

---

Subject: Re: FLOAT images instead of BYTE ones from IDL Object graphics ?

Posted by [Karl\[1\]](#) on Mon, 26 Jan 2009 17:57:36 GMT

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

---

Well, your composite function is

$$\text{dest}' = \text{src} * \text{srcalpha} + \text{dest} * (1 - \text{srcalpha})$$

srcalpha comes from your opacity table, which has a max value of 15.

So, for a voxel at max value, 255, the brightest that a single pixel can be against a black background is 15. Of course all voxels along the ray would contribute to the brightness in an additive way.

With a max opacity of 15, you are going to be getting "mostly background" with each blend. It will take a lot of high voxel values along to ray to get them to add up to close to 255.

Where did you get 15?

If you change that to something like 100, you will get closer to the full range in the final image.

Karl

On Jan 23, 9:31 am, Gianluca Li Causi <[lica...@mporzio.astro.it](mailto:lica...@mporzio.astro.it)> wrote:

```
>> You are going to have to spell this out for me. I
>> cannot imagine why you think the OPACITY_TABLE0
>> limits you to 100 values of gray scale. Sorry for
>> being on the dense side of the opacity scale. :-(
>
>> Cheers,
>
>> David
>
> Ok, so I think that a practical example could help: the following
> program makes a spherical volume made of concentric shells, following
> a sin(radius) law.
>
> I use a linear grayscale as RGB_TABLE so that the maximum value is
> white and the minimum is black.
> Then I also use a linear Opacity from 0 to 15, because I want to well
> view throug all the shells until the center.
>
> As you see, the final image is a byte array and its maximum value is
> 142.
> In my application the maximum gray is in the range 10 to 20, but I
```

```

> want an image with 256 gray levels not only 10 or 20 grays.
>
> Thanks for your help
> Gianluca
>
> Here is the sample program:
>
> x_pix = 100.
> y_pix = x_pix
> z_pix = x_pix
>
> Volume_Color = (findgen(256)/255) # [255,255,255]
> Volume_Opacity = (findgen(256)/255) * 15
>
> ;Make a spherical volume with density in shells:
> Volume_Data = fltarr(x_pix,y_pix,z_pix)
> FOR x = 0., x_pix-1 DO BEGIN
>     FOR y = 0., y_pix-1 DO BEGIN
>         FOR z = 0., z_pix-1 DO BEGIN
>             Volume_Data[x,y,z] = SQRT((x_pix/2.-x)^2 + (y_pix/2.-y)^2 +
(z_pix/
> 2.-z)^2)      ;distance from center
>             ENDFOR
>         ENDFOR
>     ENDFOR
> Volume_Data = Volume_Data * (Volume_Data LE (x_pix/2.))      ;cut at x_pix/
> 2 radius
> Volume_Data = SIN(Volume_Data)
>             ;make sinusoidal shells
>
> ;convert to byte as required by IDLgrVolume
> Volume_Data_byte = BYTE((Volume_Data / max(Volume_Data)) * 255.)
>
> ;3d graphic objects
> oWindow = OBJ_NEW('IDLgrWindow', RETAIN=2, DIMENSIONS=[400,400])
> oView = OBJ_NEW('IDLgrView', COLOR=[0,0,0])
> oModel = OBJ_NEW('IDLgrModel')
>
> ;Create Volume Object
> oVolume = OBJ_NEW('IDLgrVolume', Volume_Data_byte, LIGHTING_MODEL=0,
> INTERPOLATE=1, $
>             OPACITY_TABLE0=Volume_Opacity,
COMPOSITE_FUNCTION=0,
> ZERO_OPACITY_SKIP=1, ZBUFFER=1)
>
> oModel -> Add, oVolume
>
> ;set display window

```

```
> Display_XRANGE = [0, x_pix-1]
> Display_YRANGE = [0, y_pix-1]
> Display_ZRANGE = [0, z_pix-1]
>
> Display_xSize = x_pix
> Display_ySize = y_pix
> Display_zSize = z_pix
> Display_Diagonal = SQRT(Display_xSize^2 + Display_ySize^2 +
> Display_zSize^2)
>
> Display_xCenter = x_pix/2
> Display_yCenter = y_pix/2
> Display_zCenter = z_pix/2
>
> oModel -> TRANSLATE, -Display_xCenter, -Display_yCenter, -
> Display_zCenter
>
> oModel -> ROTATE, [1,0,0], -90
> oModel -> ROTATE, [0,1,0], -60
> oModel -> ROTATE, [1,0,0], 30
>
> oView -> SetProperty, VIEWPLANE_RECT=Display_Diagonal*[-.5,-.5,1,1],
> ZCLIP=Display_Diagonal* [.5,-.5], EYE=Display_Diagonal
>
> oView -> Add, oModel
>
> ;Display Object Hierarchy
> oWindow -> Draw, oView
>
> ;Catch the Window
> oWindow -> GetProperty, IMAGE_DATA=IMAGE
>
> help, IMAGE
> print, max(IMAGE)
>
> stop
>
> OBJ_DESTROY, oView
> OBJ_DESTROY, oWindow
```

---