Subject: Re: what is an efficient lossless compression way to store a gray-scale image
Posted by R.G. Stockwell on Tue, 26 Aug 2003 17:55:34 GMT
View Forum Message <> Reply to Message

"Xiaoying Jin" <xje4e@mizzou.edu> wrote in message
news:10ea38a6.0308260923.28c8cad9@posting.google.com...
> First I forgot to mention, the image is not necessary byte or integer.
> It could be float type. So .png file is not suitable in some cases.
> Anyway, it works great in 8 or 16 bits cases.

> Thanks for your suggestion! But if we wrote the files in .dat rather
> than the image files. Then we need to rewrite it into an image file if
> we want to look at the image. So not a good idea.

You could always whip up an IDL GUI to read and display the images.

...
>> So, the tiff command is actually pretto good, giving you a ~50% size.
> I think tiff command without compression will keep the original size
> 2.097.152bytes. Because the data is integer type, so when writing the
> tiff, /uint is used. What you got is byte type image data.

Doh! You are right, my error.

One thing that comes to mind is to hold the data in the type of variables
that are coming from the instrument that created the images. For instance,
if these are CCD images they will digitized at a certian level of precision
(i.e. 8, 10, 12, or 16 bit Analog to Digital conversion). You can make
sure you
keep the data files at this precision (notably, 12 bit images will be stored
inefficiently
in IDL normal types as a float or as an integer).
There is potentlially some room there to improve efficiency. This may
require that you
"undo" the scaling of the data, and for a lot of data, this may not really
be possible
(i.e. Dopplergrams of the sun's surface for instance).

So, if you can't make the above changes, my guess is that a 50% compression
is about as good as you are going to get in general, and I don't know of any
way to improve
on the compression routines that you can use with the write_tiff. (while
being lossless).

Other image options, You can do *.fits images, which can be read in IDL by
some user routines,
and also with other software like ds9 on linux boxes.

Cheers,
bob