
Subject: Re: convert read_bpm(file, R, G, B) to N x M x 3 array?

Posted by [David Fanning](#) on Wed, 28 Feb 2007 17:45:46 GMT

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

Mike writes:

```
> If I read a color bitmap file like this:
>
>  bmp = READ_BMP(filename, R, G, B)
>
>  bmp is an N x M array with values that are indices of the
>  RGB arrays.  For example, if bmp[5,10] = 3, the color
>  triple for that pixel is R[3], G[3], B[3].  To convert this to
>  an N x M x 3 for widget_button values, I've used a brute
>  force method:
>
>  bmp = read_bmp(filename, R, G, B)
>  s = size(bmp, /structure)
>  Nx = s.dimensions[0]
>  Ny = s.dimensions[1]
>  Nc = n_elements(R)
>  bmpR = bytarr(Nx, Ny)
>  bmpG = bytarr(Nx, Ny)
>  bmpB = bytarr(Nx, Ny)
>  for c = 0, Nc-1 do begin
>    i = where(bmp eq c, count)
>    if count gt 0 then begin
>      bmpR[i] = R[c]
>      bmpG[i] = G[c]
>      bmpB[i] = B[c]
>    endif
>  endfor
>  result = [[[bmpR]], [[bmpG]], [[bmpB]]]
>
>  It seems like there ought to be a more IDLy way to do
>  this, but my juggling skills are failing me.  Can
>  anyone suggest something that takes fewer lines of
>  code and is more consistent with The IDL Way?
```

It seems to me something like this should work:

```
  bmp = read_bmp(filename, R, G, B)
  bmp24 = [[r[bmp]], [g[bmp]], [b[bmp]]]
```

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: convert read_bmp(file, R, G, B) to N x M x 3 array?

Posted by [Wox](#) on Thu, 01 Mar 2007 09:04:17 GMT

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

On 28 Feb 2007 09:24:38 -0800, "Mike" <Michael.Miller5@gmail.com>
wrote:

```
> Dear IDL gang,  
>  
> If I read a color bitmap file like this:  
>  
>  bmp = READ_BMP(filename, R, G, B)  
>  
> bmp is an N x M array with values that are indices of the  
> RGB arrays.  For example, if bmp[5,10] = 3, the color  
> triple for that pixel is R[3], G[3], B[3].  To convert this to  
> an N x M x 3 for widget_button values, I've used a brute  
> force method:
```

You know you could do this right?

```
button=widget_button(base,value=filename,/bitmap)
```

```
> bmp = read_bmp(filename, R, G, B)  
> s = size(bmp, /structure)  
> Nx = s.dimensions[0]  
> Ny = s.dimensions[1]  
> Nc = n_elements(R)  
> bmpR = bytarr(Nx, Ny)  
> bmpG = bytarr(Nx, Ny)  
> bmpB = bytarr(Nx, Ny)  
> for c = 0, Nc-1 do begin  
>   i = where(bmp eq c, count)  
>   if count gt 0 then begin  
>     bmpR[i] = R[c]  
>     bmpG[i] = G[c]  
>     bmpB[i] = B[c]  
>   endif  
> endfor  
> result = [[[bmpR]], [[bmpG]], [[bmpB]]]  
>  
> It seems like there ought to be a more IDLy way do
```

> this, but my juggling skills are failing me. Can
> anyone suggest something that takes fewer lines of
> code and is more consistent with The IDL Way?

Why not just using this:

```
result=read_bmp(filename)
```

This will give a 3xNxM, so you can use this to get NxMx3:

```
result=transpose(result,[1,2,0])
```

Subject: Re: convert read_bpm(file, R, G, B) to N x M x 3 array?

Posted by [Mike\[2\]](#) on Thu, 01 Mar 2007 16:37:23 GMT

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

On Feb 28, 12:45 pm, David Fanning <n...@dfanning.com> wrote:

> It seems to me something like this should work:

>

> bmp = read_bmp(filename, R, G, B)

> bmp24 = [[r[bmp]], [g[bmp]], [b[bmp]]]

Thats it - well, almost:

```
bmp24 = [[[r[bmp]]], [[g[bmp]]], [[b[bmp]]]]
```

Thanks David!

Subject: Re: convert read_bpm(file, R, G, B) to N x M x 3 array?

Posted by [David Fanning](#) on Thu, 01 Mar 2007 16:58:54 GMT

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

Mike writes:

> Thats it - well, almost:

>

> bmp24 = [[[r[bmp]]], [[g[bmp]]], [[b[bmp]]]]

Oh, right. When I'm in a hurry I should just remember to make
it look like JDs stuff and I'll be all right. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: convert read_bpm(file, R, G, B) to N x M x 3 array?

Posted by [Mike\[2\]](#) on Thu, 01 Mar 2007 17:45:03 GMT

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

On Mar 1, 4:04 am, Wox <nom...@hotmail.com> wrote:

```
> You know you could do this right?
> button=widget_button(base,value=filename,/bitmap)
```

I know that, but I'm experimenting with converting my bitmaps to functions that return the bitmaps with the idea that it might result in more portable code.

```
> Why not just using this:
> result=read_bmp(filename)
> This will give a 3xNxM, so you can use this to get NxMx3:
> result=transpose(result,[1,2,0])
```

That doesn't work with all bitmaps. READ_BMP returns an NxM array for a single channel bitmap.

Mike

```
IDL> tmp = query_bmp('smaller.bmp', info)
```

```
IDL> help, info, /str
```

```
** Structure <844d21c>, 7 tags, length=40, data length=36, refs=1:
```

```
CHANNELS      LONG          1
DIMENSIONS    LONG    Array[2]
HAS_PALETTE   INT           1
NUM_IMAGES    LONG          1
IMAGE_INDEX   LONG          0
PIXEL_TYPE    INT           1
TYPE          STRING  'BMP'
```

```
IDL> help, read_bmp('smaller.bmp')
```

```
<Expression>  BYTE      = Array[24, 24]
```

```
IDL> tmp = query_bmp('alpha_left.bmp', info)
```

```
IDL> help, info, /str
```

```
** Structure <844bfc4>, 7 tags, length=40, data length=36, refs=1:
```

```
CHANNELS      LONG          3
DIMENSIONS    LONG    Array[2]
HAS_PALETTE   INT           0
```

```
NUM_IMAGES    LONG        1
IMAGE_INDEX   LONG        0
PIXEL_TYPE    INT         1
TYPE          STRING      'BMP'
IDL> help, read_bmp('alpha_left.bmp')
<Expression>  BYTE       = Array[3, 11, 11]
```
