Subject: Re: what is an efficient lossless compression way to store a gray-scale image
Posted by xje4e on Tue, 26 Aug 2003 17:23:45 GMT
View Forum Message <> Reply to Message

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.

> 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).
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.

> 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.
> 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.
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.

Regards,

Julia