

DIFFERENTIAL IMAGE COMPRESSION BASED ON ADAPTIVE PREDICTION

M.V. Gashnikov

Samara National Research University, Samara, Russia

Abstract. The paper describes the adaptive prediction algorithm for differential methods of image compression. This prediction algorithm can automatically use a variety of calculation ways, depending on the smoothness of local image area. The parameters of proposed prediction algorithm are adjusted automatically by simple recursive scheme before the actual processing. An experimental research of the proposed algorithm is performed at «Waterloo» set of images.

Keywords: digital images compression, prediction, differential compression, mean square error, maximum error.

Citation. Gashnikov MV. Differential image compression based on adaptive prediction. CEUR Workshop Proceedings, 2016; 1638: 320-326. DOI: 10.18287/1613-0073-2016-1638-320-326

Introduction

Differential methods of image compression [1-4] that are also called by differential pulse code modulation (DPCM) have many important advantages: low computational and structural complexity, the ability to control the compression ratio without the use of the buffer memory etc. Differential methods are used for the compression of remote sensing data [5-8] and are part of other, more complex methods of compression, such as the JPEG [9] and the hierarchical grid interpolation [10-13]. Now the task of improving these methods is still actually.

In differential compression the image $x(n_1, n_2)$ is processed in a progressive mode, line by line. For each pixel $x(n_1, n_2)$ the predicted value $p(n_1, n_2)$ is calculated, based on the already processed pixels. Then this predicted value $p(n_1, n_2)$ is subtracted from the original value $x(n_1, n_2)$ to obtain a difference signal $d(n_1, n_2)$. Then, the difference signal $d(n_1, n_2)$ is quantized [2] and is encoded ("packed") to an archive file or a link. By selecting different predictors, quantizers and "packers", we can design a differential method with fundamentally different characteristics. Prediction algorithm is a central element of differential compression of digital image. So, the prediction algorithm in many ways determines the effectiveness of differential compression method. In this article we propose an adaptive prediction algorithm, based on the use of different prediction ways depending on local characteristic of the surrounding area of the current pixel.

1 Prediction for differential compression of digital images

Often, one of the closest pixels already processed or the half-sum of closest pixels is used as the predicted values:

$$p^{(0)}(n_1, n_2) = r(n_1 - 1, n_2), \quad (1)$$

$$p^{(1)}(n_1, n_2) = \frac{1}{2}(r(n_1 - 1, n_2) + r(n_1, n_2 - 1)), \quad (2)$$

$$p^{(2)}(n_1, n_2) = r(n_1, n_2 - 1), \quad (3)$$

where $r(n_1, n_2)$ are already processed pixels. These predictors are linear. Therefore, these predictors have a large error on the boundaries (contours). Besides, there are also non-linear predictors, such as a Greham [3-4] predictor:

$$p^G(n_1, n_2) = \begin{cases} p^{(0)}(n_1, n_2), & \text{if } V(n_1, n_2) < H(n_1, n_2), \\ p^{(2)}(n_1, n_2), & \text{if } V(n_1, n_2) \geq H(n_1, n_2), \end{cases} \quad (4)$$

where the discrete differences are calculated in such way

$$V(n_1, n_2) = |r(n_1, n_2 - 1) - r(n_1 - 1, n_2 - 1)|, \quad (5)$$

$$H(n_1, n_2) = |r(n_1 - 1, n_2) - r(n_1 - 1, n_2 - 1)|. \quad (6)$$

Algorithm (4) predicts "along" the vertical or horizontal contour (see. Fig. 1). Therefore, the error of this predictor in such contours is much small. The predictor (2) is more accurate on the flat areas of the image due to the noise averaging. On the contrary, the predictor (4) is effective on contours.

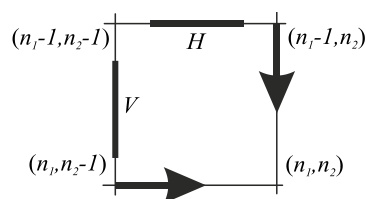


Fig. 1. Greham predictor. Arrows indicate the direction of prediction. Bold lines indicate the discrete differences

In this paper, we propose the adaptive predictor which can be automatically switched between the formulas (2), (4) at each pixel of the image according to the contour characteristics of the local neighborhood of that pixel (see Fig. 2):

$$p^A(n_1, n_2) = \begin{cases} p^{(0)}(n_1, n_2), & \text{if } \mu(n_1, n_2) < \mu^{(-)}; \\ p^{(1)}(n_1, n_2), & \text{if } \mu^{(-)} \leq \mu(n_1, n_2) \leq \mu^{(+)}; \\ p^{(2)}(n_1, n_2), & \text{if } \mu^{(+)} < \mu(n_1, n_2), \end{cases} \quad (7)$$

where

$$\mu(n_1, n_2) = V(n_1, n_2) - H(n_1, n_2) \quad (8)$$

is «feature of direction»; $\mu^{(-)}$, $\mu^{(+)}$ are «switching parameters»

$$(-x_{\max} \leq \mu^{(-)} \leq 0 \leq \mu^{(+)} \leq x_{\max}); x_{\max} = \max\{x(n_1, n_2)\}.$$

If the «feature of direction» (8) has a sufficiently large positive value, the contour in the neighborhood of the current pixel is vertical. Therefore, prediction is performed vertically (1), in this case. If the feature (8) has a sufficiently large negative value, the contour is horizontal. So, prediction is performed horizontally (3), in this case. If the absolute value of the feature (8) is small, the contour is not exists in the neighborhood of the current pixel (it is a flat area). In this case, the prediction is performed by averaging (2) for noise reduction.

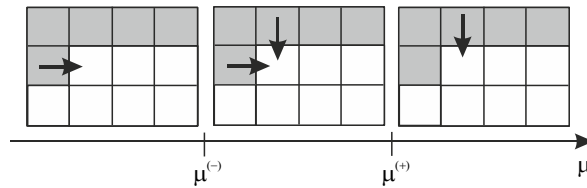


Fig. 2. Proposed adaptive predictor. Arrows indicate the direction of prediction

In other words, algorithm (7) is automatically switched between predictors (2) and (4). So, algorithm (7) combines the advantages of predictors (2) and (4): it works well on the contours and flat areas. In fact, algorithm (7) is a parameterization of predictor (4).

2 Calculation of the switching parameters

«Switching parameters» $\mu^{(-)}$, $\mu^{(+)}$ are calculated automatically by rapid recurrent procedure. This calculation is performed only once before the actual image processing. To increase the accuracy of predictions this calculation is made on the basis of minimizing the total prediction error:

$$\varepsilon(\mu^{(+)}, \mu^{(-)}) = \sum_{(n_1, n_2)} |x(n_1, n_2) - p(n_1, n_2)| \rightarrow \min_{\mu^{(+)}, \mu^{(-)}}, \quad (9)$$

where $p(n_1, n_2)$ is a predicted value. The complexity of this optimization problem stems from its two-dimensionality.

To move to the one-dimensional problem we consider the sets of indices corresponding to different signs of feature (8):

$$\begin{cases} N^{(+)} = \{(n_1, n_2) : \mu(n_1, n_2) > 0\} \\ N^{(-)} = \{(n_1, n_2) : \mu(n_1, n_2) < 0\} \\ N^{(0)} = \{(n_1, n_2) : \mu(n_1, n_2) = 0\} \end{cases} \quad (10)$$

We can see, that sets (10) form a covering of all indexes $\{(n_1, n_2)\}$ of the image $x(n_1, n_2)$. The total prediction error (9) is divided into parts that meet sets (10) of indices

$$\varepsilon(\mu^{(+)}, \mu^{(-)}) = \varepsilon^{(-)}(\mu^{(-)}) + \varepsilon^{(0)} + \varepsilon^{(+)}(\mu^{(+)}) , \quad (11)$$

where

$$\varepsilon^{(+)}(\mu^{(+)}) = \sum_{(n_1, n_2) \in N^{(+)}} |x(n_1, n_2) - p(n_1, n_2)| , \quad (12)$$

$$\varepsilon^{(-)}(\mu^{(-)}) = \sum_{(n_1, n_2) \in N^{(-)}} |x(n_1, n_2) - p(n_1, n_2)| , \quad (13)$$

$$\varepsilon^{(0)} = \sum_{(n_1, n_2) \in N^{(0)}} |x(n_1, n_2) - p(n_1, n_2)| . \quad (14)$$

We can see that each of these errors depends on no more than one parameter. Thus the 2-D problem (9) is divided into two 1-D problems:

$$\varepsilon^{(+)}(\mu^{(+)}) \rightarrow \min_{\mu^{(+)}} , \quad (15)$$

$$\varepsilon^{(-)}(\mu^{(-)}) \rightarrow \min_{\mu^{(-)}} . \quad (16)$$

Next, we consider the algorithm for solving the first problem (15), i.e., the search of switching parameter $\mu^{(+)}$. To do this we fill the two 1-D arrays $E^{(0)}$ and $E^{(1)}$ on the first pass in the image

$$E_{\mu}^{(k)} = \sum_{(n_1, n_2) \in N_{\mu}} |x(n_1, n_2) - p^{(k)}(n_1, n_2)| , \quad (17)$$

where $p^{(k)}$ are predicted values (1-3). The length of arrays (17) is $(x_{\max}+1)$. The set of indices N_{μ} is recorded as follows

$$N_{\xi} = \{(n_1, n_2) : \mu(n_1, n_2) = \xi\} . \quad (18)$$

Next, the 1-D array of prediction errors (12) can be calculated by the simple recurrent formula:

$$\varepsilon^{(+)}(x_{\max}) = \sum_{\mu=0}^{x_{\max}} E_{x_{\max}}^{(1)}; \quad \varepsilon^{(+)}(\mu) = \varepsilon^{(+)}(\mu+1) + E_{\mu}^{(2)} - E_{\mu}^{(1)}, \quad \mu = (x_{\max} - 1) \dots 0 \quad (19)$$

The optimum value of switching parameter $\mu^{(+)}$ is searched in the array (19) by simple search. The length of this array is $(x_{\max}+1)$; therefore this search is fast. So, the problem (15) is solved. The problem (16) is solved similarly. So, the optimum values $\mu^{(+)}$, $\mu^{(-)}$ are founded and the problem (9) is solved.

Importantly, the image size does not affect the computational complexity of the recursion procedure (19). The computational complexity of this procedure is $6x_{\max}$ operations for any image size. For example, for the image size is 1000×1000 , the computational complexity of procedure (19) is 0,00153 operations per pixel. This value can be neglected. So, the total computational complexity of the proposed predictor (7) is 9 operations per pixel. This value is sufficiently small for most practical problems.

3 Experimental results

As part of this article we have carried out all the software implementation of all considered algorithms in C#. With the help of the developed software we performed the computational experiments on the Waterloo-set [14] of grayscale images, which is a generally accepted standard set in the research of image compression algorithms. We used the differential method with considered predictors for compression of images of Waterloo-set.

As a measure of the error between the original image $x(n_1, n_2)$ and the reconstructed image $r(n_1, n_2)$, already passed the compression and decompression, we use the relatively square error [3-4] and the maximum error [2]

$$\varepsilon_{ratio}^2 = \frac{\varepsilon^2}{D_x} = \frac{1}{MND_x} \sum_{n_1=0}^{S_1-1} \sum_{n_2=0}^{S_2-1} (x(n_1, n_2) - r(n_1, n_2))^2$$

$$\varepsilon_{\max} = \max_{(n_1, n_2)} |x(n_1, n_2) - r(n_1, n_2)|$$

where D_x is estimation of image dispersion, ε is mean square error (MSE), $S_1 \times S_2$ are image sizes.

Some results are averaged over all images Waterloo-set, shown in Fig. 3-4. As seen in Fig. 3, the proposed adaptive predictor (7) has the distinct advantage over the averaging predictor (2) (up 7%) and the Grahm predictor (4) (up 4%) at the compression ratio. In addition, we compared the differential method of image compression with proposed predictor (7) against the JPEG compression method [9]. As shown in Fig. 4, considered implementation of the differential method has the distinct advantage (up to two times) of JPEG at the maximum error.

Thus, the proposed adaptive predictor is a promising to enhance the efficiency of the differential methods of digital image compression.

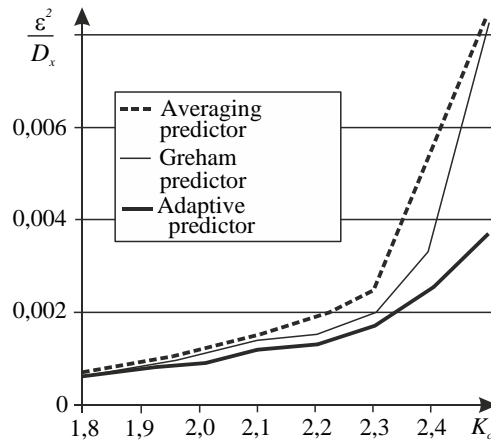


Fig. 3. The relationship ratio MSE ϵ^2/D_x from compression ratio K_c

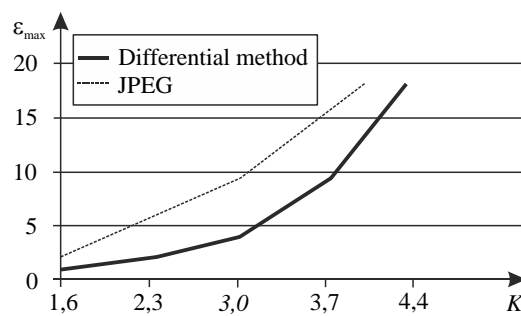


Fig. 4. The relationship maximum error ϵ_{\max} from compression ratio K_c

Conclusion

Adaptive prediction algorithm is developed for differential methods of image compression. The algorithm is based on the use of several prediction formulas, depending on the contour direction of the current pixel. Algorithm parameters are automatically adjusted to different images based on fast recursive algorithm. Computational experiments demonstrate the proposed algorithm gain from other prediction algorithms in real images.

Acknowledgements

This work was financially supported by the Russian Scientific Foundation (RSF), grant no. 14-31-00014 “Establishment of a Laboratory of Advanced Technology for Earth Remote Sensing”.

References

1. Salomon D. Data Compression. The Complete Reference. Springer-Verlag, 4ed, 2007; 1118 p.
2. Lin S, Costello D. Error Control Coding: Fundamentals and Applications, 2ed. New Jersey: Prentice-Hall, inc. Englewood Cliffs, 2004; 1260 p.
3. Pratt W. Digital image processing. Wiley, 4ed, 2007; 807 p.
4. Woods E, Gonzalez R. Digital Image Processing. Prentice Hall, 3ed, 2007; 976 p.
5. Jensen JR. Remote sensing of the environment: an Earth resource perspective (2nd ed.). Prentice Hall, 2007.
6. Chang C. Hyperspectral Data Processing: Algorithm Design and Analysis. Wiley Press, 2013; 1164 p.
7. Borengasser M, Hungate W, Watkins R. Hyperspectral Remote Sensing – Principles and Applications. CRC Press, 2004; 128 p.
8. Chang C. Hyperspectral data exploitation: theory and applications. Wiley-Interscience, 2007; 440 p.
9. Wallace G. The JPEG Still Picture Compression Standard. Communications of the ACM, 1991; 34(4): 30-44.
10. Gashnikov MV, Glumov NI. Hierarchical grid interpolation for hyperspectral image compression. Computer Optics, 2014; 38(1): 87-93. [in Russian].
11. Gashnikov MV, Glumov NI. Hierarchical GRID Interpolation under Hyperspectral Images Compression. Optical Memory and Neural Networks (Information Optics), 2014; 23(4): 246-253.
12. Gashnikov MV, Glumov NI. Hyperspectral images repository using a hierarchical compression. Posters Proceedings of 23-rd International Conference on Computer Graphics, Visualization and Computer Vision (WSCG), 2015: 1-4.
13. Gashnikov MV, Glumov NI. Hierarchical compression for hyperspectral image storage. Computer Optics, 2014; 38(3): 482-488. [in Russian].
14. Waterloo Grey Set. University of Waterloo Fractal coding and analysis group: Mayer Gregory Image Repository, 2009. URL: <http://links.uwaterloo.ca/Repository.htm>.