Subject: Re: Reading various HDF files
Posted by peter.albert@gmx.de on Wed, 09 Nov 2005 13:36:10 GMT
View Forum Message <> Reply to Message

Well, it took me a while to create MPEGs, too, but here is my recipe:
First of all, I let mpeg_put read the individual frames from a direct
graphics window. So you have to subsequently display each frame in a
window.

; In order to get the dimensions right, I'd suggest to display the
first image e.g. like

```
window, 1, xsize = 600, ysize = 400
tv, data

; First, open the mpeg_file:

mpegID = mpeg_open( $
  [!d.x_size, !d.y_s
  filename = filename
  motion_vec_length = 1, $
  iframe_gap = 3, $
 quality = 75 $
 )

; With the images I wanted to put together, simply specifying
; QUALITY brought horrible results full of
; jpeg artefacts. With MOTION_VEC_LENGTH and
; IFRAME_GAP, everything is just fine. Increasing
; IFRAME_GAP gives better results, but you pay with longer processing
time.

; O.k., now for the individual frames, assuming that you have some code
for
; reading the i-th dataset

for i = 0, n do begin
  data = read_the_data(i)
  tv, data
  mpeg_put, mpegID, $
    window = !d.window, $
    /color, $
    frame = i, $
    /order
endfor

; After all this, you have to close the file and let IDL do a lot of
compression:
```

mpeg_save, mpegID
mpeg_close, mpegID

That's it. In case you have only a small number of images to show, the MPEG file will be short, given a framerate of 24 frames per second. In that case you can just repeatedly add one and the same frame with a second loop, but make sure to correctly count the frame number as provided via FRAME = i. (i.e. use something like

```
 for j = 0, nframes-1 do begin
   f = i * (nframes) + j
  ...
  ... FRAMES = f, $
  ...
 endfor
```

Cheers,

Peter

)
