Image Compression Based on LDPC Codes

Andrey Belogolovyi
Information Systems Security Department of Saint-Petersburg State University of Airspace Instrumentation,
Saint-Petersburg, Russia
andrew@vu.spb.ru

Abstract

This paper describes the code-based approach to the still image compression using vector quantization. Low-density parity check codes (LDPC) are considered as vector quantizers, and the procedure of LDPC decoding can be used as fast search of the quantizer. For special images the compression ratio is greater than compression ratio by JPEG algorithm with equal quality of reconstructed image.

***Keywords: Image compression, Vector quantization, Coverings, Error-correcting codes, LDPC codes, Decoding*.**

1. INTRODUCTION

Many image lossy compression schemes use splitting of the whole images into domains (small parts), and then every domain is compressed. The common methods often use some spectral transformation and then – scalar quantization. The term "scalar quantization" means that the inputs are the separate values, and as the result the every image domain is processed separately.

Usually natural images contain a lot of similar regions and, therefore, there are a lot of similar domains after the splitting image into domains. It seems that usage of domains similarity properties can give compression gain if during processing the domains are considered as the whole blocks, not as the set of separate pixels.

Usage of domains similarity properties during image compression is not novel at all, and the key function of it is the vector quantization.

Vector quantization (VQ) is a procedure of "mapping" all possible input data sets into a relatively small set of symbols. In contrast to a scalar quantization, when the operation is performed with individual pixels, the vector quantization operates with groups of pixels, called vectors, so the results of applying any kind of vector quantization could be different with changing the information source. There are a lot of works dedicated to vector quantization in image compression [1-4,7].

2. VECTOR QUANTIZATION FOR IMAGE COMPRESSION

The common method of image compression using vector quantization is the following: the whole image is split into domains. The domains are considered as vectors or points in multidimensional space and then, during lossy compression, the similar domains are replaced with single pattern sample called codeword. All the patterns (codewords) are stored in codebook.

Traditionally the generalized Lloyd algorithm (GLA) [4,5] is used for vector quantization. As the result, for every vector (domain) of given image, GLA finds the codeword that is the best approximation of the domain considering given restrictions, also the codebook is built, i.e. the set of codewords that were used for given image quantization. Vector quantization itself provides $\frac{NL}{\log(K)}$ times compression, where N is the vector length (number of pixels in domain), L is the number of bits per one element in vector, K is the number of codewords in codebook. The codebook size is not considered here.

Clearly, after the vector quantization the reconstructed image will differ from original, and the quality loss (quantization errors) will be affected not only by the compression ratio, but also by the codebook vectors set itself. I.e. for one image given codebook will give good results in quality, but for another image the same codebook will lead to unacceptable distortions. That is why the codebook generation is the key task in schemes of vector quantization for image compression, and traditionally the different codebooks are built for different images. To make the image reconstruction possible the codebook ought to be stored or transmitted with the quantized image.

The codebook takes a lot of bytes by itself, and it contains a significant part of the whole number of bytes that ought to be stored after image compression. It means that considering vector quantization for image compression the compression ratio can be increased if it would be possible not to store the codebook or use the same codebook for quantization of all possible images.

3. CODE BASED APPROACH TO VECTOR QUANTIZATION

Lets say that the domain a of size $n \times n$ is covered in the code W with radius R, if there is $w \in W$, such that the Euclidean distance between w and a is not greater than R: $\sum (w_i - a_i)^2 \le R$. Then every code can be considered as the covering set for the given domains space with radius R, and the vector quantization process can be considered as the mapping of the domains into the codewords $w \in W$ (see fig. 1).

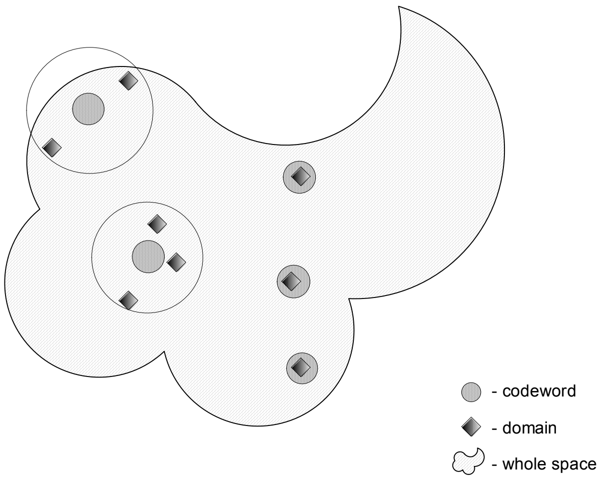

Figure 1: Adaptive covering for selected points in space

For codebook that was built with GLA the covering radius R will be minimal, because the code is built especially to cover only selected image domains and not to cover the whole space. But that code has no special properties, and that is why it cannot be compactly stored or transmitted.

If one attempts to use special structured code that covers the whole space the covering radius becomes significantly larger (see fig.2), but the structure properties of the can be used and the code could be transmitted compactly or could not be transmitted at all after the vector quantization of the image with given code.

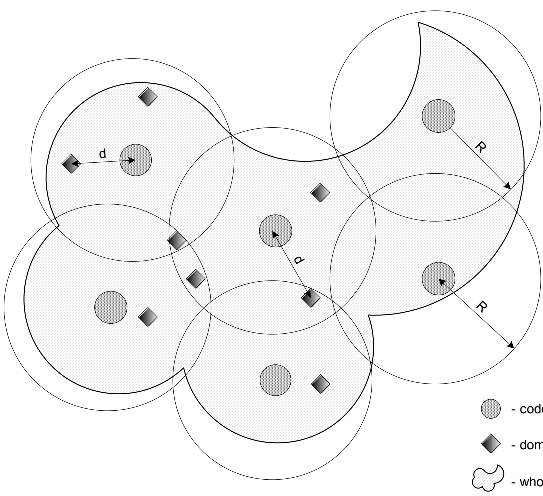

Figure 2: Using of code that covers the whole space

Let G is the linear (n,k)-code [6] over $GF(q)$, $q = 256$ in case of grayscale images (q is equal to luminance range). Let n_d be the domain side, $n_d \cdot n_d = n$.

If the points of the covered space are the domains of given image, then the vector quantization of the image with the code G can be done as follows: every domain that consists of n elements will be replaced with codeword of the code G that has the minimum Euclidean distance to original domain. From the error-correcting codes theory [6] it is known that the codeword is uniquely defined by its information set, in our case the information set has the length k, so it is enough to store k elements of the codeword to reconstruct the whole codeword later. As the result instead of storing n elements only k elements should be stored, and every domain compression ratio would be n/k (future possible entropy coding is not considered here). The main advantage of is the fact that the code G is not transmitted or stored at all. The case of using Reed-Solomon codes (RS-codes) as covering codes for image compression is described in [7]

In spite of the fact that in case using of error-correcting code for image quantization there is no need to store the given code, the overall compression is significantly less than compression with the adaptive code construction using GLA, because considering the equal quality of the reconstructed images quantized with RS-codes and with adaptive codes made by GLA the adaptive codes have significantly smaller dimension. Therefore it seems that the anyway some king of adaptation is necessary, but not only the code can be adapted to match the image. The image can also be adapted to the given code.

4. ADAPTATION OF THE IMAGE TO THE CODE

Consider the arbitrary space C. Let W is the code that covers C with radius R. It means that for any point of C there is a codeword from W whose Euclidean distance to original point is less than or equal to R: $\exists w \in W: \quad \forall c \in C, \quad d(c,w) \le R$, where $d(c,w) = \sum (w_i - c_i)^2$.

Let C' is the set of points that are the real domains of the given image, let M is the set of spatial transformations over C' with the following property: the result of transformation always belongs to the space C, $\forall c' \in C', m \in M : c = mc', c \in C$.

It means that any point from space C' can be transformed into the point from C, where it can be covered by the code W with the radius R, so the transformations can "adapt" the image to the code used (see fig.3).

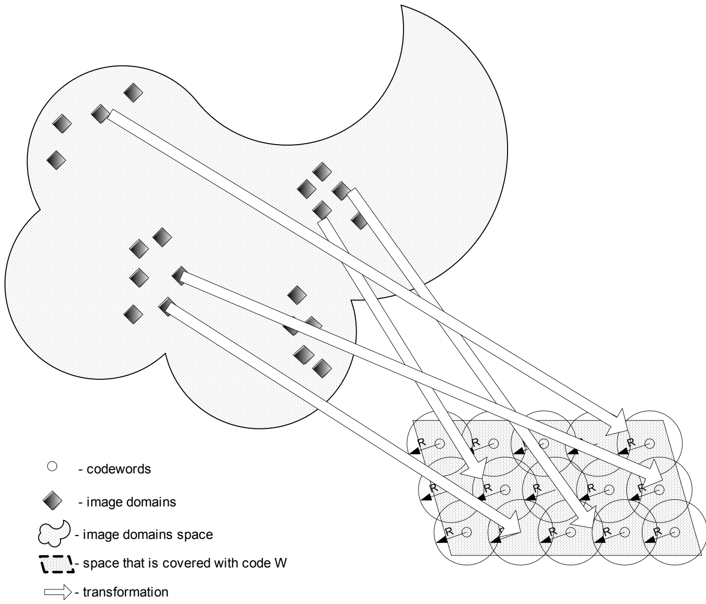

Figure 3: Transfer to code covered space using spatial transformations

5. USING THE PROPERTIES OF ERROR CORRECTING CODES IN QUANTIZATION

During vector quantization of the image the task of finding the nearest codeword for every image domain should be solved. In general case this task can be solved by exhaustive search when all the codewords are consequently tested. It was assumed that the code used has special properties, they can be used to speed up the searching procedure.

By analogy with the error correction codes theory [6] definitions any point of the space C can be considered as the codeword a with added error vector e: $\forall c \in C : c = a + e$. The weight of the error is the distance from the given vector to the zero vector (vector that consists from only zeros): $wt(e) = d(e,0)$.

Then the task of finding the codeword that has the distance to the given point less than or equal to R can be considered as the task of finding the error of weight less than or equal to R, which satisfies the equation $e : a = w + e, w \in W, wt(e) \le R$.

In terms of error correcting codes theory this task is called the task of decoding the code W in radius up to R, that is equal to finding the codeword $w \in W$, which satisfies the equation $w = c - e, wt(e) \le R$.

Now the common method of image vector quantization with error correcting codes can be described as follows:

1. Every domain x_i of the given image is mapped with transformations into the point c_i of the space C, space C should be covered by the code W.
2. In space C the decoding of vector c_i in code W is performed, the result of decoding is the codeword w_i, and the w_i is considered as vector quantizer for the given image domain x_i.
3. The information set γ_i of the codeword w_i is stored. The compression of vector quantization is the storing of the message set γ_i instead of storing the whole domain x_i.

However, the adaptation of the image to the code can be done not only via transformations.

Consider the splitting of the image domains into bitplanes. When one quantizes the bitplanes one-by-one, it is clear that the quantization errors in high-order planes leads to largest errors in reconstructed image. But it is necessary to take into account the fact that after processing the high-order bitplane the other bitplanes are to be quantized, and some of quantization errors of the high-order plane can be minimized during low-order bitplanes quantization.

Example 1. Let the original image contains two different points with brightness of 255 and 128. The high-order bitplane for both points contains “1”. The quantization error in this plane leads to equal overall quantization error with value 128, but the quantization errors in other bitplanes can increase the overall error value in first case and decrease the error value in second case. In both cases if the high-order bitplane contains the error, in the best possible quantization of the other bitplanes sets the reconstructed point brightness to the value of 127, that is very close to original value in second case.

The Example 1 shows that during quantization it is necessary to take into account the influence of quantization error in given bitplane to the possible overall error. The bits in bitplanes can be separated into two groups: one group allows the quantization error, and other group does not allow quantization errors at all. It means that if one uses the decoding of error correcting code as quantization procedure, it is necessary before decoding to specify bits allowing error and bits that should be decoded correctly.

To satisfy the conditions described the LDPC codes [7] can be used.

6. IMAGE QUANTIZATION WITH LDPC CODES

Low density parity check codes [7] are the linear codes that have the simple decoding procedure using reliabilities [7]. In case of image quantization the importance of the bits inside bitplanes is similar to reliabilities of received bits in terms of error correcting codes theory, so the decoding using reliabilities can be used as the procedure of fast finding the best quantizer for the given bitplane of domain.

Let $\{X_i\}$ be the set of brightness values of the pixels of given image, let $\{W_i\}, 1 \le i \le N$ be the set of LDPC codes, N is the number of bits required for binary representation of the brightness $\{X_i\}$. Then the algorithm of quantization with setting reliabilities will operate as follows:

1. At the first quantization stage (first, high-order bitplane) the set $\{x_i^1\}$ is formed, where x_i^1 are the high-order bits of X_i, and the set of reliabilities $\{l_i^1\}$ is made by the rule $l_i^1 = (X_i - 2^{N-1} + x_i^1)^2$. From pairs (x_i^1, l_i^1) the input vectors for the LDPC decoders are made, and then the decoding using reliabilities is performed in code W_1. The result of the decoding is the vectors consisting of $\hat{x}_i^1$, the quantized values.
2. At every following quantization stage k, $2 \le k \le N$ (following bitplanes) , the differences $r_i^k = X_i - \sum_{m=1}^{k-1} \hat{x}_i^m \cdot 2^{N-m}$ are computed and the set $\{x_i^k\}$ is formed, where $x_i^k = \begin{cases} 1, r_i^k \ge 2^{N-k} \\ 0, r_i^k < 2^{N-k} \end{cases}$, and for every x_i^k the reliability is computed as $l_i^k = (r_i^k - 2^{N-k} + x_i^k)^2$. From pairs (x_i^1, l_i^1) the input vectors for the LDPC decoders are made, and then the decoding using reliabilities is performed in code W_k. The result of the decoding is the vectors consisting of $\hat{x}_i^k$, the quantized values.

After completing all N stages all the quantized values $\hat{x}_i^k$ are obtained and they are considered as the new bitplanes of the given image.

The selection of LDPC codes parameters to be used at different quantization stages can be done in accordance to the described fact of unequal meaning of errors at different stages. The simplest way of code selection is the following: at every quantization stage the code used should have the rate less than or equal to the rate of code used on the previous stage: $\frac{k_1}{n} > \frac{k_2}{n} > \ldots$ The best results were achieved using LDPC codes (16,8), (16,8), (16,8), (16,8), (64,14), (256,29), (256,29), (256,29) at the corresponding stages. Using the codes described the standard test image "Lena" (grayscale, 512x512) was quantized with the average ratio 2.57:1 (there was no entropy coding of quantized data) and the SNR of reconstructed image was 23.61 dB (PSNR = 30.25 dB).

7. CONCLUSION

The results of the image compression method was considered in this paper are different for different domain type, also currently it is hard to achieve the high quality of reconstructed images. However the method of image compression based on LDPC codes can be used not only as single compression method but also it can be used as the prediction method for any other compression algorithm, for example, JPEG.

Let's consider the situation when the minimal possible quality of reconstruction image is fixed. Then it is possible to implement the following scheme:

1. All image domains are compressed with the standard JPEG algorithm that provides the predefined quality.
2. All image domains are compressed with the LDPC based vector quantization and then the difference between the original image and the reconstructed one is compressed with difference JPEG algorithm to achieve the predefined quality. If the total number of bits required is greater than number of bits required for standard JPEG, then the domain is still compressed with standard JPEG.

The using of LDPC code based compression as the predictor for difference JPEG on some domains gives a compression gain, and the best compression gain was achieved on texture domains (see table 1).

Filename	Percentage of domains that give compression gain when LDPC code based quantization was used as the prediction procedure	Overall compression gain in bytes	SNR
airplane.bmp	4.37 %	500	37.25
arctichare.bmp	0.07 %	0	41.84
baboon.bmp	51.10 %	16500	33.02
barbara.bmp	14.72 %	3000	33.42
boat.bmp	21.00 %	4000	33.38
cat.bmp	10.25 %	1800	35.35
fruits.bmp	4.83 %	800	35.29
lena.bmp	5.27 %	800	32.19
peppers.bmp	4.52 %	800	32.89
pool.bmp	0.76 %	0	30.96

Table 1. Results of using the LPDC code based quantization as the predictor for the difference JPEG

The best results were achieved on the test picture "baboon.bmp" due to lot of texture-like domains. The figure 4 is the standard test image "Baboon" after removing all domains 8x8 pixels on which there are no compression gain by using LDPC code based quantization as the predictor for difference JPEG.

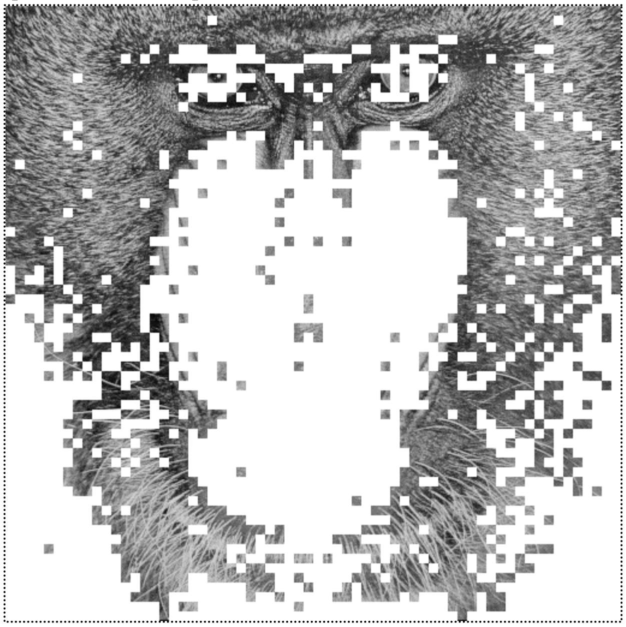

Figure 4: Test image "Baboon": the domains that give compression gain when LDPC code based quantization is used as the predictor

The scheme of joint using of LDPC code based quantization and the standard and difference JPEG can be used in very high quality image compression applications like DVD-compressors etc.

8. REFERENCES

[1] Gray R. M. *Fundamentals of Vector Quantization. http://www-isl.stanford.edu/~gray/compression.html*

[2] Cosman P. C., Oehler K. L., Riskin E. A., Gray R. M. *Using vector quantization for image processing. // Proc. of the IEEE, vol. 81, no. 9, September 1993, pp. 1326-1341*.

[3] Bayazit U., Pearlman W. A., *Variable-Length Constrained Storage Tree-Structured Vector Quantization. IEEE Trans. Image Processing, vol. 8 no. 3, March 1999, pp 321-33*1.

[4] Gersho A., Gray R.M., *Vector Quantization and Signal Compression, Kluwer Academic Publishers, 1992*.

[5] Garey M. R., Johnson D. S., Witsenhausen H. S., *The complexity of the generalized Lloyd-Max problem. // IEEE Trans. Inform. Theory, vol. 28, no. 2, 1982, pp. 255-256*.

[6] MacWilliams F.J., Sloane N,J,A., *The Theory of Error correcting codes, Bell Laboratories, Murray Hill, NJ 07974, U.S.A., 1977*.

[7] Belogolovyi A., *Application of Error Correcting Codes for Image Compression. // Proceedings of Second International School-seminar BICAMP'99,Saint-Petersburg, 1999, p. 119*.

[8] Gallager R. G., *Low-density parity-check codes. // IEEE Trans. on Inform. Theory, vol. IT-8, Jan. 1968, pp. 21–28*.

About the author

Andrey Belogolovy is an assistant professor at Saint-Petersburg State University of Airspace Instrumentation, Information Systems Security Department. His contact email is andrew@vu.spb.ru