

Re: Reduce numbers to one number

Source: <http://sci.tech-archive.net/Archive/sci.math/2006-02/msg00326.html>

- *From:* stush@xxxxxxxxxxxxxxxx
 - *Date:* 2 Feb 2006 03:22:13 -0800
-

Socrates wrote:

```
> Hello -
>
> I am developing an online and downloadable utility to enable any image
> to be expressed ultimately as a single number and or a character.
>
> At the present stage by right-clicking on any image on the web, the
> utility stores the pixel values in the image from left-to right and top
> to bottom as characters or numbers (based upon converting the image to
> 256 colours if necessary).
```

jpg is lossy in that the uncompressed image doesn't equal the original.

My first question was is your algorithm lossy or lessless, but you answer it here if the image > 256 colors. For 24-32 bit images, that could present a problem. jpg works by converting the image to wavlets and then storing an approximation of those wavlets that lends itself more to compression. By just converting an image to 256 colors vs jpg lossy compression of hi-color-resolution images, I think you will see a noticable drop of quality in the uncompress imaged. Of course you could jump up to 16 bit "characters" e.g. unicode. Another option I remember back to the days of the C64 where you could only have 4 colors per 4x8 pixels. But people where able to put out some amazing images (considering the limitation). You could break the image up into chunks and assign a 256 color palette to each chunk. The more chunks, the less loss of color quality.

Another, more common, trick you can do is split the image into 3 256 color images, one red-scale, one blue-scale, and one green-scale.

```
> An image of 500 x 375 pixels appears at www.amwl.com - there are
> 187,500 pixels which results in 187,500 numbers or characters.
>
> The file size of the jpg image is 145 kb but the resulting text file
> containing the 187,500 text characters is 187kb which can be zipped
> down to under 1kb. So far so good. However, I want to go further and
> reduce the text file even more.
```

Something doesn't seem right here. I went to that site, but couldn't find the image you are looking for.

A common term in compression theory is entropy which is how much "information" is contained in data. If you are mapping 256 pixel colors to 256 characters, you have not changed the entropy, you should not get better ratios than say zipping the image file stored as a raw image or unix pixmap.

Try storing the image as 375000 hexadecimal characters. I guessing you may actually get a little better compression if you zip it.

Re: Reduce numbers to one number

For such a small image with few colors, most of the jpg file is probably unnecessary overhead. Compare a larger, higher color file in jpg and your method.

I would also compare your method to converting the image to something like RLE or compressed BMP particular compressed raw (which is essentially what I think you have).

> One way of reducing the 184,500 numbers and characters is to apply a
> calculator that the utility will make so as to reduce the numbers to
> far fewer numbers/characters.

Which is what all compression algorithms already do. Try zipping a zip file.

> These few remaining numbers or characters can then be saved as a text
> file which will be no more than a few bytes in file size (one text
> character = 1 byte).
>
> When the recipient receives the text file (which may be no larger than
> 1kb) then the same utility will reproduce the image from the numbers or
> characters appearing in the text file.
>
> Can anyone guide me on the type of calculation that my utility will
> have to make in order to reduce the characters or numbers in the text
> file?

You should drop the idea of a text file. A text file is just a stream of bytes. The reason text files seem to compress so well is the low entropy of written language. But just converting an image to text, you do not change the entropy.

First I suggest you read up on compression theory. You should also learn the jpg algorithm. Another lossy image compression algorithm is Fractal Image Compression which is much, much better than jpg, but it is not "free" and you must pay license fees to use it. But I would suggest learning how it works.

There are two approaches to lossy image compression: one is to focus on the visual properties of the original image (to maintain as much quality as possible from the original image) and one is to just play with the numbers. Current algorithms in use are the former. If you ignore (compressed) image quality and just play with the numbers, you might have more luck as I would guess lossy data file compression has had the least attention.

> The best formula or method will be incorporated in the utility with
> credit given to the author.

.