

iTools와 키워드

마우스로 많은 일을 수행할 수 있는 iTools의 특성 때문에 iTools 각각이 수많은 키워드를 가지고 있는 프로시저라는 사실이 간과되고 있습니다. iTools의 많은 키워드 중, 매우 유용한 키워드 몇 가지를 여기에 소개합니다.

IDENTIFIER

iTools를 시동할 때 해당 iTool의 고유 ID를 부여 받습니다. Direct Graphics의 창 번호와 유사하게 사용될 수 있습니다. IDENTIFIER 키워드는 대부분의 경우 줄여서 ID라고 씁니다.

IDL> iPlot, findgen(100), IDENTIFIER=id1

IDL> iSurface, dist(40), IDENTIFIER=id2

IDL> itcurrent, id1

IDL> itdelete

IDL> itdelete, id2

ITCURRENT는 Direct Graphics의 WSET과 유사하며, ITDELETE는 WDELETE와 유사한 프로시저입니다.

OVERPLOT

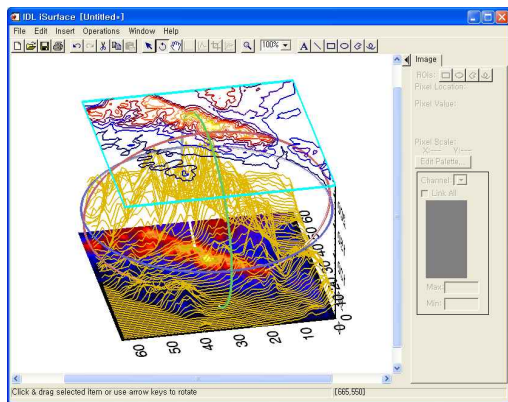


그림1. OVERPLOT 키워드의 활용

OVERPLOT=iTools_ID와 같이 ID로 지정되는 iTool에 겹쳐 그리기를 합니다. Direct Graphics의 /OVERPLOT과 유사하게 사용됩니다. iTools에서도 일반적으로 /OVERPLOT으로 사용하여 현재 iTool에 겹쳐 그리도록 사용하는 경우가 많습니다.

IDL> iSurface, data, style=4

IDL> iImage, data, ZValue=0, RGB_TABLE=5, /OVERPLOT

IDL> iContour, data, ZValue=max(data), \$
RGB_TABLE=5, N_LEVELS=15, /OVERPLOT

RGB_TABLE

iTools는 언제나 트루컬러 디스플레이 환경입니다. 그렇지만, Direct Graphics의 256컬러 환경과 같은 결과로 디스플레이 할 수 있는데, 이 때 사용하는 키워드가 RGB_TABLE입니다. 이 키워드에는 컬러 팔레트를 (3X256 또는 256X3) 배열로 직접 정의해 줄 수도 있고 IDL 컬러테이블 숫자로 지정해 줄 수도 있습니다.

VIEW_GRID, VIEW_NEXT, VIEW_NUMBER

Direct Graphics의 !p.multi로 화면을 분할하여 디스플레이하는 방법과 유사한 방법입니다. /VIEW_NEXT는 다음 뷰로 이동, VIEW_NUMBER는 1번부터 시작되는 VIEW 번호를 직접 지정합니다.

IDL> iImage, img, VIEW_GRID=[3,2] ; 6개의 View

IDL> iImage, img, RGB_TABLE=5, /VIEW_NEXT

IDL> iImage, img, RGB_TABLE=15, VIEW_NUMBER=4

IDL> iImage, img, RGB_TABLE=40, VIEW_NUMBER=3

IDL> iContour, img, VIEW_NUMBER=5

IDL> iSurface, img, /VIEW_NEXT

IDL> iContour, img, /OVERPLOT, \$

VIEW_NUM=1, COLOR=[255, 255, 0]

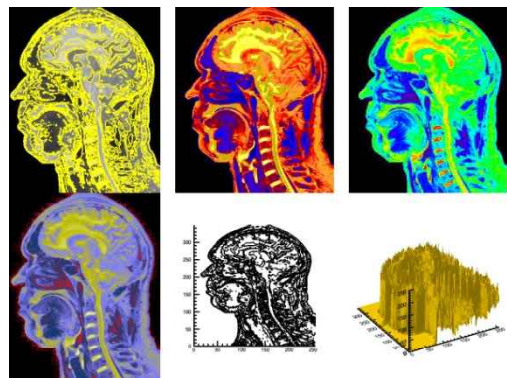


그림2. VIEW_GRID 키워드 활용

TEXTURE_IMAGE

iSurface에서 표면에 입히고자 하는 영상을 지정해 줄 수 있습니다. DEM(Surface)과 위성영상(Image)의 조합을 만들 수도 있고, 값을 높이(Surface)와 색(Surface)의 두가지로 표현할 수도 있습니다.

IDL> iSurface, dmdata, TEXTURE_IMAGE=imdata

IDL> iContour, dmdata, /OVERPLOT, \$

PLANAR=0, COLOR=[255, 255, 0]

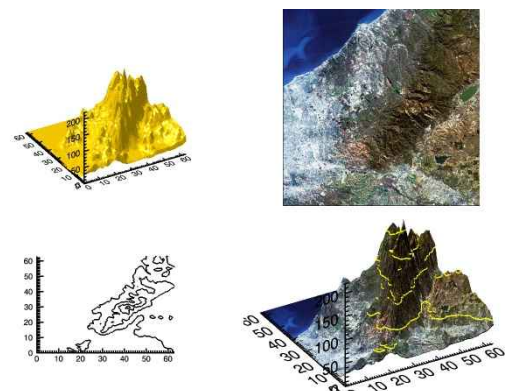


그림3. TEXTURE_IMAGE의 활용

INSERT_LEGEND, VIEW_TITLE

VIEW_TITLE 키워드를 이용하여 각 VIEW의 TITLE을 생성할 수 있습니다. INSERT_LEGEND 키워드를 이용하여 범례를 생성할 수 있으며 이 때 각 항목의 이름은 NAME 키워드의 값을 활용합니다.

```
IDL> iPlot, t, y1, name='Theoretical', $
    /INSERT_LEGEND, VIEW_TITLE=$
    'Obs. vs Theory', XTITLE='time', YTITLE='value'
IDL> iPlot, t, y2, name='Observational', $
    /INSERT_LEGEND, /OVERPLOT
```

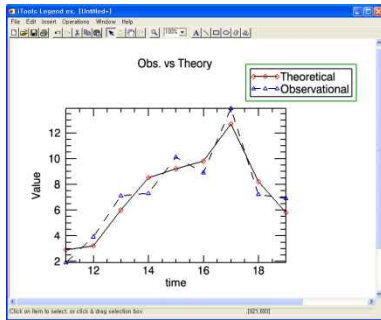


그림4. INSERT_LEGEND와 VIEW_TITLE

AUTO_COLOR, AUTO_SUBSAMPLE

iVector에서 각 벡터들의 색을 컬러테이블에 맞추어 자동으로 스케일링하는 것이 AUTO_COLOR 키워드입니다. 또한 벡터들을 적당한 간격으로 표출할 수 있도록 자동으로 샘플 수를 줄이는 키워드가 AUTO_SUBSAMPLE입니다 (X_SUBSAMPLE과 Y_SUBSAMPLE의 자동화된 키워드입니다). 처음 시험적인 표출을 해 볼 때 유용하게 사용됩니다.

```
IDL> iMap, MAP_PROJECTION='Mollweid'
IDL> iVector, u, v, x, y, /OVERPLOT, $
    /AUTO_COLOR, /AUTO_SUBSAMPLE, RGB_TABLE=39
```

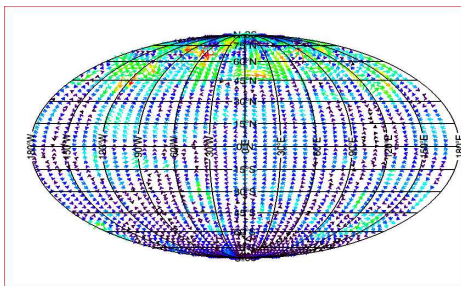


그림5. AUTO_COLOR와 AUTO_SUBSAMPLE 활용

AUTO_RENDER

볼륨 데이터의 렌더링은 연산 부하가 매우 큰 작업입니다. 이로 인해 iVolume의 디폴트 설정은 자동으로 렌더링을 하지 않게 되어 있습니다. iVolume에서 바로 렌더링하여 표출을 하기를 원한다면 AUTO_RENDER 키워드를 사용할 수 있습니다.

```
IDL> iVolume, /AUTO_RENDER, $
    subvolume=[40, 0, 0, 79, 99, 56], /AUTO_RENDER
```

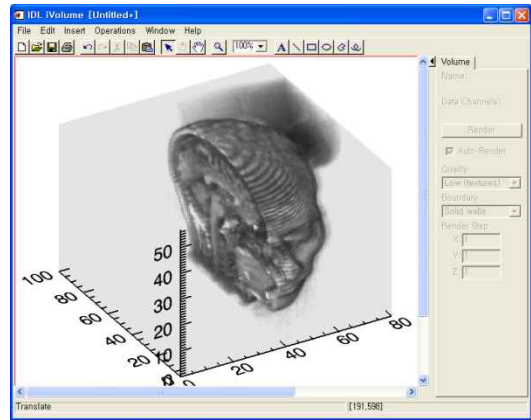


그림6. AUTO_RENDER 키워드 사용

SCATTER

iPlot의 기본 설정은 점과 점을 선으로 잇는 것입니다. 하지만 데이터의 분포를 확인하고자 할 때에 선을 그리지 않도록 하는 방법으로 SCATTER 키워드를 사용할 수 있습니다.

```
IDL> x=randomn(seed, 1000)
IDL> y=randomn(seed, 1000)
IDL> z=randomn(seed, 1000)
IDL> iPlot, x, y, z, VIEW_GRID=[2, 1]
IDL> iPlot, x, y, z, /VIEW_NEXT, /SCATTER
```

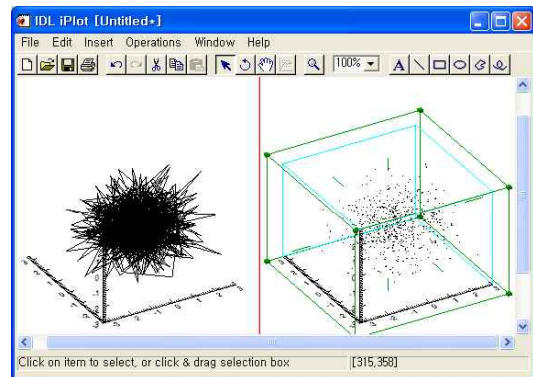


그림7. SCATTER 키워드 사용 (오른쪽)

VECTOR_STYLE

iVector의 표출을 화살표로 할 것인지, 바람깃(풍향/풍속)으로 할 것인지 결정합니다.

```
IDL> iVector, u, v, x, y, VIEW_GRID=[2, 1], $
    VECTOR_STYLE=0, AUTO_COLOR=1, RGB_TABLE=13, $
    /INSERT_COLORBAR
IDL> iVector, u, v, x, y, /VIEW_NEXT, $
    VECTOR_STYLE=1, AUTO_COLOR=1, RGB_TABLE=13, $
    /INSERT_COLORBAR
```

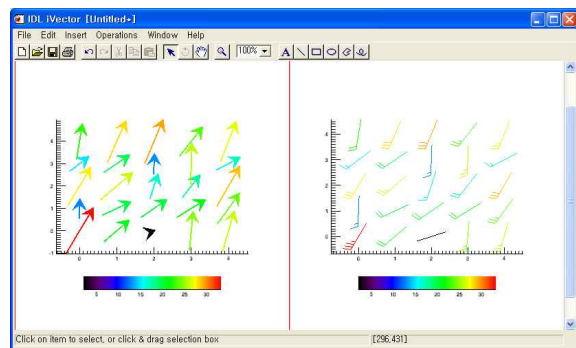


그림8. VECTOR_STYLE 0과 1의 차이