
Subject: mpeg movies from Wave
Posted by [Peter Clinch](#) on Fri, 31 Jan 1997 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Greetings all,

I need to make mpeg movies from PV~Wave. I'm assuming I'll have to export files as 2d images and combine them. Questions are...

1) is there an easier way to do it from an array of movie frames

2) what's a good freeware mpeg encoder (better still with where to get it). I'll need a SPARC Solaris version, though a SunOS one might work under binary compatibility...

Thanks, Pete.

--

Peter Clinch Dundee University & Teaching Hospitals
Tel 44 1382 660111 ext. 3637 Medical Physics, Ninewells Hospital
Fax 44 1382 640177 Dundee DD1 9SY Scotland UK
net p.j.clinch@dundee.ac.uk <http://www.dundee.ac.uk/MedPhys/>

Subject: Re: mpeg movies from Wave
Posted by [Christian Soeller](#) on Tue, 11 Feb 1997 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Included is an IDL routine that does what you wants (you have to check if it runs without modifications under wave). It uses mpeg_encode to do the actual encoding. The doc gives information on how to get it.

Cheers,

Christian

```
FUNCTION pseudo_to_true, image8
```

```
s = SIZE(image8)
IF s(0) NE 2 THEN BEGIN
  MESSAGE, 'input array must be 2D BYTE array.'
  RETURN, -1
ENDIF
```

```
width = s(1)
height = s(2)
```

```
; Load current color table into byte arrays
```

TVLCT, red, green, blue, /GET

image24 = BYTARR(3,width, height)

image24(0,*,*) = red(image8(*,*))

image24(1,*,*) = green(image8(*,*))

image24(2,*,*) = blue(image8(*,*))

RETURN, image24

END

PRO WRITE_MPEG, mpegFileName, image_array, delaft=delaft, rep=rep

;+

; NAME: WRITE_MPEG

;

;

;

;

; PURPOSE: write a sequence of images as an mpeg movie

;

;

;

; CATEGORY: utility

;

;

;

;

; CALLING SEQUENCE:

; WRITE_MPEG,'movie.mpg',ims

;

;

; INPUTS:

; ims: sequence of images as a 3D array with dimensions [sx, sy, nims]

; where sx = xsize of images

; sy = ysize of images

; nims = number of images

;

; OPTIONAL INPUTS:

;

;

;

;

; KEYWORD PARAMETERS:

; delaft: if set delete temporary array after movie was created,

; you should actually always do it otherwise you get

; problems with permissions on multiuser machines (since

; /tmp normally has the sticky bit set)

;

; OUTPUTS: None

;

; OPTIONAL OUTPUTS:

```

;
; COMMON BLOCKS:
;
; SIDE EFFECTS:
;     creates some files in TMPDIR which are only removed when
;     the DELAFT keyword is used
;
;
; RESTRICTIONS:
;     depends on the program mpeg_encode from University of
;     California, Berkeley, which must be installed in /usr/local/bin
;     You find mpeg_encode at
;     ftp://mm-ftp.cs.berkeley.edu/pub/multimedia/mpeg/encode
;     (they even have binaries for a number of platforms).
;
;
; PROCEDURE:
;     writes a parameter file based on the dimensions of the image
;     array + the sequence of images in ppm format into a
;     temporary directory; finally spawns mpeg_encode to build the
;     movie
;
;
; EXAMPLE:
;
;
; MODIFICATION HISTORY:
;
;     Mon Nov 18 13:13:53 1996, Christian Soeller
;     <csoelle@mbcsg1.sghms.ac.uk>
;
;     grabbed original from the net and made slight modifications
;
;-

if n_elements(rep) eq 0 then rep=0

movieSize = SIZE(image_array)
xSize = movieSize(1)
ySize = movieSize(2)
nFrames = movieSize(3)*rep

nDigits = 1+FIX(ALOG10(nFrames))
formatString = STRCOMPRESS('(i'+STRING(nDigits)+'.'+STRING(nDigits)$
+      ')/REMOVE_ALL)
ON_IOERROR, badWrite

```

```

; Make a temporary directory if necessary or clear it otherwise'
TMPDIR = '/tmp/idl2mpeg.frames'
SPAWN, 'if test -d ' + TMPDIR + '; then echo "exists"; fi', result, /SH
dirExists = result(0) EQ 'exists'
IF dirExists THEN command = 'rm ' + TMPDIR + '/*' $
  ELSE command = 'mkdir ' + TMPDIR
SPAWN, command

; Write each frame into TMPDIR as a 24-bit .ppm image file
framenum=0
FOR ino = 0, movieSize(3)-1 DO BEGIN
  image = pseudo_to_true(image_array(*,*,ino))
  for j=0,rep-1 do begin
    fileName = TMPDIR + '/frame.' + STRING(frameNum,FORMAT=formatString)$
      + '.ppm'
    WRITE_PPM, fileName, image
    PRINT, 'Wrote temporary PPM file for frame ', frameNum+1
    framenum=framenum+1
  endfor
ENDFOR

; Build the mpeg parameter file
paramFile = TMPDIR + '/idl2mpeg.params'
OPENW, unit, paramFile, /GET_LUN
PRINTF, unit, 'PATTERN IBBBBBBBBBBBP'
PRINTF, unit, 'OUTPUT ' + mpegFileName
PRINTF, unit, 'GOP_SIZE 16'
PRINTF, unit, 'SLICES_PER_FRAME 5'
PRINTF, unit, 'BASE_FILE_FORMAT PNM'
PRINTF, unit, 'INPUT_CONVERT *'
PRINTF, unit, 'INPUT_DIR /tmp/idl2mpeg.frames'
PRINTF, unit, 'INPUT'
PRINTF, unit, 'frame.*.ppm [' + string(FORMAT=formatString,0) + $
  ' ' + string(FORMAT=formatString,nFrames-1) + ']'
PRINTF, unit, 'END_INPUT'
PRINTF, unit, 'PIXEL FULL'
PRINTF, unit, 'RANGE 5'
PRINTF, unit, 'PSEARCH_ALG LOGARITHMIC'
PRINTF, unit, 'BSEARCH_ALG SIMPLE'
PRINTF, unit, 'IQSCALE 6'
PRINTF, unit, 'PQSCALE 6'
PRINTF, unit, 'BQSCALE 6'
PRINTF, unit, 'REFERENCE_FRAME ORIGINAL'
PRINTF, unit, 'FORCE_ENCODE_LAST_FRAME'
FREE_LUN, unit

; spawn a shell to process the mpeg_encode command
SPAWN, '/usr/local/bin/mpeg_encode ' + paramFile

```

```
IF KEYWORD_SET(delaf) then $  
  SPAWN, 'rm -r ' + TMPDIR
```

```
RETURN
```

```
badWrite:  
alert, 'Unable to write MPEG file!'
```

```
END
```
