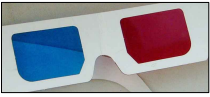


3D Anaglyph in Object Graphics

Anaglyph



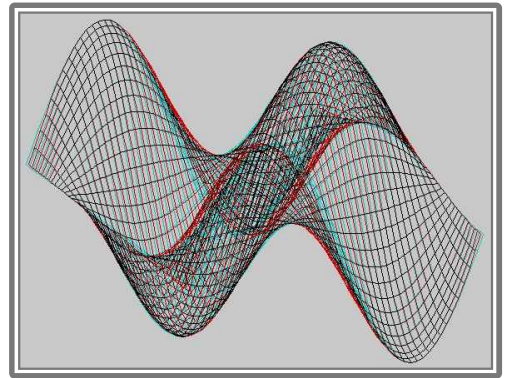
Anaglyph은 두 눈이 빨간색과 파란색을 따로 보게 하여 구현하는 입체시각 효과입니다. 보통은 빨간색과 파란색의 셀로판지로 안경을 만들어 보게 되며, 최근 3D 영화로 유행하는 편광의 원리를 이용하는 방식에 비해 색감 면에서는 뒤떨어지지만, 특수한 장비 없이 일반 용지, 일반 모니터에서도 구현할 수 있어, 대중성 면에서는 압도적이라고 할 수 있겠습니다.

IDL의 Object Graphics 시스템은 입체적 요소를 표출 할 때, 사용자의 시선 방향을 지정하면 지정 시선에서 보는 것처럼 렌더링을 하도록 설계되어 있습니다. 그러므로 왼쪽 눈의 시선으로 볼 때의 입체 형태와 오른쪽 눈의 시선으로 볼 때의 입체 형태를 따로 그려낸 뒤, 이를 각각 빨간색과 파란색 밴드로 할당하여 하나의 이미지로 합쳐 놓게 되면 어렵지 않게 3D 입체시를 구현할 수 있습니다.

Michael Galloy의 Anaglyph 클래스

Object Graphics 시스템에서는 표출 요소들을 IDL 메모리에 입체로 구현한 뒤, 최종적인 표출 대상(화면, 클립보드, 프린터 등)을 사용자가 지정하면 이에 맞추어 그림으로 생산해 냅니다. 일반적인 모니터 화면의 그래픽 창은 IDLgrWindow라는 클래스를 통하여 구현하는데요, Michael Galloy는 IDLgrWindow를 확장하여 MGgrWindow3D라는 클래스를 만들어 공개하고 있습니다. 이 클래스를 이용하여 최종 표출을 하면, anaglyph 입체시 효과를 비교적 간단히 구현할 수 있게 됩니다. (<http://michaelgalloy.com> 에서 anaglyph를 검색해 보세요).

mggrwindow3d_define.pro 와 mggr3dconverter_define.pro 두 개의 파일을 IDL 경로에 포함하면 사용할 준비가 완료된 것입니다. 함께 배포하는 mg_3d_demo.pro와 mg_3dwidget_demo.pro 두 실행 파일을 살펴 보면, 일반 프로그래밍과 GUI 프로그래밍에서 각각 어떻게 사용하는지 감을 잡을 수 있을 것입니다.



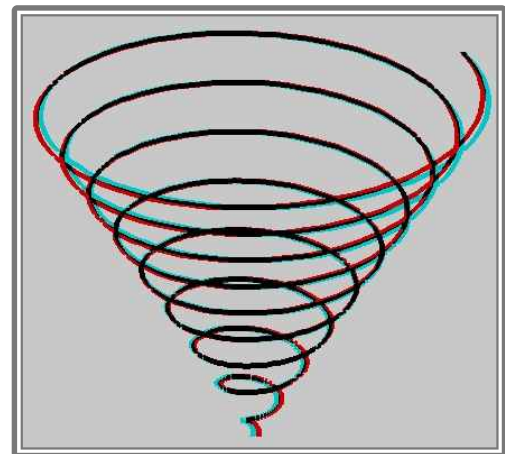
그물형 Surface 예제

그물형 Surface 예제

```
x=findgen(50)#replicate(1, 50)
y=replicate(1, 50)#findgen(50)
z=sin(x/25*!pi)*cos(y/25*!pi) ;Direct Graphics에서는 Surface, Z 로 그림니다.
oAtom=obj_new('IDLgrSurface', z, style=1)
```

```
;이후 과정은 거의 비슷하게 진행됩니다.
oView=obj_new('IDLgrView', color=[200, 200, 200])
oModel=obj_new('IDLgrModel')
oView->add, oModel
oLight = obj_new('IDLgrLight', type=2, location=[-1, -1, 1])
oModel->add, oLight
oModel->add, oAtom
oAtom->getProperty, xrange=xr, yrange=yr, zrange=zr
xc=norm_coord(xr)
yc=norm_coord(yr)
zc=norm_coord(zr)
xc[0] -= 0.5 & yc[0] -= 0.5 & zc[0] -= 0.5
oAtom->setProperty, xcoord_conv=xc, ycoord_conv=yc, zcoord_conv=zc
oModel->rotate, [1, 0, 0], -90
oModel->rotate, [0, 1, 0], 30
oModel->rotate, [1, 0, 0], 30
```

```
;Object Graphics는 일반적으로 IDLgrWindow를 통해 내보내지만, Anaglyph를 만들기
;위해 이 대신 MGgrWindow3d를 사용합니다(IDLgrWindow의 확장 클래스).
oWindow = obj_new('MGgrWindow3d', dimension=[500, 500], eye_separation=3)
oWindow->draw, oView
```



Polyline 예제

메모리에 잔류하는 오브젝트 제거

OBJ_NEW() 함수로 생성된 오브젝트들은 메모리에 계속 남아있게 되며, 명시적으로 메모리에서 제거하지 않으면 메

모리 누수라고 하는 - 사용하지도 않는 객체가 메모리에 잔류하며 관리도 되지 않는 - 바람직하지 않은 상태가 됩니다. 여기 문서에서 보여지는 정도의 간단한 예제라면 메모리에 남아 봐야 크게 무리가 되지 않지만, 실제 프로그래밍을 할 때는 분명히 오브젝트를 제거해 주어야 합니다. 그러므로 예제를 실행해 결과를 확인한 후에는 다음과 같이 메모리 상의 오브젝트를 제거해 줍니다.

```
IDL> obj_destroy, oWindow
IDL> obj_destroy, oView
```

반복되는 코드는 배치(Batch) 프로세스로 처리

위 예제에서 반복되는 부분은 배치 스크립트(Batch Script)로 만들어 이후 작업을 편하게 하겠습니다.

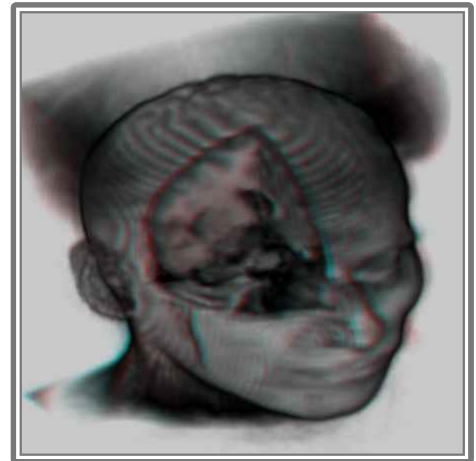
"draw_anaglyph.pro" 라는 이름으로 코드의 반복되는 부분을 저장합니다.

앞의 예제 코드에서 `oView=obj_new('IDLGrView', color=[200, 200, 200])` 부터 `oWindow->draw, oView` 까지 이렇게 해 놓으면 이후에 다음과 같이 중복되는 부분을 한번에 실행할 수 있습니다.

```
IDL> @draw_anaglyph
```

Polyline 데이터 예제

```
IDL> t=findgen(900)
IDL> x=t*cos(t/50!*pi)
IDL> y=t*sin(t/50!*pi)
IDL> oAtom=obj_new('IDLGrPolyline', x, y, t, thick=5)
IDL> @draw_anaglyph
;결과 확인 후
IDL> obj_destroy, oWindow & obj_destroy, oView
```



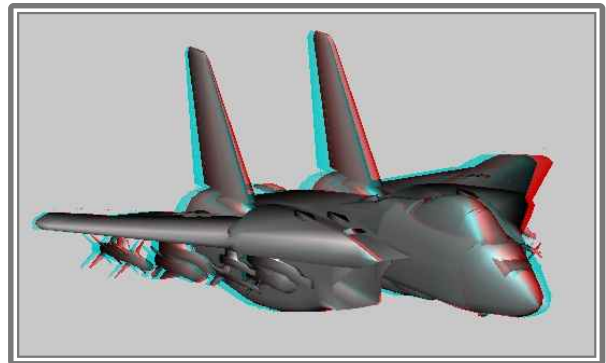
Volume 데이터 예제

Volume 데이터 예제

```
IDL> file=filepath('head.dat', subdir=['examples', 'data'])
IDL> data=read_binary(file, data_dims=[80, 100, 57])
IDL> data[0:39, 0:49, 29:56]=0
IDL> oAtom=obj_new('IDLGrVolume', data, /INTERP)
IDL> @draw_anaglyph
;결과 확인 후
IDL> obj_destroy, oWindow & obj_destroy, oView
```

Polygon 데이터 예제

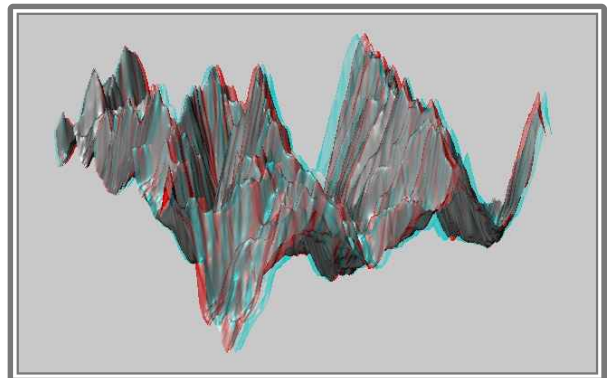
```
IDL> file='C:\Users\yi\Desktop\F-14.dxf'
IDL> oDXF=obj_new('IDLGrDXF')
IDL> status=oDXF-> read(file)
IDL> contents=oDXF->getcontents(10)
IDL> data=oDXF->getEntity(contents)
IDL> obj_destroy, oDXF
IDL> vertices = *data.vertices
IDL> polygons = *data.connectivity
IDL> oAtom=OBJ_NEW('IDLGrPOLYGON', vertices, $
POLYGONS=polygons)
IDL> oAtom->setProperty, AMBIENT=[99, 124, 180]
IDL> oAtom->setProperty, DIFFUSE=[129, 129, 154]
IDL> oAtom->setProperty, SPECULAR=[155, 155, 155], $
SHININESS=21.2
IDL> oAtom->setProperty, Shading=1
IDL> @draw_anaglyph
;결과 확인 후
IDL> obj_destroy, oWindow & obj_destroy, oView
```



Polygon 데이터 예제

DEM Surface 예제

```
IDL> file=filepath('marbells.dat', subdir=['examples', 'data'])
IDL> restore, file
IDL> oAtom=obj_new('IDLGrSurface', elev, style=2)
IDL> oAtom->setProperty, AMBIENT=[99, 124, 180]
IDL> oAtom->setProperty, DIFFUSE=[129, 129, 154]
IDL> oAtom->setProperty, SPECULAR=[25, 255, 255], $
SHININESS=10
IDL> oAtom->setProperty, Shading=1
IDL> @draw_anaglyph
;결과 확인 후
IDL> obj_destroy, oWindow & obj_destroy, oView
```



DEM Surface 예제