Subject: Mismatch between video frame dimensions and stream using Coyote graphics

Posted by BLesht on Sun, 08 Jan 2017 18:13:42 GMT

View Forum Message <> Reply to Message

I've run into a problem creating some mp4 videos from a series of images. I've been following David Fanning's example here: http://www.idlcoyote.com/gallery/mpg_movie.pro The problem seems to be related to how, once the horizontal dimension of the video (width) is specified, the vertical dimension is calculated from the original aspect ratio.

My code is generalized to accommodate different sizes of input images. The problem is that the code works fine except for one case when I get the error. In the case that fails the frame size (determined from the input image size) is 800x924 but the image added to the stream is 800x925. The actual(real) value of the vertical dimension (800/aspect) is 923.913 (see code below). The Coyote code uses Imagemagik's convert command so it seems that the problem is a difference in how the vertical dimension of the frame is computed from the aspect ratio, but I'm not sure. Any insight would be very welcome.

What is do is

1. Calculate the aspect ratio of the original image (plus some padding in the horizontal and vertical dimensions for annotations)

```
dims = SIZE(indata)      ; indata is an 2-d array
xSize = dims[1]
ySize = dims[2]

data  = FLTARR(xSize, ySize)
aspect_rat = (xSize+xpad)/FLOAT(ysize+ypad)    ; xpad and ypad are constants
```

2. Start the video processing

```
video = IDLffVideoWrite(movie_name, FORMAT='mp4')
framerate = 6
framedims = [800, ROUND(800/aspect_rat)]


stream = video.AddVideoStream(framedims[0],framedims[1],framerate)
cgProgressBar = Obj_New("CGPROGRESSBAR", /Cancel, Title = 'Creating movie...')
cgProgressBar -> Start
```

3. Loop through the data

```
FOR i=start, end DO BEGIN
 OPENR, lun, infiles[i], /GET_LUN
 READU, lun, data
 CLOSE, lun
 FREE_LUN, lun
```

4. Open a postscript file

```
 cgPS_Open, 'movie.ps', /QUIET
 cgDisplay, framedims[0], framedims[1]

 scldImage = BYTSCL(data, MIN=0.0, MAX=cmax, TOP=nlevels-1, /NAN)
 scldImage[boundary_index] = 254
 cgImage, scldImage, POSITION=mapPos, OPOSITION=opos, /Keep_Aspect_Ratio, /NOERASE, /Order
```

5. Close the postscript file and output to png

```
  cgPS_Close, /PNG, Width=800, /Delete_PS
```

6. Read the png add it to the video

```
  image=READ_PNG('movie.png')

  File_Delete, 'movie.png'
  void = video -> Put(stream, image)
  IF cgProgressBar -> CheckCancel() THEN BEGIN
    ok = Dialog_Message('The user cancelled operation.')
    video -> Cleanup
  ENDIF ELSE cgProgressBar -> Update, (i+1)/FLOAT(nfiles)*100.

ENDFOR
```