
Subject: Re: Videotape Animation

Posted by [kreardon](#) on Thu, 13 May 1993 03:24:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

In article <C6IK1r.KrE@csn.org>, mayer@teal.csn.org (Mike Mayer) writes:

|> Hello.

|>

|> I have someone who wishes to take raster images

|> (512x512 byte images) from PV-WAVE CL and pass

|> them on to their RasterOps frame buffer for

|> subsequent recording to videotape. They have

|> all the required hardware and blank tapes, just

|> wondering how to pass an image from WAVE to

|> the RasterOps board. The board can accept

|> Targa format image files, but WAVE does not

|> write to Targa format. Has anyone attempted

|> this or have any WAVE routines that take an arbitrary

|> bitmap and dump it to a Targa format file?

|>

|> Thanks.

|>

|> Mike Mayer

Here is a program to save an array in standard TARGA format.

It has some features specific to our system (i.e. the LISTFILE thang),

but the guts of determining and writing the TARGA header is all here.

This program only accepts 8-bit images, which sounds like what you want,

but I also have versions that accept 16- and 32-bit images (and also

a program to read a TARGA file and header in IDL).

Have Fun.

kevin reardon

```
;-----  
;  
;+  
; NAME:  
;   TARGAWRITE  
; PURPOSE:  
;   To write out a file in TARGA format  
; CATEGORY:  
;   Movie-making Software  
; CALLING SEQUENCE:  
;   targawrite,file,d  
; INPUTS:  
;   file = output TARGA file name      in  
;   d=data; must be byte (as of 9/92)  in  
; KEYWORD PARAMETERS:
```

```

; Keywords:
; ERROR=err. 0=ok, 1=data error, 2=file not opened
; /BYTESCALE - Enables automatic optimal byte scaling
; /NOERROR - Disables automatic termination of program on error
; /XDR - Uses XDR format
; LISTFILE = File into which lists of files are saved, default=listfile
; MACHINE = machine through which PC can access hard disk
; default=mamane
; ID = user comment to be included in header.
; RED = red color table array (byte array)
; GREEN = green color table array (byte array)
; BLUE = blue color table array (byte array)
; INDEX = number which to tack onto end of 'file'
; /NEW - overwrite existing 'listfile' file instead of appending to it
; OUTPUTS:
; file is written to disk
; COMMON BLOCKS:
; NOTES:
; This program takes a supplied two-dimensional image, byte scales
; that image if BYTESCALE is set, creates a TARGA header,
; and writes the image and header out to disk. A list of the created
; files is put into the same directory (for use on the PC in transferring
; the files.
;
; MODIFICATION HISTORY:
; k.reardon, 10 sep, 1992
;
; -
;-----

pro targawrite,file,d,id=id,help=hlp,error=err,bytescale=bytescale, $
noerror=noerror,listfile=listfile,machine=machine, $
red=red,green=green,blue=blue,index=index,new=new

    if (n_params(0) lt 2) or keyword_set(hlp) then begin
        print,'Write out a file in TARGA format,'
        print,'for transferring to laser disc and videotape.'
        print,' targawrite, file, d'
        print,' file = TARGA file name. in'
        print,' d = data. Must be byte. in'
        print,' KEYWORDS:'
        print,' /XDR use XDR format.'
        print,' ERROR=err. 0=ok, 1=data type error, 2=file not opened.'
        print,' /BYTESCALE disables automatic optimal byte scaling.'
        print,' /NOERROR disables automatic termination of program on error'
        print,' LISTFILE = File into which list of files are saved.'
        print,' MACHINE = disk-access machine; default = mamane.'
        print,' ID = user defined comment to be saved with file.'
    
```

```

print,' RED = red color table array (byte array)'
print,' GREEN = green color table array (byte array)'
print,' BLUE = blue color table array (byte array)'
print,' INDEX = number appended to end of output filename'
print,' /NEW overwrites existing -listfile- instead of appending'
return
goto,finishup
endif

;----- open file -----
on_ioerror, err1
if (n_elements(index) ne 0) then begin
    fil = file+strcompress(index,/rem)
endif else fil = file
get_lun, lun
print,'Output File - ',fil
if keyword_set(xdr) then begin
    openw, lun, fil, /xdr, /stream ; Open unformatted file (XDR).
endif else begin
    openw, lun, fil ; Open unformatted file.
endelse
on_ioerror, err2
if (not keyword_set(listfile)) then begin ; find path for 'file'
    sep = '/'
    brek = 0
    while (brek ge 0) do begin ; find last backslash
        lst = brek
        brek=strpos(fil,sep,brek+1)
    endwhile
    if (lst eq 0) then preface = " $ ; create name for 'listfile'
    else preface = strmid(fil,0,lst+1)
    fillist = preface+'listfile'
    endif else fillist=listfile
get_lun,lun2
filexist = findfile(fillist,count=count) ;find out whether file exists
if ((count ge 1) and (not (keyword_set(new)))) then begin
    openu,lun2,fillist,/append
    print,'List of Output Files appended to - ',fillist
endif else begin
    openw,lun2,fillist
    print,'List of Output Files - overwritten - ',fillist
endelse
; get machine name to append to filename
if (not keyword_set(machine)) then begin
    mach = getenv('HOST')
    if (n_elements(mach) le 0) then mach='mamane'
    endif else mach = machine
;mach = 'mamane' ;for now Pili can only access mamane

```

on_ioerror, null

;----- read color table -----

maxsz = 255 ;maximum value in color table

if (not(keyword_set(red) or keyword_set(blue) or keyword_set(green))) \$

then tvlct,red,green,blue,/get

szr=size(red) ;Because of Windows, color table is < 255 colors

szg=size(green)

szb=size(blue)

if ((szr(0) ne 1) or (szg(0) ne 1) or (szb(0) ne 1)) then begin

print,'Color tables must be one-dimensional arrays.'

print,'I am sorry, but I cannot continue. Exiting. No file written.'

goto,finishup

endif

if ((szr(1) ne szg(1)) or (szr(1) ne szb(1))) then begin

print,'Color table array must all be of the same size.'

print,'I am sorry, but I am unable to continue.'

print,'Exiting. No file written.'

goto,finishup

endif

if (max(red) gt maxsz) or (max(green) gt maxsz) or (max(blue) gt maxsz)\$

then begin

print,'At this time, the color tables are at maximum 8 bits.'

print,'This means that the maximum value in the color table'

print,'array must be less than 256.'

print,'Sorry. I cannot go on. Exiting. No file written.'

goto,finishup

endif

ctab = bytarr(3,szr(1))

ctab(0,*)=byte(blue) ; why this order

ctab(1,*)=byte(green) ; hell if I know.

ctab(2,*)=byte(red)

;----- check data array -----

darr = d

xsx = 640 ;x size of display

ysz = 480 ;y size of display

; scale image to use all available colors (and no more)

if keyword_set(bytescale) then begin

print,'Byte scaling array as per your request.'

darr = bytscl(darr,top=byte(szr(1)))

endif

sizim = size(darr)

if (sizim(sizim(0)+1) ne 1) then begin ;check for byte format

print,'The data is supposed to be in byte format, bub.'

print,'I must byte scale data to conform to required format.'

if (not keyword_set(noerror)) then begin

print,'Exiting, no file written.'

```

    goto,finishup
    endif
endif
if (sizim(0) ne 2) then begin ;check number of dimensions
    print,'Data must be a two-dimensional array.'
    print,'Exiting. No file written.'
    goto,finishup
    endif
;check to see if image is bigger than the video display
if (sizim(1) gt xsz) or (sizim(2) gt ysz) then begin
    print,'Array does not conform to 640x480 video screen.'
    print,'Part of the image will be lost around the edges.'
    if (not keyword_set(noerror)) then begin
        print,'Exiting, no file written.'
        goto,finishup
    endif
endif
;check to see if image is too small and center it if so
if (sizim(1) lt xsz) or (sizim(2) lt ysz) then begin
    print,'Undersized image - centering and adding background.'
    dd=bytarr(xsz,ysz)
    if (sizim(1) lt xsz) then begin
        xbord = fix(((xsz-sizim(1))/2.+.5)
    endif else xbord = 0
    if (sizim(2) lt ysz) then begin
        ybord = fix(((ysz -sizim(2))/2.+.5)
    endif else ybord = 0
    if (xbord eq 0) then begin
        dd(*,ybord:ybord+sizim(2)-1) = darr(0:xsz-1,*)
    endif else begin
        if (ybord eq 0) then begin
            dd(xbord:xbord+sizim(1)-1,*) = darr(*,0:ysz-1)
        endif else begin
            dd(xbord:xbord+sizim(1)-1,ybord:ybord+sizim(2)-1) = darr
        endelse
    endelse
    darr = dd
    endif

;----- make header -----
sizim = size(darr) ;size of data
;print,sizim
maxlen=228 ;maximum length of user defined header information
if (not keyword_set(id)) then id=' '
if (id eq "") then id = ' '
usr=getenv('USER')
ll=strlen(usr)
for tt = ll+1,10 do usr = usr+' '

```

```

showdate,dt,/noprint
datelen = strlen(dt(1))+strlen(dt(2))+strlen(dt(3))+strlen(dt(5))
    xlen = strlen(id)
if (xlen gt maxlen) then begin
    print,'Maximum length for extra header information is 228 characters.'
    print,'Truncating excesses.'
    id = strep(id,'D',maxlen,xlen-maxlen,flg)
    endif
nm=strlen(usr)

    idlen = byte(nm+datelen+xlen);Number of bytes in Image ID Field
cmaptyp = 1b                ;Color map type, 1=color map included
imtyp = 1b                  ;Image type, 1=uncompressed, colormapped
cmapindx = fix(0)           ;Color map, first entry index
cmaplen = fix(sizr(1))       ;Size of color map
cmapsiz = 24b               ;Color map entry size, LUT=24 bits
    xorig=fix(2)             ;X-origin of image
yorig=fix(1)                ;Y-origin of image
imwid=fix(sizim(1))          ;Image width
imhgt=fix(sizim(2))          ;Image height
pixdep=8b                   ;Pixel depth, only 8-bit for now
b0=0
b1=0
b2=0
b3=0
b4=0
b5=0
b6=0
b7=0
imdesc=(b7*128)+(b6*64)+(b5*32)+(b4*16)+(b3*8)+(b2*4)+(b1*2) +b0
    ;Image Descriptor
byteorder,cmapindx
byteorder,cmaplen
byteorder,xorig
byteorder,yorig
byteorder,imwid
byteorder,imhgt

;----- write header -----
writeu,lun,idlen,cmaptyp,imtyp,cmapindx
    writeu,lun,cmaplen,cmapsiz,xorig,yorig
writeu,lun,imwid,imhgt,pixdep,byte(imdesc)
writeu,lun,byte(usr),byte(dt(2)),byte(dt(1)),byte(dt(5))
    writeu,lun,byte(dt(3)),byte(id),ctab,darr
; write out machine and file name to listfile
printf,lun2,mach+'.'+strcompress(fil,/rem)+string(13b)

;print,idlen,cmaptyp,imtyp,cmapindx,cmaplen,cmapsiz

```

```
;print,xorig,yorig ,imwid,imhgt,pixdep,byte(imdesc)
;print,(usr),(dt(2)),(dt(1)),(dt(5))
;print,(dt(3)),(id)
;----- Finish up -----
```

```
done:
close,lun
free_lun,lun
free_lun,lun2
print,'TARGA file '+fil+' written to disk.'
err=0
```

```
goto,finishup
```

```
err1:
print,'Error in targawrite: Could not open file '+fil
err=2
goto,finishup
```

```
err2:
print,'Error in targawrite: Could not open file '+fillist
err=2
goto,finishup
```

```
finishup:
end
```
