

투명도를 적용한 구름의 표출



투명도(Transparency)

영상에서 투명도라는 것은 주로 두 영상을 포개어 표출할 때에 등장하는 용어이며, 위쪽에 얹어지는 영상이 아래쪽에 깔리는 영상을 얼마나 비쳐 보여 주는가를 의미하는 용어입니다. A영상 위에 B영상을 중첩하여 표출할 때, B영상의 불투명도를 0.0~1.0 사이의 수로 나타낸 값을 α 라 한다면, 표출 결과는 다음과 같이 구해질 수 있을 것입니다.

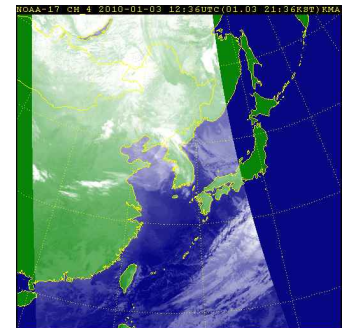
$$\text{Result} = A * (1 - \alpha) + B * (\alpha)$$

```
IDL> file=filepath('people.dat', subdir=['examples', 'data'])
IDL> people=read_binary(file, data_dims=[192, 192, 2])
IDL> ali=people[*,* ,0]
IDL> david=people[*,* ,1]
IDL> T=0.6 ;투명도 0.6(불투명도 0.4) 즉, Ali : David = 60% : 40%
IDL> tvscl, ali*T+david*(1-T)
```



픽셀마다 다른 투명도를 적용해야 합니다.

오른쪽 NOAA 구름 영상처럼, 구름을 사실적으로 표현하기 위해서는 앞에서와 같이 일괄적인 투명도를 적용하는 방식을 적용할 수 없습니다. 구름이 짙을수록 불투명하고, 구름이 옅을수록 투명하게 처리해야 합니다. 즉, 구름 영상의 각 픽셀마다 구름이 얼마나 짙은가에 상응하는 투명도가 적용되어야 하는 것이죠. 결국, 구름 영상의 크기가 [Xsize, Ysize]라면, 투명도 정보 역시 [Xsize, Ysize] 크기의 배열로 존재해야 한다는 것입니다.



기상청 웹사이트(www.kma.go.kr)를 통해 제공되는 NOAA 적외선(구름) 영상. 구름이 없는 부분을 투명하게 처리하였기 때문에 구름 아래의 육지와 바다가 보이고 있습니다.

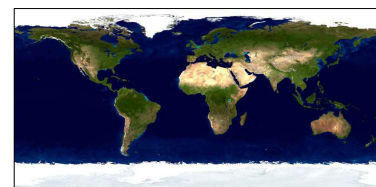
알파(alpha) 채널

투명도 정보의 크기가 영상과 같아보니, 기존의 Red, Green, Blue 채널에 한 채널 더 붙여서 투명도 채널로 쓰게 되었는데, 이 투명도 정보 채널을 보통 **알파(alpha) 채널**이라고 하고, IDL 뿐 아니라 포토샵 등 거의 모든 영상 처리 소프트웨어에서 사용하는 용어입니다. 원래 alpha 채널의 값의 범위는 0.0~1.0으로 0.0은 완전 투명, 1.0은 완전 불투명을 의미합니다(이렇게 따지자면, **알파 채널은 불투명도 채널**이라고 해야 맞을 것 같습니다). alpha 채널을 포함한 트루컬러 영상은 그러므로, [4, Xsize, Ysize]의 크기를 가지는 3차원 배열이 됩니다. 배열이란, 그 안에 포함되는 데이터 타입이 모두 같아야 하거든요. 하지만 보통 0~255의 정수값인 R, G, B 영상 데이터(Byte 타입)와 0.0~1.0의 실수 값(Float 타입)인 불투명도 데이터는 다른 데이터 타입입니다. 그래서, 알파 채널의 값 역시 0~255의 정수값으로 맞추어 쓰기도 합니다. **0이면 완전 투명, 255면 완전 불투명**을 의미하는 값이 되는 것이죠.

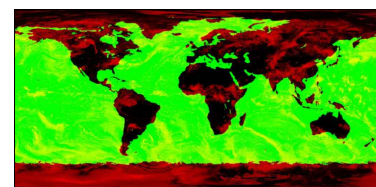
실습에 사용할 영상

IDL 배포판에는 Land Cover 영상인 Day.jpg와 구름 정보 영상인 Clouds.jpg가 포함되어 있습니다. 두 영상 모두 3채널 True Color 영상인데, Clouds.jpg의 Red 채널이 구름의 농도 정보입니다.

```
IDL> land_file=filepath('Day.jpg', subdir=['examples', 'data'])
IDL> cloud_file=filepath('Clouds.jpg', subdir=['examples', 'data'])
IDL> read_jpeg, land_file, land
IDL> read_jpeg, cloud_file, cloud
IDL> help, land, cloud
LAND   BYTE   =ARRAY[3, 1024, 512]
CLOUD  BYTE   =ARRAY[3, 1024, 512]
IDL> iimage, day
IDL> iimage, cloud
```



Day.jpg 구름 아래로 보일 영상입니다.



Clouds.jpg Red 채널이 구름 정보입니다.

중첩용 구름 영상 만들기

Clouds.jpg에서 구름 농도인 Red Channel 만을 뽑아 구름 영상을 만들어야 합니다.

```
IDL> iimage, reform(cloud[0, *, *])
```

위와 같이 0번 채널 정보를 뽑아낼 수 있습니다. 이렇게 뽑아낸 영상은 단일 채널(2차원 배열) 영상이 됩니다. 그렇지만 이 영상에 일괄적인 투명도를 적용하는 것은 우리가 바라는 바가 아닙니다. 구름 표출에서 각 픽셀마다의 투명도가 다르도록 처리해야 한다면 제 4의 채널인 alpha 채널이 꼭 필요합니다.

4채널 영상을 담기 위한 배열 cloud4를 생성합니다.

```
IDL> cloud4=bytarr(4, 1024, 512)
```

구름을 모두 순백색이라고 하고, R, G, B 모두 255인 순백의 영상을 만들어 보겠습니다(구름은 흰색이라고 가정하는 겁니다).

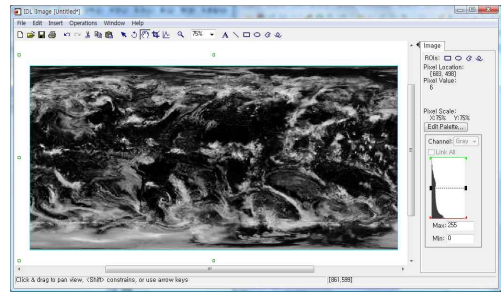
```
IDL> for i=0, 2 do cloud4[i, *, *]=255
```

alpha 채널(불투명도)에는 구름 영상을 끼워 넣습니다. 구름이 없는 곳(0)은 투명하여 아래가 비쳐 보이고, 구름이 가득한곳(255)은 불투명하여 흰색으로 보일 것입니다. 그 중간 단계들은 구름의 농도에 상응하는 불투명도를 가질 것입니다.

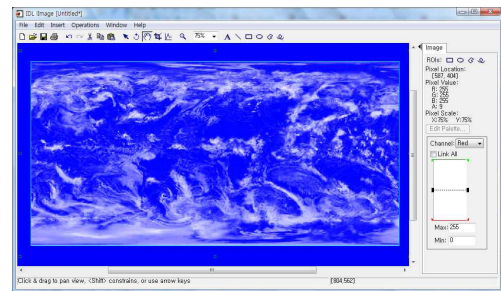
```
IDL> cloud4[3, *, *]=cloud[0, *, *]
```

4채널 영상은 iImage에서 바로 alpha 채널을 인식합니다. 투명도를 확인하기 위해 배경을 파란색(0, 0, 255)으로 설정해 표출해 보겠습니다.

```
IDL> iimage, cloud4, background_color=[0, 0, 255]
```



이 영상의 배경은 흰색입니다. 검은색은 투명한 것이 아니라 진짜 검은색입니다.



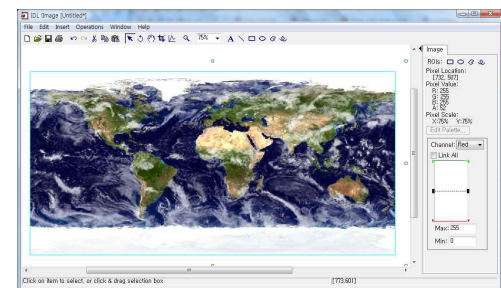
이 영상의 배경은 파란색입니다. 구름이 열수록 배경이 잘 비쳐 보이는 것을 확인할 수 있습니다.

iImage를 이용한 구름 영상 표출

4채널인 cloud4 데이터는 투명도 정보를 포함한 영상이므로 그냥 없으면 됩니다. 다음을 시도해 보세요.

```
IDL> iimage, day
```

```
IDL> iimage, cloud4, /OVER
```



IMap을 이용한 구름 영상 표출

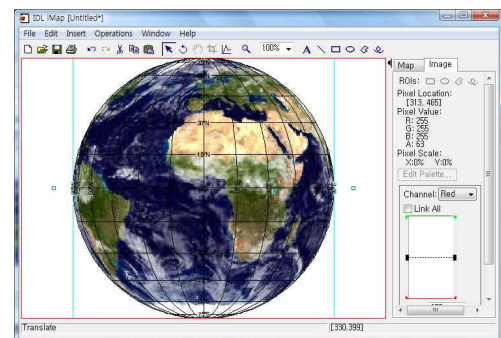
한 장 강의 시리즈의 "지도와 영상 데이터(2)"를 응용하여 구름 영상을 지도 투영법에 맞추어 디스플레이할 수 있습니다. iMap도 내부적으로는 iImage를 호출하므로 일반적인 사용법은 똑같습니다. 다만, 영상의 각 픽셀이 지도 좌표로 어디인지를 알려 주는 것이 중요합니다.

```
IDL> lon=findgen(1024)*360./1023. - 180.
```

```
IDL> lat=findgen(512)*180./512. - 90.
```

```
IDL> imap, land, lon, lat, grid_units=2, $
      map_projection='Orthographic'
```

```
IDL> imap, cloud4, lon, lat, grid_units=2, /OVER
```



그밖에...

- 구름의 색이 꼭 순백색일 필요는 없습니다. 필요에 따라 청색, 적색, 회색 등 어떤 색이든 쓸 수 있습니다.
- 불투명도가 꼭 선형일 필요는 없습니다. 몇가지 함수를 시도해 보세요. 다만, 최종 불투명도의 범위는 0~255가 되도록 맞추어 주어야 합니다. 예를 들어, 0~10의 11단계로 구름 농도를 기록한 데이터라면, BYTSC() 함수를 이용하여 불투명도 범위를 재조정할 필요가 있습니다.