University of Brescia
School of Engineering
Master Course in Communication Technologies and Multimedia

Multimedia Communication Services

Laboratory Experience, No.9

Students:
Casali Nicolò
Getahun Kebede
Ghitti Alberto
Leali Alberto
Lombardi Stefano
Rocco Davide
Tigist Getachew

Academic Year 2013/2014

Contents

List of Figures

Abstract

Laboratory Experience, No 9

In this experience we focused on the generalized Lloyd-Max algorithm in order to implement a vector quantization scheme for images.

After applying the algorithm, we have reconstructed the image at each iteration, using a different and improved codebook for each iteration and also computed the reconstruction error. We have repeated the procedure many times with different initialization conditions.

Vector Quantization: Theoretical introduction

The Lloyd-Max algorithm consists of several step:

1. Get a set of training vectors U_k (we use 8x8 pixel blocks) to be quantized vectorially with a dictionary of $L = 128$ words;

2. Set L initial codewords $w_i^{(0)}$ to form $C^{(0)}$;

3. For each u_k, it has to be found which codeword minimizes the distance block-codeword. Therefore each codeword matches with many blocks u_k. This set of blocks represents the region r_i ;

$$r_i^{(j)} = \{\ u_k\ \mid\ d(u_k, w_i^{(j)}) < d(u_k, w_l^{(j)}), i \neq l \quad \text{or} \quad d(u_k, w_i^{(j)}) = d(u_k, w_l^{(j)}), i < l\ \}$$

4. Now it has to be compute a new value of $w_i^{(j+1)}$ as the center of mass of $r_i^{(j)}$ (using blocks matched in that region);

$$w_i^{(j+1)} = \frac{1}{|r_i^{(j)}|} \sum_{u_k \in r_i^{(j)}} u_k$$

5. Continue to step **3)** as long as $w_i^{j+1} = w_i^{(j)}\ \forall i$ or the distortion stops decreasing "significantly".

The Figure 1 shows two reconstructed images.
The first image is quantized using the initial random codebook and the second one using the codebook computed by the algorithm at the first step. It is easy to see the ability of the algorithm in the codebook optimization.

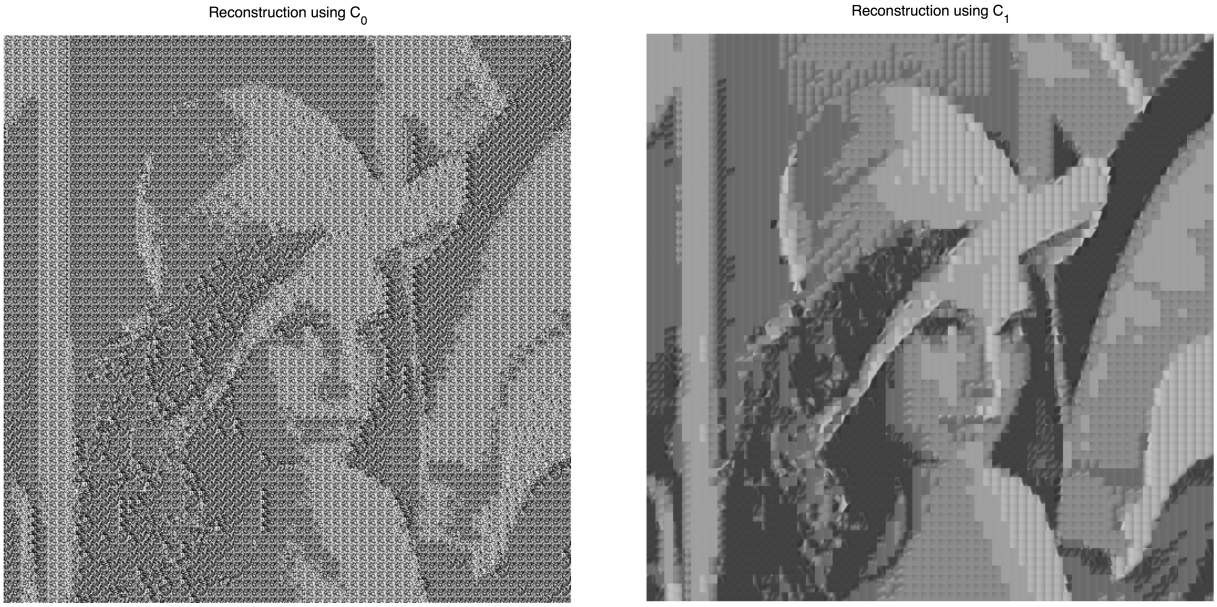

Figure 1: Codebook optimization algorithm.

Random Initial Codebook

In the first part of this laboratory we have set the initial codewords generating them randomly over R^N, using the same block dimension of the image (8x8 pixels).
Iterating the procedure, the reconstructed image converges to the original one (2).

Figure 2: Iteration of reconstructed images using an random initial codebook.

It is possible to see significant changes in the first three step and then the convergence becomes slow.
This is more clear looking the energy of the residual image for each step (3).

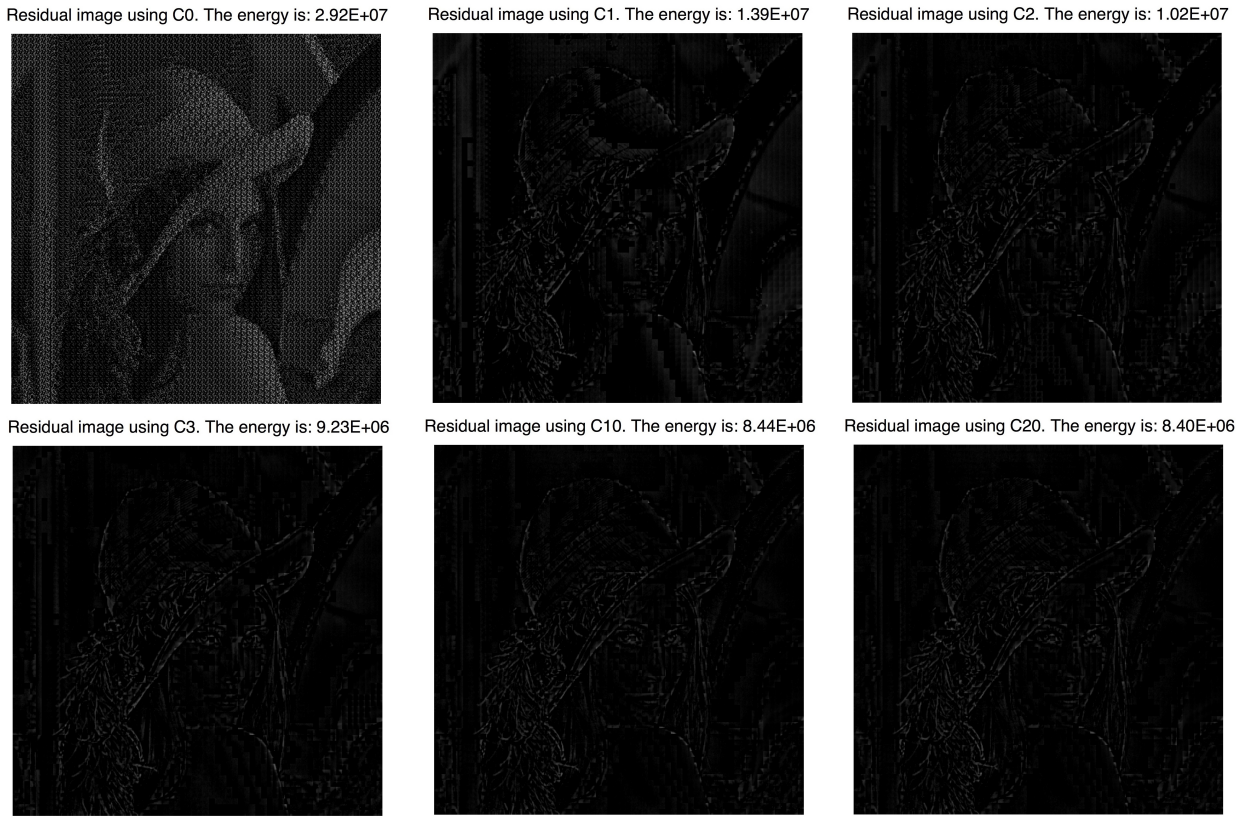

Figure 3: Residual image and relative energy for each iteration, using an random initial codebook.

Sensible Initial Codebook

Using a smarter initial codebook, it is possible to reach better results.
The Figure 4 shows the reconstructed image using 128 blocks of the original image like initial codebook.

Figure 4: Iteration of reconstructed images using a sensible initial codebook.

The reconstructed image is more similar to the original one, also in the first step, with respect to the previous part of the experience.
Looking the residual images (5) it is possible to see that the error energy is lower than the previous case at each iteration.
This big difference between the two initial codebooks is due to the fact that the random one has some codewords far from all blocks of the original image and they are not used for the computation. This means that the quantization isn't done with 128 words.

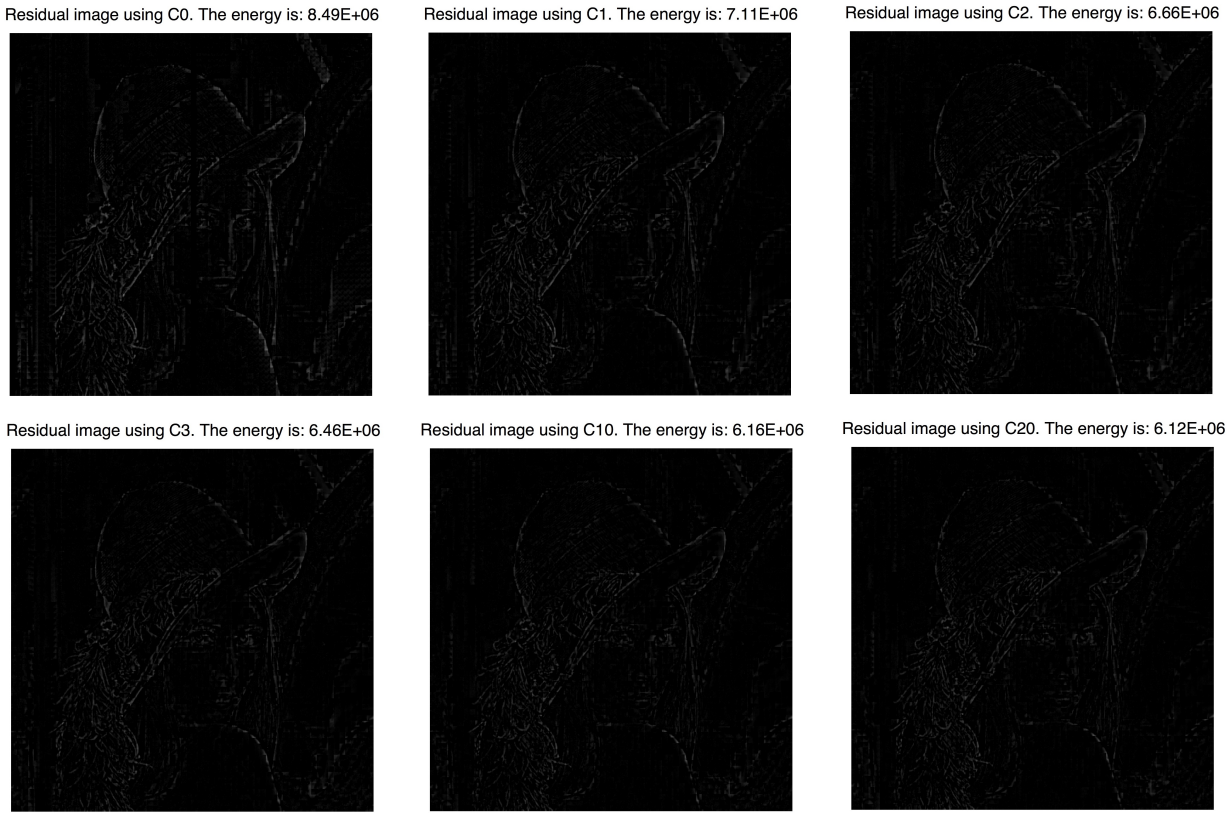

Figure 5: Residual image and relative energy for each iteration, using a sensible initial codebook.

This fact is shown in the Figure 6 where it is possible to see the codeword distributions used for 1^{st} and 20^{th} steps.
While a codebook took from the original image guarantees that no codewords are far away from the image. So all codewords is used at least one times.
The distribution shows that there are not unused codewords and this means that the quantization is really on 128 words codeword (7).

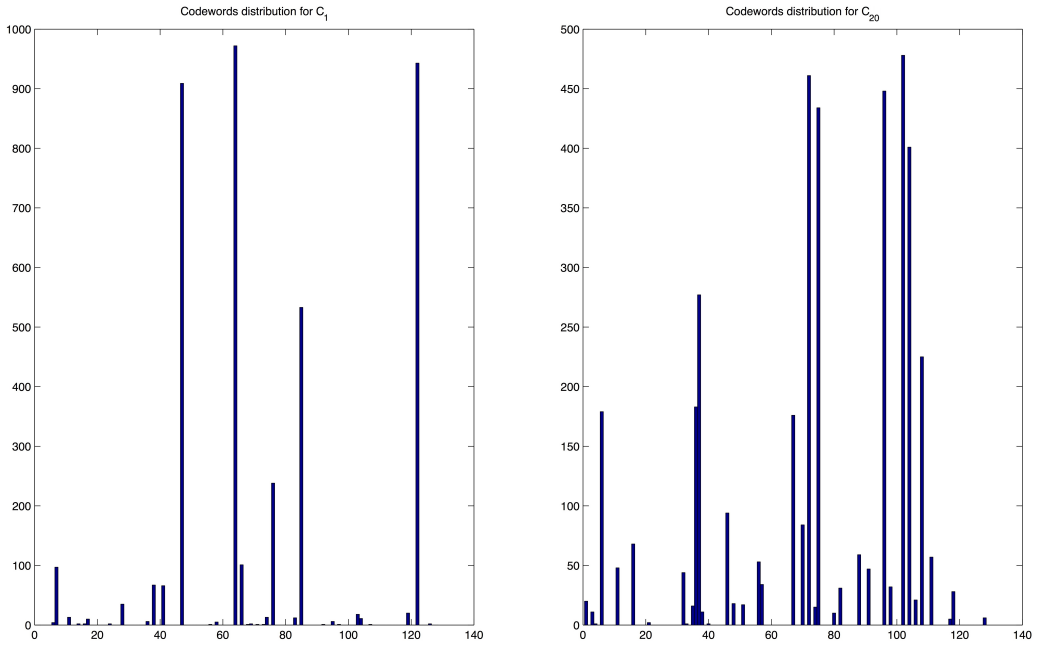

Figure 6: Codeword distribution for 1^{st} and 20^{th} steps, using a random initial codebook.

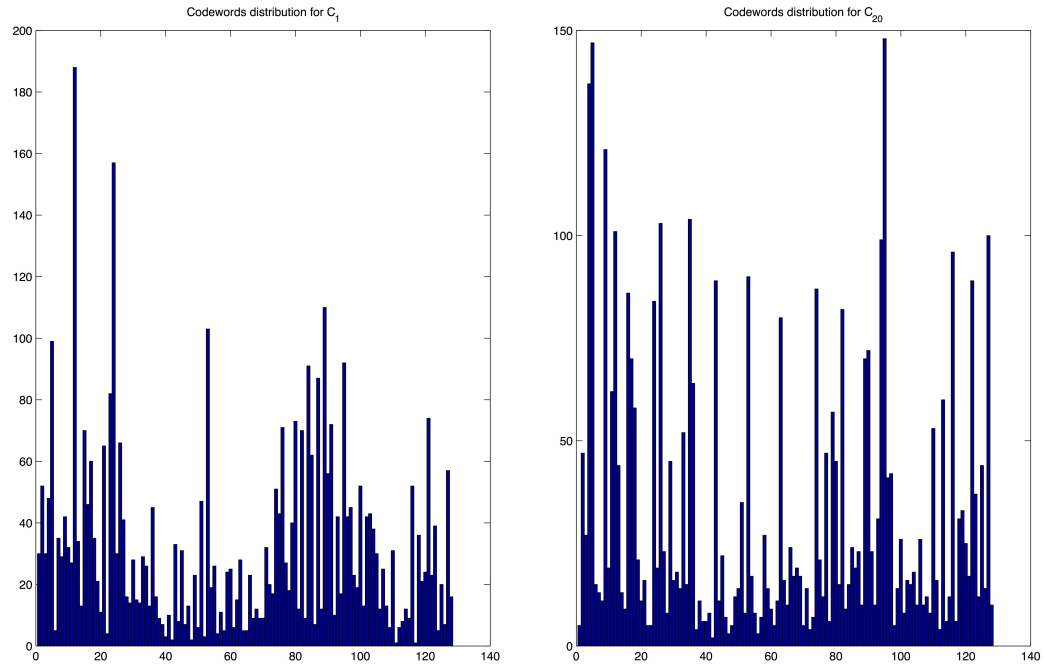

Figure 7: Codeword distribution for 1^{st} and 20^{th} steps, using a sensible initial codebook.

Conclusion

This algorithm had shown the ability to reach the convergence starting from each possible codebook but doing a smart initial choice it's possible to use all codewords and reach lower error in few step.
This problem can be solved adding a control step that reuses in some way the unused codewords but the complexity could increase a lot.