

# NEAR-LOSSLESS LZW IMAGE COMPRESSION

**Macarie BREAZU, Antoniu PITIC, Daniel VOLOVICI, Remus BRAD**  
**University “Lucian Blaga” of Sibiu, Computer Science Department**  
**Emil Cioran 4, Sibiu, Romania**  
**email: macarie.breazu / antoniu.pitic / daniel.volovici / remus.brad @ulbsibiu.ro**

**Abstract:** Classical image compression methods are based on measuring the error only at entire image level. In some areas there is an obvious need for getting an upper bound for the error at the pixel level. In the paper we propose such a near-lossless method based on LZW dictionary algorithm. The modifications needed to adapt LZW to become a near-lossless method are presented. As far as we know this is the first attempt to use LZW as a near-lossless method. Experimental results done and presented in the paper prove that the method gives better than the one based on quadtree partitioning so the proposed method is promising.

**Key words:** image, compression, near-lossless, quadro, LZW.

## 1. INTRODUCTION

Today, in the age of computers and communication, there is an obvious need for data compression and, especially, for image compression. Because the compression methods where no error is accepted (called lossless) gives poor results on images we have to accept some error in the decompressed image in order to get much higher compression ratios. Those methods (called lossy) become very popular especially with the multimedia development.

In the classic approach the criteria that is minimized by the image compression methods is:

$$E_1(R, O) = \sum (R_{ij} - O_{ij})^2$$

where  $R_{ij}$  represents the restored (decompressed) image and  $O_{ij}$  represents the original (uncompressed) image and the sum is made over the entire image. Here we are interested on the error at the entire **image level**. The error expression corresponds to the classic Euclidean distance measure.

Another class of applications is the one where it is very important to get an upper limit on the error at each pixel. Now the criteria to be minimized is:

$$E_2(R, O) = \max |R_{ij} - O_{ij}|$$

where  $R_{ij}$  and  $O_{ij}$  have the same meanings and the maximum is searched also over the entire image. Here we are interested on the error at the **pixel level**. The error expression to be minimized corresponds to the classic Manhattan distance measure.

The first approach (based on  $E_1$ ) is the classic one and a lot of research has been done during years on that topic. The most popular method is the one based on DCT and stated in the JPEG standard. The main drawback of the scheme is that we can not have any guarantee about the error at pixel level. In some applications (satellite images, medical images, etc.), there is a need for some guarantees at pixel level because the artifacts introduced by the compression methods are unacceptable (without an upper limit). Therefore, in such area image compression was very rarely accepted (or not accepted at all).

In the last years the interest for such compression methods (based on  $E_2$ ) grows quickly and, therefore, the area requires a lot of research (Weinberger et al. 1998; Vleuten R.J., 2001; Yamauchi et al. 2001). Because usually we are interested only in small values for the acceptable error limit ( $\pm 1, \pm 2, \dots, \pm 10$ ) the method is called **near-lossless (NL)**. Certainly, if the acceptable error level become 0 we get the classical lossless compression.

It is very interesting to observe that such a small change, only in distance measure, will put us in front of a new area, all the old results (based on  $E_1$ ) being now irrelevant (for  $E_2$ ).

It is proven that the task of finding the best representation (and compression) within the acceptable error  $\pm k$  is a NP-complete task.

## 2. QUADRO PARTITIONING METHOD

One of the methods very popular in image processing is quadro (quadtree) partitioning (Salomon 1998). It consists in dividing and representing a square image area by its 4 quadrants. The process is repeated recursively until the square area is uniform and then is represented by that value. The main advantages of the method are that it is very simple (implementation is almost always recursive), it is adaptive to the image and the amount of

extra information needed to store the partitioning information (only 1 bit for describing a partitioning decision) is very low comparing to other partitioning methods. Even if its main application is adaptive partitioning the method is popular also as a representation (and compression) method.

If the initial image is not square the image is decomposed in squares and the method is applied individually on each of them.

### 3. LZW COMPRESSION ALGORITHM

LZW is the most popular algorithm from the LZ family. It was proposed by Lempel, Ziv and modified to this form by Terry Welch (Welch 1984). As in all dictionary-based methods, a dictionary of previously encoded strings is kept. The size of the dictionary is usually in the area between 512 (index represented on 9 bits) and 16384 (index represented on 14 bits). The first 256 entries are completed from the beginning with the 0-255 symbols.

Considering “x” being the current input stream character and I the current string the encoding process can be stated as follows:

```

I = empty
while not end of input
  read next character x
  if Ix exists in dictionary
    I = Ix
  else
    emits index of I to the decoder
    add Ix to the dictionary
    I = x

```

The decoder operates accordingly. The only difference is that, when receiving an index and adding Ix to the dictionary, the symbol x is, for the moment, unknown. It will be known as the first symbol of the next encoded string.

The main advantage of the LZW method over other LZ-family methods is that there is no need to transmit to the decoder anything else than the index of the string in dictionary (not even the new character). Even if it was not design specifically for image compression it can easily be used on images by considering the images a stream on bytes (for example, by scanning de image line by line).

### 4. NEAR-LOSSLESS EXTENSIONS

In order to use them in NL environments the previous methods have been adapted accordingly.

Regarding the QUADRO method we have to change the criteria used to stop the partitioning process. The original one, of stopping only when the entire remaining image square is identical, is replaced by the one of stopping

Accepted error (Err)	CR
0	0,968
1	1,035
2	1,216
3	1,482
4	1,839
5	2,308
6	2,677
7	3,107
8	3,511
9	3,982
10	4,472

Table 1 – Results for QUADRO method

when the difference between the maximum and the minimum values inside the image block is less or equal two times the accepted error. In that case the partitioning process is stopped and the entire block is represented by the mean value of the minimum and the maximum one.

Regarding the LZW method we have to search in the dictionary for the longest existing entry that meets the maximum accepted error requirements. Different from the original LZW method, now we can find more than one such entry. In that case a decision of choosing one of them has to be done. We select always the entry that, for the same maximum accepted error, presents the smallest mean error among the bytes of the entry. If there are still more such entries we pick the first one. The resulted method is labeled in the following chapters NL-LZW.

An interesting aspect is that, when adding an entry to the decoder’s dictionary, the real value of the next character is not known exactly (but only within the accepted error level). So, the coder and decoder dictionaries are not in synchronism! This is not really a problem (as it might look at the first sight) because the search process is done only at the coder where the dictionary is based on exact values. Certainly, at decoding, different data streams will be decoded (based on different dictionaries) but this is OK, the error is in the accepted level.

Even if LZW was previously used in a lossy purpose (Pigeon 2001) (the lossy aspect come only from restricting the dictionary search) this is the first attempt to use LZW as a near-lossless compression method.

### 5. EXPERIMENTAL RESULTS

In order to test the proposed NL-LZW method we have compared it to the QUADRO method. The tested methods are the ones described in the previous paragraph. Experiments have been done on the standard 8-bit grayscale 512\*512 LENA image. The accepted error is considered in the 0-10 domain (as usual in the near-lossless image compression papers).

For the QUADRO method we have considered the maximum size of the block being 16. Accepting a larger block size will overload the compressed stream with



Figure 1 Original test image LENA

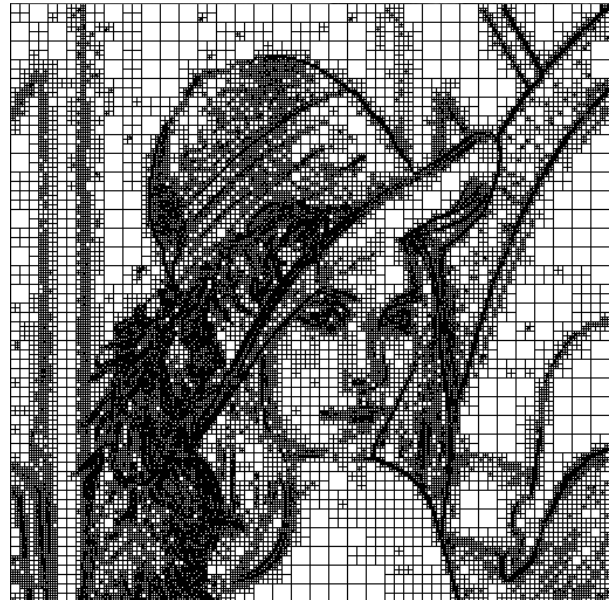


Figure 2 Quadro partitioning

partitioning bits without any CR improvement. Experimental results (not presented here) have proven that 16 is the best option to be taken. The values for the compression ratio (CR) obtained are listed in Table 1.

For the NL-LZW method the CR results are presented in Table 2. Regarding the dictionary size, the number of 512, 1024, 2048 and 4096 entries are considered (corresponding to their index representation on 9 to 12 bits).

To visually inspect the decompressed images we present in Figure 2 and Figure 3 the image partitioning and the decompressed image for the QUADRO method and in Figure 4 the decompressed image for the NL-LZW method. In both methods the level of accepted error considered is 10 (for smaller values the image is visually identical with the original and useless for direct comparison). The obtained CR's are almost the same (4.47 vs 4.50). Even in that case, when the CR advantage of our proposed NL-LZW method is gone, we notice that the decompressed image looks clearly better, without suffering for the block artifacts of the QUADRO method.



Figure 3 Decoded image for QUADRO method



Figure 4 Decoded image for NL LZW method

Accepted error (Err)	Dictionary size			
	9 bits	10 bits	11 bits	12 bits
0	0,977	0,999	1,089	1,165
1	1,187	1,412	1,459	1,542
2	1,478	1,653	1,796	1,900
3	1,784	1,986	2,153	2,287
4	2,041	2,273	2,448	2,589
5	2,376	2,616	2,800	2,921
6	2,523	2,867	3,037	3,189
7	2,789	3,177	3,364	3,597
8	3,000	3,422	3,623	3,782
9	3,277	3,682	3,937	4,177
10	3,482	4,155	4,274	4,507

Table 2 – Experimental results in CR for NL-LZW

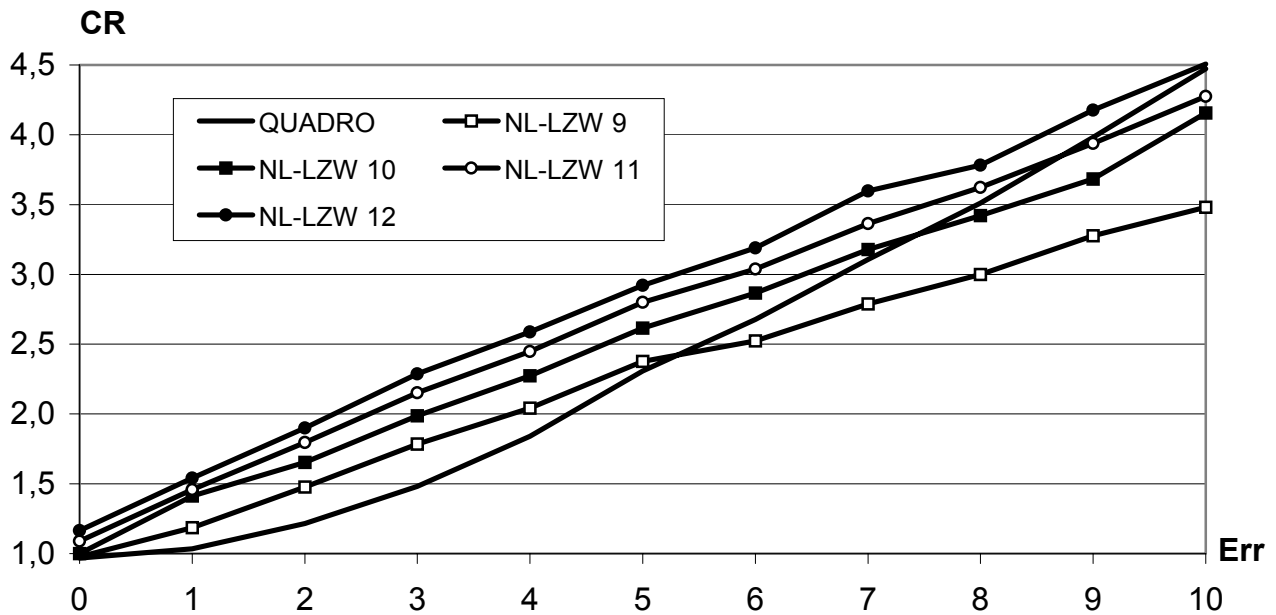


Figure 5 Rate-distortion curves obtained experimentally

In Figure 5 the corresponding rate-distortion curves are presented (CR is the compression ratio and Err represents the accepted error). We notice that:

- By increasing the size of the dictionary the NL-LZW method performances are always increased.
- The NL-LZW method always outperforms the QUADRO method for small level of the accepted error – where in fact the near-lossless methods are of maximum interest. For the 12 bits case the NL-LZW gives better for the entire tested area.

## 6. CONCLUSIONS AND FUTURE WORK

We have proposed a new method based on the use of the well-known lossless LZW method in a near-lossless compression scheme – as far as we know an original approach.

Results presented in the paper proven that the presented NL-LZW method outperforms the QUADRO method for small levels of the accepted error, exactly in the domain of maximum interest for near-lossless image compression. Even more, our method did not present the block artifacts that appear in the quadro-based method.

Certainly, further research has to be done, especially dealing with:

- Testing other solutions for the situation when the dictionary become full, i.e. clearing the entire dictionary or using an LRU-like policy for finding a replaceable entry in the dictionary. In our tests we freeze the dictionary when it becomes full.

- Optimizing the encoding program to allow us to test our method for larger dictionaries (i.e. represented on 13 bits or more), not done now due to computational requirements.
- Other decision that can be taken in consideration when choosing the best dictionary entry for the current input stream. The most interesting one is not to consider strictly the longest entry but the one that gives us (considering also the future data stream) the smallest number of strings (entries) coded.

## 7. REFERENCES

- Pigeon S., 2001 - An Optimizing Lossy Generalization of LZW, Data Compression Conference (DCC '01), Snowbird, Utah, USA
- Salomon D., 1998 - Data Compression, Springer Verlag
- Vleuten R. J., 2001 - Low-Complexity Lossless and Fine-Granularity Scalable Near-Lossless Compression of Color Images, Data Compression Conference (DCC '02), Snowbird, Utah, USA
- Weinberger M., Seroussi G., Sapiro G., 1998 - The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS, Computer Systems Laboratory, HPL-98-193
- Welch, T.A., 1984 – A technique for high-performance data compression, IEEE Computer 17(6)
- Yamauchi M., Wakatani A., 2001 - A New Lossless Compression Scheme for Medical Images by Hierarchical Segmentation, Data Compression Conference (DCC '01), Snowbird, Utah, USA