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 16:42:07 GMT
View Forum Message <> Reply to Message

"Xiaoying Jin" <xje4e@mizzou.edu> wrote in message
news:10ea38a6.0308260652.6a1e1b9a@posting.google.com...
> Hi, there,
>
> For a gray-scale image, I haven't found a good way to store it yet. It
> seems that there is no way to store gray-scale images in IDL
> efficiently. If so, that will be too bad, because all the satellite
> images we processed are very large and occupy a lot of space.
>
> I tried 'tiff' format with Packbits compression, but it does not help
> for gray-scale image. Sometimes the file size of the compressed image
> is even larger than the original raw data!!!
>
> Can anyone give me some hint? Thank you very much!
>
> Regards,
>
> Julia

Hi Julia,
this is not trying to be a flippant reply, but the easiest solution is to
simply buy more hardrives.
A couple of 200G drives fore a couple hundred each might solve the problem.

Or, offhand I would say the best you can do is probably directly writing a
binary file
of the appropriate precision (and use the compress keyword on the openw
procedure).
For instance, if you have 16 bit numbers, write an array of integers.
A quick look seems to show that your tiff NONcompressed files are pretty
good.
(of course, the compression you actually gets depends on the data)

I was curious, so I made a little example.
here, the data is 1024 x 1024 16 bit integers, so it should
be about 2Megs in size 2,097,152 bytes.

```
len = 1024
randomdata = fix(100*randomn(seed,len,len))
regulardata = indgen(len,len)
openw,lun,'randomdata_compress.dat',/get_lun,/compress
```

```
 writeu,lun,randomdata
free_lun,lun
openw,lun,'randomdata.dat',/get_lun
 writeu,lun,randomdata
free_lun,lun
openw,lun,'regulardata_compress.dat',/get_lun,/compress
 writeu,lun,regulardata
free_lun,lun
openw,lun,'regulardata.dat',/get_lun
 writeu,lun,regulardata
free_lun,lun
write_tiff,'tiff_compress_random',randomdata,compression=2
 write_tiff,'tiff_compress_regular',regulardata,compression=2
write_tiff,'tiff_regular',regulardata,compression=0
```

These commands give the following file sizes:

```
08/26/2003  10:36a         1,477,230 randomdata_compress.dat
08/26/2003  10:36a         2,097,152 randomdata.dat
08/26/2003  10:36a         1,905,228 regulardata_compress.dat
08/26/2003  10:36a         2,097,152 regulardata.dat
08/26/2003  10:31a         1,058,072 tiff_compress
08/26/2003  10:36a         1,058,072 tiff_compress_random
08/26/2003  10:36a         1,058,062 tiff_compress_regular
08/26/2003  10:36a         1,049,862 tiff_regular
```

So, the tiff command is actually pretto good, giving you a ~50% size, and it works
beter than the gzip compression in the openw command.
So perhaps your best bet is just to buy more disk space, or reduce your data based on
some other criteria (i.e. bin the data or downsample to a larger sampling size in space, or bin/downsample
in time if that is an appropiate for your applications)

Cheers,
bob
