 조화시킨다. 다음과 같은 15 개의 변수가 이루는 그래픽의 조형미를 연출하면 Cuboid의 역할을 화상형태로 애니메이션 할 수 있다(한재영, 1999).

```
In[17] = efgp[{{a_,b_,c_},{d_,e_,f_},{g_,h_,i_},{p_,q_,r_},{x_,y_,z_}}]:=
  Show[ Graphics3D[
  Table[{{Table[{EdgeForm[RGBColor[p,q,r], Thickness[0.01]],
  FaceForm[RGBColor[x,y,z]], Cuboid[{s,t,u]},{u,0,a,d}]}],{s,0,b,e},{t,0,c,f}}],
  Lighting->False, ViewPoint->{g,h,i}, AspectRatio->Automatic]
```

2.1.2. Cuboid의 모서리와 면의 형태를 다양하게 연출하는 프로그램의 조합이다.

```
In[18] = Show[ GraphicsArray[{
  efgp[{{(1,12,12)},{1,1,1},{-2.794,2.090,1.551}},{1,1,1},{0.05t,0.05t,0.05t}}],
  efgp[{{(1,12,12)},{1,1,1},{-2.794,2.090,1.551}},{1,0,1},{0.05t,0.05t,0.05t}}],
  efgp[{{(1,12,12)},{1,1,1},{-2.794,2.090,1.551}},{0,1,0.05t},{0.05,0.05,0.05}}}],
  Boxed->False, GraphicsSpacing->{0.5,0.3}]
```



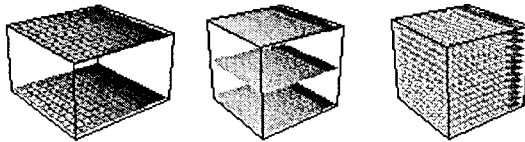
2.2. Cuboid의 다양한 효능

2.2.2. Cuboid의 대각선 위의 점들 사이의 거리를 적절히 조정하면 공간에서 평면을 구현하는 효과를 얻는다.

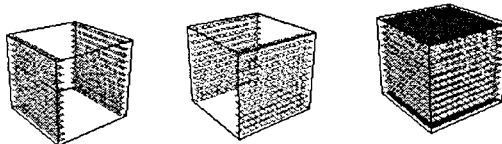
```
n[19] = mefgp[{{a,b,c},{d,e,f},{g,h,i},{p,q,r},{x,y,z}}]:=
Show[ Graphics3D[
Table[{{Table[{EdgeForm[RGBColor[p,q,r], Thickness[0.01]],
FaceForm[RGBColor[x,y,z], Cuboid[{s-1,t-1,u},{s,t,u}]}],{u,0,a,d}}},
{s,0,b,e},{t,0,c,f}], Lighting->False, ViewPoint->{g,h,i},
AspectRatio->Automatic]
```

2.2.3. 직육면체의 가로, 세로, 높이를 조절하여 얻는 그래픽 효과의 실제 상황을 표출한 것이다.

```
In[20] = Show[ GraphicsArray[
mefgp[{{(12,12,12),(8,1,1},{-2.794,2.090,1.551},{1,0,1},{0.5t,0.05t,0.5t}}},
mefgp[{{(12,12,12),(6,1,1},{-2.794,2.090,1.551},{1,0.05t,1},{0.05t,0.5t,0.05t}}},
mefgp[{{(12,12,12),(1,1,1},{-2.794,2.090,1.551},{0,1,0.05t},{0.5t,0.5t,0.05t}})]]]
```



```
In[21] = Show[{{mefgp[{{(12,12,12),(12,1,1},{-2.794,2.090,1.551},{1,0,1},{1,0,1}})}},
{mefgp[{{(12,12,12),(1,1,12},{-2.794,2.090,1.551},{1,0,1},{0,1,21}})}},
{mefgp[{{(12,12,12),(1,12,1},{-2.794,2.090,1.551},{1,0,1},{1,1,0}})}]}]
```



2.3. Cuboid에 의한 현실 상황의 연출

2.3.1. 직육면체들로 바둑판이나 바둑알과 같은 형태를 출현하는 재미있는 프로그램이다.

```
In[22] = badukpan = mefgp[{{1,17,17},{1,1,1},{0.827,-0.894,3.157},{0,0,0},{1,1,0}}]
```

```
In[23] = game = Show[ Graphics3D[
  Table[{{Table[{RGBColor[0,0,0], PointSize[0.1], Point[{s,18Abs[Sin[s]],2}]}],
  Table[{RGBColor[1,0,0], PointSize[0.1],
  Point[{s,18Abs[Cos[s]],2}]}],{m,0,1,1}}],{s,0,18,2},{t,0,18,3}]],
  Lighting->False, Boxed->False, ViewPoint->{0.827,-0.894,3.157},
  AspectRatio->Automatic];
```

```
In[24] = Show[ {badukpan,game}, Lighting->False, ViewPoint->{0.031,0.000,2.310},
  Boxed->False, AspectRatio->Automatic]
```

```
In[25] = Show[ GraphicsArray[{{%%,%%,%%}}, Boxed->False,
  GraphicsSpacing->{0.5,0.3}]
```



3. 정육면체의 각 면이 연출하는 실용적인 형상

3.1.1. 정육면체의 각 면을 구성하는 프로그램이다.

```
In[26] = msso = {EdgeForm[RGBColor[1,1,1], Thickness[0.0001]],
```

```
  FaceForm[RGBColor[1,1,0]], Cuboid[{0,0,0},{s,s,0}];
```

```
msos = {EdgeForm[RGBColor[1,1,1], Thickness[0.0001]], FaceForm[RGBColor[1,0,1]],
  Cuboid[{0,0,0},{s,0,s}];
```

```
moss = {EdgeForm[RGBColor[1,1,1], Thickness[0.0001]], FaceForm[RGBColor[0,1,1]],
  Cuboid[{0,0,0},{0,s,s}];
```

```
mssi = {EdgeForm[RGBColor[1,1,1], Thickness[0.0001]],
  FaceForm[RGBColor[0.5,0.5,0.5]], Cuboid[{s-1,s-1,1-0.001},{s,s,1}];
```

```
msis = {EdgeForm[RGBColor[1,1,1], Thickness[0.0001]], FaceForm[RGBColor[0,0,1]],
  Cuboid[{s-1,1-0.001,s-1},{s,1,s}];
```

```
miss = {EdgeForm[RGBColor[1,1,1], Thickness[0.0001]], FaceForm[RGBColor[1,0,0]],
```

```

Cuboid[{1-0.001,s-1,s-1},{1,s,s}];
In[27] = nsti = {EdgeForm[RGBColor[1,1,1], Thickness[0.0001]],
  FaceForm[RGBColor[0.5,0.5,0.5]], Cuboid[{s-1,t-1,1-0.001},{s,t,1}];
nsiu = {EdgeForm[RGBColor[1,1,1], Thickness[0.0001]], FaceForm[RGBColor[0,0,1]],
  Cuboid[{s-1,1-0.001,u-1},{s,1,u}];
nstu = {EdgeForm[RGBColor[1,1,1], Thickness[0.0001]], FaceForm[RGBColor[1,0,0]],
  Cuboid[{1-0.001,t-1,u-1},{1,t,u}];

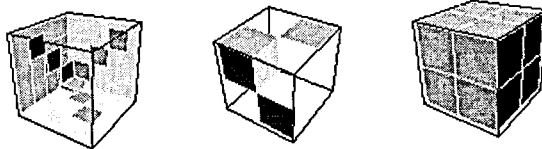
```

3.1.1. 컴퓨터 프로그램이 연출하는 도형의 기하학적 분해와 조립, 파생물질의 기묘한 현상이다 (Thom, 1944).

```

In[28] = Show[ Graphics3D[Table[{{msso,msos,moss,mssi,msis,miss}},
  {s,1,4,1},{t,0,4,1},{u,0,0,1}]], ViewPoint->(2.215,1.561,1.286),
  Lighting->False, AspectRatio->Automatic];
In[29] = Show[ Graphics3D[Table[{{msso, msos, moss, mssi, msis, miss}},
  {s,0,1,1},{t,0,1,1},{u,0,1,1}]], ViewPoint->(-1.505,2.178,2.108),
  Lighting->False, AspectRatio->Automatic];
In[30] = Show[ Graphics3D[Table[{{msso, msos, moss, nsti, nsiu, nstu}},
  {s,0,1,1},{t,0,1,1},{u,0,1,1}]], ViewPoint->(2.215,1.561,1.286),
  Lighting->False, AspectRatio->Automatic];
In[31] = Show[ GraphicsArray[{{%%,%%,%%,%%}}, Boxed->False,
  GraphicsSpacing->(0.5,0.3)]

```



3.1.2. Cuboid의 각 요소를 변환하여 얻는 조형물의 가치를 음미할 수 있는 형상이다.

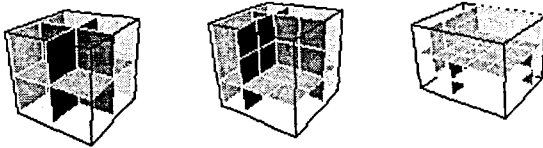
```

In[32] = Show[ Graphics3D[Table[{{msso, msos, moss, nsti, nsiu, nstu}},
  {s,1,2,1},{t,1,2,1},{u,1,2,1}]], ViewPoint->(2.215,1.561,1.286),
  Lighting->False, AspectRatio->Automatic];
In[33] = Show[ Graphics3D[Table[{{msso, msos, moss, nsti, nsiu, nstu}},
  {s,1,3,1},{t,1,3,1},{u,1,3,1}]], ViewPoint->(2.215,1.561,1.286),
  Lighting->False, AspectRatio->Automatic];
In[34] = Show[ Graphics3D[Table[{{msso, msos, moss, nsti, nsiu, nstu}},

```



```
{s,1,4,1},{t,0,4,1},{u,0,1,1}], ViewPoint->{2.215,1.561,1.286},
Lighting->False, AspectRatio->Automatic];
In[35] = Show[ GraphicsArray[{"%%%,%%%,%"}], Boxed->False,
GraphicsSpacing->{0.5,0.3}]
```



III. 결론

특수한 기능을 가진 소프트웨어를 이용하여 수학이 의도하는 여러 측면을 화상으로 보여주는 작업은 수학 그 자체를 멀티미디어 교육 방법으로 보편적 학문으로 인식하게 하는데 기여하고 있다. 수학적 지식의 다양한 활용은 보다 현실적인 문제에서 출발하여 확산과 분해 그리고 재형성의 과정을 걸쳐 학문 전반에 살아 움직이는 분야로 정착할 단계에 와있다. 이 논문의 내용은 이런 작업의 일부분으로 공학계열의 각 분과에 수학의 효용성을 제공하고 있다. 수학은 그 자체의 심오한 구조적 특성과 함께 기호언어로써 여러 학문에 기초를 형성하는 단계별 표현과 향상의 지름길을 여는 것으로 수학 교육적 측면이 중요시되고 있다. 생각하는 수학이 있는 반면에 보는 수학이 있다는 인식을 제고하기 위하여 이런 작업을 수행하고 있다는 뜻이다.

참 고 문 헌

- 한재영 (1999). 수학적 표현론, 서울: 교우사.
- 한재영 (1988). 대한수학교육학회 논문집 8(2), pp.269-278.
- James K. & Millianne Lehmann (1992). *Exploring Calculus with Mathematica*, Addison-Wesley, New York.
- Eugene W.J. (1995). *Linear Algebra with Mathematica*, Brooks/Cole Publishing Company, Cincinnati, Washington.
- Marta, L.A. & James, P. (1977). *Differential Equation with Mathematica*, Academic Press, London, New-York.
- Stwphan W. (1988). *The Mathematica*(3rd, ed.), Cambridge Univ. Press.
- Thom W.J. (1944). *Mathematica Graphics*, Springer-Velag.