ENTROPY CONSTRAINED OVERCOMPLETE-BASED CODING OF NATURAL IMAGES

Andre Filgueiras de Araujo, Maryam Daneshi, Ryan Peng

afaraujo, mdaneshi, rppeng@stanford.edu
Stanford University

ABSTRACT

We introduce an Entropy-Constrained Overcomplete-Based coding scheme for natural images. The traditional overcomplete-based framework for compression is improved in its main components. The main contribution of the work is a new dictionary learning algorithm for overcomplete-based compression, referred as Entropy-Constrained Dictionary Learning. We show that the presented scheme outperforms a basic DCT coder with gains of up to 2 dB.

1. INTRODUCTION

Recently, there has been a wide interest in sparse coding of signals using overcomplete dictionaries. The term *overcomplete* refers to a set of basis vectors with a size bigger than the one needed to span the data subspace. For a signal in $\mathbb{R}^N$, such an approach employs K basis vectors, with $K > N$ (instead of N). The objective of such methods is then to have more basis vector options to select from and therefore a better chance of finding a smaller number of basis whose linear combination matches the signal vector. Consequently, at the entropy coding stage, we expect that fewer bits will be needed to code the resulting coefficients.

In usual lossy compression, with the help of transform coding, the energy of the signal vector gets compacted in few coefficients. Quantization follows, setting many of the coefficients to zero, being the only stage of the pipeline which introduces distortion. On the other hand, in the overcomplete approach, another lossy stage is introduced: the system being underdetermined, we usually can only get an approximation of the signal to represent, due to the constraints imposed on finding the solution. However, we will see that these two stages can be actually integrated in only one, in which we trade-off distortion and complexity in a Lagrangian cost framework. As we show in this work, that's the key idea of using the overcomplete framework for compression.

Some work has been done on the compression of images using overcomplete dictionaries [1, 2]. They compare favorably in some cases with respect to a basic DCT/JPEG framework. Nevertheless, the better performance is achieved by a restriction on the nature of the data: in [1], for example, the application of facial images is considered. In this work, we analyze the use of overcomplete dictionaries for compression of natural images.

There are two stages in a general overcomplete coding method: *Sparse Coding* and *Dictionary Learning*. Sparse coding assumes an overcomplete dictionary is known and seeks the best representation of the input signal. Dictionary learning is the process that builds an overcomplete dictionary.

Knowing the overcomplete dictionary, the sparse coder finds a set of bases among all the dictionary vectors to represent the given signal. That is to say, given an overcomplete dictionary $A \in \mathbb{R}^{N \times K}$, the sparse coder outputs a representation of the input signal vector $y \in \mathbb{R}^N$ as the vector of coefficients $x \in \mathbb{R}^K$, such that $y \approx Dx$ or $\|y - Dx\|^2 \leq \epsilon$.

Dictionary Learning works by training a dictionary based on a set of examples. That is, for a given input set with P samples $Y = y_{i=1}^P$, it searches the dictionary A which will lead to the best representation of Y using a set of coefficients $X = x_{i=1}^P$.

In this work, we propose a compression scheme using the overcomplete approach. This comprises a proposed entropy-constrained dictionary learning method and a rate-distortion-optimized sparse coding method, which was initially envisaged in [3].

After briefly introducing the basic sparse coding methods in section 2, we present a rate-distortion-based sparse coding algorithm. In section 3, we describe general dictionary learning techniques, discuss their properties and propose an entropy-constrained dictionary learning method. In section 4, we present simulation results of the proposed compression scheme for a set of natural images and compare it to a basic DCT-based framework as well as to JPEG and JPEG2000 standards.

2. SPARSE CODING

As previously discussed, sparse coding is the process of finding the best representation of a given signal based on a known dictionary. The problem of finding the optimal set of basis vectors from a dictionary, according to a certain constraint, is known to be an NP-hard problem [4], so approximate solutions are needed. One of the highly used greedy optimization techniques for solving this problem is Matching Pursuit

(MP) [5]. We will briefly describe MP and some variants of it in the following subsections.

2.1. Matching Pursuit

Matching Pursuit handles the problem of finding the best coefficient vector x for the given input signal y and dictionary A through a greedy iterative process. At each iteration, MP projects the residual r (initially, $r = y$) on all basis vectors of A. It chooses the basis that gives maximum projection. The projection result is then the coefficient, which is added to the current vector x. The residual is updated as $r = y - Ax$, and this process is repeated by projecting the new residual and picking one new nonzero coefficient in each step. Two stopping criteria are commonly used for MP: 1) a predefined number of non-zero coefficients is reached, or 2) the norm of the residual achieves a predefined threshold ϵ.

At each stage of MP, the orthogonalization of the residual with respect to each new selected dictionary component can introduce components of previously selected basis vectors into the residual. To avoid that, Orthogonal MP (OMP) has been proposed by [6]. In OMP, the least-squares approximation of y using all currently selected vectors is found. The residual is calculated as before, with this new set of coefficients (the previously selected coefficients also change when adding a new nonzero coefficient). Compared to MP, this method has shown to result in a smaller number of coefficients for a fixed ϵ. OMP is the most employed algorithm for sparse coding in the literature. In this work, we employ OMP as the basic sparse coding algorithm, due to its simplicity and good performance.

2.2. Rate-Distortion OMP

MP and its variants have been highly used for sparse coding. The selection process of these, however, is based on the minimization of a simple distortion cost. For compression purposes, it is know that a better approach is to also consider the rate cost of the selections, with the widely used Lagrangian cost $J = D + \lambda R$ being its standard formulation. To deal with this issue, Rate-Distortion OMP (RD-OMP) was first suggested in [3] (however, to the best of our knowledge, it was not used in recent overcomplete-based compression schemes). In such a scheme only a subset of the coefficient vector has nonzero value (only these will be coded); therefore for every block, the rate is calculated as: $R_{block} = R_{ind} + R_{coeffs} + R_{EOB}$. In each block, we need to send the indexes of the nonzero coefficients (R_{ind}), their values (R_{coeffs}), and some control bits to indicate the end of the block (R_{EOB}). As R_{EOB} is naturally fixed, it doesn't need to be taken into account in the sparse coding process. RD-OMP selection process stops when the improvement on the overall J cost is very small (no need to impose a predefined number of nonzero coefficients). We will employ RD-OMP in the proposed compression scheme, and discuss its performance in section 4.

3. DICTIONARY LEARNING

Most of the dictionary learning algorithms in the literature are a generalization of the K-means algorithm used in vector quantization training [7]. The learning approach is performed in a two-step process. The first step finds the coefficients for a given dictionary (sparse coding). Then, in the second step, given the coefficient results of the first step, the dictionary gets updated. This process is repeated until a stopping criteria is reached. Among many learning methods, we have chosen K-SVD and MOD algorithms because they are the most widely employed algorithms and have shown to provide good results with a simple framework. In the following subsections we will briefly introduce these algorithms and, based on them and Entropy-Constrained Vector Quantization (EC-VQ), propose the "Entropy-Constrained Dictionary Learning" method.

3.1. K-SVD

K-SVD [8] is one of the highly deployed dictionary learning algorithms. K-SVD has two main parts, the sparse coding stage and the dictionary update stage, as shown in Algorithm 1.

Algorithm 1 K-SVD

Initialization: Set the dictionary matrix $A^{(0)} \in \mathbb{R}^{N \times K}$ with l^2 normalized columns
Set $n = 1$
Repeat until stopping rule is satisfied:
1.Sparse Coding: Compute the coefficient vectors x_i for each y_i, by approximating the solution of:
$\min_{x_i} \|y_i - Ax_i\|^2, i = 1, 2, ..., P$ subject to
$NNZ(x_i) \leq T_0$

2.Dictionary Update: For each basis vector $k = 1, 2, ..., K$ in $A^{(n-1)}$
2.1 Define the group of input vector that uses this basis vector, $w_k = \{i \mid 1 \leq i \leq P, x_T^k(i) \neq 0\}$
2.2 Compute the overall representation error matrix $E_k = Y - \sum_{j \neq k} d_j x_T^j$
(x_T^k: vector of coefficients that uses basis k)
2.3 Restrict E_k to the input vectors corresponding to w_k, and obtain E_k^R
2.4 Apply SVD decomposition $E_k^R = U \Delta V^T$.
2.5 Choose the updated dictionary basis vector $\tilde{d_k}$ to be the first column of U. Update the coefficient vector x_R^k to be the first column of V multiplied by $\Delta(1,1)$

Set $n = n + 1$

In the sparse coding stage, any approximation pursuit method, i.e. OMP, can be used as long as the resulting solution satisfies the constraint of fixed and predetermined number of nonzero coefficients, $NNZ(x) \leq T_0$. It is worth

mentioning that these algorithms can be adapted to use an error-based stopping criterion.

In the "Dictionary Update" stage of K-SVD, the multiplication AX can be decomposed as the sum of K rank-1 matrices. $K-1$ of these terms are assumed to be fixed while k^{th} term gets modified. The SVD decomposition finds the closest rank-1 matrix which approximates the term E_k. This operation minimizes the error $\|E_k - d_k x_T^k\|^2$. The SVD decomposition of E_k may violate the sparsity constraint so E_k^R is constructed based on w_k, the group of indexes of Y that uses d_k, i.e., the input samples that use the k^{th} basis vector are fixed (the nonzero elements of x_T^k remain in the same position), but the coefficient associated to each of these changes. As stated in the algorithm, the SVD decomposition will result in an updated dictionary vector $\tilde{d_k}$. The corresponding coefficient vector gets updated and the same process gets repeated for every basis vector of the dictionary.

3.2. Method of Optimal Directions (MOD)

The other highly used overcomplete dictionary learning approach in the literature is the Method of Optimal Directions (MOD) [9]. As shown in Algorithm 2, MOD follows the same sparse coding approach as K-SVD. However, in the dictionary updating stage, MOD assumes the coefficient vector x_i for the input vector y_i is fixed while the dictionary is updated. The overall Mean Square Error (MSE) is given by:

$$\|E\|^2 = \|e_1, e_2, ..., e_P\|^2 = \|Y - AX\|^2 \quad (1)$$

Taking the derivative of the above formula with respect to A (X and Y are fixed), $(Y - AX)X^T = 0$, results in the following dictionary update expression:

$$A^{(i+1)} = YX^{(i)^T}.(X^{(i)}X^{(i)^T})^{(-1)} \quad (2)$$

MOD derives the best possible dictionary adjustment (in a MSE sense) based on (2) for a fixed coefficient matrix X.

Algorithm 2 MOD

Initialization: Set the dictionary matrix $A^{(0)} \in \mathbb{R}^{N\times K}$ with l^2 normalized columns
Set $n = 1$
Repeat until stopping rule is satisfied:
Sparse Coding: Compute the coefficient vectors x_i for each y_i, by approximating the solution of:
$\min_{x_i}\{\|y_i - Ax_i\|^2, i = 1, 2, ..., P$ subject to $NNZ(x_i) \leq T_0$

Dictionary Update: Given the input data Y and coefficient matrix $X^{(n)}$, update the dictionary as
$A^{(n+1)} = YX^{(n)^T}.(X^{(n)}X^{(n)^T})^{(-1)}$

Set $n = n + 1$

3.3. Entropy-Constrained Dictionary Learning

We propose an Entropy-Constrained Dictionary Learning (EC-DL) algorithm which employs a more appropriate scheme in the context of image compression. This is the main contribution of this work. As shown in Algorithm 3, in addition to "Sparse Coding" and "Dictionary Update", EC-DL introduces a third stage called "Rate Cost Update". We will discuss each stage of the algorithm in more details. EC-DL operates based on a rate-distortion Lagrangian cost and stops the iterative learning process if the cost doesn't significantly decrease any further.

As discussed in previous subsections, most of the existing dictionary learning algorithms are designed based on distortion-based sparse coding. In section 2, we presented RD-OMP, an evolution of OMP using a Lagrangian rate-distortion framework. This is the algorithm that we use for the "Sparse Coding" stage of EC-DL.

Algorithm 3 Entropy Constrained Dictionary Learning

Initialization: Set the dictionary matrix $A^{(0)} \in \mathbb{R}^{N\times K}$ with l^2 normalized columns
Set $n = 1$, $J_0 = \infty$
Sparse Coding: Use RD-OMP, as presented in section 2.2.

Rate Cost Update: Update probability mass functions of coefficients and indexes. Estimate the rate cost of each of them using the relation $l_m = -\log_2 p(m)$

Dictionary Update: Given the input data Y and coefficient matrix $X^{(n)}$, update the dictionary as
$A^{(n+1)} = YX^{(n)^T}.(X^{(n)}X^{(n)^T})^{(-1)}$

Lagrangian cost update: $J^{(n)} = D^{(n)} + \lambda R^{(n)}$

if $\frac{J^{(n-1)} - J^{(n)}}{J^{(n-1)}} < \epsilon$ **then**
Stop the algorithm
else
Set $n = n + 1$
go to **Sparse Coding**
end if

Similarly to EC-VQ for training data, after determining the coefficient matrix, the probability mass functions of the non-zero coefficients and their corresponding indexes are updated in the "Rate Cost Update" stage. Based on these probabilities, the codeword length of the coefficients and the indexes are determined using $l_m = -\log_2 p(m)$.

In the dictionary update stage of K-SVD, as the dictionary gets modified, the coefficients will vary; so, it is not possible to control the rate in this stage, therefore reduction of the Lagrangian cost function is not assured. On the other hand, MOD updating method provides the optimal adjustment while preserving the coefficients. The latter is the employed

method in EC-DL. It is interesting to notice again the similarity with EC-VQ, which also assumes fixed coefficients while updating the representative levels (centroid calculation).

All the presented dictionary learning methods have non-guaranteed convergence. Since the performance of the "Sparse Coding" stage is not optimal, its resulting cost (Lagrangian or distortion only) might increase. We use a set of techniques to overcome this difficulty. First, we build the initial dictionary by picking vectors of the input matrix, Y, in equally spaced positions, so the dictionary is more representative of the input vectors (as opposed to the use of the first K input samples, which could lead to an unsatisfactory local minimum, especially for images). Second, we always keep the best dictionary found so far and update it to a newer one only if the latter is less costly than the former. Third, we allow the dictionary search to grow in cost up to some threshold because some oscillations might occur, and a better match might be found after them.

4. EXPERIMENTS

Our experiments consisted of simulations with the test images *Lena*, *Boats*, *Harbour* and *Peppers* of resolution 128×128. To provide a fair comparison, the entropy coding was performed identically for every method: using the optimal coder for each subband. We used blocks of size 8×8 as input data and quantization step, q, varying from 8 to 128 in all experiments. The λ factor was calculated using the expression $\lambda = 0.2q^2$. Three dictionary sizes (K) were employed: 128, 256 and 512. In the following subsections, we present results for each of the enhanced algorithms and for the overall compression scheme.

4.1. Sparse Coding comparison

We implement OMP and RD-OMP for a performance comparison. OMP is used with NNZ varying from 5 to 15. Results are provided in Figure 1, for the image *Lena*, with K fixed to 128. We observe that RD-OMP clearly outperforms all other algorithms for the entire bitrate range, with a performance gain of up to 2 dB. RD-OMP is employed in the schemes described in the following subsections.

4.2. Dictionary Learning comparison

In this experiment, K-SVD, MOD and EC-DL have been used to train overcomplete dictionaries. The dictionaries were all given as input to RD-OMP, and the results were compared. We observe that the proposed scheme outperforms the other two approaches, with gains of the order of 2 dB. Results for the image *Peppers*, with $K = 256$ are shown in Figure 2.

4.3. Compression scheme comparison

As shown in Figure 3, two compression schemes using EC-DL for dictionary learning and RD-OMP for sparse coding were employed. In scheme 1, for any input image a dictionary was generated from that same image and then used to perform sparse coding, followed by the entropy coding of the resulting coefficients. In such a scenario, the dictionary would need to be transmitted to the decoder, along with the coefficients. In scheme 2, as a more practical scheme, a set of training images was used to build an overcomplete dictionary. The resulting dictionary was used to encode the test images, which were not included in the training set. In this experiment, the set of training images consisted of 18 Kodak natural images downsampled to 128×128.

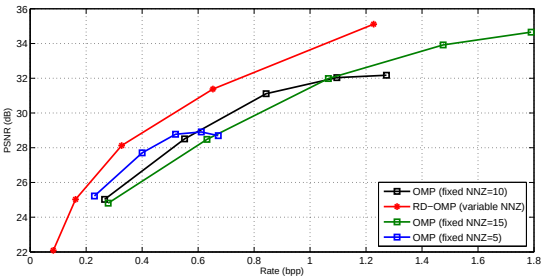

Fig. 1. Performance of RD-OMP vs. OMP, $K = 128$, image Lena

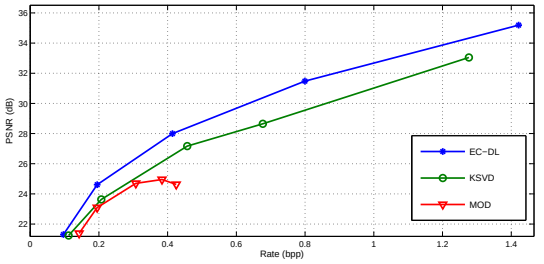

Fig. 2. Comparison of dictionary learning methods, $K = 256$, image Peppers

In Figures 4 and 5, we present the PSNR-rate comparison of images *Harbour* and *Lena* for the two mentioned schemes. We observe that for both images, scheme 1 outperforms other results. In the presented results, we are not considering the required bitrate for transmitting the dictionary to the decoder and these results can be considered as an upper bound of the performance which can be obtained by the use of a trained dictionary. According to our calculations, the gain in rate using this scheme (around 1 bit per pixel) is not enough to transmit the dictionary to the decoder, therefore this is not a practical scheme.

In Figures 4 and 5 we also present the results of scheme 2 with dictionary sizes of 128, 256 and 512. As the dictionary size grows, the performance of the encoding scheme increases. For a large enough dictionary, EC-DL will have up to 2 dB gain compared to DCT.

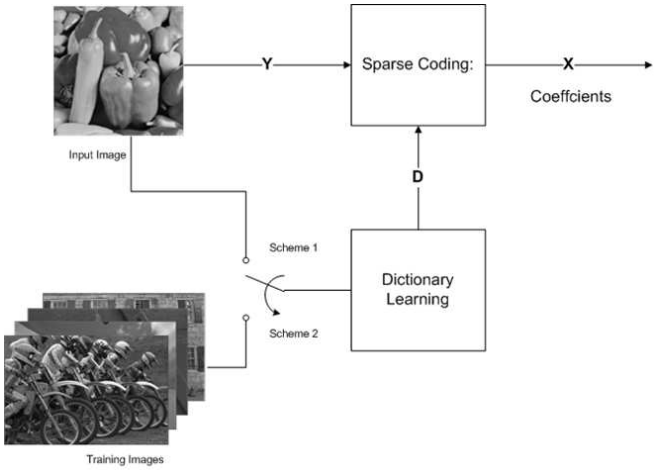

Fig. 3. Employed compression schemes. Scheme 1: training and coding with the same image. Scheme 2: training with a set of natural images and then using the resulting dictionary to a test image

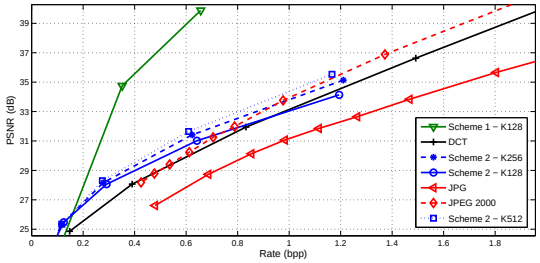

Fig. 4. Comparison of compression schemes, Image: Harbour

We also compared scheme 2 against JPEG and JPEG2000. We observe comparable results to JPEG2000 and performance gain of $4\ dB$ with respect to JPEG. It is important to mention that, for the presented results, JPEG and JPEG2000 effectively implement an entropy coder, while our scheme (as well as the basic DCT framework we use) assumes the utilization of an optimal entropy coder.

5. CONCLUSION

In this work, we investigated the use of entropy-constrained overcomplete-based schemes for compression of natural images. The results show that the presented methods outperform the ones based on the commonly employed approaches for overcomplete-based compression. RD-OMP provides a gain of up to $2\ dB$ with respect to OMP. EC-DL, a method introduced in this work and its main contribution, improves the overcomplete dictionary learning process for compression, with gains in the order of $2\ dB$. Finally, the overall compression scheme employing a trained dictionary outperforms a basic DCT scheme when the number of basis vectors is sufficiently large, with gains of up to $2\ dB$. As future work, we intend to: a) implement an entropy coder for the proposed scheme, b) investigate further the trade-offs between K and N, c) extend this scheme to video coding, d) reduce the complexity of the proposed algorithms and e) evaluate our scheme against directional transforms.

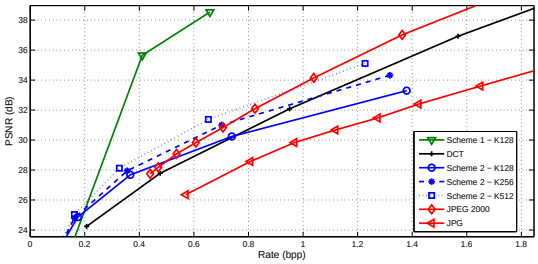

Fig. 5. Comparison of compression schemes, Image: Lena

6. REFERENCES

[1] O. Bryt, and M. Elad, "Compression of facial images using the K-SVD algorithm," *Journal of Visual Communication and Image Representation*, vol. 19, no. 4, pp. 270–282, May 2008.

[2] K. Engan, J. H. Husoy, and S. O. Aase. "Frame based representation and compression of still images," *International Conference on Image Processing*, vol. 2, no., pp. 427–430, Oct 2001.

[3] M. Gharavi-Aikhansari, "A model for entropy coding in matching pursuit," *International Conference on Image Processing*, vol. 1, pp. 778–782, Nov. 1998.

[4] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *J. Constructive. Approximation*, vol. 13, pp. 57–98, 1997

[5] S. G. Mallat, and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. on Signal Processing*, vol. 41, pp. 3397–3415, Dec. 1993.

[6] G. Davis, S. Mallat, and Z. Zhang, "Adaptive time-frequency decomposition," *Optical Engineering*, vol. 33, pp. 2183–2191, July 1994.

[7] A. Gersho, and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer Academic, 1991.

[8] M. Aharon, M. Elad, and A. M. Bruckstein. "The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representations," *IEEE Trans. on Signal Processing*, vol. 19, pp. 4311–4322, Nov. 2006.

[9] K. Engan, S. O. Aase, and J. H. Husoy. “Method of optimal directions for frame design,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, no. 5, pp. 2443–2446, 1999.

Appendix

Breakdown of project preparation:

Andre: General simulations, implementations, presentation slides and project report

Maryam: General simulations, implementations, presentation slides and project report

Ryan: Sparse coding simulations, presentation slides and proejct report