Dithering as a method for image data compression

Pavel Slavík (slavik@cslab.felk.cvut.cz),
Jan Přikryl (prikryl@sgi.felk.cvut.cz)

Czech Technical University Prague
Faculty of Electrical Engineering
Department of Computer Science
Karlovo náměstí 13, 121 35 Praha 2, Czech Republic

Keywords: *data compression, dithering, parallel processing.*

1 Introduction

Data compression in general is not a new problem. This problem has to be solved always when problems with "the size" of information occur (e.g. size of files, volume of information transmitted through networks etc.). Generally, all data contain some redundant information. The objective of a data compression algorithm is to transform redundant data into a form which is smaller leaving out the redundant information. In case of textual (one-dimensional) information we assume that the information compressed retains the same information. As we will see in further this is not always the case when compressing the image data.

Many methods for information compression have been already developed. They posses various attributes and properties according to which they are classified (e.g. static – dynamic [1]). In this paper we will deal with image data compression. We will discuss various approaches to the image compression and we will present a new method which does not suffer from some disadvantages of methods widely used.

2 Traditional methods for image data compression

Any scheme to compress a given message may be viewed as consisting of two steps: source modelling and coding. The task of a model is to determine the characteristics of data to be compressed. The characteristics obtained helps to develop an efficient algorithm for data compression. The first algorithms used for image data compression were slightly modified algorithms for string compression.

In such a case we speak about sequential encoding: each image component is encoded in a single left-to-right, top-to-bottom scan. The 2D image is thus transformed in a sequence of scan lines. These scan lines are encoded by means of methods formerly used for string compression like RLE, Huffman coding, LZW etc. These encodings are usually lossless: the image is encoded to guarantee exact recovery of every source image sample value.

There is also another class of compression methods used for image data compression: lossy methods. These methods allow to compress image with "some acceptable" information loss. This results in higher compression ratio in comparison with the lossless compression algorithms. The lossy compression algorithms have usually two-dimensional character as they investigate a two-dimensional area in the image.

Example of such a method is the Discrete Cosine Transformation (DCT). In this case the image is divided into pixel blocks consisting of 8×8 pixels. The DCT transforms the "locational" information into "frequency" information. The coherence between pixels in the block results in the situation when the value of coefficients (obtained by means of the transformation) corresponding to higher frequencies are either zero or near to zero. These coefficients can be neglected. This represents the compression. The Inverse DCT (IDCT) outputs 8×8 sample blocks to form the reconstructed image.

Another example of a lossy compression method is fractal compression. As in the previous example the image is divided in pixel blocks. A representation of a pixel block by means of IFS theory should be found. This approach allows to obtain high compression rates — compressed image retains less than 1/100th of the space needed by other methods. The description of the pixel block has a form of a set of transformation coefficients for appropriate IFS functions.

3 Alternative approach to the image data compression

Some of the above listed methods suffer from several disadvantages. There are either lossless compressions (but the compress ratio is low) or lossy compression (with the high compression ratio but with lower quality of the decompressed image). It is obvious that for many applications the lower quality of the decompressed image is acceptable. The problem which has to be solved concerns the strategy how the information from image can be omitted.

In case of lossy methods the principle of methods is to find proper values which describe with "acceptable accuracy" the pixel block (e.g. 8×8 pixels) without respect to the situation in the surrounding pixel blocks. A method should be found where the picture is processed as a whole with the respect to human eye properties.

Such a method has been already known for couple of years. We speak about dithering. This method has been used to display images with high number of grey levels on a device with black and white pixels only [2]. The image information contained in the image is reduced. By this reduction an error in each pixel occurs. This error can be distributed in surrounding pixels. In such a way the "local errors" are not so obvious as in the case when local image processing takes place. It is obvious that the information is considerably reduced — e.g. when using the dithering method for a gray level out of 16 gray levels - then after dithering one pixel corresponds to one bit only (instead of former 4 bits). That means that the compression ratio is 1:4.

From experiments with dithering methods it is clear that certain amount of information is lost what results in the loss of tiny details in the image. When investigating other compression methods we can see that there are two stages:

- image compression
- image decompression.

In the case of dithering only the first stage has been covered. We should investigate the possibility to obtain the original image (or the image as close as possible to the original one). It is necessary to use the inverse process — "undithering" [3].

Before the explanation what the undithering is we should remind methods used for dithering. The two most common methods are ordered dither and Floyd–Steinberg dither. Ordered dither uses a cleverly chosen set of black-and-white patterns to represent the different gray levels. The algorithm uses a thresholding scheme to replace each gray pixel with a black or white pixel [2]. Floyd–Steinberg dither is error diffusion method that processes the pixels of each scan-line from left to right and top to bottom. Each pixel is examined and rounded to black or white. We have to compensate the error — we distribute this error into neighbouring pixels. In such a way the information is not lost. The Floyd–Steinberg dither is considered to be better method than ordered dither as it does a better job of representing fine lines.

Both methods take into account the process of human eye perception. The dithered image consists of black and white pixels. The reason the dithering works is that the eye blurs the dots into shades of gray, so the first step in undithering is mechanically smooth or blur the image.

Computerized blurring is done by replacing each pixel's value with the average of the pixel in a small region around it. Usually the 4×4 square is used. This square averages an even number of pixels and produces "correct" gray levels for pixels investigated. To get 4×4 average we can use the convolve function. Convolution "rolls together" the pixels in a region as a weighted average and places it in the center pixel.

The result of the blurring usually produces an almost realistic gray-scale image. When investigating the obtained picture carefully we can see (in many cases) that the image often looks mottled and the smoothing has blurred some details. The mottling occurs because there are not enough grays in the gray scale (only 17 shades compared to the original 256). To smooth out the mottled areas without disturbing the high-contrast areas (that represent the edges of objects). We can use an optimal filtering technique: Lee's local statistics method [3].

This statistics method has been used in many applications — e.g. when measurement is corrupted by noise. We can use this method to estimate the true gray levels of an image that has been corrupted by dithering. After the first stage of our undithering process we have a picture with 17 gray levels instead of the original 256 levels. We can assume that each original gray level had been rounded to the nearest of the 17 gray levels. The difference between the original value and the rounded value is the noise.

The method used uses a least-mean-square estimator. To apply this method, we can use estimates of the mean and variance of both the true signal (the original image) and the noise (the error introduced by the dithering and blurring). None of these means or variances is known to us directly. We can estimate the mean and variance of the original image at each point by calculating the mean and variance of a small square centered at the point in the blurred image (therefore the "local statistics").

In works dealing with the information filtering we can meet filters which allow to sharpen edges etc. These filters are also convolutions whose kernels can be described as matrices. During practical experiments we have found out that the matrix element values are highly dependent on the image processed. In difference from previous methods the usage of this method does not lead in general to a better image.

4 Experiments with image data compression and decompression

The above described method has been implemented and extensive experiments have been made. For this article three example pictures (a raytraced picture of two spheres and scanned photographs) were used.

The raytraced picture is a typical synthetic picture without use of textures. Therefore we can find several areas of almost the same gray level in the picture which makes the compression ration slightly higher than when compressing typical scanned photograph.

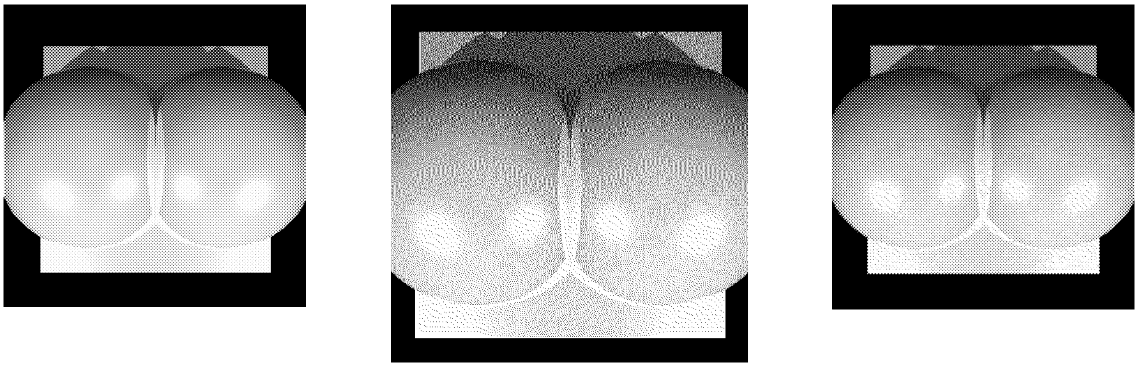

Fig.1: Original, dithered and undithered picture 'spheres'.

To evaluate the quality of the decompressed picture in comparison to the original one a proper method had to be developed. It would be complicated to develop a procedure which would include both the formal similarity of original and uncompressed picture (from the mathematical point of view) and the similarity from the point of human perception. In general we can say that decompressed pictures are quite near to the original ones — they are a bit blurred but the loss of information is not very substantial.

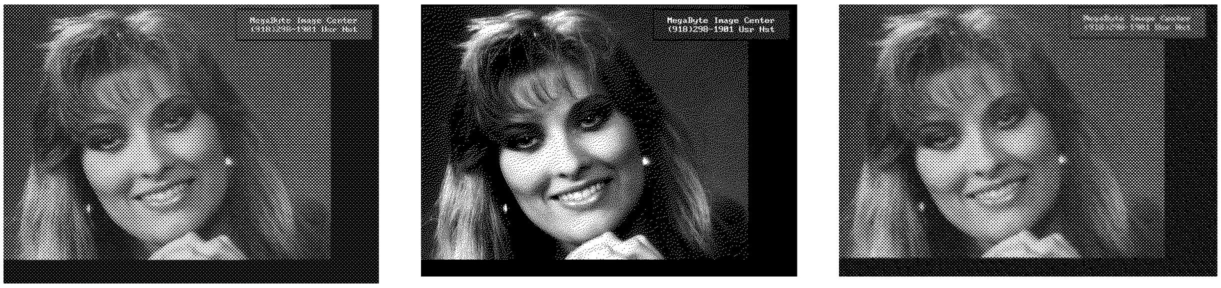

Fig.2: Original, dithered and undithered picture 'photo1'.

As the first criterion we have defined the loss of average brightness of the decompressed picture. The brightness of all pixels have been summed and then divided by the number of pixels. The results are shown in Tab.1.

picture name	original brightness	undithered brightness	loss
spheres	100%	90%	10%
photo1	80%	70%	15%
photo2	85%	72%	14%

Tab.1: Loss of picture brightness

We can make a conclusion that the loss of brightness is more significant in pictures with minor average brightness. The loss of brightness corresponds to the loss of tiny details in the reconstructed picture.

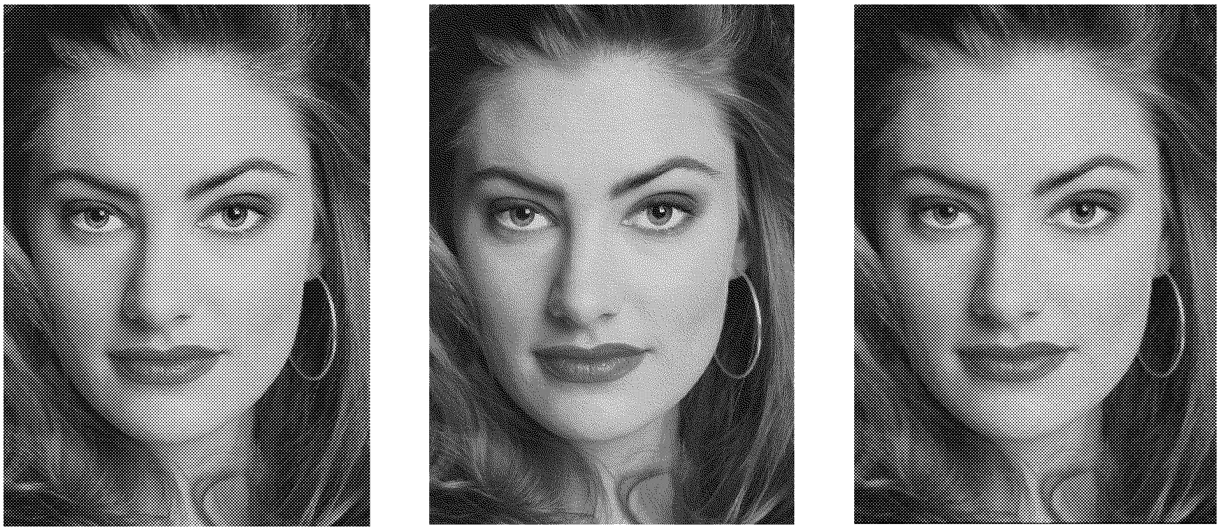

Fig.3: Original, dithered and undithered picture 'photo2'.

From the table can be seen that the average decrease in the brightness is about 2%. Experiments have shown that the 2-5% increase of brightness in the whole reconstructed picture will show some details which could not be seen before. It will be necessary to continue in experiments where the quality of human perception will be considered.

The most important characteristics of image data compression algorithms is the compression ratio. To improve it we have used the second stage of compression after the dithering process. Here we have used a frequently used GIF picture format which uses the loosless LZW compression method in order to decrease the volume of picture data. We should have in mind that dithering itself makes significant data compression. In case when 256 gray shades are used in the original picture the compression ratio is 1:8. The typical compression ratio for the grayscale GIF images is 1:4 to 1:5.

When combining dithering (what is in our case lossy compression method) with above given compression methods (lossless ones) we can obtain image data compression with relatively high compression ratio with a good quality of the decompressed picture. Tab.2 gives us an idea about the quality of compression in comparison to the size of original files.

name of the picture	greyscale size	GIF size	bitmap size	bitmap GIF size	compression ratio
spheres	467810	336783	58680	53338	1:8.77
photo1	306560	142740	38320	21431	1:14.30
photo2	409600	55402	51200	26455	1:15.48

Tab.2: Quality of compression

We can see that on average the volume of data when using our combined method is 3 to 4 times smaller in comparison with the original GIF files. The decrease in the quality is not so dramatic.

5 Future work

One of the directions for the future work is the investigation of transformation methods to obtain better quality of decompressed pictures. We have followed different line of research. During the compression and decompression the speed of operations is critical especially when a sequence of pictures is transmitted via network. To speed up the computation parallel methods can be used for both the compression and decompression of the picture.

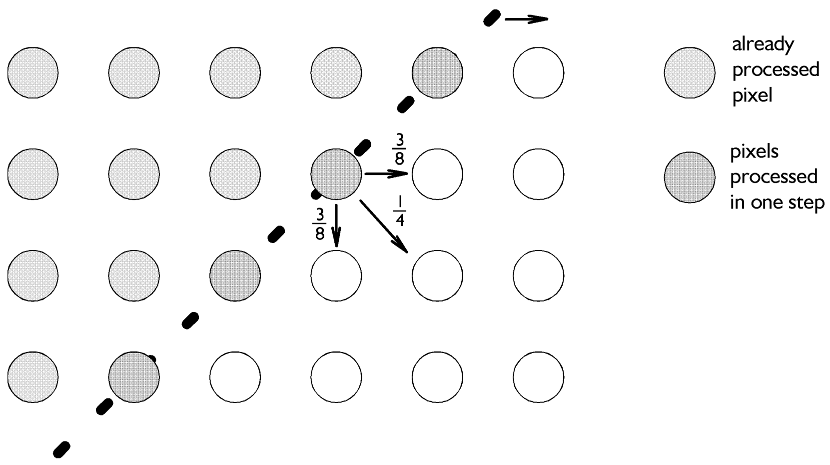

Fig. 4 Scheme for parallel dithering

In case of the ordered dither it would be an easy task as each pixel can be processed independently what leads to parallel processing. As we used the Floyd–Steinberg dither (because of the better quality) we had to discuss the possibility of parallelizing this method. We have used the simple version of Floyd–Steinberg dither where the error is distributed into east, south and south-east pixels (relatively to the investigated pixel). Then we have chosen a scheme where all the pixels which lay on the "diagonals" of the picture (lines slope of which is 45 degrees) can processed in parallel. The number of pixels processed in one step varies in time but this method gives a considerable speed-up of the computation.

This method has been tested in practice. As no physical device with hundreds of processors was available we used the simulation system PARALAXIS developed at the University of Stuttgart [4]. This system is a model for massively parallel computing. The tasks suitable for the SIMD computer architecture can be modelled on this system. The programming language is a parallel extension of the MODULA–2 language. It allows the user to define connections between processors and thus define the data exchange between them. The program which performed the above given method was surprisingly short (about 80 statements including the description of configurations of processors) and the results obtained have shown that the implementation was correct [5].

The speed-up of the dithering process when using parallel computation can be seen in Tab.3.

size of picture	number of steps	average number of processors in one step	number of pixels
64×64	128	32	4096
128×128	256	64	16384
256×256	512	128	65536
512×512	1024	256	262144

Tab.3: Speed-up of the parallel dithering process.

The number of pixels represents also the number of steps in case when the computations are performed serially.

The decompression (undithering) can be also performed in parallel. The parallelization of the undithering process is possible in all phases of the process. In case of blurring process it is possible to compute the average of the area surrounding the investigated pixel in parallel for many pixels at the same time. Similar approach can be used for the local statistics method.

6 Conclusion

The method described has several aspects. It uses two compression steps: dithering and a traditional loosless compression method (LZW). This approach (two compression steps) has been also used in other systems. The use of dithering has an advantage that after the first decompression the picture (in its low quality form) can be displayed. This is especially important when performing interaction via networks as there occurs the possibility to perform a "raw" interaction with pictures in the dithered form. When the layout of subpictures in the picture is satisfactory then the second decompression process can take place. This would speed up considerably the graphical interaction in networks.

References

[1] Melichar, B., Pokorny, J.: Data Compression, Report DC-92-04, Czech Technical University Prague, Dept. of Comp. Sci., 1992

[2] Rogers, D.F.: Procedural Elements for Computer Graphics, McGraw Hill 1985

[3] Stenger, A.: Converting Dithered Image Back to Gray Scale, Dr. Dobb's Journal, Nov. 1992, pp. 64-68.

[4] Barth, I., Braunl, T., Engelhardt, S., Sambach, F.: Parallaxis Version 2 - User Manual, Comp. Sci. Report 2/92, Uni. Stuttgart 1992

[5] Felkel, P.: Dithering and Parallel Processing, student project, CTU Prague 1993.