Image Compression Using Sparse Representations and the Iteration-Tuned and Aligned Dictionary

Joaquin Zepeda, Christine Guillemot, and Ewa Kijak

***Abstract*—We introduce a new image coder which uses the Iteration Tuned and Aligned Dictionary (ITAD) as a transform to code image blocks taken over a regular grid. We establish experimentally that the ITAD structure results in lower-complexity representations that enjoy greater sparsity when compared to other recent dictionary structures. We show that this superior sparsity can be exploited successfully for compressing images belonging to specific classes of images (e.g., facial images). We further propose a global rate-distortion criterion that distributes the code bits across the various image blocks. Our evaluation shows that the proposed ITAD codec can outperform JPEG2000 by more than 2 dB at 0.25 bpp and by 0.5 dB at 0.45 bpp, accordingly producing qualitatively better reconstructions.**

***Index Terms*—Image coding, learned dictionaries, matching pursuit algorithms, sparse representations, transform coding.**

I. Introduction

A SPARSE representation of a signal consists of a linear combination of vectors known as *atoms* taken from a predefined, generally overcomplete transform known as the *dictionary*. The representation is called *sparse* because it employs only a small number of atoms from the dictionary. Signals that can be well represented sparsely are termed *compressible* under the given dictionary. Knowing beforehand that a class of signals is compressible in a given dictionary has proven to be a very powerful tool. For example, an added sparsity prior can be exploited to render an undercomplete linear system solvable. A good example of a practical application of this property is inpainting [1], where a missing signal region can be reconstructed from a neighborhood. Other common applications of sparsity include image denoising [2] and texture separation [3].

The application considered in this paper is image compression, where sparse representations consisting of only a few nonzero coefficients are ideal to produce compact representations of image blocks. Indeed the JPEG [4] standard is based on the premise that natural images are compressible in the DCT basis. Its successor, the JPEG2000 standard, instead substitutes a wavelet basis that better satisifies the compressibility requirement for natural images. An adaptation of the JPEG standard is still very much in use for intra-coding of video frames in the H.264 standard [5]. However, note that compression algorithms based on DCT or wavelet transforms are generic, in the sense that they are not tailored to particular classes of images.

On the other hand, recent research effort has been dedicated to learning dictionaries which would thus be adapted to a signal class for the purpose of image compression. By using a learned dictionary, the image encoder can benefit from the ensuing greater compressibility of the considered signal class. An example of this approach is embodied in the facial image codec based on the K-SVD dictionary introduced by Bryt and Elad [6]. Their approach nonetheless employs a piecewise-affine warping of the face that ensures that the various facial features coincide with those of a pre-specified face template. Each block of the face template (corresponding roughly to a facial feature such as the nose) defines a class of signals that is then represented with a corresponding K-SVD dictionary. Thus, the compressibility of the image blocks in that approach relies, to a large extent, not on the K-SVD dictionary but rather on the affine warping procedure. This warping procedure in turn increases the codec complexity and is further sensitive to image variations encountered in practical scenarios (e.g., in lighting conditions, pose and particularities of the subject).

Another example of an image compression system based on trained overcomplete dictionaries is that developed by Sezer *et al.* [7]. Their dictionary structure consists of a concatenation of orthogonal bases. A single one of these bases is selected to encode any given image block of fixed size. This approach has the advantage that it reduces the atom-index coding overhead, yet this comes at the cost of reduced effective dictionary size.

Recently, the Iteration-Tuned Dictionary (ITD) framework has been introduced [8], [9] consisting of a dictionary structure adapted to the iterative nature of greedy pursuit algorithms such as those of the matching pursuit family [10]–[12]. In the ITD framework, the dictionary used at the different iterations of the iterative pursuit algorithm vary along with the iterations. These dictionaries are trained to be adapted to the characteristics of the residues produced in sucessive iterations. The Iteration-Tuned and Aligned Dictionary (ITAD) [9] is a particular ITD variant which implements a tree-structured ITD scheme, employing a set of rotation matrices to reduce the storage overhead of the tree structure. The resulting scheme shares some similarities to gain-shape vector quantization (GSVQ) and multi-stage vector quantization (MSVQ) [13]. Relationships between Sparse Representations (SR) and GSVQ have previously been pointed out in [14]: in both cases, *atoms* (*codevectors* in vector quantization terms) are selected (one for GSVQ, multiple for SR) each along with a real-valued representation coefficient. ITAD is further similar to MSVQ in that the ith atom selection is done using the $(i-1)$th residual as an input to the ith dictionary *stage* (called a *layer* in the context of ITAD).

Manuscript received October 11, 2010; revised January 21, 2011; accepted March 09, 2011. Date of publication April 05, 2011; date of current version August 17, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Michael Elad.

The authors are with IRISA/INRIA, 35042 Rennes Cedex, France (e-mail: my_joaqal@gmail.com; christine.guillemot@irisa.fr; ewa.kijak@irisa.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2011.2135332

In this paper, we introduce a new image codec for image classes based on the ITAD transform trained for the particular image class. In the codec we introduce, the ITAD transform is used to encode the mean-removed image blocks, while the block-mean is encoded using a common DPCM-based arrangement. The ITAD transform coefficients are encoded using a simple uniform quantizer/entropy encoder combination, while the atom indices are encoded using a fixed-length code. We further introduce a new global method for jointly selecting the sparsity of the image blocks based on a rate-distortion criterion. The proposed ITAD codec is shown to outperform the JPEG and JPEG2000 compression standards in quantitative and qualitative evaluations when used on specific classes of images such as facial images, for which the transform can be well adapted.

We begin in the following (Section II) by first reviewing the ITAD structure that is at the core of our codec. Then, in the subsequent (Section III), we introduce the codec structure and, in particular, develop a global (image-wide) rate-distortion criterion that allows us to better select the sparsity of each of the component image blocks. In the results (Section IV), we first compare the ITAD dictionary that is at the heart of our codec to other recent (learned) dictionaries and thus demonstrate the enhanced compressibility that it provides. We then carry out both quantitative and qualitative evaluations of our codec, comparing it against both JPEG and JPEG2000 standards. Finally, we provide some concluding remarks in the conclusion (Section V).

II. Background: The Iteration-Tuned and Aligned Dictionary (ITAD)

Given a dictionary $\mathbf{D} \in \mathbb{R}^{d\times N}$, a sparse representation of a signal vector $\mathbf{y} \in \mathbb{R}^d$ defines a linear combination of L atoms $\mathbf{d}$ (assumed unit norm) which are defined as columns of matrix $\mathbf{D}$, such that $\mathbf{D} = [\mathbf{d}_a]_{a=1}^N$. The sparsity L is small compared to the dimension d of input signal vectors, and generally either fixed or determined for each vector $\mathbf{y}$ based on a distorsion error criterion.

For each iteration i of an iterative matching pursuit (MP) decomposition [10], the atom $\mathbf{d}_{a_i}$ of the dictionary $\mathbf{D}$ that best approximates the input vector is selected and the associated coefficient γ_i is computed, until the sparsity L is reached. Denoting an input vector of the ith iteration as $\mathbf{r}^{i-1}$, and starting with $\mathbf{r}^0 = \mathbf{y}$, the column index a_i in the dictionary of the selected atom at the ith iteration and its associated coefficient γ_i are computed as follows:

$$a_i = \underset{a}{\operatorname{argmax}} \left|(\mathbf{d}_a)^\top \mathbf{r}^{i-1}\right| \tag{1a}$$

$$\gamma_i = (\mathbf{d}_{a_i})^\top \mathbf{r}^{i-1} \tag{1b}$$

where we assume that $\|\mathbf{d}_a\| = 1$. Then at the output of the ith iteration, the residual $\mathbf{r}^i$ is defined as

$$\mathbf{r}^i = \mathbf{r}^{i-1} - \gamma_i \mathbf{d}_{a_i} \tag{2}$$

and the signal approximation $\hat{\mathbf{y}}^i$ satisfies

$$\hat{\mathbf{y}}^i = \mathbf{y} - \mathbf{r}^i. \tag{3}$$

The proposed ITDs are overcomplete dictionaries that are structured to better suit the iterative nature of greedy pursuit decomposition algorithms such as those of the matching pursuit family [10]–[12]. This means that a different dictionary is used at each iteration i of the sparse decomposition. In the simplest case, one can consider one unique dictionary $\mathbf{D}^i$ per iteration i (also referred as layer i) [8]. In a more general way, there can be several *candidate dictionaries* per layer i. Then ITDs consist of a sequence of layers, with each layer i containing a set of K_i dictionaries $\{\mathbf{D}^{i,k}\}_{k=1}^{K_i}$. For simplicity, we assume throughout that every dictionary $\mathbf{D}^{i,k}$ of any layer contains the same number N of atoms.

An ITD-based matching pursuit (ITD-MP) decomposition proceeds as for the case of traditional fixed-dictionary decompositions until L pairs (atom-index, coefficient) are chosen, except that atoms selected at each iteration belong to different dictionaries. Thus, for each iteration i, a candidate dictionary $\mathbf{D}^{i,k_1}$ is first chosen, and then an atom $\mathbf{d}_{a_i}^{k_i} \in \mathbf{D}^{i,k}$ and a coefficient γ_i are selected according to the MP selection rules (1). One problem with this approach lies in the selection of the dictionary $\mathbf{D}^{i,k_1}$ among the K_i candidates at iteration i. We address this problem in the following while defining a relation between the dictionaries of the different layers based on a tree-structure.

A. Tree-Structured Iteration-Tuned Dictionary

The Tree-Structured Iteration-Tuned Dictionary (TSITD) [9] is an ITD variant in which each candidate dictionary $\mathbf{D}^{i,k}$ of a layer i is the child of an atom $\mathbf{d}_a^{i-1,l} \in \mathbf{D}^{i-1,l}$, $l \in \{1,\ldots,K_{i-1}\}$ from the previous layer $i-1$. Thus the candidate dictionary selected at the ith iteration of the pursuit decomposition is the child of the atom chosen in the previous iteration. Hence, each dictionary $\mathbf{D}^{i,k}$ is uniquely specified by the ordered sequence of *ancestor atoms* $(\mathbf{d}_{a_j}^{j,k_j})_{j=1}^{i-1}$, selected in all previous layers. Then to simplify notations, each dictionary $\mathbf{D}^{i,k}$ can be uniquely designed by a path through the tree defined by the sequence of column-index of its ancestor atoms $\mathcal{A}_i = (a_1,\ldots,a_{i-1})$, with $\mathcal{A}_1 = \emptyset$ and $\mathbf{D}^{\mathcal{A}_1}$ being the unique dictionary in the first layer.

Using these notations, the MP atom-index and coefficient selection rules at the ith TSITD layer can be written as

$$a_i = \underset{a}{\operatorname{argmax}} \left|\left(\mathbf{d}_a^{\mathcal{A}_i}\right)^\top \mathbf{r}^{i-1}\right|, \quad \text{with } \mathbf{d}_a^{\mathcal{A}_i} \in \mathbf{D}^{\mathcal{A}_i} \tag{4a}$$

$$\gamma_i = \left(\mathbf{d}_{a_i}^{\mathcal{A}_i}\right)^\top \mathbf{r}^{i-1} \tag{4b}$$

where $\mathbf{r}^i = \mathbf{r}^{i-1} - \gamma_i \mathbf{d}_{a_i}^{\mathcal{A}_i}$, $\mathbf{r}^0 = \mathbf{y}$ and again we have assumed that $\|\mathbf{d}_a^{\mathcal{A}_i}\| = 1$.

We group the atoms selected up to the ith pursuit iteration to form the *selected-atoms matrix*

$$\mathbf{S}^{\mathcal{A}_i} = [\,\mathbf{d}_{a_1}^{\mathcal{A}_1} \quad \ldots \quad \mathbf{d}_{a_i}^{\mathcal{A}_i}\,]. \tag{5}$$

Accordingly grouping the representation coefficients γ_i to form the coefficients vector $\mathbf{\Gamma}^{\mathcal{A}_i} = [\gamma_1 \ \ldots \ \gamma_i]^\top$, we denote the resulting approximation at the output of the ith layer as

$$\hat{\mathbf{y}}^i = \mathbf{S}^{\mathcal{A}_i}\mathbf{\Gamma}^{\mathcal{A}_i}. \tag{6}$$

1) TSITD Training: The TSITD structure training is performed according to a top-down approach in which each TSITD

layer i is trained using the $(i-1)$th residuals of a training set. According to the MP decomposition rules, at each layer i a single atom from $\mathbf{D}^{\mathcal{A}_i}$ will be chosen to approximate a layer's input residual. Then the output residuals of the ith layer are partitioned into N subsets according to which atom-index a_i has been selected. Each of the resulting residuals subsets $\{\mathbf{r}^i\}_{A_{i+1}}$ is used to train the corresponding dictionary $\mathbf{D}^{\mathcal{A}_{i+1}}$ in the next iteration.

The training procedure of each $\mathbf{D}^{\mathcal{A}_i}$ is similar to the K-SVD algorithm [6] when the sparsity constraint L is equal to 1, and proceeds as follows. First, $\mathbf{D}^{\mathcal{A}_i}$ is initialized with a random selection, without repetition, of N training vectors from the set of training residues $\{\mathbf{r}^{i-1}\}_{A_i}$ produced by the parent atom $\mathbf{d}_{a_{i-1}}^{\mathcal{A}_{i-1}}$ (by convention, $\{\mathbf{r}^{i-1}\}_{A_1} = \{\mathbf{y}\}$). Then, a classification and an update step are repeated until convergence. During the classification step, each training vector $\mathbf{r}^{i-1} \in \{\mathbf{r}^{i-1}\}_{A_i}$ is assigned to an atom $\mathbf{d}_a^{\mathcal{A}_i} \in \mathbf{D}^{\mathcal{A}_i}$ according to (4a). We define the class matrices $\mathbf{R}_a^{\mathcal{A}_i}, a = 1,\ldots,N$ having as columns the training residual vectors $\mathbf{r}^{i-1}$ that are projected onto $\mathbf{d}_a^{\mathcal{A}_i}$.

Each atom $\mathbf{d}_a^{\mathcal{A}_i}$ is then updated using the first left singular vector of the corresponding training class matrix $\mathbf{R}_a^{\mathcal{A}_i}$, where the singular vectors are assumed ordered according to decreasing magnitude of their singular value. Once the algorithm has converged, the coefficients γ_i are obtained from (4b), and the residues $\mathbf{r}^i$ used as input for the following layer can be computed (2).

2) TSITD Properties: The tree-structure of the TSITD scheme combined with the particular dictionary training scheme results in two implications [9] that will be useful in the remainder of the paper. First, TSITD candidates produce selected-atoms matrices $\mathbf{S}^{\mathcal{A}_i}$ that are orthogonal:

$$(\mathbf{S}^{\mathcal{A}_i})^\top \mathbf{S}^{\mathcal{A}_i} = \mathbf{I} \quad \forall i. \tag{7}$$

Second, it follows from the above that all the atoms of a dictionary $\mathbf{D}^{\mathcal{A}_i}$ must be orthogonal to all its ancestor atoms appearing in $\mathbf{S}^{\mathcal{A}_{i-1}}$

$$(\mathbf{S}^{\mathcal{A}_{i-1}})^\top \mathbf{D}^{\mathcal{A}_i} = \mathbf{0}. \tag{8}$$

This last expression implies that the rank of $\mathbf{D}^{\mathcal{A}_i}$ is $d-i+1$, and thus, using an adequate rotation matrix, the atoms of $\mathbf{D}^{\mathcal{A}_i}$ (which are contained in the *signal space* $\mathbb{R}^d$) can be expressed equivalently in the *reduced space* $\mathbb{R}^{d-i+1}$. Formally, this relation can be stated as follows for a given candidate dictionary $\mathbf{D}^{\mathcal{A}_i}$:

$$\mathbf{D}^{\mathcal{A}_i} = \mathbf{\Phi}^{\mathcal{A}_{i-1}} \mathbf{D}^{\mathcal{A}_i\prime} \tag{9}$$

where $\mathbf{Phi}^{\mathcal{A}_{i-1}} \in \mathbb{R}^{d\times(d-i+1)}$ is a rotation matrix and $\mathbf{D}^{\mathcal{A}_i\prime} \in \mathbb{R}^{(d-i+1)\times N}$ is the reduced-space representation of the candidate dictionary.

B. ITAD Structure

The ITAD structure aims to reduce the storage footprint of the TSITD tree by constraining all $\mathbf{D}^{\mathcal{A}_i\prime}$ of a given layer i to be equal. We use $\mathbf{D}^{i\prime}$ to denote this reduced dictionary common to all nodes of layer i and refer to it as the *prototype dictionary* of

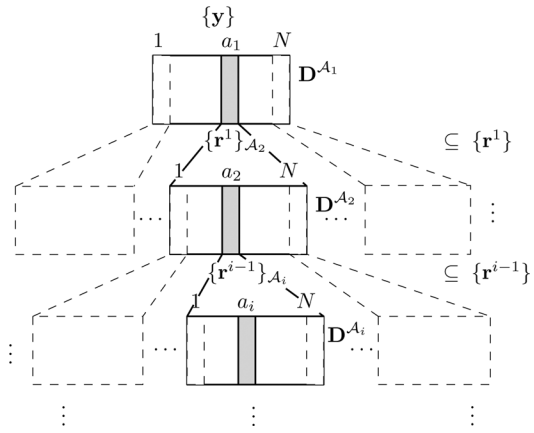

Fig. 1. TSITD structure and training residual subsets.

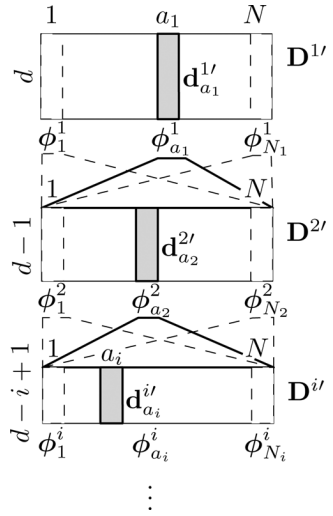

FIg. 2. The ITAD structure. The equivalent signal-space representation is illustrated in Fig. 1. We use subscript a_i indices to differentiate the atoms $\mathbf{d}^{i\prime}$ (and their corresponding alignment matrices $\boldsymbol{\phi}^i$) of the same prototype dictionary $\mathbf{D}^{i\prime}$.

the ith layer. The storage gain follows from the fact that a single prototype dictionary $\mathbf{D}^{i\prime}$ needs to be stored per layer as opposed to one reduced dictionary $\mathbf{D}^{\mathcal{A}_i\prime}$ per node.

The rotation matrices $\mathbf{\Phi}^{\mathcal{A}_{i-1}}$ can also be stored efficiently by factoring them as $\mathbf{\Phi}^{\mathcal{A}_{i-1}} = \boldsymbol{\phi}_{a_1}^1 \cdot \ldots \cdot \boldsymbol{\phi}_{a_1}^{i-1}$, where $\{\boldsymbol{\phi}_{a_j}^j\}_{j=1}^{i-1}$ are called *alignment matrices* and carry out the dimensionality reduction of residuals at the output of successive layers.[1] Each alignment matrix $\boldsymbol{\phi}_{a_j}^j \in \mathbb{R}^{(d-j+1)\times(d-j)}$ is uniquely associated to the atom $\mathbf{d}_{a_j}^{j\prime} \in \mathbb{R}^{(d-j+1)}$ of the prototype dictionary $\mathbf{D}^{j\prime}$.

This is illustrated in Fig. 2.

C. Signal Decompositions and Reconstruction Using ITAD

The ITAD structure illustrated in Fig. 2 can be used directly to implement an ITAD adaptation of the MP decomposition algorithm by first defining the reduced residuals as follows:

$$\mathbf{r}^{i\prime} \triangleq (\boldsymbol{\phi}^i)^\top \left(\mathbf{r}^{(i-1)\prime} - \gamma_i \mathbf{d}_{a_i}^{i\prime}\right) \in \mathbb{R}^{d-i} \tag{10}$$

[1]Any $\boldsymbol{\phi}^j \in \mathbb{R}^{(d-j+1)\times(d-j)}$ satisfying $(\boldsymbol{\phi}^j)^\top \mathbf{d}^{j\prime} = 0$ will accomplish this task, and thus the ITAD alignment matrices are further chosen (cf. Section II-E) to better structure the residuals for representation with the single prototype dictionary of the next layer.

where, by convention, we let $\mathbf{r}^{0\prime} = \mathbf{y}$. Using this definition, the MP atom and coefficient selection rules at the ith ITAD layer follow directly:

$$a_i = \underset{a}{\operatorname{argmax}} \left| \left(\mathbf{d}_a^{i\prime}\right)^{\mathsf{T}} \mathbf{r}^{(i-1)\prime} \right|, \quad \text{with } \mathbf{d}_a^{i\prime} \in \mathbf{D}^{i\prime} \tag{11a}$$

$$\gamma_i = \left(\mathbf{d}_{a_i}^{i\prime}\right) \mathbf{r}^{(i-1)\prime}. \tag{11b}$$

The structure in Fig. 2 can also be used directly to carry out the reconstruction operation by defining the reduced accumulator vector

$$\check{\mathbf{r}}^{(i-1)\prime} \triangleq \gamma_i \cdot \mathbf{d}^{i\prime} + \boldsymbol{\phi}^i \check{\mathbf{r}}^{i\prime} \in \mathbb{R}^{d-i+1} \tag{12}$$

where, by convention, we let $\check{\mathbf{r}}^{L\prime} = \mathbf{0}$. The final reconstruction is given by $\check{\mathbf{r}}^{0\prime}$.

D. Reduced-Space/Signal-Space Representations

The ITAD structure based on reduced candidate dictionaries (i.e., prototype dictionaries) $\mathbf{D}^{i\prime}$ illustrated in Fig. 2 has two practical advantages over the equivalent representation that uses the signal-space tree structure: First, the ITAD structure in Fig. 2 has a storage footprint that is significantly smaller than that of the signal-space tree-structured representation. Second, carrying out the decomposition and reconstruction operations in the reduced spaces (as discussed in Section II-C) decreases the combined decomposition/reconstruction complexity for high i, making it even lower than that of traditional fixed-dictionary schemes using OMP.

While the signal-space dictionaries $\mathbf{D}^{\mathcal{A}_i}$ will not enjoy these advantages, we will find them, nonetheless, useful for analysis purposes in Section III and onwards given that the signal-space decomposition and reconstruction expressions [(4) and (6)] are simpler than the corresponding, mathematically equivalent, reduced-space operations [respectively, (10), (11) and (12)].

E. ITAD Training

The ITAD structure in Fig. 2 is trained using the top-down approach described in Section II-A1. Differences lie in that the $(i-1)$th reduced residuals $\mathbf{r}^{(i-1)\prime} \in \mathbb{R}^{d-i+1}$ are used as training set for the ITAD layer i, and that since all reduced residuals reside in the same subspace, the next layer dictionary is trained using all the residuals simultaneously, without partitioning them.

In addition, during the training, not only the atoms $\mathbf{d}_a^{i\prime}$ are updated using the first left singular vector of the corresponding training class matrix $\mathbf{R}_a^{i\prime}$. The alignment matrices $\boldsymbol{\phi}_a^i$ corresponding to atoms $\mathbf{d}_a^{i\prime}$ are also updated using all remaining left singular vectors, except the first. The coefficients γ_i are obtained from (11b), and the residuals $\mathbf{r}^{i\prime}$ used as input for the following layer from (10).

This approach ensures firstly that the chosen atom is the best atom for single-atom approximations of the corresponding subclass and second that the corresponding aligned residuals of each class share a common left singular vectors basis (and accordingly, a more similar structure) at the input of the subsequent layer $i+1$.

An example of the ITAD candidate dictionaries produced by the above described algorithm is shown in Fig. 3, represented in signal-space for visualization purposes.

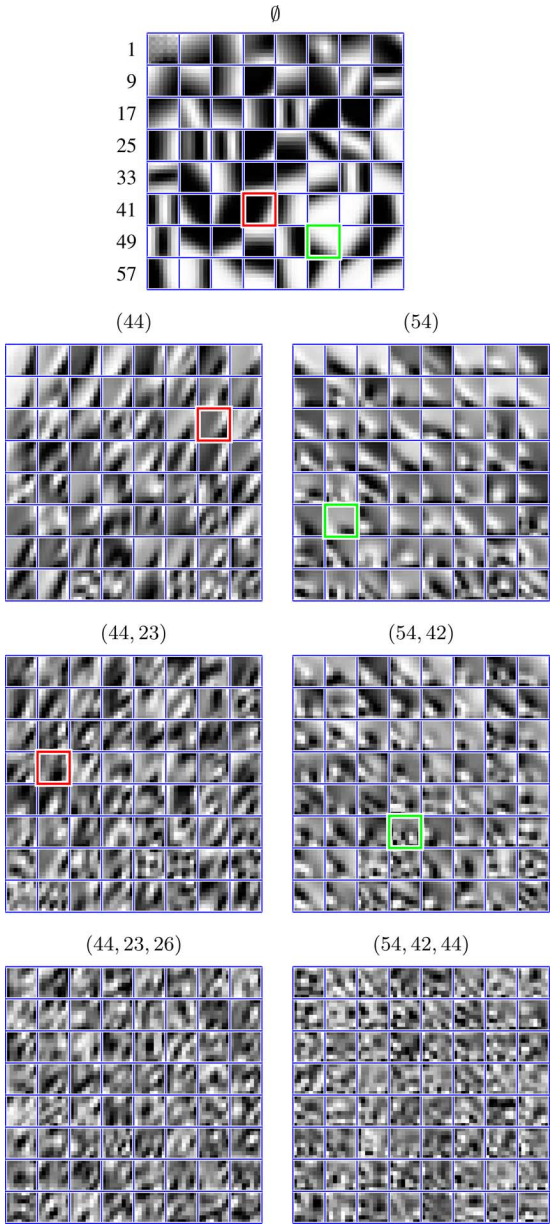

Fig. 3. Example of ITAD candidate dictionaries found along two branches across the first four layers (represented in signal-space). The ancestor atom indices $\mathcal{A}_i = (a_1, \ldots, a_{i-1})$ of each dictionary are noted on top of each figure. The immediate parent atom of each dictionary is the highlighted atom found in the candidate dictionary directly above.

III. Proposed Image CODEC

We now present the proposed ITAD image codec that is the main contribution of this paper. The encoder uses the ITAD transform discussed in the background section to compress the mean-removed component of image blocks taken over a regular grid; we refer to the mean of a block and a block's mean-removed version as its DC and AC components, respectively. In order to determine the sparsity of each individual block, we derive a global sparsity criterion that accounts for the global image rate and distortion.

A. Block Slicer and AC/DC Splitter

The block diagram in Fig. 4 illustrates the major components of our proposed image encoder. The first step of the process

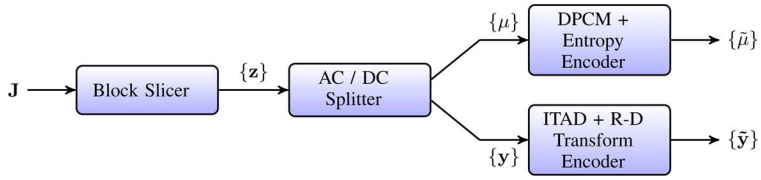

Fig. 4. The ITAD-based image codec: The input image $\mathbf{J}$ is split into disjoint blocks $\mathbf{z}$. The DC and AC components μ and $\mathbf{y}$ of each block $\mathbf{z}$ are then encoded separately. The operation of the bottom-right AC encoding block *ITAD/R-D Transform Coding* is described in Figs. 5 and 6.

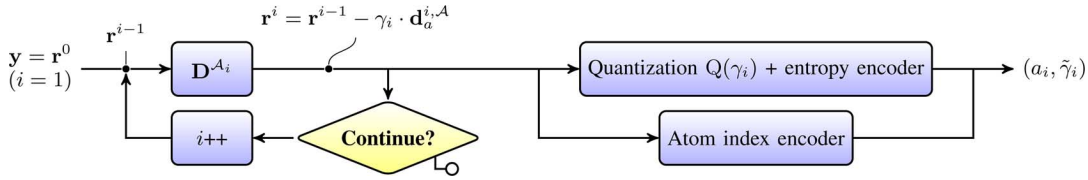

Fig. 5. The ITAD Block Coder (BC): Sparse decomposition at the ith ITD-MP iteration and subsequent encoding of the atom-index/coefficient pair.

consists of slicing a given input image $\mathbf{J}$ into non-overlapping $b \times b$ blocks using a regular grid. Vector $\mathbf{z} \in \mathbb{R}^d$ (with $d = b^2$) denotes the vectorized version of one of these blocks, and thus the input image is represented at the output of the block-slicer by an ordered set of blocks $\mathbf{z}$:

$$\mathbf{J} \iff \{\mathbf{z}\}. \tag{13}$$

As illustrated in Fig. 4, each block $\mathbf{z}$ is subsequently split into a DC and an AC component, and the resulting AC and DC streams are encoded separately. We let μ denote the mean component of each block, given by

$$\mu = \frac{1}{d}\sum_{k=1}^{d} \mathbf{z}(k) \tag{14}$$

$\mathbf{y}$ will denote the AC (mean-removed) version obtained by subtracting μ from each entry of $\mathbf{z}$.

B. DPCM Coding of DC Components

The DC components μ are encoded using an approach similar to that of various block-based codecs including the JPEG image coder [4]. The approach exploits the spatial correlation of the DC coefficients by means of a differential pulse code modulation (DPCM) applied to the DC stream $\{\mu\}$. By convention, we order the DC stream using row-wise rastering and reversal of order from line to line to better exploit the spatial correlation of the DC coefficients. The DPCM symbols are subsequently encoded using an entropy encoder. The resulting coded version of the DC values is denoted $\tilde{\mu}$.

Note that, following removal of the DC components, all AC blocks $\mathbf{y}$ are contained in the orthogonal complement of the all-ones vector $\mathbf{1}$. The fact that AC and DC components are orthogonal means that the distortion at the output of the codec is the sum of the distortions of the two components. For this reason, we can disregard the DC component in the subsequent discussion on AC encoding and instead focus on minimizing the AC distortion.

C. ITAD-Based Transform Coding of AC Components

The remaining AC component blocks $\mathbf{y}$ are compressed using an encoder based on the ITAD transform in Fig. 2. The proposed encoder distributes bits to the various AC blocks $\mathbf{y}$ using a rate/distortion criterion that we will derive shortly. Throughout the discussion we assume that the ITAD structure has been trained using a large number of examples $\mathbf{y}$ taken from a set of training images $\mathbf{J}$ according to the algorithm described in Section II-E.

1) ITAD Sparse Decomposition of AC Blocks: In the left-hand side of Fig. 5 we illustrate the ITAD atom index/coefficient selection process. For ease of analysis, we have used the signal-space representation of the residual vectors $\mathbf{r}^i$ and of the ITAD candidate dictionaries $\mathbf{D}^{\mathcal{A}_i}$. The sparse decomposition of the signal blocks $\mathbf{y}$ proceeds iteratively, with each iteration of the loop selecting one atom $\mathbf{d}_{a_i}^{\mathcal{A}_i}$ and a corresponding coefficient γ_i from the ith ITAD layer and accordingly producing the ith residual vector $\mathbf{r}^i$.

2) Quantization of the Coefficients: The coefficients γ_i selected using the ITAD structure need to be quantized in order to produce a compact representation of each $\mathbf{y}$. We use $\tilde{\gamma}_i$ to denote the quantized version of γ_i and, accordingly, we let $\tilde{\mathbf{\Gamma}}^i = [\tilde{\gamma}_1 \ \ldots \ \tilde{\gamma}_i]$ denote the quantized version of the coefficients vector $\mathbf{\Gamma}^i$.

The coefficient encoding strategy consists of one uniform scalar quantizer common to all layers i of the ITAD structure:

$$\tilde{\gamma}_i = \mathrm{Q}(\gamma_i). \tag{15}$$

The quantized symbols $\tilde{\gamma}_i$ are then encoded using a Huffman entropy encoder unique to each layer i. The Huffman code table is constructed offline on the training set of images.

Previous work [15] on quantization of coefficients from overcomplete transforms has considered adding the quantization step in the atom selection process illustrated in the left-hand side loop of Fig. 5 by substituting $\tilde{\gamma}_i$ when building the residual $\mathbf{r}^i$ at the input of the following iteration. For the case of general overcomplete transforms, this approach produces better sparse representations under quantization because the latter iterations of the decomposition process consider the non-orthogonality between residual and selected atom(s) resulting form the finite precision of the quantized projection coefficients $\tilde{\gamma}_i$. However, as in the case of general orthogonal transforms, the orthogonality of the ITAD selected-atoms matrices implies that ITAD sparse decompositions do not require this extra consideration at the encoder.

3) Atom Encoding: The decoder needs to know which atoms $\mathbf{d}_{a_i}^{\mathcal{A}_i}$ have been chosen at the encoder, and thus the encoder needs

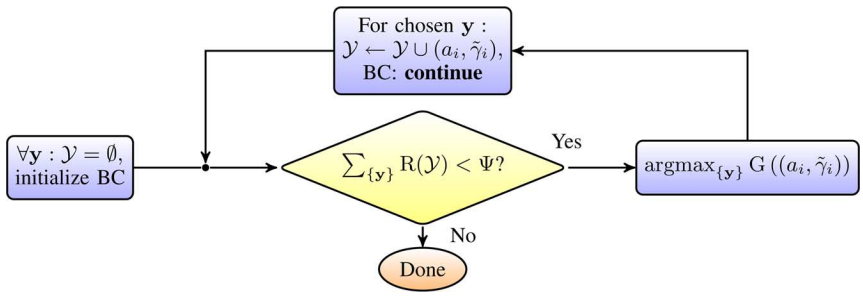

Fig. 6. Rate-distortion transform coding using ITAD. Each of the image blocks $\mathbf{y}$ is assumed to have a corresponding block coder (BC) (illustrated in Fig. 5).

to transmit the index a_i of each atom selected. The order of these atom indices is important, as we let the position of each index a_i in the ordered set $(a_1, \ldots, a_i, \ldots a_{L_\mathbf{y}})$ (where the sparsity $L_\mathbf{y}$ denotes the number of atoms used in the approximation of the input vector $\mathbf{y}$) specify the layer i to which it refers.

We use a fixed length code to encode each atom index a_i, since we have observed that there is only a small gain resulting from using an entropy code for the atom indices. Assuming that the ith ITAD dictionary contains N atoms, each atom index thus incurs a rate penalty of

$$\mathrm{R}(a_i) = \log_2(N). \tag{16}$$

D. Block Sparsity Selection Using a Global Rate-Distortion Criterion

Any image compression system aims to maximize reconstruction fidelity under a rate constraint. In order to arrive at a formal expression for this goal in the context of ITAD, we denote the ith approximation (6) under coefficient quantization as

$$\tilde{\mathbf{y}}^i = \mathbf{S}^{\mathcal{A}_i} \tilde{\mathbf{\Gamma}}^i. \tag{17}$$

At the output of the encoder, each image block $\mathbf{y}$ is then represented by the approximation $\tilde{\mathbf{y}}^{L_\mathbf{y}} = \mathbf{S}^{\mathcal{A}_{L_\mathbf{y}}} \tilde{\mathbf{\Gamma}}^{L_\mathbf{y}}$, where the sparsity value $L_\mathbf{y}$ generally differs for each block $\mathbf{y}$. Each approximation $\tilde{\mathbf{y}}^{L_\mathbf{y}}$ is in turn completely specified by an ordered set of atom-index/quantized-coefficient pairs

$$\mathcal{Y} = \{(a_i, \tilde{\gamma}_i)\}_{i=1}^{L_\mathbf{y}} \tag{18}$$

that incurs a rate penalty of $\mathrm{R}(\mathcal{Y})$.

1) Problem Formulation: Using this notation, we can express the initially stated rate-constrained representation fidelity goal as the following constrained optimization problem:

$$\underset{\{\mathcal{Y}\}}{\mathrm{argmin}} \sum_{\{\mathbf{y}\}} \|\mathbf{y} - \tilde{\mathbf{y}}^{L_\mathbf{y}}\|^2 \,\mathrm{s.t.}\, \sum_{\{\mathbf{y}\}} \mathrm{R}(\mathcal{Y}) \leq \Psi \tag{19}$$

where Ψ denotes the maximum rate allocated to the image.

2) Solution Strategy: The above stated problem is difficult to solve exactly and would likely require an untractable combinatorial approach. We thus consider the following strategy (illustrated in Fig. 6) to approximate the solution: we build the reconstructed image by first initializing the approximations of all blocks to zero:

$$\tilde{\mathbf{y}}^0 = \mathbf{0},\ \mathcal{Y} = \emptyset, \quad \forall \mathbf{y} \tag{20}$$

accordingly initializing one block coder (illustrated in Fig. 5) per image block. We then select one image block $\mathbf{y}$ at a time and improve its approximation $\tilde{\mathbf{y}}^{i-1}$ (note from Fig. 5 that i is initialized to unity) by adding a single $(a_i, \tilde{\gamma}_i)$ pair to its representation $\mathcal{Y}$, repeating the block selection and improvement process as long as the rate constraint in (19) has not been reached.

At any one time, the block $\mathbf{y}$ chosen for improvement of its approximation will be the one that best fits the aim of the constrained problem in (19). Noting that the new $(a_i, \tilde{\gamma}_i)$ pair will incur a rate penalty of $\mathrm{R}((a_i, \tilde{\gamma}_i))$ bits, a good, low-complexity estimate of the block that best corresponds to (19) is that producing the largest reduction in the error per bit

$$\mathrm{G}((a_i, \tilde{\gamma}_i)) = \frac{\|\mathbf{y} - \tilde{\mathbf{y}}^{i-1}\|^2 - \|\mathbf{y} - \tilde{\mathbf{y}}^{i}\|^2}{\mathrm{R}((a_i, \tilde{\gamma}_i))}. \tag{21}$$

To simplify this above expression, we recall that at the dth decomposition iteration (where d is the signal dimension) the approximation $\hat{\mathbf{y}}^d$ of $\mathbf{y}$ is exact: $\mathbf{y} = \mathbf{S}^{\mathcal{A}_d} \mathbf{\Gamma}^d$. The ith residual vector $(1 \leq i < d)$ under the influence of quantization can be expressed using (3) as

$$\tilde{\mathbf{r}}^i = \mathbf{y} - \tilde{\mathbf{y}}^i = \mathbf{S}^{\mathcal{A}_d} \mathbf{\Gamma}^d - \mathbf{S}^{\mathcal{A}_i} \tilde{\mathbf{\Gamma}}^i \tag{22}$$

$$= \mathbf{S}^{\mathcal{A}_i} (\mathbf{\Gamma}^i - \tilde{\mathbf{\Gamma}}^i) + \bar{\mathbf{S}}^{\mathcal{A}_{i+1}} \bar{\mathbf{\Gamma}}^{i+1} \tag{23}$$

where $\bar{\mathbf{S}}^{\mathcal{A}_{i+1}} = [\mathbf{d}_{a_{i+1}}^{\mathcal{A}_{i+1}} \ \ldots \ \mathbf{d}_{a_d}^{\mathcal{A}_d}]$ and $\bar{\mathbf{\Gamma}}^{i+1} = [\gamma_{i+1} \ \ldots \ \gamma_d]^\mathsf{T}$ contain the atoms and coefficients corresponding to layers $(i+1), \ldots, d$.

From the orthogonality of the matrix $\mathbf{S}^{\mathcal{A}_d}$ we know that the right-hand side term of this last expression contains the component of $\mathbf{y}$ existing in the orthogonal complement space of the selected atoms matrix $\mathbf{S}^{\mathcal{A}_i}$ corresponding to the ith residual $\mathbf{r}^i$. We further define the quantization error vector $\mathbf{q}^i \triangleq (\mathbf{\Gamma}^i - \tilde{\mathbf{\Gamma}}^i)$ and, using $\mathbf{r}^i = \bar{\mathbf{S}}^{\mathcal{A}_{i+1}} \bar{\mathbf{\Gamma}}^{i+1}$, thus rewrite (23) as

$$\tilde{\mathbf{r}}^i = \mathbf{S}^{\mathcal{A}_i} \mathbf{q}^i + \mathbf{r}^i. \tag{24}$$

The numerator of (21) can then be expanded as follows:

$$\begin{aligned}&\|\mathbf{y}-\tilde{\mathbf{y}}^{i-1}\|^2-\|\mathbf{y}-\tilde{\mathbf{y}}^{i}\|^2\\&\quad=\|\mathbf{S}^{\mathcal{A}_{i-1}}\mathbf{q}^{i-1}+\bar{\mathbf{S}}^{\mathcal{A}_i}\bar{\mathbf{\Gamma}}^{i}\|^2\\&\qquad-\|\mathbf{S}^{\mathcal{A}_i}\mathbf{q}^{i}+\bar{\mathbf{S}}^{\mathcal{A}_{i+1}}\bar{\mathbf{\Gamma}}^{i+1}\|^2.\end{aligned}\tag{25}$$

Letting q_j denote the jth element of $\mathbf{q}$, the above expression can be further expanded as

$$\begin{aligned}&\left(\sum_{j=1}^{i-1}q_j^2+\sum_{j=i}^{d}\gamma_j^2\right)-\left(\sum_{j=1}^{i}q_j^2+\sum_{j=i+1}^{d}\gamma_j^2\right)\\&\quad=-q_i^2+\gamma_i^2&(26)\\&\quad=\gamma_i^2-(\gamma_i-\tilde{\gamma}_i)^2.&(27)\end{aligned}$$

Plugging this last result into (21), the distortion gain per bit resulting from adding a new $(a_i,\tilde{\gamma}_i)$ pair to a given approximation $\tilde{\mathbf{y}}^{i-1}$ of a block $\mathbf{y}$ is given by

$$\mathrm{G}((a_i,\tilde{\gamma}_i))=\frac{\gamma_i^2-(\gamma_i-\tilde{\gamma}_i)^2}{\mathrm{R}((a_i,\tilde{\gamma}_i))}.\tag{28}$$

As illustrated in Fig. 6, we use this last expression as a control criterion to choose which block approximation $\tilde{\mathbf{y}}^{i-1}$ will get a new $(a_i,\tilde{\gamma}_i)$ pair.

On the left-hand column of Fig. 7 we illustrate the performance of the proposed rate-distortion based global sparsity criterion above described by plotting, from top to bottom, 1) a reconstructed image and 2) its atom-distribution map and 3) RMSE per-block map. On the right-hand column of the same figure, we show the same three graphics obtained when using a common sparsity-selection approach based on an RMSE threshold:

$$\underset{L_{\mathbf{y}}}{\operatorname{argmin}}\,L_{\mathbf{y}}\ \text{s.t.}\ \|\mathbf{y}-\hat{\mathbf{y}}^{L_{\mathbf{y}}}\|^2\leq d\cdot\epsilon^2.\tag{29}$$

Note that the proposed sparsity criterion distributes atoms more uniformly than the scheme based on (29), while the resulting RMSE per block of the proposed scheme is less uniform. For the same coding rate (0.5 bpp), the proposed scheme offers an advantage of 0.63 dB.

E. Bit-Stream Format

In Fig. 8, we propose a simple bit-stream format for the ITAD codec discussed above. The structuring of the bit-stream is carried out using a one-bit end-of-block (EOB) flag which is inserted after each coefficient/index pair. If the flag is 0, then it is not the last pair of the block, if it is 1, it is the last pair of the block. Thus, the rate corresponding to the transmission of this flag has to be taken into account when calculating the rate of an $(a_i,\tilde{\gamma}_i)$ pair to compute the block selection criterion in (28):

$$\mathrm{R}((a_i,\tilde{\gamma}_i))=\mathrm{R}(a_i)+\mathrm{R}(\tilde{\gamma}_i)+\mathrm{R}(\texttt{EOB})\tag{30}$$

where $\mathrm{R}(a_i)$ is given in (16) and $\mathrm{R}(\tilde{\gamma}_i)$ is the length in bits of the codeword representing $\tilde{\gamma}_i$. The corresponding rate for the AC component of an image block is given by $\mathrm{R}(\mathcal{Y})=\sum_i\mathrm{R}((a_i,\tilde{\gamma}_i))$.

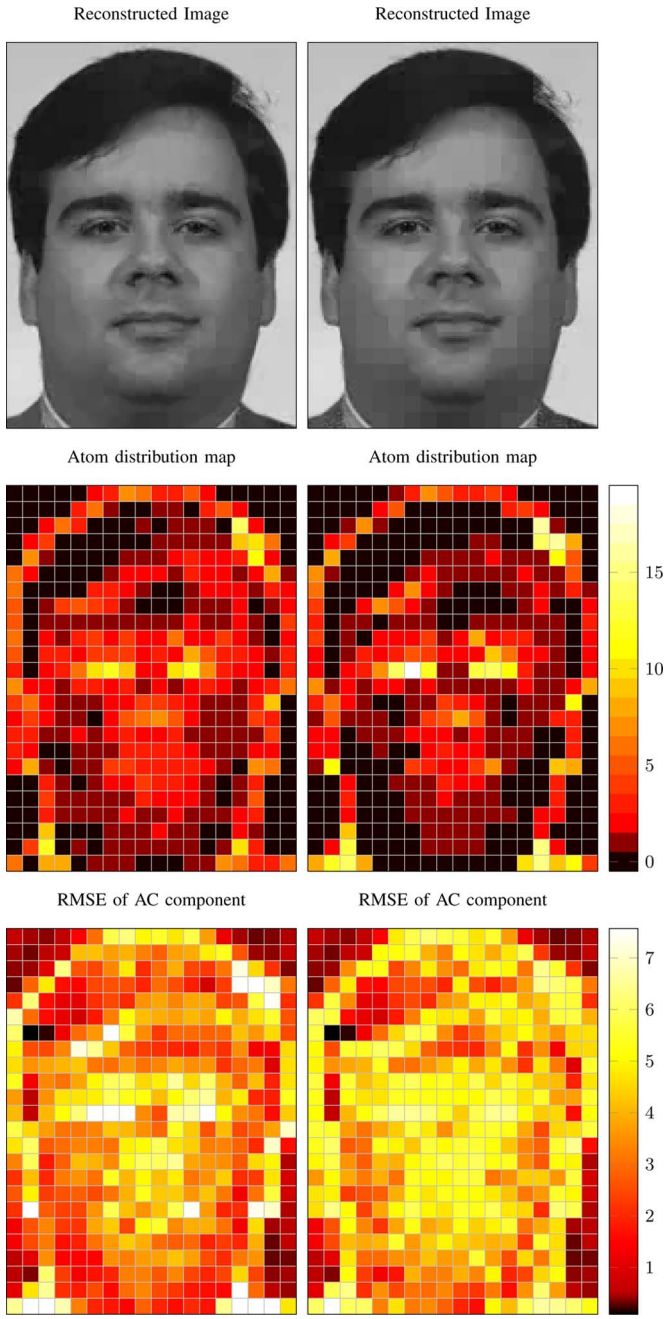

Fig. 7. Example of (*top*) the reconstructed image (*middle*) the encoder distribution of atoms-per-block and (*bottom*) the RMSE of the AC component of each block when using sparsity-selection criteria based on (*left*) a global rate-distortion scheme as illustrated in Fig. 6 or (*right*) a constant RMSE threshold ϵ as in (29). Both setups correspond to 0.5 bpp; the resulting PSNR and mean block-sparsity are: (*left*) 36.40 dB and 2.07 atoms-per-block; (*right*) 35.77 dB and 1.92 atoms-per-block. The original image used is shown in the second row of Fig. 14.

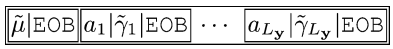

Fig. 8. Bit-stream used to represent the set $\mathcal{Y}$ of $(a_i,\tilde{\gamma}_i)$ pairs defining the approximation $\tilde{\mathbf{y}}$ of an image block and the DPCM coded DC component $\tilde{\mu}$.

IV. Results

In the current section, we evaluate the proposed image codec in compression of the class of facial images, comparing it

against the state-of-the art in image compression. Our evaluations show that our codec can offer an advantage that can be as high as several dBs for certain coding rates.

A. Experimental Setup

We use an image dataset consisting of frontal pose images of 764 different subjects: 664 of these images are used to train the image codec while the remaining (mutually exclusive) 100 images are used as a test set. The images are high-resolution uncompressed images taken from the FERET image dataset [16], manually cropped to focus on the face and re-sized to a uniform size of 192×144 pixels. When plotting avergae rate-distortion performance for all 100 test images, we carry out the following interpolation strategy: For each image, we first compute the distortion at some nominal rates. Since it is not always possible to achieve a given rate exactly, we interpolate the rate-distortion curves to obtain the distortion at the exact nominal rates. The rate-distortion curves plotted throughout correspond to the nominal rates versus the interpolated distortion averaged over all 100 test images.

As a comparison reference we use the state-of-the-art JPEG2000 [17] image encoder and its widely used predecessor, the JPEG encoder.

B. Choice of ITAD Dictionary

In order to justify the choice of the ITAD dictionary, we compare it to the (over)complete DCT and several other recently proposed overcomplete dictionaries. The reference dictionaries are the (over)complete DCT basis and three recent trained overcomplete dictionaries: the K-SVD dictionary of [14], the sparse dictionary (SD) of [18] and the online learned dictionary (ONLD) of [19]. Decompositions using these reference dictionaries are carried out using OMP. The reference trained dictionaries are trained using the software made available by the authors [20]–[22] with a training RMSE theshold ϵ_T [cf. (29)] chosen to yield the best curve when varied over a range centered on ϵ_T and unit steps [9]. For the case of SD, we similarly selected the best sparsity L_a of the dictionary columns [18]. We summarize the selected parameters for the reference dictionaries in Table I.

From Fig. 9 we compare the various dictionaries by plotting the PSNR of the reconstruction as a function of the mean number of atoms $L_{\mathbf{y}}$ required. We note that ITAD dictionaries with candidates having only 64 atoms per candidate can outperform all reference fixed dictionaries having 256 atoms. The reduced number of atoms of the ITAD candidates implies that the improved PSNR is attained with reduced complexity [9]. The improved compressibility that can be achieved with ITAD while incurring lower complexity justifies its use in the proposed codec.

C. ITAD Codec Construction

To construct the ITAD codec presented in previous sections, we begin by first extracting non-overlapping image blocks $\mathbf{z}$ from all 664 training images using a regular grid. To test the influence of the block size on the results, we build codecs using three different block sizes: 8×8, 12×12, and 16×16. This

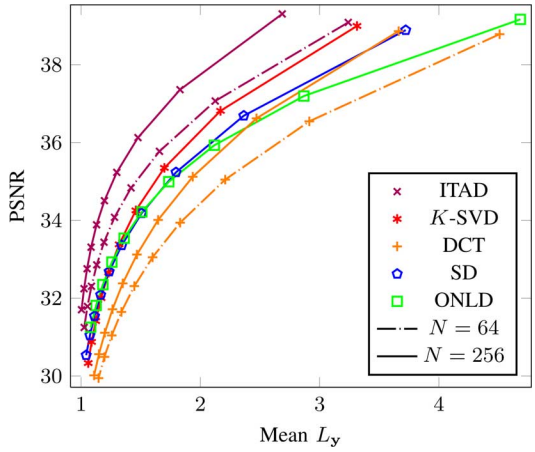

Fig. 9. Compressibilities achievable with ITAD and the reference K-SVD, SD, ONLD, and (over)complete DCT dictionaries. The original, uncropped FERET images withouth re-sizing are used in this experiment. The chosen training/testing FERET images are the same images used in the rest of the experiments (664 training images, 100 testing images). The results plotted are averaged over all 100 test images. For all reference dictionaries, N represents the total number of atoms, whereas for ITAD it represents the number of atoms per layer (i.e., per prototype dictionary).

TABLE I
PARAMETERS USED FOR REFERENCE TRAINED DICTIONARIES IN FIG. 9

Dictionary	Parameters
K-SVD [14]	$\epsilon_T = 8$; 40 iterations; DCT initialization
Sparse Dictionary (SD) [18]	$\epsilon_T = 14$; atom sparsity $L_a = 15$; 40 iterations; random initialization; DCT base dictionary of same size
Online Learned Dictionary (ONLD) [19]	$\epsilon_T = 10$; 1000 seconds training duration on 15-core computer

produces, respectively, training sets containing $2.9 \times 10^5, 1.27 \times 10^5$ and 0.71×10^5 vectors.

For each of these training sets, we first extract all the means μ and quantize them using a uniform quantizer with unit quantization step and dynamic range between 0 and 255. The resulting streams $\{\tilde{\mu}\}$ (one per training image) are DPCM encoded, and the residual of the DPCM prediction (for all training images) is used to design the corresponding entropy coder.

For simplicity, the AC coefficients γ_i are quantized using a single uniform quantizer that is common across all layers. The AC quantization step Δ is defined in terms of the per-pixel RMSE, which, due to the orthonormality property of the selected atoms, can be equivalently computed as

$$\epsilon_\Delta = \left(\frac{1}{b^2} \int_{-\Delta/2}^{\Delta/2} x^2 p(x) \mathrm{d}x \right)^{\frac{1}{2}}$$

where x denotes the representation coefficients. When the quantization error is uniformly distributed (i.e., $p(x) = 1/\Delta$), this per-pixel RMSE is computed as

$$\Delta = \epsilon_\Delta \cdot b \cdot \sqrt{12} \tag{31}$$

where the block is assumed to be $b \times b$. While we use an AC quantizer that is common to all layers, the subsequent entropy encoder is layer dependent. We use the same set of per-layer encoders i for all test images; each encoder i is designed using

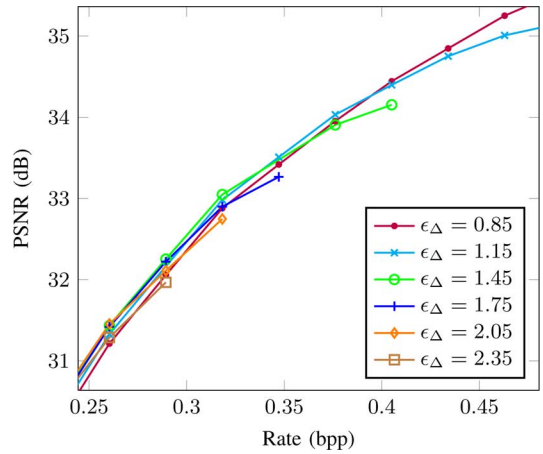

Fig. 10. Experimental rate-distortion curves for the ITAD-codec for various quantization steps when using $N = 128$ atoms (constant for all layers) per ITAD prototype dictionary. The values displayed are averaged (using interpolation) over the test images. We only plot data points where the rate is achievable by at least 90 of the 100 test images.

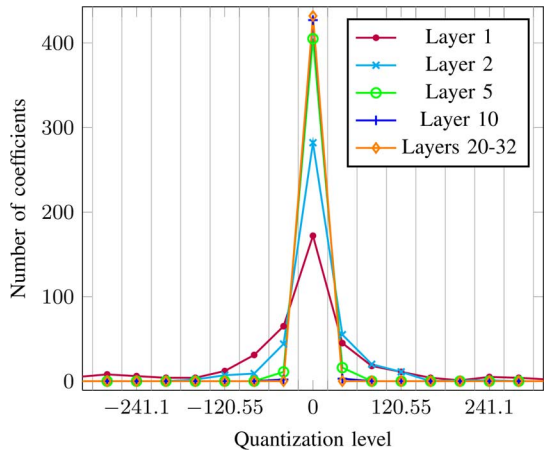

Fig. 11. Histogram of quantized coefficient values for all 432 blocks of one test image using the $\epsilon_\Delta = 1.45$ quantizer. Each image block is decomposed using the first 32 ITAD layers and all 32 resulting coefficients are quantized. A layer's histogram is then computed on all the 432 coefficients thus produced by that layer. Histogram values (denoted by the data markers) are connected by straight lines for illustrative purposes. Vertical grid lines denote quantization boundaries.

the empirical probabilities of the quantization symbols $\tilde{\gamma}_i$ of the training set at the corresponding layer i.

We use Huffman codes [23] for entropy encoding of both AC and DC (DPCM residual) symbols.

D. Quantitative Experiments

We carry out three different quantitative experiments: In the first two experiments we evaluate, respectively, the influence of 1) the quantization step and of 2) the number of prototype dictionary atoms N (kept constant for all ITAD layers). In the third experiment, we compare our image codec to the state-of-the-art JPEG2000 image encoder and its predecessor the JPEG image encoder on a specific class of imaged (set of facial images). For all three experiments we plot the average PSNR as a function of the true bit-rate, where the average is taken over all 100 test

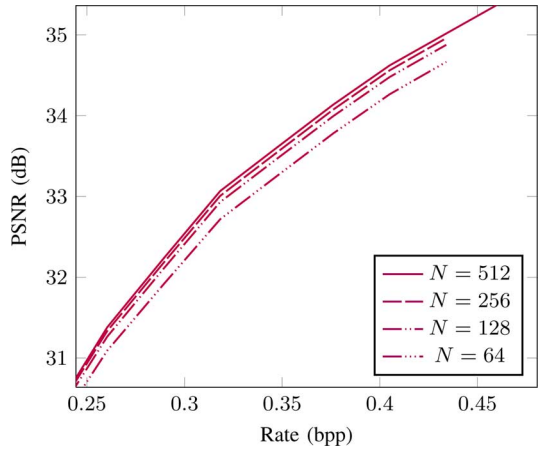

Fig. 12. Experimental rate-distortion curves for the ITAD-codec as a function of the ITAD prototype dictionary size N (constant for all layers) when using $\epsilon_\Delta = 0.91$. The values displayed are averaged (using interpolation) over the 100 test images.

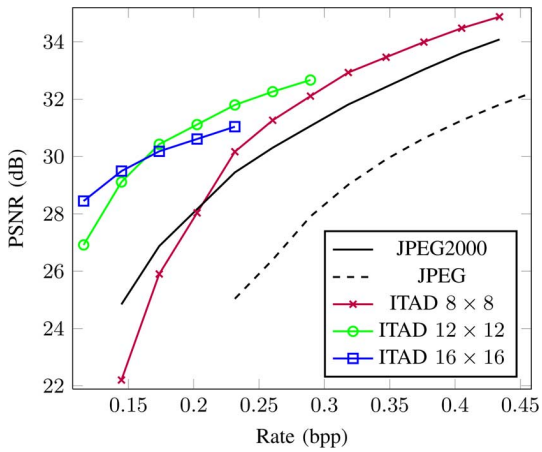

Fig. 13. Experimental rate-distortion curves for the ITAD-codec using various block sizes versus JPEG2000 and JPEG. The values displayed are averaged over all 100 test images [9].

images. From Fig. 8 and (30) the total bit-rate for a given image is given by

$$\sum_{\{\mathbf{z}\}} \left(\mathrm{R}(\tilde{\mu}) + \mathrm{R}(\texttt{EOB}) + \sum_i \mathrm{R}((a_i, \tilde{\gamma}_i)) \right). \qquad (32)$$

The rate plotted is the corresponding value expressed in bits-per-pixel (bpp).

In Fig. 10, we carry out the first experiment evaluating the quantization step-size (expressed in terms of the per-pixel RMSE ϵ_Δ). We use an ITAD structure having $N = 128$ atoms per layer and blocks of size 8×8. As illustrated in the figure, the optimal quantization step depends on the coding rate, yet, for simplicity, we use a constant quantization step for all rates (for a given block size). Note that some curves (e.g., for $\epsilon_\Delta = 1.45$) appear truncated in the figure. The reason for this is that it is not possible to achieve all rates for a given step size since, for sufficiently large i, all coefficients γ_i are quantized to zero. We illustrate this in Fig. 11, where we plot histograms of quantized coefficients (for $\epsilon_\Delta = 1.45$) for selected layers. Note that, by the tenth layer, only a small

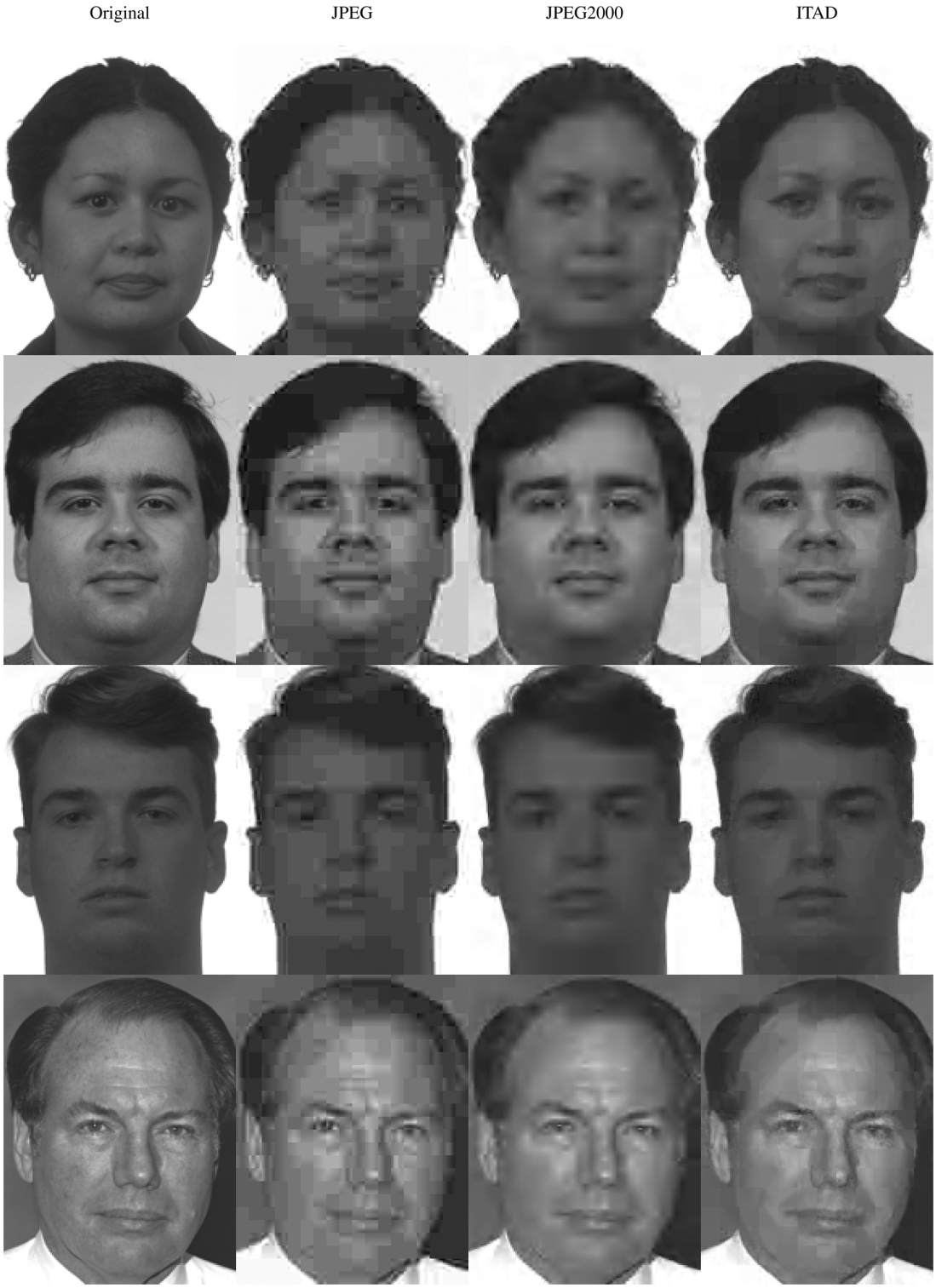

Fig. 14. Qualitative evaluation of the ITAD codec at a nominal rate of 0.3 bpp. From left to right, the first column contains the original images and the remaining three columns contain, respectively, the decoder-output image for JPEG, JPEG2000 and the ITAD codec. The exact rate and distortion for the rows of the last three columns are given in the respective cells of Table II.

fraction of coefficients are quantized to bins other than the zero bin. From layers 20 through 32, all coefficients are quantized to the zero bin. This problem could be addressed by decreasing the quantizer step size along with increasing layer index, but we did not pursue this approach in the present work.

In Fig. 12, we carry out the second experiment (again using blocks of size 8×8) which evaluates the performance of the ITAD codec as a function of the total number of atoms N in each layers (using $\epsilon_{\Delta} = 0.91$). Note that larger dictionaries result in improved performance even though the atom index coding penalty $\log_2(N)$ increases with N. The increased spar-

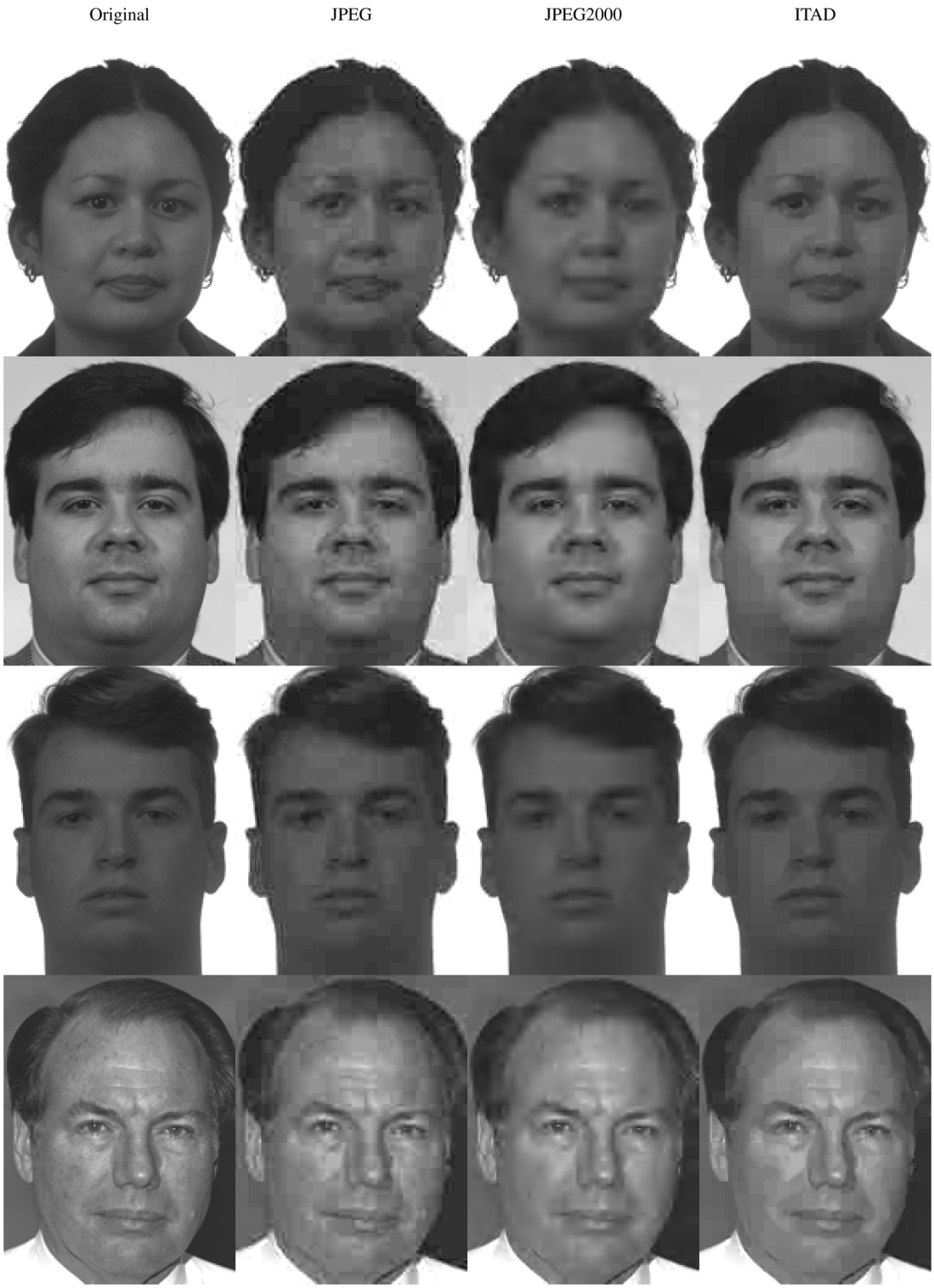

Fig. 15. Qualitative evaluation of the ITAD codec at a nominal rate of 0.4 bpp. From left to right, the first column contains the original images and the remaining three columns contain, respectively, the decoder-output image for JPEG, JPEG2000 and the ITAD codec. The exact rate and distortion for the rows of the last three columns are given in the respective cells of Table III.

sity of the representation indeed succeeds in overcoming the related atom-index penalty.

In Fig. 13, we compare our proposed ITAD codec against the JPEG and JPEG2000 image encoders. The three different curves shown for the ITAD codec correspond to three different block sizes (8×8, 12×12, and 16×16); we used the same experiment as in Fig. 10 to choose the corresponding ϵ_Δ value (respectively, 0.91, 0.8, and 0.72). We fixed the number of atoms N to 128 in all cases as this value is sufficiently low to facilitate training and keep codec complexity low. From Fig. 12, it is also evident that increasing N results in marginal gain.

The results in Fig. 13 show that different ITAD codecs are capable of outperforming the two reference codecs at all plotted rates by a wide margin. The 8×8 ITAD codec, for example,

TABLE II
RATE (bpp) AND DISTORTION (dB) FOR THE IMAGES IN FIG. 14

JPEG		JPEG2000		ITAD	
bpp	dB	bpp	dB	bpp	dB
0.32	28.93	0.31	30.73	0.31	33.99
0.33	29.15	0.31	31.46	0.31	33.90
0.32	31.20	0.32	35.58	0.32	36.59
0.32	29.78	0.31	31.71	0.31	32.91

TABLE III
RATE (bpp) AND DISTORTION (dB) FOR THE IMAGES IN FIG. 15

JPEG		JPEG2000		ITAD	
bpp	dB	bpp	dB	bpp	dB
0.40	31.09	0.40	33.26	0.40	36.49
0.40	31.18	0.40	33.48	0.40	35.10
0.40	33.75	0.39	37.33	0.40	38.45
0.40	31.35	0.39	32.99	0.39	33.79

outperforms JPEG2000 for all rates above 0.23 bpp by at least 0.5 dB. At 0.4 bpp, the ITAD Codec gain is 0.9 dB. The ITAD codecs based on the two larger block sizes offer gains of several dB for lower bitrates. For example, at 0.28 bpp, the 12×12 codec offers a gain of 1.5 dB.

E. Qualitative Experiments

We now carry out a qualitative comparison of the ITAD codec and the JPEG2000 and JPEG codecs on four images chosen from our test set. The results are illustrated in Figs. 14 and 15. As indicated in the figures, each of the four columns of either figure corresponds, respectively, to 1) the original image, 2) the JPEG decoded image 3) the JPEG2000 decoded image, and 4) the ITAD decoded image. The difference between the two figures is in the nominal rate used for all encoders: In Fig. 14, we use a nominal rate of 0.3 bpp and in Fig. 15 we use a nominal rate of 0.4 bpp. Using Fig. 13, we choose an ITAD block size of 12×12 for the 0.3 bpp rate and of 8×8 for the 0.4 bpp. Note that blocking artificacts will be more visible for the 0.4 bpp images because of the greater number of block boundaries resulting from the smaller block size.

From the illustrations, one can observe that the images at the output of the ITAD codec indeed display an improved visual quality relative to either of the reference codecs. The JPEG encoder suffers from a very pronounced blocking artifact, particularly in the low-rate figure. The JPEG2000 images, on the other hand, suffer from blurring of the facial features. This is especially noticeable (in all images) around the eyes and nose, which are a lot sharper in the ITAD decoded images. For completeness, we provide the exact rates and PSNRs for both qualitative comparisons in Tables II and III, with each row of the table corresponding, respectively, to a row of Figs. 14 and 15.

V. CONCLUSION

In this paper, we have shown how the superior sparse approximation capability of the ITAD dictionary can be leveraged to construct an image codec capable of outperforming state-of-the art algorithms such as JPEG2000, when used on specific classes of images, such as facial images. The codec selects the sparsity of the various blocks using a rate-distortion criterion. Coding of the ITAD coefficients is then carried out using a standard quantizer/entropy encoder combination. The evaluations we carried out show that our proposed codec can outperform the state of the art by at least 0.5 dB over a large range of rates. We further showed that the reconstructed image is more clear and better preserves details.

REFERENCES

[1] O. Guleryuz, "Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-part I: Theory," *IEEE Trans. Image Process.*, vol. 15, no. 3, pp. 539–554, Mar. 2006.

[2] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[3] M. Elad, J.-L. Starck, P. Querre, and D. Donoho, "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)," *Appl. Comput. Harmon. Anal.*, vol. 19, no. 3, pp. 340–358, Nov. 2005.

[4] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, no. 4, pp. 30–44, 1991.

[5] G. J. Sullivan and T. Wiegand, "Video compression—From concepts to the H.264/AVC standard," *Proc. IEEE*, vol. 93, no. 1, pp. 18–31, Jan. 2005.

[6] O. Bryt and M. Elad, "Compression of facial images using the K-SVD algorithm," *J. Vis. Commun. Image Represent.*, vol. 19, no. 4, pp. 270–282, 2008.

[7] O. G. Sezer, O. Harmanci, and O. G. Guleryuz, "Sparse orthonormal transforms for image compression," in *Proc. IEEE Int. Conf. Image Process.*, 2008, pp. 149–152.

[8] J. Zepeda, C. Guillemot, and E. Kijak, "The iteration-tuned dictionary for sparse representations," in *Proc. IEEE Int. Workshop MMSP*, 2010.

[9] J. Zepeda, "Novel sparse representation methods and their application in image compression and indexing" Ph.D. dissertation, Univ. de Rennes 1/Université Européenne de Bretagne, Rennes, France, 2010 [Online]. Available: http://hal.archives-ouvertes.fr/tel-00567851/

[10] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[11] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Asilomar Conf. Signals, Syst., Comput.*, A. Singh, Ed., Los Alamitos, CA, 1993, IEEE Comput. Soc. Press.

[12] L. Rebollo-Neira and D. Lowe, "Optimized orthogonal matching pursuit approach," *IEEE Signal Process. Lett.*, vol. 9, no. 4, pp. 137–140, Apr. 2002.

[13] K. Sayood, *Introduction to Data Compression*, 2nd ed. San Francisco, CA: Morgan Kaufmann, 2000.

[14] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[15] V. K. Goyal and M. Vetterli, "Dependent coding in quantized matching pursuit," in *Proc. SPIE—Visual Commun. Image Process.*, 1997, pp. 2–12.

[16] P. J. Phillips, H. Moon, P. J. Rauss, and S. Rizvi, "The FERET evaluation methodology for face recognition algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1090–1104, Oct. 2000.

[17] M. D. Adams, "The JPEG-2000 still image compression standard," Dept. of Elect. and Comput. Eng., Univ. of Victoria, Victoria, BC, Canada, Tech. Rep., 2005.

[18] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1553–1564, 2010.

[19] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 10–60, 2010.

[20] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD Toolbox," 2006 [Online]. Available: http://www.cs.technion.ac.il/~elad/software/

[21] R. Rubinstein, "Sparse K-SVD Toolbox," 2010 [Online]. Available: http://www.cs.technion.ac.il/~ronrubin/software.html

[22] J. Mairal, "SPArse Modeling Software (SPAMS)," 2010 [Online]. Available: http://www.di.ens.fr/willow/SPAMS/

[23] D. Huffman, "A method for the construction of minimum-redundancy codes," *Proc. IRE*, pp. 1098–1102, 1952.

Joaquin Zepeda received the B.Eng. and M.Eng. degrees from McGill University, Montreal, QC, Canada, in 2003 and 2006, respectively, and the Ph.D. degree from the University of Rennes 1, Rennes, France, in 2010.

He currently holds a post-doctoral position in the Technicolor Research and Innovation Center, Rennes, France, where he works on content-based video search and sparse representations. Previously, he worked in INRIA, Rennes, France, on topics including sparse representations, image description, and compression. His research interests have included real-valued codes, filterbanks, and wireless communications.

Christine Guillemot received the Ph.D. degree from the École Nationale Supérieure des Télécommunications (ENST), Paris, France, and the Habilitation for Research Direction from the University of Rennes, Rennes, France.

She is currently *Directeur de Recherche* at INRIA, Rennes, France, in charge of a research team dealing with image and video modeling, processing, coding, and communication. From 1985 to October 1997, she was with FRANCE TELECOM, where she has been involved in various projects in the domain of image and video coding for TV, HDTV, and multimedia applications. In particular, she has been involved in several European and national projects, and has managed several EU and national projects in the area of multimedia and audio-visual processing. From January 1990 to mid 1991, she has worked at Bellcore, NJ, as a Visiting Scientist. She has authored or coauthored 15 patents, 8 book chapters, 50 journal papers and 140 conference papers.

Dr. Guillemot has served as an Associated Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING (2000–2003), the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (2004–2006), and the IEEE TRANSACTIONS ON SIGNAL PROCESSING (2007–2009). She is currently an Associate Editor for the EURASIP *Journal on Image Communication*. She has been a member of the IEEE IMDSP and MMSP technical committees and serves on the committee assigning the Best Paper Award of the EURASIP *Journal on Image Communication* (JoIC).

Ewa Kijak received the Ph.D. degree in video processing from the University of Rennes 1, Rennes, France, in 2003.

She is currently an Associate Professor at the University of Rennes 1. She worked at Thomson Corporate Research, in collaboration with the IRISA Research Unit, while pursuing the Ph.D. degree. She worked on video indexing and multimedia information fusion. She then was an Associate Professor at the University of Paris 6, working in the Computer Science Laboratory (LIP6) of the university for two years, and joined the IRISA laboratory in 2006 within the TEXMEX team. Her research interests are in the fields of images and videos description and indexing.