We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter

On the Application of Dictionary Learning to Image Compression

Ali Akhari and Maria Trocan

IntechOpen

Abstract

Signal models are a cornerstone of contemporary signal and image-processing methodology. In this chapter, a particular signal modelling method, called synthesis sparse representation, is studied which has been proven to be effective for many signals, such as natural images, and successfully used in a wide range of applications. In this kind of signal modelling, the signal is represented with respect to dictionary. The dictionary choice plays an important role on the success of the entire model. One main discipline of dictionary designing is based on a machine learning methodology which provides a simple and expressive structure for designing adaptable and efficient dictionaries. This chapter focuses on direct application of the sparse representation, i.e. image compression. Two image codec based on adaptive sparse representation over a trained dictionary are introduced. Experimental results show that the presented methods outperform the existing image coding standards, such as JPEG and JPEG2000.

Keywords: image compression, dictionary learning, sparse representation

1. Introduction

Signal models are fundamental tools for efficiently processing of the signals of interesting, including audio recordings, natural images, video clips, and medical scans, to name just a few. A signal model formulates a mathematical description of the family of signals of interesting in a way which faithfully captures their behaviour. Designing accurate signal models, which efficiently capture useful characteristics of the signals, has been a crucial aim in the signal processing area for so many years and a variety of mathematical forms has been proposed. Sparsity-based modelling has been used in many applications in which each signal is represented in terms of linear combinations of an underlying set, called *dictionary*, of elementary signals known as atoms, resulting in simple and compact models. The driving force behind this model is *sparsity*, i.e. the rapid decay of the representation coefficients over the dictionary. In this signal modelling, the dictionary plays an important role for the success of entire model in an efficient representation of the signal.

Finding appropriate dictionaries with good predictive power of various signal classes of interest and high compactness ability, especially natural images, has been an active field of research during past decades. The early attempts for designing dictionaries were based on building the model using harmonic analysis of the signal classes and extracting some mathematical functions, resulting in a fixed off-the-shelf dictionary called *analytic* or mathematical dictionary. The sparse

representation using these fixed mathematical dictionaries is called *analysis* sparse modelling. The long series of works on designing the analytic dictionaries lead to appearing various transforms such as Fourier and its discrete version, discrete cosine, wavelets, curvelets, contourlets, bandlets, and steerable wavelets.

A significantly different approach to the sparse modelling, originally introduced by Olshausen and Field [1], consists of learning a dictionary from some training data. The sparse representation using this trained dictionary is called, *synthesis* sparse modelling. The trained dictionaries, also called *synthetic* dictionary, could efficiently capture the underlying structures in the natural image patches and are well adapted to a large class of natural signals.

The analysis and synthesis sparse signal modelling has led to the design effective algorithms for many image-processing applications, such as compression [2–8] and solving inverse problems [9–22]. A straightforward application of the sparse signal modelling in the field of image processing has been image compression due to providing a compact representation of the signal. This chapter presents the theoretical and practical aspects of using the dictionary-based sparse signal modelling for image compression. We address one important question: How to efficiently represent an image over a trained dictionary in order to improve the performance of the image compression? Based on this concept, we present two novel image compression methods. Experimental results show that the introduced methods outperform the existing image coding standards, such as JPEG and JPEG2000.

The remainder of chapter is organized as follows: It starts with introducing the concept of sparsity in signal processing. The use of sparse representation modelling with respect to trained dictionaries is presented. Next, a brief description of a well-known and yet effective dictionary learning algorithm, introduced by Aharon et al. [23], is given. The end of chapter is devoted to introducing two generic image compression schemes. Finally, we conclude the chapter.

2. Sparsity-based signal models

One of the well-known methods in designing of the signal models is linear approximation. In this modelling technique, given a set of vectors $\left[\mathbf{d}_k \in \mathbb{R}^N\right]_{k=0}^{K-1}$, a signal $\mathbf{x} \in \mathbb{R}^N$ is represented as a linear combination of K basis,

$$\mathbf{x} \simeq \sum_{k=0}^{K-1} c_k \mathbf{d}_k, \tag{1}$$

where set $[c_k]_{k=0}^{K-1}$ consists of representation coefficients. The signal approximation Eq. (1) can be reformulated in a matrix form as

$$\mathbf{x} \simeq \mathbf{D}\mathbf{c}, \tag{2}$$

where $\mathbf{c} = [c_1, c_2, \cdots c_k]^T \in \mathbb{R}^K$ is the coefficients vector. The matrix $\mathbf{D} = [\mathbf{d}_1 \mathbf{d}_2 \cdots \mathbf{d}_K] \in \mathbb{R}^{N \times K}$ is called dictionary and its columns constitute the dictionary atoms.

With the right choice of dictionary, the coefficients vector $\mathbf{c}$ is expected to be sparse, in the sense that its sorted coefficients decay rapidly. Motivated by this idea, the design of efficient complete dictionaries, in which, i.e., $K = N$ was an active area of research during the last decades of twentieth century. The well-known Fourier transform [24], used in the JPEG compression standard [25], and wavelet transform [2], used in JPEG2000 compression standard [26], are the results of these

works which sparsify uniformly the smooth signals. However, this signal modelling loses its optimality in representation of the image signals due to existence of curve singularities (elongated edges) in these types of signals [27]. As an instance, the images encoded by the JPEG2000 standard suffer from the ringing (smoothing) artefacts near edges.

In an attempt to minimize this weakness of the dictionaries, the design of more general over-complete dictionaries, which have more atoms than the dimension of signal, i.e. $K>N$, has been investigated over the past decades, and is still intensely ongoing. These dictionaries have a more descriptive ability to represent a wide range of interesting signals, in comparison with the invertible complete dictionaries.

2.1 Sparse modelling using over-complete dictionaries

Compared to the complete case, representation with over-complete dictionaries must be more carefully defined. There are two distinct paths for representing a signal using the over-complete dictionaries: analysis path and synthesis path. The *analysis* sparse modelling relies on the classical basics of the signal modelling in which the representation of the signal is identified as a linear combination of atoms,

$$\mathbf{x} \simeq \mathbf{D}\mathbf{c}_a, \tag{3}$$

where the coefficients vector $\mathbf{c}_a$ is obtained via the inner products of the signal and the dictionary $\mathbf{c}_a = \mathbf{\Omega}\mathbf{x} = \mathbf{D}^T\mathbf{x}$. This method has the advantage of providing a simple and efficient way to achieve sparse representation over the dictionary. In this case, every signal has an unique representation as a linear combination of the dictionary atoms.

Increasing sparsity, in order to obtain a well-defined representation, requires departure from this linear representation towards a more flexible and non-linear representation. Each signal is represented using a different set of atoms from a pool, called dictionary, in order to achieve the best sparsity. Thus, the approximation process becomes

$$\mathbf{x} \simeq \mathbf{D}\mathbf{c}_s, \tag{4}$$

where the coefficients vector $\mathbf{c}_s$ is obtained via a non-linear approach, in contrast to the linear-based approach in the analysis path. This signal modelling approach, called *synthesis* sparse representation, needs further refinement to find the well-defined representation due to degrees of freedom identified by the null-space of $\mathbf{D}$ [27], which leads to a non-unique choice of $\mathbf{c}_s$ in Eq. (4), as opposed to the analysis sparse modelling which has a unique solution. In order to find the most informative representation, the coefficients vector $\mathbf{c}_s$ is obtained with respect to some cost function $\mathbf{F}(\cdot)$, which minimizes the sparsity of the coefficients vector $\mathbf{c}_s$ under a reconstruction constraint:

$$\mathbf{c}_s = \underset{\mathbf{c}\in\mathbb{R}^K}{\arg\min}\mathbf{F}(\mathbf{c}) \; \textit{subject to} \;\; \|\mathbf{x} - \mathbf{D}\mathbf{c}\|_2^2 \leq \epsilon, \tag{5}$$

where ϵ is the prior knowledge about the noise level. The penalty function $\mathbf{F}(\cdot)$ is defined in a way that is tolerant to the large coefficients and aggressively penalizes the small coefficients [27]. The normal choice for this function is ℓ_p norm with $0 \leq p \leq 1$. Of specific interest is the ℓ_0 case, i.e., $\mathbf{F}(\mathbf{c}) = \|\mathbf{c}\|_0$, which counts the number of non-zeros in the representation. For this case, the problem Eq. (5) becomes

$$\mathbf{c}_s = \arg\min_{\mathbf{c}\in\mathbb{R}^K} \|\mathbf{c}\|_0 \quad \textit{subject to} \quad \|\mathbf{x} - \mathbf{D}\mathbf{c}\|_2^2 \leq \epsilon. \tag{6}$$

This problem, known to be NP-hard in general, can be efficiently approximated based on the idea of iterative greedy pursuit. The earliest and yet effective one includes the orthogonal matching pursuit (OMP) [28].

This formulation Eq. (6) has gained a large success beyond the statistics and signal processing communities and has been extensively employed in different signal processing algorithms. In the image-processing applications, since the size of natural images is too large, it is chosen to partition the image into blocks and the sparse modelling is done on the set of image blocks $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2 \cdots x_L]$, each of size $\sqrt{N} \times \sqrt{N}$ pixels, where $\sqrt{N}$ is an integer value and $\mathbf{x}_i \in \mathbb{R}^N$ is lexicographically stacked representation of the i-th image patches.

2.2 Dictionary choice

In the discussion so far, it is assumed that the dictionaries of the analysis and synthesis models are known. Choosing the dictionary carefully is an important and involving task, in which substantial research has been invested. Based on analysis and synthesis models, the scientific community has developed two main routes for designing the dictionaries [27].

The first one is *analytic* dictionaries derived from a set of mathematical assumptions made on the family of the signals. The dictionaries of this type are generated by finding the appropriate mathematical functions through harmonic analysis of the interesting signals for which an efficient representation is obtained. For instance, Fourier basis is designed for optimal representation of smooth signals, while the wavelet dictionary is more suitable for piecewise-smooth signals with point singularities.

Designing analytic over-complete dictionaries are formulated as $\mathbf{D}\mathbf{D}^T\mathbf{x} = \mathbf{x}$ for all $\mathbf{x}$. Then, the approach tries to establish an appropriate dictionary by analysing the behaviour of $\mathbf{D}^T\mathbf{x}$ and establishing a decay rate. The curvelet [29], contourlet [30], and bandlet [31] transforms are some of the analytic dictionaries which provide comprehensive frameworks in order to handle the multi-dimensional signals.

Finding the more compact sparse representation has been a major driving force for the continued development of more efficient dictionaries. The synthesis formulation of the sparse representation paved the way to the design of an efficient dictionary, called *synthetic* dictionaries, from signal realizations via machine-learning techniques. The basic assumption behind this approach is that the structure of the complex natural signals can be more accurately extracted directly from the data than by using a general mathematical model [27]. In fact, this approach replaces prior assumptions on the signal behaviour with a training process which constructs the dictionary based on the observed signal properties. Compared to the analytic dictionaries, the synthetic dictionaries deliver an increased flexibility and the ability to adapt to specific signals and are superior in terms of representation efficiency at the cost of a non-structured and substantially more complex dictionary.

In this approach, a dictionary is trained for the sparse representation of small patches collected from a number of training signals. The desire to efficiently train a dictionary for the sparse representation led to developing some algorithms so far [1, 23, 32–34]. The earlier works on the dictionary learning mostly focused on statistical methods. Given the training image patches $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2 \cdots \mathbf{x}_L]$, where L is

the number of training patches, this method finds a dictionary which either maximizes the likelihood of the training data $P(\mathbf{X}|\mathbf{D})$ [35] or the posterior probability of the dictionary $P(\mathbf{D}|\mathbf{X})$ [36]. These formulations lead to some optimization problems that are solved in an expectation-maximization fashion, alternating estimation of the sparse representation and the dictionary using gradient descent or similar methods.

In neuroscience area, Olshausen and Field [1] proposed a significantly different approach for designing the dictionary using the training data, benefiting from modelling the receptive fields of simple cells in the mammalian primary visual cortex. In another attempt, the K-SVD algorithm introduced by Aharon et al. [23] is one of the well-known methods of dictionary learning. Given a set of examples $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2 \cdots \mathbf{x}_L]$, the goal of the K-SVD algorithm is to search the best possible dictionary $\mathbf{D} \in \mathbb{R}^{N \times K}$ for the sparse representation of the training set $\mathbf{X}$ through the optimization problem of Eq. (7):

$$\underset{\mathbf{C},\ \mathbf{D}}{\arg\min} \sum_{i=1}^{L} \|\mathbf{c}_i\|_0 \quad \textit{subject to} \quad \|\mathbf{X} - \mathbf{DC}\|_2^2 \leq \epsilon, \tag{7}$$

where ϵ is a fixed small value and $\mathbf{C} = [\mathbf{c}_1\ \mathbf{c}_2 \cdots \mathbf{c}_L]$ is a matrix of size $K \times L$, consisting of the representation coefficients vectors $[\mathbf{c}_i]_{i=1}^{L}$ of the training samples. The expression in Eq. (7) is performed iteratively. First, by considering an initial dictionary, the algorithm tries to find the best coefficients matrix $\mathbf{C}$ that can be found. Once $\mathbf{D}$ is known, the penalty, posed in Eq. (6), reduces to a set of L sparse representation operations, like the ones seen in (Section 2.1). The OMP algorithm [28] is used for the near-optimal calculation of the coefficients matrix $\mathbf{C}$.

At the next stage, the columns of dictionary are sequentially updated and relevant coefficients in the matrix $\mathbf{C}$ are simultaneously changed. At a time, one column is updated and the process of updating one column is based on the singular value decomposition (SVD) on the residual data matrices, computed only on the training samples that use this atom. The K-SVD algorithm includes a mechanism to control and rescale the ℓ_2-norm of the dictionary elements. Indeed, without such a mechanism, the norm of $\mathbf{D}$ would arbitrarily go to infinity. For more details, refer to [23].

In the image-processing applications, since the size of natural images is too large for learning a full matrix $\mathbf{D}$, it is chosen to learn the dictionary on a set of natural image patches $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2 \cdots \mathbf{x}_L]$, each of size $\sqrt{N} \times \sqrt{N}$ pixels, where $\sqrt{N}$ is an integer value and $\mathbf{x}_i \in \mathbb{R}^N$ is lexicographically stacked representations of the i-th image patches.

2.3 Analysis versus synthesis

As mentioned before and also outlined in [27], some of the most important elements of effective dictionary design include localization, multi resolution, and adaptivity. Modern dictionaries typically provide localization in both the analysis and synthesis routes. However, multi-resolution property is usually better supported by the analytic structures, whereas adaptivity is mostly found in the synthetic methods.

The most important advantage of the analytic dictionaries is the easy and fast implementation. On the other hands, the main advantage of the trained dictionaries is their ability to provide a much higher degree of specificity to the particular signal properties, allowing them to produce better results in many practical applications such as image compression, feature extraction, content-based image retrieval and others. However, some drawbacks might arise after the compactness introduced by

synthesis scheme. In this case, only if a few atoms for presenting a signal, the importance of all atoms largely varies. Consequently, any wrong choice of one atom could potentially lead to additional erroneous atoms that are selected as compensation, deviating further from the desired reconstruction. This weakness is usually appeared in the ℓ_0-norm-based non-convex optimization problem of Eq. (6). The convex relaxation approaches from ℓ_0 to ℓ_1 are more stable for the sparse representation at the expense of computational complexity. In the analysis formulation, however, all atoms take an equal part in describing the signal, thus minimizing the dependence on each individual one, and stabilizing the recovery process.

3. Image compression

Reducing the cost for storage or transmission of image signals with negligible degradation in the quality is the main goal of an lossy image compression algorithm. It tries to remove the redundancies among the image data by adopting different ways, such as transferring the image into a transform domain with compressible coefficients. In the transform domain, a few significant coefficients capture a large part of the image information. A typical lossy image compression algorithm usually encodes these significant transform coefficients to reduce the requirements for image storage or transmission. The analysis and synthesis sparse modellings are two powerful tools for transforming the image into a compressible domain. The JPEG [25] and JPEG2000 standards [26] are the results of using the analysis sparse representation of the image by designing analytic dictionaries, e.g. discrete cosine transform (DCT) and discrete wavelet transform (DWT), respectively. Since analytic dictionaries-based image sparse representation is typically over-simplistic, it fails to represent the high-textured images efficiently [27]. Due to this weakness of analysis sparse modelling in the efficient expressiveness, an extensive body of literature has recently focused on various applications of the synthesis sparse signal modelling via a trained dictionary. In this way, the performance can be significantly improved for the image compression application, benefiting the sparse representation of the image over a dictionary specifically adapted to it.

In order to improve the limitations of the traditional sparse representation approaches over a trained dictionary, several studies have been proposed in the literature [37–39]. In [37], the authors train a set of dictionaries. In order to improve the compression performance, the sparse representation is achieved by choosing an optimal dictionary among the trained dictionaries. In [38], a set of dictionaries in a tree structure is trained. At each level of tree, a dictionary is learned via generating the image residual. The residual is obtained by difference between the original and recovered images using the trained dictionary at the previous level of tree. Sparse representation is done by selection of one atom from each dictionary at each tree level. The total number of atoms is determined according to the sparsity level. Authors in [39] introduce the concept of multi-sample sparse representation (MSR) and incorporate it into the dictionary learning process. Each image patch is encoded with a certain sparsity level. To do this purpose, multiple neighbouring image patches are considered during the sparse representation to explore different sparsity levels. Based on this concept, an MSR-based image coding approach is proposed in [39].

Using a trained dictionary, each image is represented over the dictionary. The coefficients vector can be obtained via different algorithms such as the basis pursuit algorithms, matching pursuit techniques and other schemes [40]. These conventional approaches usually consider constant sparsity level, i.e. a fixed number of dictionary atoms is considered for representing all the image patches with different characteristics. This approach leads to a weak image compression performance.

In this section, an adaptive sparse representation approach is presented. It is based on this fact that the visual significance, called visual saliency, of each image patch varies with its location within the image [41]. In other words, the human visual system (HVS) usually focuses on some parts of the image (salient regions), while other parts of the image have a lower level of visual interest. Therefore, designing an adaptive sparse representation scheme by considering the HVS characteristics plays an important role in designing an efficient image compression algorithm.

3.1 Dictionary learning-based image codec

Figure 1 presents the block diagram of the dictionary learning-based image coding (DLC) framework [7]. The DLC mainly has four main parts, pre-processing, dictionary learning, adaptive sparse representation and entropy coding. First, at the pre-processing step, the input image is divided into L non-overlapping image patches $\mathbf{X} = [\mathbf{x}_i]_{i=1}^{L}$. $\mathbf{x}_i \in {}^{N}$ represents a $B \times B$ image patch vectorized into a vector of size N. As other coding algorithms, the mean values of image patches (DC components) $\mathbf{M} = [m_i]_{i=1}^{L}$ and AC components $\mathbf{Y} = [\mathbf{y}_i]_{i=1}^{L}$ are separately encoded. The AC components are represented with respect to a trained dictionary $\mathbf{D}$ of size $N \times K$ learned by the K-SVD dictionary learning algorithm [23]. As other dictionary-based image compression methods, the trained dictionary should be shared between encoder and decoder. The OMP algorithm is used for sparse representation step in an adaptive manner using the visual saliency information. Incorporating this adaptive sparse representation step into the image coding framework aims to further reduce the reconstructed errors. Finally, the DC elements $\mathbf{M} = [m_i]_{i=1}^{L}$ and representation coefficients $\mathbf{C} = [\mathbf{c}_i]_{i=1}^{L}$ are entropy coded, where $\mathbf{c}_i \in {}^{K}$ denotes the coefficients vector of the i-th image patch. In the following sections, the adaptive sparse representation and entropy coding steps are detailed. We ignore the details of decoder. However, it should be noted that the decoder can retrieve the image by a minor application of the above steps.

3.1.1 Adaptive synthesis sparse representation

The sparse representation step has a strong impact on effectively encoding of image patches and thus the rate-distortion performance. Graph-based visual saliency (GBVS) model, proposed in [42], has an capability to extract the saliency map of an image, i.e. locations within an image where have a high visual interest to a human observer. By incorporating this model into the sparse representation step, we build up an image coding scheme which compresses the image more efficiently than the traditional coders. Please refer to [42] for the details of the GVBS model. This saliency map of the image is exploited to determine the visual significance of the image patches in order to allocate different sparsity levels to each patch according to its visual significance to the HVS.

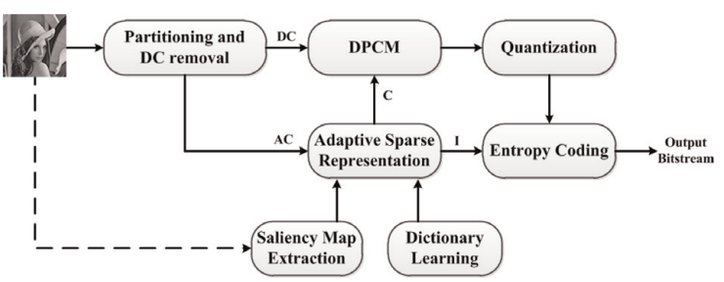


Figure 1.
Block diagram of the dictionary-based image coding (DLC) framework.

It should be noted that we normalize the saliency map values to the range $[0, 1]$. Therefore, the intensity of each pixel of this normalized saliency map stands for the probability belonging that pixel to a salient region. Some examples of saliency maps are shown in **Figure 2**. In the second row, brighter regions represent the salient locations which a human observer pays more attention to, while the darker areas represent the less-saliency regions.

To obtain the saliency value of each image patch, first, the saliency map is partitioned into non-overlapping blocks of size $B \times B$ pixels. Then, the average of the saliency values of pixels belonging to each block is considered as the saliency value of the corresponding patch. Let H_i denotes the saliency value of i-th image patch. Since saliency value of each image patch varies with its position within the image; therefore, by assigning different sparsity level to each block according to its saliency value leads to an improvement in the rate-distortion performance.

Consider the sparsity level of i-th image patch as:

$$S_i = \alpha_i S. \tag{8}$$

Given the target sparsity level S, we aim to allocate a different sparsity level S_i to the i-th image patch. It should be noted that the average of sparsity levels of all image patches must be equal (or slightly inferior) to the target sparsity level S. The factor α_i can be easily obtained via:

$$\alpha_i = \frac{L \times H_i}{\sum_{i=1}^{L} H_i}. \tag{9}$$

where $[H_i]_{i=1}^{L}$ denotes the set of saliency values. As a result, a set of sparsity levels $[S_i]_{i=1}^{L}$ is obtained via Eq. (8). Based on these new obtained sparsity levels, the sparse representation of each patch x_i over the dictionary $\mathbf{D}$ is achieved by:

$$\underset{\mathbf{c}_i}{\text{argmin}} \|\mathbf{x}_i - \mathbf{D}\mathbf{c}_i\|_2 \text{ Subject To } \|\mathbf{c}_i\|_0 \leq S_i. \tag{10}$$

The OMP method in [43] is used to solve this problem. By assigning different sparsity level S_i to each block, a more effective sparse representation of the image is obtained.

3.1.2 Quantization step and entropy coding

Differential pulse-coded modulation (DPCM) scheme is used to quantize the DC elements. Let $e_i = m_i - m_{i-1}$ denotes the residual between DC values of two

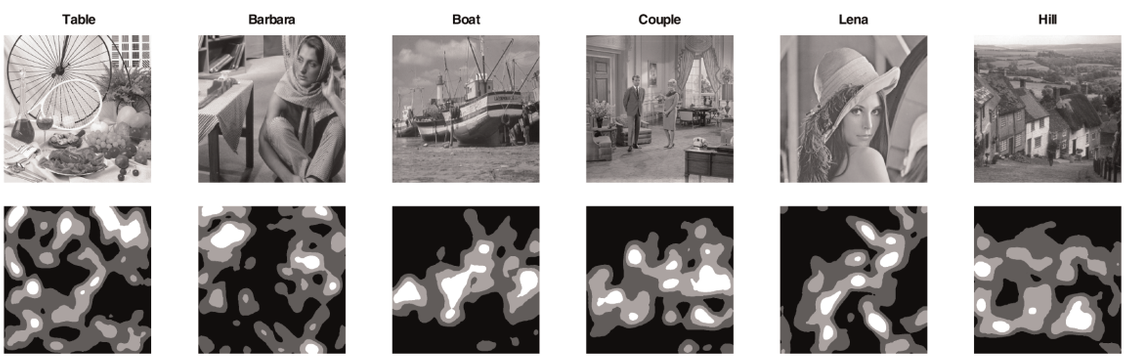


Figure 2.
The original images and their saliency maps.

neighbouring patches i and $i-1$ (here we set m_0 as 0). Instead of encoding $\mathbf{M} = [m_i]_{i=1}^{L}$ directly, the residuals between subsequent DC elements, i.e. $\mathbf{E} = [e_i]_{i=1}^{L}$ are quantized and encoded. The quantization is achieved via $round(e_i/b)$ where b is a constant value. A dead-zone quantizer is used for quantization of the residuals [43].

In order to further remove the redundancy, the Huffman coder is used to entropy-code the quantized DC values. During the entropy coding step, pre-defined code-word tables are used built offline. The same procedure is employed for quantizing and encoding the nonzero coefficients of $\mathbf{C} = [\mathbf{c}_i]_{i=1}^{L}$.

A big part of the bit-stream generated by the above scheme is occupied by encoding the indices of the representation coefficients, $\mathbf{I}$. The reason behind this fact is that the non-zero coefficients in $\mathbf{C} = [\mathbf{c}_i]_{i=1}^{L}$ have a random structure. To efficiently encode this random pattern, a quad-tree splitting algorithm is employed. First, each vector $\mathbf{c}_i$ is divided into two equal parts. At the next step, a binary evaluation is achieved on each part: if each part consists of one nonzero coefficient, the encoder outputs 1; otherwise, the encoder sends 0. Each part including one nonzero coefficient at least, is then divided into two separate parts and the binary evaluation is again performed. This process continues until the maximum depth of partitioning is reached. In comparison with other methods like fixed length coding [39], this quad-tree splitting algorithm encodes the indices of nonzero coefficients more efficient.

3.2 Down-sampling-based codec

In a basic image compression approach, a down-sampling strategy is employed at the encoder side, as another lossy operation. Then, at the decoder side, an up-scaling algorithm is employed to reconstruct the high resolution (HR) image. In this technique, the high-frequency details within the image are removed by employing the down-sampling operator before the quantization lossy operator. Thus low frequency information within an image, where contain most of the energy image, is encoded by higher bit-rate. This bit allocation scheme leads to the better rate-distortion performance at the low bit-rates. Although the down-sampling operator improves the rate-distortion performance at low bit-rates, it eliminates the high frequency (HF) details. These information should be reproduced in order to recover a high quality image at the higher bit-rates.

In the DLC encoder, presented in previous section, the quantization and sparse representation are the main compression tools. In this section, the down-sampling operator is incorporated into the DLC codec as another lossy operator to further remove redundancies existing within an image. Instead of encoding the whole image, its low-resolution (LR) version is encoded. At the decoder side, an up-scaling algorithm is carefully designed to recover the high-frequency details. As discussed before, this strategy enhances the coding efficiency at the low bit-rates. However, designing an efficient up-scaling algorithm is an important step at the decoder side by which the high frequency information can be correctly recovered. A failure in designing a good up-scaling algorithm leads to a weak rate-distortion performance at the high bit-rates. The presented up-scaling scheme in this section recovers the HR image by encoding and sending the residual image, i.e. difference between the original image and the up-scaled one, as side information. The encoder generates the residual image and encodes it by the sparse representation of the residual image over a trained dictionary. Decoder recovers the final image by combination of the LR decoded image and the reconstructed residual. Both dictionaries used for the sparse representation of LR residual images are trained by a bi-level dictionary learning algorithm. Further, an image analyser is designed and

incorporated into the encoder. The goal of this image analyser is for designing an adaptive sparse representation in order to assign higher bit-rate to the salient parts of the image. Experimental results illustrate that this down-sampling-based image compression scheme based on sparse representation (DCSR) achieves better rate-distortion performance when compared with the conventional codecs, such as JPEG and JPEG2000, and the DLC codec, described in Section 3.1.

The block diagram of the DCSR encoder is shown in **Figure 3**. This framework mainly consists of five parts, including down-sampling, up-scaling, sparse representation, quantization and entropy coding [8]. An image analyser is also introduced to the core of codec. At the pre-processing step, the down-sampling operator uses a blurring convolution kernel and a simple decimator by three to produce the LR image with only 1/9 of total pixels of the original image. At the up-scaling step, the down-sampled image is restored back to its original resolution. The goal of the up-scaling step is designing an accurate algorithm to improve the rate-distortion performance at the higher bit-rates. A joint sparse representation method is employed to solve this ill-posed, complex inverse problem. This method guarantees to restore the high frequency details which are enough and sufficient for reconstruction of the HR image.

Assume the LR image, X^L, is obtained via $\mathbf{X}^L = \mathbf{H}\mathbf{X}^H$, where $\mathbf{H}$ is the down-sampling operator and $\mathbf{X}^H$ denotes the HR image. After the image recovery via the up-scaling algorithm, the relationship between the HR and the reconstructed images is described as $\mathbf{X}^H = \tilde{\mathbf{X}}^H + \mathbf{E}$, where $\mathbf{E}$ is the residual image. The LR image $\mathbf{X}^L$ and the residual image $\mathbf{E}$ are separately partitioned into non-overlapping image patches of size $B \times B$. Let us assume $[\mathbf{x}_i]_{i=1}^{T_L}$ and $[\mathbf{e}_i]_{i=1}^{T_E}$ denote the vectorized image blocks in the $\mathbf{X}^L$ and $\mathbf{E}$, respectively, where T_L and T_E are the number of patches in the $\mathbf{X}^L$ and $\mathbf{E}$, respectively. Moreover, note that $T_L = 1/9T_E$.

At the sparse representation step, the LR image patches $[\mathbf{x}_i]_{i=1}^{T_L}$ are sparsely represented over an over-complete dictionary $\mathbf{D}^L$ of size $N \times K$ using the OMP algorithm. An image analyser is carefully designed to use the visual saliency information at the sparse representation step. The goal of the analyser is to further reduce the reconstructed errors in the areas within the image where have higher visual interest to the HVS. The same procedure is also applied for the sparse representation of the image patches of the residual image $[\mathbf{e}_i]_{i=1}^{T_E}$. A different over-complete dictionary $\mathbf{D}^E$ of size $N \times K$ is used for the sparse representation of the residual patches. The two dictionaries $\mathbf{D}^L$ and $\mathbf{D}^E$ are trained offline using a bi-level dictionary learning algorithm. It is assumed that these dictionaries are shared between the encoder and decoder. Finally, the obtained sparse coefficients of LR patches and residual patches, i.e. $\mathbf{C}^L = \left[\mathbf{c}_i^L\right]_{i=1}^{T_L}$ and $\mathbf{C}^E = \left[\mathbf{c}_i^E\right]_{i=1}^{T_E}$ respectively, are quantized and entropy coded in order to obtain the bit-stream. $\mathbf{c}_i^L \in \mathbb{R}^K$ and $\mathbf{e}_i^L \in \mathbb{R}^K$

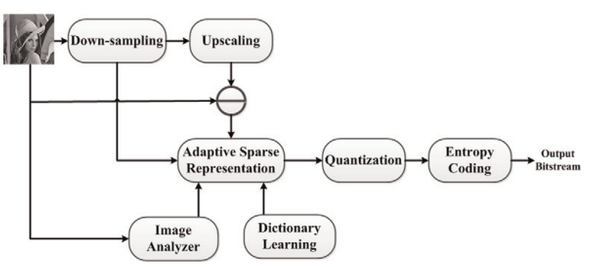


Figure 3.
Block diagram of the DCSR codec.

denote the coefficients vector of the i-block in the LR image $\mathbf{X}^L$ and the residual image $\mathbf{E}$, respectively.

At the decoder side, the image is reconstructed by a minor application of the above steps. First, the reconstructed LR image $\hat{\mathbf{X}}^L$ is up-scaled by the up-scaling operator to restore the image $\hat{\mathbf{X}}^H$ of size of the original image. Then, the final restored image is obtained via $\mathbf{X}^D = \hat{\mathbf{X}}^H + \hat{\mathbf{E}}$, where $\hat{\mathbf{E}}$ denotes the decoded residual image.

3.2.1 Image analyser for adaptive sparse representation

The main goal of image analyser is to enhance the performance of the DCSR encoder. It reduces the reconstructed error in some parts of the image which have higher visual interest to a human observer. Therefore, the image analyser extracts the salient regions within the image according to its visual significance to the HVS. Then, it allocates higher rates to the salient regions by assigning different sparsity levels to the image patches, located within the salient parts, in both LR and residual images. As mentioned before, the GBVS model [42] is used to obtain the salient regions within an image. Some examples of saliency maps are shown in **Figure 2**.

For finding the sparsity levels of the image patches in the LR image, the obtained saliency map is re-sized to the same size of the LR image. Then, it is divided into $B \times B$ blocks to obtain the corresponding saliency value H_i of the i-th block by taking the average of the saliency values of pixels within that block. The goal is assign a different sparsity level S_i to each block. Note that the average of the sparsity levels assigned to all blocks within the image should be equal (or slightly inferior) to the target sparsity level S. Given the target sparsity level S and the set of saliency values $[\mathbf{H}_i]_{i=1}^{T_L}$, the sparsity level of the i-th block is obtained by:

$$S_i^L = \frac{T_L \times H_i}{\sum_{i=1}^{T_L} H_i} S, \tag{11}$$

where T_L is the number of blocks in the LR image. Based on this strategy, different sparsity levels are allocated for the sparse representation of the image patches in the LR image. Given the obtained sparsity levels $\left[S_i^L\right]_{i=1}^{T_L}$, the sparse representation of the LR patch $\mathbf{x}_i$ in the LR image over the dictionary $\mathbf{D}^L$ is achieved by [44]:

$$\underset{\mathbf{c}_i^L}{\operatorname{argmin}} \|\mathbf{x}_i - \mathbf{D}^L \mathbf{c}_i^L\|_2, \text{ s.t. } \|\mathbf{c}_i^L\|_0 \leq S_i^L. \tag{12}$$

The same procedure is applied to obtain the sparsity levels $\left[S_i^E\right]_{i=1}^{T_E}$ for the image patches in the residual image. The set of saliency values $[\mathrm{H}_i]_{i=1}^{T_E}$ is obtained using the saliency map of the original image, as described before. The sparse representation of each patch e_i in residual image with respect to the dictionary $\mathbf{D}^E$ is achieved by:

$$\underset{\mathbf{e}_i^E}{\operatorname{argmin}} \|\mathbf{e}_i - \mathbf{D}^E \mathbf{c}_i^E\|_2, \text{ s.t. } \|\mathbf{c}_i^E\|_0 \leq S_i^E. \tag{13}$$

The OMP method is used to solve these problems. At the low bit-rates, DCSR encodes only the LR image. At the higher bit-rates, DCSR encodes the residual image and send it as side information in order to improve the coding efficiency. In

this case, the patches within the residual and LR images are represented over the corresponding dictionaries with the same target sparsity level S.

3.2.2 Joint dictionary-mapping learning for up-scaling

The goal of up-scaling algorithm is to recover the high frequency information eliminated during the down-sampling operation. One well-known up-scaling approach is to learn a function which maps the LR and HR image patches. This mapping function can be obtained using two training databases of the LR and HR image patches. Instead of finding the direct mapping function between the LR and HR patches, authors in [44] show that the mapping function can be efficiently learned in the transform domain. In this approach, first, the LR and HR patches are transferred into the representation space over the trained dictionaries and then the relationship between representation coefficients is learned.

Let two sets $\mathbf{X}$ and $\mathbf{Y}$ consist of N training LR and corresponding HR image patches, respectively. Suppose $\mathbf{D}_x \in \mathbb{R}^{M \times K}$ and $\mathbf{D} \in \mathbb{R}^{M_y \times K}$ denote the trained dictionaries for the sparse representation of the LR and HR patches. These two dictionaries are trained jointly such that the sparse representation of all training LR and HR patches are mapped to each other via a linear function $\mathbf{M} \in \mathbb{R}^{K \times K}$. The dictionaries and mapping matrix are obtained by solving the following minimization problem:

$$\begin{aligned} \underset{\mathbf{D}_x,\, \mathbf{D}_y,\, \mathbf{\Lambda}_x,\, \mathbf{\Lambda}_y,\, \mathbf{M}}{\arg\min} \; & \|\mathbf{X} - \mathbf{D}_x \mathbf{\Lambda}_x\|_2^2 + \left\|\mathbf{Y} - \mathbf{D}_y \mathbf{\Lambda}_y\right\|_2^2 + \gamma \left\|\mathbf{\Lambda}_x - \mathbf{M}\mathbf{\Lambda}_y\right\|_2^2 \\ & + \lambda \|\mathbf{\Lambda}_x\|_1 + \lambda \left\|\mathbf{\Lambda}_y\right\|_1 + \lambda_m \|\mathbf{M}\|_2^2, \end{aligned} \tag{14}$$

where $\mathbf{\Lambda}_x \in \mathbb{R}^{K \times N}$ and $\mathbf{\Lambda}_y \in \mathbb{R}^{K \times N}$ represent the corresponding sparse representation matrices. The terms γ, λ and λ_m denote regularization parameters. As proposed in [45], the minimization problem Eq. (14) is solved in an iterative approach with respect to a variable while all the other variables are considered as fixed parameters.

After training process, the up-scaling is achieved by the dictionary pair (i.e. $\mathbf{D}_x$ and $\mathbf{D}_y$) and the mapping matrix $\mathbf{M}$. First, the LR image is divided into overlapping blocks of size $B_s \times B_s$. Then, the sparse representation vector $\boldsymbol{\alpha}_y$ of an LR patch $\mathbf{y}$ is obtained with respect to the trained dictionary $\mathbf{D}_y$. In the minimization problem Eq. (14), we use ℓ_1-norm to train the dictionaries. Given these trained dictionaries, the up-scaled algorithm uses ℓ_0-norm, instead of ℓ_1 norm, in order to improve the rate-distortion performance. Thus, the sparse representation of LR patch $\boldsymbol{\alpha}_y$ is obtained by solving the following minimization problem:

$$\boldsymbol{\alpha}_y = \underset{\boldsymbol{\alpha}}{\arg\min} \left\|\mathbf{y} - \mathbf{D}_y \boldsymbol{\alpha}\right\|_2^2 + \delta \|\boldsymbol{\alpha}\|_0, \tag{15}$$

where δ denotes the regularization parameter. At the next step, we derive $\hat{\boldsymbol{\alpha}}_x = \mathbf{M}\boldsymbol{\alpha}_y$ and restore the HR patch via $\hat{\mathbf{x}} = \mathbf{D}_x \hat{\boldsymbol{\alpha}}_x$. At the last step, the restored up-scaled image $\hat{\mathbf{X}}^H$ is obtained by replacing the recovered image patches into the whole image grid and taking the average over the overlapped pixels. For more details about solving the above optimization problems, please refer to [44].

3.2.3 Bi-level dictionary learning algorithm for image compression

The up-scaling algorithm, described in previous section, leads to a bi-level dictionary learning algorithm which used for image compression at the DCSR encoder.

Note that the trained dictionary D^L and the up-scaling operator are used to estimate a large part of signal energy. The remained energy of residual is captured by representing the residual over another trained dictionary. This residual is encoded as side information to improve the quality of the restored image at the higher bit-rates. The bi-level dictionary learning algorithm is achieved by the following steps: first, the training images are down-sampled by a factor 3. Second, these LR images are employed to train a dictionary D^L by the K-SVD algorithm [23]. Third, the LR training images are represented with respect to the trained dictionary D^L. Forth, the reconstructed LR images are then up-scaled by the algorithm described in previous section. Fifth, these up-scaled images are used to create the training residual images which are subsequently used to train the second dictionary D^R by the K-SVD algorithm.

3.2.4 Quantization and entropy coding

Differential pulse-coded modulation (DPCM) is used to quantize the non-zero elements in the sparse representation vectors C^L and C^E. A dead-zone quantizer is used for quantization. Further, the classical Huffman coding is employed to entropy-code the quantized sparse coefficients. Predefined code-word tables, needed for Huffman encoder, are constructed offline and stored at both encoder and decoder sides.

The location of non-zero coefficients in C^L and C^E follow a random pattern. Therefore, encoding the indices of these coefficients occupies a big part output bit-stream. For efficiently encoding these indices I, the quad-tree splitting algorithm, explained in Section 3.1.2, is employed.

3.3 Experimental results

In this section, the rate-distortion performance of the DCSR codec is examined by performing a suite of experiments on a set of 8-bit grey-scale standard images. All evaluated images are re-sized to size of 528×528 pixels in order to produce the LR image with only 1/9 of total pixels of the original image. The well-known peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) index are chosen as a measure in the experiments. Further, the rate distortion performance is compared with the JPEG and JPEG2000 standards, as well as the DLC codec, presented in Section 3.1.

Block size is set as 8×8 for encoding the LR and residual images. Further, two dictionaries D^L and D^E of size 64×440 are trained using the bi-level dictionary learning algorithm presented in Section 3.2.3. We use the images from the CVG-Granada dataset[1] for training. All these training images are partitioned into 8×8 image patches and 12,000 ones are randomly selected for training. One hundred epochs, each processing 12,000 training vectors, are considered during training process. For up-scaling, the dictionary pair of size 25×256 and the mapping matrix of size 256×256 are trained using the algorithm introduced in Section 3.2.2. The regularization parameters γ, λ, and λ_m are empirically set as 0.1, 0.1, and 0.01, respectively.

Next, we present the rate-distortion graph for the test images in **Figures 4** and **5**. Note that we also provide the results for different baseline algorithms, i.e. JPEG,[2]

[1] 1http://decsai.ugr.es/cvg/dbimagenes

[2] http://www.ijg.org

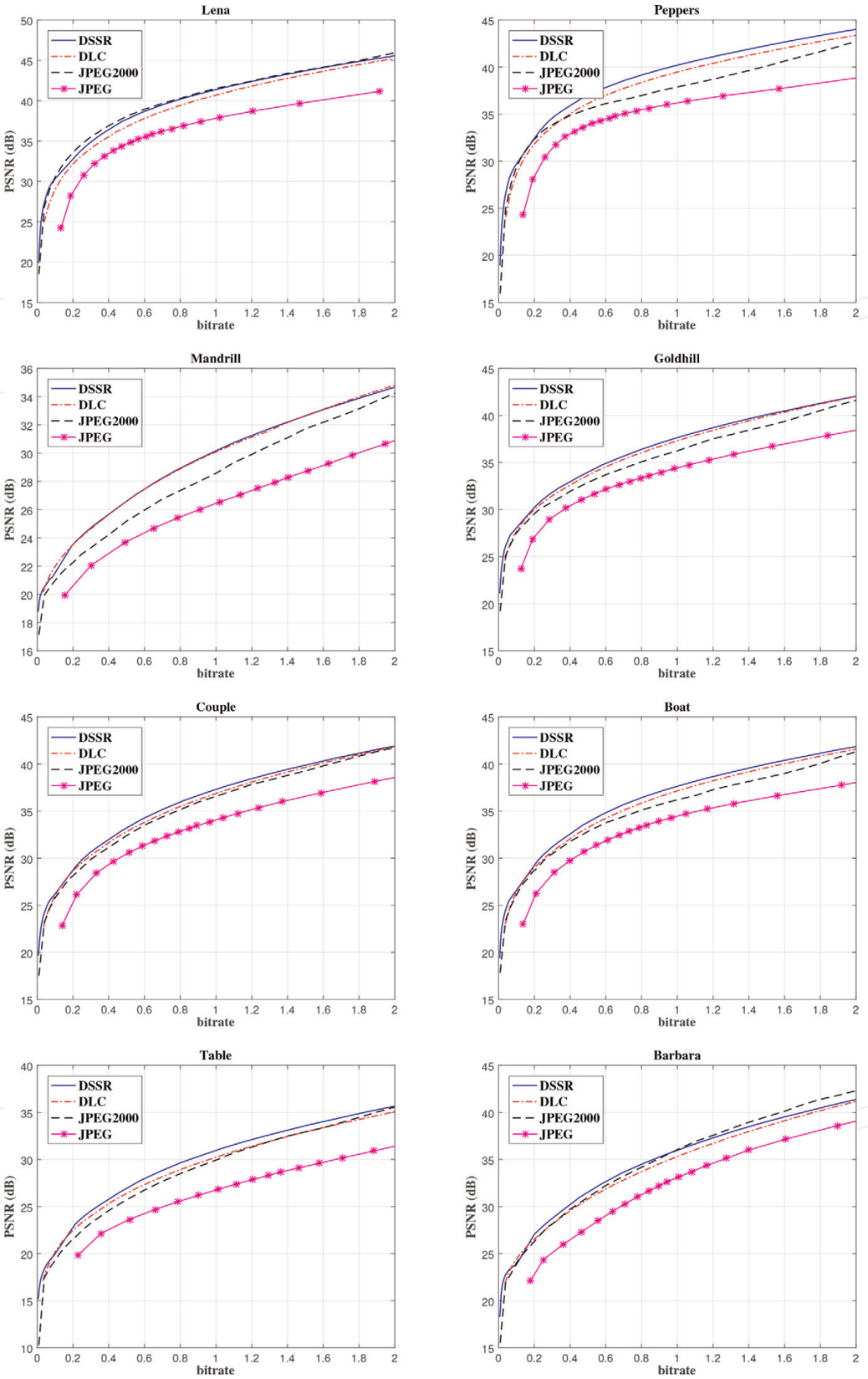


Figure 4.
Rate-distortion performance of the DCSR codec compared to JPEG, JPEG2000 and DLC codecs in terms of PSNR for several test images (size 528 × 528, grey-level).

JPEG2000,[3] and DLC algorithm [12]. As these figures shows DLC and DCSR codecs provide a better performance (in terms of PSNR and SSIM) than the available image coding standards JPEG and JPEG2000. Interestingly, as the **Figure 4** demonstrates,

[3] http://www.openjpeg.org

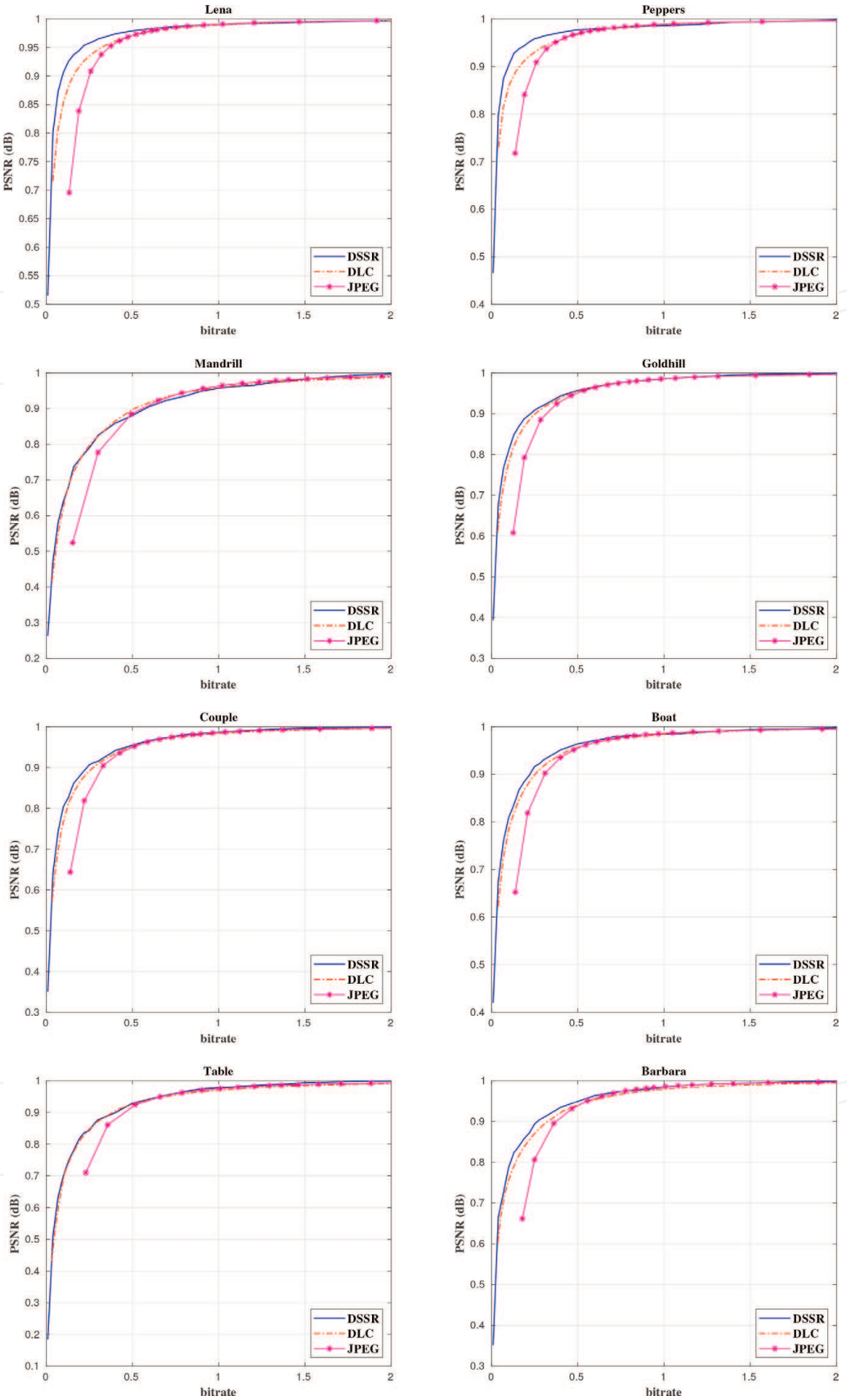


Figure 5.
Rate-distortion performance of the DCSR codec compared to JPEG and DLC codecs in terms of MSSIM for several test images (size 528 × 528, grey-level).

the DCSR algorithm enhances the quality of the image over the JPEG2000 codec. The enhancement depends on the statistics of the image, so different enhancement is observed in different images. Please note that SSIM shows the perceived quality



Figure 6.
Compressed images at bit-rate (0.15) using JPEG (left) and DCSR (right) scheme.

of the compressed image, therefore, SSIM curves for DCSR and JPEG2000 are very close to each other. For a better visualization, we removed the rate-distortion performance for JPEG2000 in **Figure 5**. As the trained dictionary performs a better capture and presents the contours more accurate, DCSR codec presents a higher performance compared to the analytic dictionaries such as the wavelet transform basis. Moreover, a 0.2 dBs improvement is observed using the DLC algorithm, marginally better for simpler images, like *Mandrill*.

Several compressed images are shown in **Figure 6** for visual comparisons. All images are coded with the same bit-rate (0.15) using JPEG and DCSR algorithm. As it can be seen, the JPEG standard fails to reconstruct images compressed at very low bit rates. In comparison, DCSR algorithm can preserve more image details. It is more effective in reconstruction of both the smooth area and the complex regions, including texture and edges, leading to visually much more pleasant recovery.

As a consequence, the down-sampling-based coding results in the high-frequency details in the image before performing the quantization lossy, hence larger number of bits are assigned low frequency information occupying more energy in the image. The presented bit assigning strategy brings a higher

performance for images coded at low bit-rate. In higher bit-rates, the efficiency of coding is enhanced after the residual image is encoded as side information.

4. Conclusions

A brief summary of the signal modelling methodology has been given at the first part of chapter. We continued with the explicit and straightforward formulation of the sparse representation being more suitable for the compression tasks. We have just focused on the synthesis-based signal modelling because of being mature of the image compression using the analysis-based sparse signal modelling. An adaptive sparse representation over a trained over-complete dictionary was presented to compress the images. More specifically, given the saliency map of the image to be encoded, an image patch could be well represented with the linear combination of atoms selected from an over-complete and trained dictionary based on the sparsity level. Finally, at the end part of this chapter, a down-sampling operation is incorporated into the codes in order to improve the compression performance. The experimental results demonstrated that the presented image compression frameworks outperform image coding standards, such as JPEG and JPEG2000, which use an analytic dictionary.

Author details

Ali Akbari[1,2*] and Maria Trocan[2]

1 University of Surrey, Guildford, UK

2 Institut supérieur d'électronique de Paris, Paris, France

*Address all correspondence to: ali.akbari@isep.fr

IntechOpen

© 2019 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. CC BY

References

[1] Olshausen A, Field DJ. Sparse coding with an overcomplete basis set: A strategy employed by v1? Vision Research. 1997;**16**(4):3311-3325

[2] Mallat S. A Wavelet Tour of Signal Processing. 3rd ed. Academic Press; 2008

[3] Pennec EL, Mallat S. Bandelet image approximation and compression. SIAM Multiscale Modeling & Simulation. 2005;**4**(3):992-1039

[4] Villegas OOV, Elias RP, Villela PR, Sanchez VGC, Salazar AM. Edging out the competition: Lossy image coding with wavelets and contourlets. IEEE Potentials. 2008;**27**(2):39-44

[5] Wang S, Shu Z, Zhang L, Liu G, Gan L. Iterative image coding with overcomplete curvelet transform. In: Proceedings of IEEE Congress on Image and Signal Processing (CISP), vol. 1; May 2008; Hainan, China. pp. 666-670

[6] Beferull-Lozano B, Ortega A. Coding techniques for oversampled steerable transforms. In: Proceedings of Asilomar Conference on Signals, Systems, and Computers, vol. 2; October 1999; Pacific Grove, CA. pp. 1198-1202

[7] Akbari A, Mandache D, Trocan M, Granado B. Adaptive saliency-based compressive sensing image reconstruction. In: Proceedings of IEEE International Conference on Multimedia Expo Workshops (ICMEW); July 2016; Seattle, WA. pp. 1-6

[8] Akbari A, Trocan M. Downsampling based image coding using dual dictionary learning and sparse representations. In: Proceedings of IEEE International Workshop on Multimedia Signal Processing (MMSP); August 2018. pp. 1-5

[9] Akbari A, Trocan M. Sparse recovery-based error concealment for multiview images. In: Proceedings of IEEE International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM); October 2015; Prague, Czech Republic. pp. 1-5

[10] Akbari A, Trocan M, Granado B. Synthesis sparse modeling: Application to image compression and image error concealment. In: Proceedings of Signal Processing with Adaptive Sparse Structured Representations Workshop (SPARS); June 2017; Lisbon, Portugal

[11] Mandache D, Akbari A, Trocan M, Granado B. Image compressed sensing recovery using intra-block prediction. In: Proceedings of IEEE International Conference on Telecommunications Forum (TELFOR); November 2015; Belgrade, Serbia. pp. 748-751

[12] Akbari A, Trocan M, Granado B. Image compression using adaptive sparse representations over trained dictionaries. In: Proceedings of IEEE International Workshop on Multimedia Signal Processing (MMSP); September 2016; Montreal, Canada. pp. 1-6

[13] Akbari A, Trocan M, Granado B. Image error concealment using sparse representations over a trained dictionary. In: Proceedings of IEEE Picture Coding Symposium (PCS); December 2016; Nuremberg, Germany. pp. 1-5

[14] Akbari A, Trocan M, Granado B. Residual based compressed sensing recovery using sparse representations over a trained dictionary. In: Proceedings of International ITG Conference on Systems, Communications and Coding (SCC); February 2017; Hamburg, Germany. pp. 1-6

[15] Akbari A, Trocan M, Granado B. Sparse recovery-based error

concealment. IEEE Transactions on Multimedia. 2017;**19**(6):1339-1350

[16] Akbari A, Trocan M, Granado B. Joint-domain dictionary learning-based error concealment using common space mapping. In: Proceedings of IEEE International Conference on Digital Signal Processing (DSP); August 2017. pp. 1-5

[17] Akbari A, Trocan M, Granado B. Image error concealment based on joint sparse representation and non-local similarity. In: Proceedings of IEEE Global Conference on Signal and Information Processing (GlobalSIP); November 2017

[18] Akbari A, Trocan M, Sanei S, Granado B. Joint sparse learning with nonlocal and local image priors for image error concealment. IEEE Transactions on Circuits and Systems for Video Technology. 2019

[19] Akbari A, Trocan M, Granado B. Image compressed sensed recovery using saliency-based adaptive sensing and residual reconstruction. In: Compressed Sensing: Methods, Theory and Applications. New York: Nova Science Publishers; 2018

[20] Akbari A, Trevisi M, Trocan M, Carmona-Galán R. Compressive imaging using rip-compliant cmos imager architecture and landweber reconstruction. IEEE Transactions on Circuits and Systems for Video Technology. 2019:1-1

[21] Akbari A, Trocan M. Robust image reconstruction for block-based compressed sensing using a binary measurement matrix. In: 2018 25th IEEE International Conference on Image Processing (ICIP); October 2018. pp. 1832-1836

[22] Akbari A, Trevisi M, Trocan M. Adaptive compressed sensing image reconstruction using binary measurement matrices. In: 2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS); December 2018. pp. 659-660

[23] Aharon M, Elad M, Bruckstein A. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing. 2006;**54**(11): 4311-4322

[24] Ahmed N, Natarajan T, Rao KR. Discrete cosine transform. IEEE Transactions on Computers. 1974;**C-23** (1):90-93

[25] G. K. Wallace, "The jpeg still picture compression standard," IEEE Transactions on Consumer Electronics, vol. 38, no. 1, pp. xviii-xxxiv, 1992

[26] Skodras A, Christopoulos C, Ebrahimi T. The jpeg 2000 still image compression standard. IEEE Signal Processing Magazine. 2001;**18**(5):36-58

[27] Rubinstein R, Bruckstein AM, Elad M. Dictionaries for sparse representation modeling. Proceedings of the IEEE. 2010;**98**(6):1045-1057

[28] Pati YC, Rezaiifar R, Krishnaprasad PS. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In: Proceedings of Asilomar Conference on Signals, Systems and Computers; November 1993; Pacific Grove, CA. pp. 40-44

[29] Candés E, Demanet L, Donoho D, Ying L. Fast discrete curvelet transforms. SIAM Multiscale Modeling & Simulation. 2006;**5**(3):861-899

[30] Do MN, Vetterli M. Contourlets: A new directional multiresolution image representation. In: Proceedings of Asilomar Conference on Signals, Systems and Computers, vol. 1; November 2002; Pacific Grove, CA. pp. 497-501

[31] Pennec EL, Mallat S. Sparse geometric image representations with bandelets. IEEE Transactions on Image Processing. 2005;**14**(4):423-438

[32] Engan K, Skretting K, Husoy JH. Family of iterative LS-based dictionary learning algorithms, ILS-DLA, for sparse signal representation. Digital Signal Processing. 2007;**17**(1):32-49

[33] Vidal R, Ma Y, Sastry S. Generalized principal component analysis (GPCA). IEEE Transactions on Pattern Analysis and Machine Intelligence. 2005;**27**(12): 1945-1959

[34] Sulam J, Ophir B, Zibulevsky M, Elad M. Trainlets: Dictionary learning in high dimensions. IEEE Transactions on Signal Processing. 2016;**64**(12): 3180-3193

[35] Lewicki MS, Sejnowski TJ. Learning overcomplete representations. Neural Computation. 2000;**12**(2):337-365

[36] Kreutz-Delgado K, Murray JF, Rao BD, Engan K, Lee TW, Sejnowski TJ. Dictionary learning algorithms for sparse representation. Neural Computation. 2003;**15**(2): 349-396

[37] Gurumoorthy KS, Rajwade A, Banerjee A, Rangarajan A. A method for compact image representation using sparse matrix and tensor projections onto exemplar orthonormal bases. IEEE Transactions on Image Processing. 2010;**19**(2):322-334

[38] Mazaheri JA, Guillemot C, Labit C. Learning a tree-structured dictionary for efficient image representation with adaptive sparse coding. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); May 2013; Vancouver, Canada. pp. 1320-1324

[39] Sun Y, Tao X, Li Y, Lu J. Dictionary learning for image coding based on multisample sparse representation. IEEE Transactions on Circuits and Systems for Video Technology. 2014;**24**(11): 2004-2010

[40] Peyre G. A review of adaptive image representations. IEEE Journal of Selected Topics in Signal Processing. 2011;**5**(5):896-911

[41] Borji A, Cheng MM, Jiang H, Li J. Salient object detection: A benchmark. IEEE Transactions on Image Processing. 2015;**24**(12):5706-5722

[42] Harel J, Koch C, Perona P. Graph-based visual saliency. In: Proceedings of Neural Information Processing Systems (NIPS); 2006. pp. 545-552

[43] Skretting K, Engan K. Image compression using learned dictionaries by rls-dla and compared with K-SVD. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); May 2011; Prague, Czech Republic. pp. 1517-1520

[44] Wang S, Zhang L, Liang Y, Pan Q. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR); June 2012; Providence, RI. pp. 2216-2223

[45] Huang DA, Wang YCF. Coupled dictionary and feature space learning with applications to cross-domain image synthesis and recognition. In: Proceedings of IEEE Conference on Computer Vision (ICCV); December 2013; Sydney, Australia. pp. 2496-2503