Subject: Re: Efficient IDL programming
Posted by deutsch on Fri, 03 Dec 1993 22:49:58 GMT
View Forum Message <> Reply to Message

In article 86221@yuma.ACNS.ColoState.EDU, dean@phobos.cira.colostate.edu writes:
>
> I would like to thank everbody who responded to my request for help in
> "extracting bits from bytes". Below is a test PRO that I made to read in
> the graphic file. It reads, converts, expands, enhances, and reverses my
> file (from (512,64) to (512,512)) in about 30 seconds.
>
> I started with DEC2BIN.PRO posted by Bill Thompson. This worked, but it
> took awhile to go thru 32,768 calculations. Both Chris Chase and Dr. Marty
> Ryba suggested "masks" which speed things up considerably.
>
> I just wanted to check to see if anyone would know if I can illiminated
> the FOR DO BEGIN loops to make this PRO a little more efficient.
>
> Thanks again guys,
>
> Kelly Dean

Try using the following example to vectorize your code. I think this should
be much more efficient at decoding your images since it contains no loops.
Note that I'm using an orientation where the bits are encoded into bytes
in the X direction; you seem to have done something a little differently,
so my example may need to be modified for encoding direction.

```
xsize=512
ysize=512

input_img=byte(indgen(xsize/8,ysize))  ; sample input image

tmp=lindgen(1.0*ysize*xsize)   ; 1.0 to avoid int wrap
tmp2=tmp-(tmp/8)*8    ; create mask input
tmp3=reform(tmp2,xsize,ysize)   ; reform to correct dim's
mask=2L^tmp3     ; create bitmask

work=congrid(inimg,xsize,ysize)   ; replicate each byte 8 times

outimg=(work and mask) ne 0   ; perform the mask

outimg=byte(outimg)*244b   ; enhance value

end
```

I hope this works.. I'd be interested in hearing how much faster this works..
I get the conversion to run in about 5 seconds on SPARC 2.

cheers,
Eric

Eric Deutsch Email: deutsch@astro.washington.edu
Department of Astronomy FM-20 Voice: (206) 543-1979
University of Washington FAX: (206) 685-0403
Seattle, WA 98195 Johnson Hall, Room 226

```
>
>  ============================================================
==================
> pro test
>       head = bytarr(56)
>       premature_EOF = 1
>       ON_IOERROR, SHORT_GRF
> ;
> ;     Read "in house" graphic file.
> ;
>       OPENR, unit, 'dtopo:gms512.grf', /GET_LUN
>       READU, unit, head
>       chead = STRING(head)
>       ck_imx = STRMID(chead,0,6)
> ;
> ;     Verify that it is an IMX graphic file before proceeding
> ;
>       IF ( ck_imx EQ '%IMAGE' ) THEN BEGIN $
>             head_lgth = STRMID(chead,28,7)
>             IF ( head_lgth GT 56 ) THEN BEGIN $
>                   rem_head = bytarr(head_lgth-56)
>                   READU, unit, rem_head
>             ENDIF
>             xsize = strmid(chead,36,6)
>             ysize = strmid(chead,43,6)
>             imxgrf = bytarr(xsize,ysize/8)
>             graphic = bytarr(xsize,ysize)
>             readu, unit, imxgrf
>             premature_EOF = 0
>             SHORT_GRF: IF premature_EOF THEN PRINT, 'Short graphic'
>             close, unit
> ;
> ;     Create mask.
> ;
>             tmp = lindgen(8)
>             mask = 2L^tmp
> ;
> ;     Expand array(xsize,ysize/8) to array(xsize,ysize).
```

```
> ;
>             yyy = 0
>             FOR y = 0,(ysize/8)-1 DO BEGIN
>             xxx = 0
>             FOR x = 0,xsize-1 DO BEGIN
>               IF ( xxx EQ xsize ) THEN BEGIN $
>                    xxx = 0
>                    yyy = yyy + 1
>               ENDIF
> ;      Perform the conversion ( Dr. Marty Ryba (MIT) suggestion).
>               graphic(xxx,yyy) = (imxgrf(x,y) and mask) ne 0
>               xxx = xxx + 8
>             ENDFOR
>             yyy = yyy + 1
>             ENDFOR
> ;
> ;      Enhance value so you can see it and reverse image, then display.
> ;
>             graphic(Where(graphic EQ 001b )) = 244b
>             graphic = rotate(graphic,7) ; Transpose 270 deg, (Xo,-Yo)
>             tv, graphic
>       ENDIF
> END
```