Identification of transform coding chains [1]

Marco Tagliasacchi[1], Marco Visentini-Scarzanella[2], Pier Luigi Dragotti[2], Stefano Tubaro[1]

***Abstract*—Transform coding is routinely used for lossy compression of discrete sources with memory. The input signal is divided into N-dimensional vectors, which are transformed by means of a linear mapping. Then, transform coefficients are quantized and entropy coded. In this paper we consider the problem of identifying the transform matrix as well as the quantization step sizes. First, we study the case in which the only available information is a set of P transform decoded vectors. We formulate the problem in terms of finding the lattice with the largest determinant that contains all observed vectors. We propose an algorithm that is able to find the optimal solution and we formally study its convergence properties. Three potential realms of application are considered as example scenarios for the proposed theory: parameter retrieval in presence of a chain of two transform coders, image tampering identification and parameter estimation for predictive coders. We show that, despite their differences, all three scenarios can be tackled by applying the same fundamental methodology. Experiments on both synthetic data and real images validate the proposed approach.**

I. Introduction

Transform coding has emerged over the years as the dominating compression strategy. Transform coding is adopted in virtually all multimedia compression standards including, among others, image compression standards such as JPEG [3] and video compression standards such as, for example, H.264/AVC [4] and HEVC [5]. This is due to the fact that transform coders are very effective and yet computationally inexpensive since the encoding operation is divided into three relatively simple steps: linear transformation of the data, scalar quantization of each coefficient, and entropy coding.

Due to its centrality to any type of multimedia data, transform coding theory is now extensively used in a new range of applications that rely on the possibility of reverse-engineering complex chains of operators starting from the available output signals. Indeed, the lifespan of a multimedia signal is virtually unbounded. This is due to the ability of creating copies and the availability of inexpensive storage options. However, signals seldom remain identical to their original version. As they pass through processing chains, some operators, including transform coding, are bound to leave subtle characteristic footprints on the signals, which can be identified in order to uncover their processing history. This insight might be extremely useful in a wide range of scenarios in the field of multimedia signal processing at large including, e.g.,: i) forensics, in order to address tasks such as source device identification [6] or tampering detection [7][8]; ii) quality assessment, to enable no-reference methods that rely solely on the received signals [9][10]; iii) digital restoration, which requires prior knowledge about the operations that affected a digital signal [11].

In this context, several works have exploited the footprints left by transform coding. In [12], a method was proposed to infer the implementation-dependent quantization matrix template used in a JPEG-compressed image. Double JPEG compression introduces characteristic peaks in the histogram of DCT coefficients, which has been used, e.g, for tampering localization [13][8]. More recently, similar techniques were applied to video signals for the cases of MPEG-2 [14][15], MPEG-4 [16][17] and H.264/AVC [18].

All the aforementioned works assume prior knowledge of the specific transform in use, whereas the quantization steps need to be estimated. Some efforts in the direction of detecting the standard being considered can be found in [19], which leverages the differences in the way transform coding is implemented to discriminate between MPEG-2, MPEG-4 and H.264/AVC coded video. However, the work in [19] implies a closed-group assumption, where the codec to be identified belongs to a set whose elements are known *a priori*. Moreover, enumerating the possible codecs may be impractical, since recent coding architectures are more diversified in terms of both the type of transform being used and the block size.

Given the specific nature (e.g., only images or only videos) and the dependence on heuristics of the aforementioned methods, it is therefore natural to try and develop a universal theory of transform coder identification that is independent of the specific application at hand. To this end, in this paper we consider a general model of transform coding that can be tailored to describe a large variety of practical implementations that are found in lossy coding systems, including those adopted in multimedia communication. From the output produced by a transform coding chain, we investigate

[1] Dipartimento di Elettronica e Informazione, Politecnico di Milano, P.zza Leonardo da Vinci, 32 20133 - Milano, Italy - E-mail: marco.tagliasacchi@polimi.it, stefano.tubaro@elet.polimi.it, [2] CSP Group, EEE Department, Imperial College London, Exhibition Road, London SW7-2AZ, United Kingdom, E-mail: p.dragotti@imperial.ac.uk, marcovs@imperial.ac.uk. A summary of the results presented in this manuscript was presented at IEEE ICASSP 2013 [1] and IEEE ICIP 2013 [2].

the problem of identifying its parameters, by proposing an algorithm that receives as input a set of P transform decoded vectors embedded in a N-dimensional vector space and produces as output an estimation of the transform adopted, as well as the quantization step sizes. We leverage the intrinsic discrete nature of the problem, by observing the fact that these vectors are bound to belong to a N-dimensional lattice. Hence, the problem is formulated in terms of finding a lattice that contains all observed vectors.

Lattice theory has been widely used for source and channel coding (e.g., [20], [21], [22]). However, to the best of the authors' knowledge, this theory has not been employed to address the problem of identifying a linear mapping using the footprint left by quantization. Only [23] uses similar principles but their goal is to investigate the color compression history, i.e., the colorspace used in JPEG compression. Therefore, the solution proposed is tailored to work in a 3-dimensional vector space, thus avoiding the challenges that arise in higher dimensional spaces. A refined version of [23] was proposed in [24] based on the notion of dual lattice.

Efforts to solve the problem of finding a basis from a set of generating vectors stemmed directly from the related problem of lattice reduction [22]. Lattice reduction techniques aim to find, given a basis for a lattice, an equivalent basis matrix with favorable properties. Usually, such basis consists of vectors that are short and with improved orthogonality. Out of the several definitions of lattice reduction techniques, the most popular one is the Lenstra-Lenstra-Lovász (LLL) reduction [25], which can be interpreted as the Gauss reduction to lattices of rank greater than 2. However, one of the disadvantages of the LLL algorithm is that the input vectors must be linearly independent [26]. Therefore, the LLL algorithm as originally formulated cannot be directly applied to the task of finding a basis from a set of randomly observed, possibly linearly dependent vectors. The algorithms in [26] and [27] address this problem by iteratively testing subsets of the observed vectors for linear dependence during the lattice reduction procedure. At their core, however, they still employ the original LLL reduction concept. This implies having to consider all the observed vectors simultaneously, which results in a high computational complexity (see [27][28][29] for details). Further efforts were devoted to improve the computational efficiency, with the works in [28][30] proposing an incremental version of the algorithm in [26] where observed vectors are individually input in the system until convergence is reached. While the proposed method achieves a better performance in terms of computational complexity it requires that $P \gg N$ and still relies on LLL.

Conversely, we propose a technique that is not based on the computationally intensive LLL, and instead we provide a number theoretical intuition to tackle this problem. Indeed, our proposed method is a higher-dimensional extension of Euclid's algorithm, which is routinely used to find the greatest common divisor (GCD) in a set of integers. Specifically, whenever $N = 1$ and $P = 2$, the proposed method coincides with Euclid's algorithm. In our proposed technique, a method for basis orthogonalization is only seldom used whenever the candidate basis exceeds a threshold orthogonality defect. It has to be stressed that this does not have to be accomplished exclusively via LLL, but alternative *ad hoc* methods can be chosen instead. Moreover, the proposed algorithm is incremental, with a complexity that is shown to grow linearly with N. We prove its convergence and show that its probability of successfully identifying the correct lattice approaches 1 whenever there is a moderate excess of 6-7 vectors beyond the dimension of the space N, thus reducing the need for techniques such as [28][30] that require $P \gg N$.

Then, as an example of the practical implications for the proposed theory, we show three separate realms of application: transform parameter retrieval for processing chains of two transform coders, image tampering identification, and parameter estimation in the presence of predictive coding. The first scenario was studied in [31] for the case of a 2-dimensional transform, in order to determine the analytical conditions under which it is possible to navigate back up the signal's history to the first coding stage and determine the first encoder's exact transform parameters. Instead, in this work we address the more challenging case of N-dimensional transforms, with $N \geq 2$. Our main contribution is a method that is able to identify the parameters used by the first transform coder (i.e., the adopted transform and quantizer), when observing the output of the second transform coder, thus generalizing the results obtained in the case of a single transform coder. Importantly, this extension also enables to analyze the conditions under which our proposed algorithm can operate correctly under quantization noise, thus relaxing the strict requirements of integral vectors [29] or exact rational arithmetic [28] present in earlier works.

The other two application scenarios hinge on the ability of the proposed framework to robustly handle outliers. We show how with a RANSAC-like procedure it is possible to correctly accomplish the proposed tasks with a high probability of success, which we characterise as a function of the main operational parameters.

The rest of this paper is organized as follows. Section II introduces the necessary notation and formulates the transform identification problem and Section III provides the background on lattice theory. First, we consider a chain that consists of a single transform coder in Section IV, we propose a method to identify the transform and provide a theoretical analysis of its con-

vergence properties. Then, we focus on a chain with two transform coders in Section V and we show how it is possible to reuse the method in Section IV, provided that the input is properly de-quantized. The performance of the algorithms is evaluated empirically in Section VII on both synthetically generated and real datasets. Finally, Section VIII concludes the paper, indicating the open issues and stimulating further investigations.

II. Problem statement

The symbols x, $\mathbf{x}$ and $\mathbf{X}$ denote, respectively, a scalar, a column vector and a matrix. A $M \times N$ matrix $\mathbf{X}$ can be written either in terms of its columns ($\mathbf{x}_j$) or rows ($\bar{\mathbf{x}}_i^T$). Specifically,

$$\mathbf{X} = [\ \mathbf{x}_1 \quad \mathbf{x}_2 \quad \dots \quad \mathbf{x}_N\ ] = \begin{bmatrix} \bar{\mathbf{x}}_1^T \\ \bar{\mathbf{x}}_2^T \\ \dots \\ \bar{\mathbf{x}}_M^T \end{bmatrix}. \quad (1)$$

Let $\mathbf{x}$ denote a N-dimensional vector and $\mathbf{W}$ a transform matrix, whose rows represent the transform basis functions.

Transform coding is performed by applying scalar quantization to the transform coefficients $\mathbf{y} = \mathbf{W}\mathbf{x}$. Let $\mathcal{Q}_i(\cdot)$ denote the quantizer associated to the i-th transform coefficient. We assume that $\mathcal{Q}_i(\cdot)$ is a scalar uniform quantizer with step size Δ_i, $i = 1, \dots, N$. Therefore, the reconstructed quantized coefficients can be written as $\tilde{\mathbf{y}} = [\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_N]^T$, with

$$\tilde{y}_i = \mathcal{Q}_i(y_i) = \Delta_i \cdot \text{round}\left[\frac{y_i}{\Delta_i}\right], \quad i = 1, \dots, N. \quad (2)$$

The reconstructed block in the original domain is given by $\tilde{\mathbf{x}} = \mathbf{W}^{-1}\tilde{\mathbf{y}}$.

Let $\{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_P\}$ denote a set of P observed N-dimensional vectors, which are the output of a transform coder. Due to quantization, the unobserved vectors representing quantized transform coefficients $\{\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_P\}$ are constrained to belong to a lattice $\mathcal{L}_y$ described by the basis $\mathbf{B}_y = \text{diag}(\Delta_1, \Delta_2, \dots, \Delta_N)$. Therefore, the observed vectors $\{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_P\}$ belong to a lattice $\mathcal{L}_x$ described by the basis:

$$\mathbf{B}_x = [\mathbf{b}_{x,1}, \dots, \mathbf{b}_{x,N}] = \mathbf{W}^{-1}\mathbf{B}_y, \quad (3)$$

with $\mathbf{b}_{x,i} = \Delta_i \hat{\mathbf{w}}_i$, $i = 1, \dots, N$, $\mathbf{W}^{-1} = [\hat{\mathbf{w}}_1, \dots, \hat{\mathbf{w}}_N]$.

First, we study the problem of determining $\mathbf{B}_x$ from a finite set of $P \geq N$ distinct vectors $\{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_P\}$. That is, we seek to determine the parameters of a *single* transform coder based on the footprints left on its output. In Section IV we propose an algorithm to solve this problem and we study its convergence properties. In addition, we show that the probability of correctly determining $\mathbf{B}_x$ (or, equivalently, another basis for the lattice $\mathcal{L}_x$) is monotonically increasing in the number of observations P, and rapidly approaching one when $P > N$. Note that when determining $\mathbf{B}_x$, the proposed method does not make any assumption on the structure of the transform matrix $\mathbf{W}$. In the general case, given $\mathbf{B}_x$, it is not possible to uniquely determine $\mathbf{W}$ and the quantization step sizes Δ_i, $i = 1, \dots, N$. Indeed, the length of each basis vector $\mathbf{b}_{x,i}$ can be factored out as $\|\mathbf{b}_{x,i}\|_2 = \Delta_i \|\hat{\mathbf{w}}_i\|_2$. However, in the important case in which $\mathbf{W}$ represents an orthonormal transform, the quantization step sizes Δ_i, $i = 1, \dots, N$, and the transform matrix $\mathbf{W}$ can be immediately obtained from $\mathbf{B}_x$. Indeed, $\mathbf{W}^{-1} = \mathbf{W}^T$, $\hat{\mathbf{w}}_i = \bar{\mathbf{w}}_i$, $i = 1, \dots, N$, with $\|\bar{\mathbf{w}}_i\|_2 = 1$. Therefore:

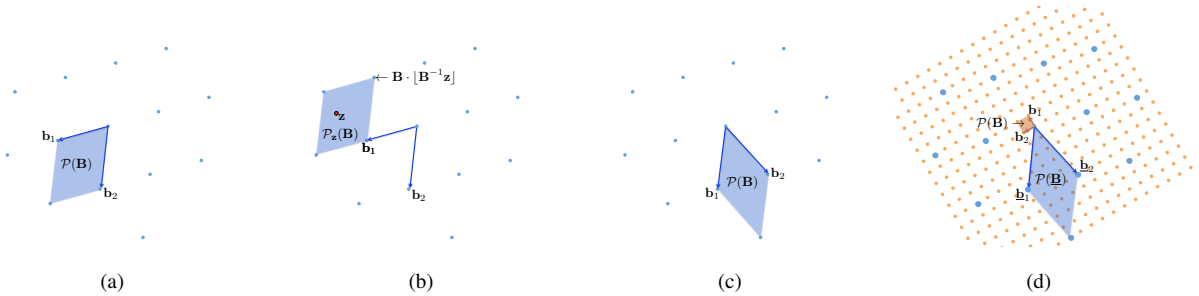
$$\Delta_i = \|\mathbf{b}_{x,i}\|_2, \quad i = 1, \dots, N, \quad (4)$$

$$\bar{\mathbf{w}}_i = \mathbf{b}_{x,i} / \|\mathbf{b}_{x,i}\|_2 \quad i = 1, \dots, N. \quad (5)$$

Then, we consider the more challenging case in which an input signal is processed as illustrated in Figure 1, by cascading *two* transform coders characterized, respectively, by the transform matrices $\mathbf{W}_a$ and $\mathbf{W}_b$ and quantizers $\mathcal{Q}_{a,i}(\cdot)$, $\mathcal{Q}_{b,i}(\cdot)$, $i = 1, \dots, N$. We assume that both transform coders work on vectors having the same size N, and that the signal is not shifted or resampled in between the two transforms. Let $\{\tilde{\mathbf{u}}_1, \dots, \tilde{\mathbf{u}}_P\}$ denote a set of P observed N-dimensional vectors, which are the output of the second transform coder. We assume that the second transform coder is completely known. Indeed, whenever this is not the case, it can be identified with the method described in Section IV. Therefore, without loss of generality, we will consider the set of observed vectors $\{\tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_P\}$, such that $\tilde{\mathbf{z}}_j = \mathbf{W}_b \tilde{\mathbf{u}}_j$.

Due to quantization, the observed vectors $\{\tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_P\}$ are constrained to belong to a lattice $\mathcal{L}_z$ described by the basis $\mathbf{B}_z = \text{diag}(\Delta_{b,1}, \dots, \Delta_{b,N})$. Similarly, the unobserved transform coefficients $\{\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_P\}$ of the first coder belong to a lattice $\mathcal{L}_y$ described by the basis $\mathbf{B}_y = \text{diag}(\Delta_{a,1}, \dots, \Delta_{a,N})$. Hence, the unobserved vectors $\{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_P\} \in \mathcal{L}_x$ with basis $\mathbf{B}_x = \mathbf{W}_a^{-1}\mathbf{B}_y$.

In Section V we study the problem of determining $\mathbf{B}_x$ from a finite set of $P \geq N$ distinct vectors $\{\tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_P\}$. That is, we seek to determine the parameters of the first transform coder in a chain of two transform coders, when we observe the output of the second one. This is a much more challenging problem than the one addressed in Section IV, since the observed vectors do not lie on the lattice $\mathbf{B}_x$. As a consequence, a direct application of the method in Section IV to the set of vectors $\{\tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_P\}$ would identify $\mathbf{W}_b$ rather than $\mathbf{W}_a$. We show how to solve the problem in the case the transform matrices are orthonormal, i.e., $\mathbf{W}_a^T\mathbf{W}_a = \mathbf{I}$ and $\mathbf{W}_b^T\mathbf{W}_b = \mathbf{I}$. For the sake of simplifying the notation, we assume that the same step size is used to quantize the transform coefficients, i.e., $\Delta_{a,i} = \Delta_a$ and $\Delta_{b,i} = \Delta_b$, $i = 1, \dots, N$. However, this condition can be relaxed.



(a) (b) (c) (d)

Fig. 2. Examples of lattices. (a) The fundamental parallelotope of a lattice defined by a basis $\mathbf{B}$. (b) Parallelotope enclosing an arbitrary vector $\mathbf{z}$. (c) Another (equivalent) basis for the lattice in (a). (d) An example of a sub-lattice of the lattice $\mathcal{L}(\mathbf{B})$.

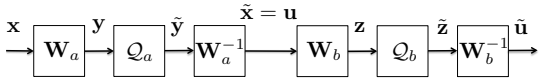


Fig. 1. Block diagram of a chain of two transform coders.

III. Background on Lattice Theory

In this section we provide the necessary background on lattice theory. Further details can be found, e.g., in [32][33][22]. Let $\mathcal{L}$ denote a lattice of rank K embedded in $\mathbb{R}^N$. Let $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_K]$ denote a basis for the lattice $\mathcal{L}$. That is,

$$\mathcal{L} = \{\mathbf{x} | a_1\mathbf{b}_1 + a_2\mathbf{b}_2 + \ldots + a_K\mathbf{b}_K, a_i \in \mathbb{Z}\}. \quad (6)$$

In order to make the mapping between a basis and the corresponding lattice explicit, the latter can be expressed as $\mathcal{L}(\mathbf{B})$.

Any lattice basis also describes a fundamental parallelotope according to

$$\mathcal{P}(\mathbf{B}) = \left\{\mathbf{x} | \mathbf{x} = \sum_{i=1}^{K} \theta_i \mathbf{b}_i, 0 \le \theta_i < 1\right\}. \quad (7)$$

When $K = 2, 3$, $\mathcal{P}(\mathbf{B})$ is, respectively, a parallelogram or a parallelepiped. As an example, Figure 2(a) shows the fundamental parallelotope corresponding to a lattice basis $\mathbf{B}$ when $K = 2$.

Given a point $\mathbf{z} \in \mathbb{R}^K$, let $\mathcal{P}_{\mathbf{z}}(\mathbf{B})$ denote the parallelotope enclosing $\mathbf{z}$. $\mathcal{P}_{\mathbf{z}}(\mathbf{B})$ is obtained by translating $\mathcal{P}(\mathbf{B})$ so that its origin coincides with one of the lattice points. More specifically,

$$\mathcal{P}_z(\mathbf{B}) = \left\{\mathbf{x} | \mathbf{x} = \mathbf{B}\cdot\left\lfloor\mathbf{B}^{-1}\mathbf{z}\right\rfloor + \sum_{i=1}^{K} \theta_i \mathbf{b}_i, 0 \le \theta_i < 1\right\}. \quad (8)$$

Figure 2(b) illustrates $\mathcal{P}_{\mathbf{z}}(\mathbf{B})$ for an arbitrary vector $\mathbf{z}$.

Different bases for the same lattice lead to different fundamental parallelotopes. For example, Figure 2(a) and Figure 2(c) depict two different bases for the same lattice, together with the corresponding fundamental parallelotopes. However, the volume of $\mathcal{P}(\mathbf{B})$ is the same for all bases of a given lattice. This volume equals the so-called *lattice determinant*, which is a lattice invariant defined as

$$|\mathcal{L}| = \sqrt{\det(\mathbf{B}^T\mathbf{B})}. \quad (9)$$

If the lattice is full rank, i.e., $K = N$, the lattice determinant equals the determinant of the matrix $\mathbf{B}$, $|\mathcal{L}| = |\det(\mathbf{B})|$.

Let $\underline{\mathcal{L}}$ denote a sub-lattice of $\mathcal{L}$. That is, for any vector $\mathbf{x} \in \underline{\mathcal{L}}$, then $\mathbf{x} \in \mathcal{L}$. A basis $\underline{\mathbf{B}}$ for $\underline{\mathcal{L}}$ can be expressed in terms of $\mathbf{B}$ as

$$\underline{\mathbf{B}} = \mathbf{B}\mathbf{U}, \quad (10)$$

where $\mathbf{U}$ is such that $u_{ij} \in \mathbb{Z}$. Moreover, let $\det(\mathbf{U}) = \pm m$, then

$$\frac{|\underline{\mathcal{L}}|}{|\mathcal{L}|} = |\det(\mathbf{U})| = m \quad (11)$$

and we say that $\underline{\mathcal{L}}$ is a sub-lattice of $\mathcal{L}$ of index m. For example, Figure 2(d) shows two lattices $\underline{\mathcal{L}}$ and $\mathcal{L}$, such that $\underline{\mathcal{L}} \subset \mathcal{L}$. In this case, $\underline{\mathcal{L}}$ is a sub-lattice of index $m = 19$.

IV. An Algorithm for Transform Identification

In this section we propose an algorithm that is able to determine the parameters of a transform coder from its output, i.e., a set of observed vectors $\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_P\}$. This is accomplished by finding a suitable lattice $\mathcal{L}^*$ such that $\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_P\} \subset \mathcal{L}^*$. We will show later that, with probability approaching one, $\mathcal{L}^* \equiv \mathcal{L}_x$, provided that $P - N > 0$.

The problem of determining a basis for the lattice $\mathcal{L}_x$ is complicated by the fact that we typically observe a finite (and possibly small) number of vectors P embedded in a possibly large dimensional space. More precisely, $\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_P\}$ belong to a bounded lattice, in virtue of the fact that each transform coefficient y_i is quantized with a finite number of bits R_i, to one of 2^{R_i} reconstruction levels. Let $\bar{R}$ denote the average number

of bits allocated to transform coefficients. The number of potential lattice points is equal to

$$\prod_{i=1}^{N} 2^{R_i} = 2^{\sum_{i=1}^{N} R_i} = 2^{N\bar{R}}, \tag{12}$$

and only P of them are covered by observed vectors. Thus, we note that, given $\bar{R}$, the number of lattice points increases exponentially with the dimension N and that in most cases of practical relevance $P \ll 2^{N\bar{R}}$.

Another issue arises from the fact that, for a set of vectors $\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_P\}$, there are infinitely many lattices that include all of them. Indeed, any lattice $\bar{\mathcal{L}}$ such that $\mathcal{L}_x \subset \bar{\mathcal{L}}$ is compatible with the observed set of vectors. Note that any basis of the form $\mathbf{B} = \mathbf{B}_x\mathbf{U}^{-1}$, with $\det(\mathbf{U}) = \pm m$, with m an integer greater than one defines a compatible lattice $\bar{\mathcal{L}}$. A simple example is obtained setting $\mathbf{U} = a\mathbf{I}$, $a \in \mathbb{N}, a > 1$.

In order to resolve this ambiguity, we seek the lattice $\mathcal{L}^*$ that maximizes the lattice determinant $|\mathcal{L}|$, within this infinite set of compatible lattices. That is,

$$\begin{array}{ll} \underset{\mathcal{L}(\mathbf{B})}{\text{maximize}} & |\mathcal{L}(\mathbf{B})| \\ \text{subject to} & \{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_P\} \subset \mathcal{L}(\mathbf{B}). \end{array} \tag{13}$$

For example, for the set of observed points $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \tilde{\mathbf{x}}_3\}$ depicted in Figure 3(a), Figure 3(g) illustrates a basis for the lattice that is the optimal solution of (13). In contrast, the lattice in Figure 3(h) is a feasible solution of (13), but it is not optimal, since it is characterized by a lower value of the lattice determinant.

The proposed method used to solve the problem above is detailed in Algorithm 1. The method constructs an initial basis for an N-dimensional lattice (line 1). This is accomplished by considering the vectors in $\mathcal{O}$ until N linearly independent vectors are found. These vectors are used as columns of the starting estimate $\mathbf{B}^{(0)}$ and to populate the initial set of visited vectors $\mathcal{S}$. We denote with $\mathcal{U}$ the set of vectors in $\mathcal{O}$ that have not been visited yet. Then, the solution of (13) is constructed iteratively, by considering the remaining vectors in $\mathcal{U}$ one by one. At each iteration, the function `recurseTI` returns a basis for a lattice that solves (13), in which the constraint is imposed only on the subset of visited vectors $\mathcal{S}$, that is, $\mathcal{S} \subset \mathcal{L}(\mathbf{B})$. As such, the algorithm starts finding the solution of an under-constrained problem and additional constraints are added as more vectors are visited.

Figure 3 shows an illustrative example when $N = 2$ and three vectors $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \tilde{\mathbf{x}}_3\}$ are observed (Figure 3(a)). The initial basis (line 1) is constructed using $\tilde{\mathbf{x}}_1$ and $\tilde{\mathbf{x}}_2$, since they are linearly independent (Figure 3(b)). Then, the point $\tilde{\mathbf{x}}_3$ is selected (line 6 and Figure 3(c)) and the function `recurseTI` (line 9) returns a basis that solves (13), i.e., a basis with the largest lattice determinant that includes all observed vectors. Figure 3(f) illustrates such a basis, and Figure 3(g) shows an equivalent basis obtained after lattice reduction.

ALGORITHM 1: `TI` algorithm

Input: *Set of observed vectors* $\mathcal{O} = \{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_P\}$
Output: *A basis* $\mathbf{B}$ *of the lattice solution of* (13)
1) $\mathbf{B}^{(0)} = \texttt{initBasis}(\mathcal{O})$;
2) $\mathcal{S} = \{\mathbf{b}_1, \ldots, \mathbf{b}_N\}$;
3) $\mathcal{U} = \mathcal{O} \setminus \mathcal{S}$;
4) $r = 0$
5) **while** card$\{\mathcal{U}\} > 0$;
6) Pick $\tilde{\mathbf{x}}$ in $\mathcal{U}$;
7) $\mathcal{U} = \mathcal{U} \setminus \{\tilde{\mathbf{x}}\}$;
8) $\mathcal{S} = \mathcal{S} \cup \tilde{\mathbf{x}}$;
9) $\mathbf{B}^{(r+1)} = \texttt{recurseTI}(\mathbf{B}^{(r)}, \mathcal{S})$;
10) $r = r + 1$
11) **end**

A. The implementation of `recurseTI`

The function `recurseTI` receives as input a set of visited vectors $\mathcal{S}$ and the current estimate of a basis $\mathbf{B}$ for the lattice $\mathcal{L}(\mathbf{B})$. If $\mathcal{S} \subset \mathcal{L}$, i.e., all the vectors in $\mathcal{S}$ belong to the lattice defined by $\mathbf{B}$, the recursion is terminated (line 1 in Algorithm 2). Otherwise, one of the vectors $\mathbf{z}$ that does not belong to $\mathcal{L}$ is selected (line 4) and the parallelotope which encloses it is identified (line 5). Then, a vector $\mathbf{d}$ is computed as the difference between $\mathbf{z}$ and one of the vertices of the parallelotope (line 7).

Specifically, given a basis $\mathbf{B}$ as input, we compute the vector $\mathbf{v} = \mathbf{B} \cdot \text{round}(\mathbf{B}^{-1}\mathbf{z})$, which represents one of the vertices of the parallelotope enclosing $\mathbf{z}$. In order to prevent numerical instability induced by the inversion of the matrix $\mathbf{B}$, we perform basis reduction according to the LLL algorithm and we find a nearly orthogonal basis which is equivalent to $\mathbf{B}$, but has a smaller orthogonality defect. In our implementation, we perform basis reduction only when the condition number is greater than a threshold T, which was set equal to 10^4.

Although any candidates $\mathbf{z} \in \mathcal{S} \setminus \mathcal{L}(\mathbf{B})$ can be selected, we pick the one that minimizes the distance from the corresponding vertex $\mathbf{v}$, so as to minimize the length of the new basis vector $\mathbf{d}$. The intuition here is to capture a short vector that cannot be represented by the current lattice, and to modify the current basis in such a way that (upon convergence) it can be represented.

Hence, the updated basis is constructed by replacing one of the columns of $\mathbf{B}$ with $\mathbf{d}$ (line 8). Among the N possible cases, any choice such that $\mathbf{B}_i$ is non-singular represents a valid selection (line 9). The choice of the new basis among the set of (up to) N candidate bases $\mathbf{B}_i$ is implemented as selecting the one that leads to the smallest lattice determinant, after excluding those that

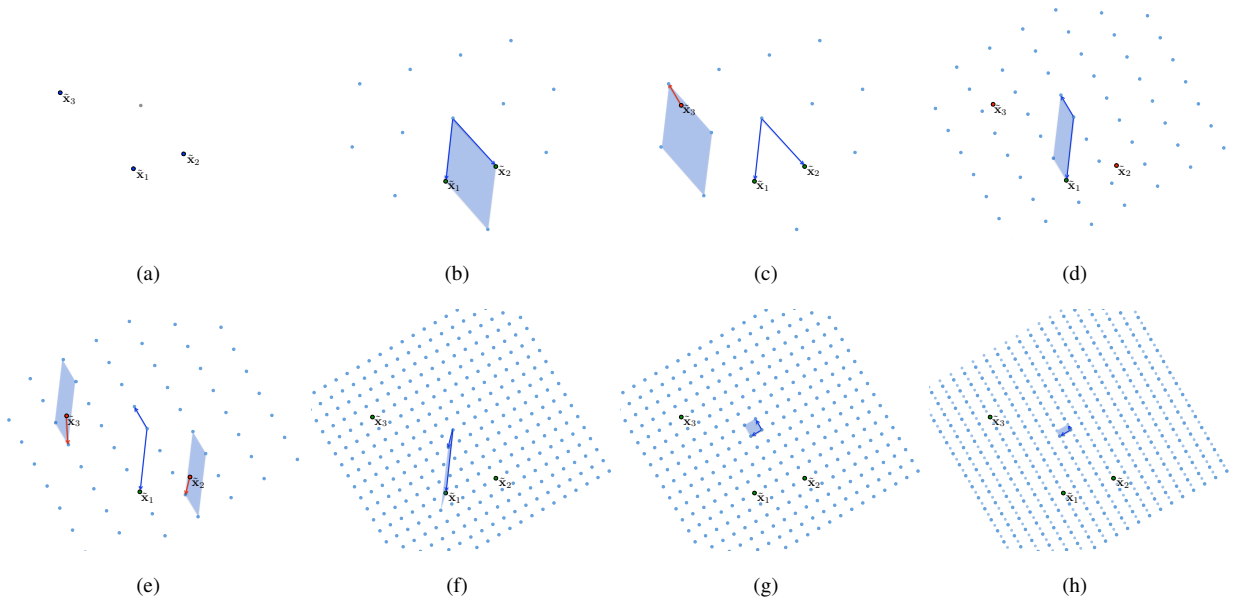


Fig. 3. An example of transform identification. A set of three observed vectors is given in (a). Then, (b)-(h) show, step-by-step, how the solution to problem (13) is sought by Algorithm 1.

do not have rank N. From Cramer's rule, it follows that $\det(\mathbf{B}_i) = \theta_i \det(\mathbf{B})$, where $\boldsymbol{\theta} = \mathbf{B}^{-1}\mathbf{d}$ is the expansion of $\mathbf{d}$ in the basis $\mathbf{B}$. Hence, we replace the l-th column of $\mathbf{B}$, which is the one corresponding to the entry of $\boldsymbol{\theta}$ with the least strictly positive absolute value. That is,

$$l = \arg \min_{j \in \{p | \theta_p \neq 0\}} |\theta_j|. \tag{14}$$

In the example in Figure 3, two recursive steps are performed before terminating `recurseTI`. In the first call, it is verified that $\tilde{\mathbf{x}}_3$ does not belong to the lattice defined by the current basis (Figure 3(c)), and the updated basis is constructed (Figure 3(d)) by replacing one of the two basis vectors with the difference vector between $\tilde{\mathbf{x}}_3$ and one of the vertices of $\mathcal{P}_{\tilde{\mathbf{x}}_3}(\mathbf{B})$. In the second call it is verified that neither $\tilde{\mathbf{x}}_3$ nor $\tilde{\mathbf{x}}_2$ belong to the updated lattice. Therefore, one of the two difference vectors (e.g., the one representing the difference between $\tilde{\mathbf{x}}_2$ and one of the vertices of $\mathcal{P}_{\tilde{\mathbf{x}}_2}(\mathbf{B})$) is used to replace one of the two basis vectors. In the third call the recursion is terminated, because all points in $\mathcal{S}$ belong to the lattice.

ALGORITHM 2: `recurseTI`$(\mathbf{B}, \mathcal{S})$

Input: *Set of vectors* $\mathcal{S} = \{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_S\}$, *a basis* $\mathbf{B}$ *of a lattice.*
Output: *A basis of a lattice* $\mathcal{L}$ *with maximum determinant* $|\mathcal{L}|$, *such that* $\mathcal{S} \subset \mathcal{L}$

```
if S ⊂ L(B)
    return B
else
    Pick z ∈ S \ L(B).
    Determine P_z(B).
    Pick a vertex v of P_z(B).
    Compute d = z − v.
    Compute B_i, replacing the i-th column of B with d.
    Pick an index l, such that det(B_l) ≠ 0.
    recurseTI(B_l, S);
end
```

B. Analysis

1) Convergence: In this section, we prove that the proposed algorithm converges in a finite number of recursive steps to the solution $\mathcal{L}^*$ of (13). Let $\mathbf{B}^{(0)}$ denote the initial estimate of a basis of the lattice. Hence, each vector of $\mathbf{B}^{(0)}$ can be expressed as a linear combination with integer coefficients of the columns of $\mathbf{B}_x$. Thus, we can write $\mathbf{B}^{(0)} = \mathbf{B}_x\mathbf{A}$, with $\det(\mathbf{A}) = m$ and $m \in \mathbb{Z} \setminus \{0\}$. From this, it follows that $|\mathcal{L}(\mathbf{B}^{(0)})| = m \cdot |\mathcal{L}_x|$ and $|\mathcal{L}_x| \leq |\mathcal{L}(\mathbf{B}^{(0)})|$

Let $\mathbf{B}^{(r)}$ denote the estimate obtained after the r-th call of the recursive function `recurseTI`. It is possible to prove the following lemma:

Lemma 4.1: $|\mathcal{L}(\mathbf{B}^{(r+1)})| \leq |\mathcal{L}(\mathbf{B}^{(r)})|$, with equality if and only if $\mathcal{S} \subset \mathcal{L}(\mathbf{B}^{(r)}) = \mathcal{L}(\mathbf{B}^{(r+1)})$

With Lemma 4.1 it is possible to prove the convergence of the proposed method

Theorem 4.2: Algorithm 1 converges to the solution of (13) in a finite number of steps.

Proofs are reported in the Appendix.

2) Rate of convergence: It is possible to prove that the proposed method converges in a number of steps

that is upper bounded by $\lceil \log_2(|\mathcal{L}(\mathbf{B}^{(0)})|/|\mathcal{L}_x|) \rceil$. To show this, it suffices to demonstrate that the value of the lattice determinant is (at least) halved between two consecutive calls of `recurseTI`, as stated by the following theorem.

Theorem 4.3: If $\mathcal{S} \not\subset \mathcal{L}(\mathbf{B}^{(r)})$, then $\frac{|\mathcal{L}(\mathbf{B}^{(r+1)})|}{|\mathcal{L}(\mathbf{B}^{(r)})|} \leq \frac{1}{2}$

The proof is reported in the Appendix.

Based on Theorem 4.3,

$$|\mathcal{L}(\mathbf{B}^{(r)})| \leq \left(\frac{1}{2}\right)^r |\mathcal{L}(\mathbf{B}^{(0)})|, \quad \forall r > 0, \mathcal{S} \not\subset \mathcal{L}(\mathbf{B}^{(r)}) \tag{15}$$

Hence, convergence is achieved in up to

$$\left\lceil \log_2 \frac{|\mathcal{L}(\mathbf{B}^{(0)})|}{|\mathcal{L}_x|} \right\rceil \tag{16}$$

number of steps.

3) Probability of success: In Section IV-B1, we showed that the proposed method converges to the optimal solution $\mathcal{L}^*$ of (13). In this section, we show that it converges to the correct (and unique) lattice $\mathcal{L}_x$ (i.e., $\mathcal{L}^* \equiv \mathcal{L}_x$) with high probability, provided that the number of observed vectors P is greater than N.

Given a lattice $\mathcal{L}_x$ of rank N embedded in $\mathbb{R}^N$, there is more than one sub-lattice $\underline{\mathcal{L}}$ of $\mathcal{L}$ of index m. It can be shown that the number of sub-lattices is equal to [34]

$$f_N(m) = \prod_{i=1}^{q} \prod_{j=1}^{N-1} \frac{p_i^{t_i+j} - 1}{p_i^j - 1} = \prod_{i=1}^{q} \prod_{j=1}^{t_i} \frac{p_i^{N+j-1} - 1}{p_i^j - 1}, \tag{17}$$

where $m = p_1^{t_1} \cdots p_q^{t_q}$ is the prime factorization of m. That is, $p_1, \ldots, p_q$ are the prime factors of m, and t_s is the multiplicity of the factor p_s.

In order to determine analytically a lower bound on the probability of converging to the correct solution, we need to prove the following lemma, which provides bounds on the number of sub-lattices.

Lemma 4.4: Given a lattice $\mathcal{L}_x$ of rank N embedded in $\mathbb{R}^N$, the number $f_N(m)$ of sub-lattices of index m is bounded by

$$m^{N-1} < f_N(m) < m^N. \tag{18}$$

The proof is reported in Appendix.

Now, consider a specific sub-lattice $\underline{\mathcal{L}} \subset \mathcal{L}_x$ of index m and a set of P vectors from the original lattice $\mathcal{L}_x$. In the case of uniformly distributed vectors, the probability that one vector belong to the sub-lattice $\underline{\mathcal{L}}$ is equal to $(1/m)$. Thus, the probability that all P vectors belong to the same sub-lattice $\underline{\mathcal{L}}$ is equal to $(1/m)^P$, assuming statistical independence among the set of vectors.

Let $p_{\text{fail}}(N, P)$ denote the probability of failing to detect the underlying lattice $\mathcal{L}_x$ of rank N, when P points are observed. Then, $p_{\text{succ}}(N, P) = 1 - p_{\text{fail}}(N, P)$. A failure occurs whenever all P vectors fall in any of the sub-lattices of index m. Hence, we can write

$$\begin{aligned} p_{\text{fail}}(N, P) &< \sum_{m=2}^{\infty} f_N(m) \left(\frac{1}{m}\right)^P < \sum_{m=2}^{\infty} m^N \left(\frac{1}{m}\right)^P \\ &= \sum_{m=2}^{\infty} \frac{1}{m^{P-N}} = \zeta(P-N) - 1 \end{aligned} \tag{19}$$

The first inequality is a union bound, i.e., the probability of failure is upper bounded by the sum of the probabilities of observing all P vectors in a given sub-lattice. The second inequality follows from the upper bound given by Lemma 4.4. The last expression contains $\zeta(\cdot)$, which is the Riemann's zeta function. That is,

$$\zeta(s) = \sum_{m=1}^{\infty} \frac{1}{m^s}. \tag{20}$$

Note that the infinite series converges when the real part of the argument s is greater than 1. In our case, this requires $P - N > 1$ or $P > N + 1$. Then, the probability of success is lower bounded by

$$p_{\text{succ}}(N, P) > 2 - \zeta(P - N). \tag{21}$$

It is interesting to observe that the probability of failure/success depend solely on the difference $P - N$. Hence, the number P of observed vectors needed to correctly identify the underlying lattice grows linearly with the dimensionality N of the embedding vector space, despite the number of potential lattice points grows exponentially with N, as indicated in Section IV.

V. Handling a chain of two transform coders

In this section we propose an algorithm that is able to determine the parameters of the first transform coder in a processing chain of two transform coders, by observing the output of the second. The key idea of the proposed method is to de-quantize the observed vectors before applying the algorithm described in Section IV. To this end, we proceed according to the following steps:

A. We consider a subset of $D \leq P$ observed vectors $\{\tilde{\mathbf{z}}_1, \ldots, \tilde{\mathbf{z}}_{l_D}\}$ and for them we recover both the distances from the origin and the distances between pairs of vectors, as they were before the application of the second transform coder.

B. Given the recovered distances, we computed a set of de-quantized vectors $\{\hat{\mathbf{z}}_{l_1}, \ldots, \hat{\mathbf{z}}_{l_D}\}$, which lie exactly on a lattice whose basis can be expressed as $\mathbf{B}_{\hat{z}} = \mathbf{R}\mathbf{W}_b\mathbf{B}_x$. The orthonormal matrix $\mathbf{R}$ represents the ambiguity introduced by the denoising procedure.

C. We compute an estimate $\hat{\mathbf{R}}$ of $\mathbf{R}$ by formulating an orthogonal Procrustes problem, which

seeks the best matching between the set of de-quantized vectors $\{\hat{\mathbf{z}}_{l_1}, \ldots, \hat{\mathbf{z}}_{l_D}\}$ and the observed ones $\{\tilde{\mathbf{z}}_{l_1}, \ldots, \tilde{\mathbf{z}}_{l_D}\}$.

D. We adopt the method described in Section IV to the vectors $\{\hat{\mathbf{x}}_{l_1}, \ldots, \hat{\mathbf{x}}_{l_D}\}$, $\hat{\mathbf{x}}_j = \mathbf{W}_b^{-1}\hat{\mathbf{R}}^{-1}\hat{\mathbf{z}}_{l_j}$, $j = 1, \ldots, D$, to obtain an estimate $\hat{\mathbf{B}}_x$ of $\mathbf{B}_x$.

Then, in Section V-E, we describe an iterative method that can be adopted whenever it is not possible to de-quantize D observed vectors at once, due to, e.g., the use of a large quantization step size by the second transform coder.

A. Exact recovery of vector distances

The denoising operation indicated in Step 1 exploits the orthogonality of the transform $\mathbf{W}_a$ to determine the quantization step size Δ_a and, consequently, to recover inter-vector distances exactly. Indeed, it is possible to express a constraint on the lengths $\tilde{\delta}_j$ of the unobserved vectors $\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_P\}$ as well as on the lengths $\tilde{\delta}_{j_1,j_2}$ of vector differences. That is,

$$\tilde{\delta}_j^2 = \|\tilde{\mathbf{x}}_j\|^2 = a_j\Delta_a^2, \quad a_j \in \mathbb{N}, \quad j = 1, \ldots, P, \tag{22}$$

$$\begin{aligned}\tilde{\delta}_{j_1,j_2}^2 &= \|\tilde{\mathbf{x}}_{j_1} - \tilde{\mathbf{x}}_{j_2}\|^2 = a_{j_1,j_2}\Delta_a^2,\\ a_{j_1,j_2} &\in \mathbb{N},\, j_1, j_2 = 1, \ldots, P.\end{aligned} \tag{23}$$

In practice, both a_j and a_{j_1,j_2} belong to the subset of integer numbers corresponding to those that can be written as the sum of (up to) N squares. However, when $N \geq 4$, this subset coincides with the set of integer numbers, as proven by Lagrange's four square theorem [35].

In order to determine Δ_a, which is unknown, we first note that $\|\tilde{\mathbf{x}}_i\| = \|\mathbf{z}_i\|$, $i = 1, \ldots, P$, since $\mathbf{W}_b$ is orthonormal. Therefore, Δ_a is the square root of the greatest common divisor of $\{\|\mathbf{z}_1\|^2, \ldots, \|\mathbf{z}_P\|^2\}$. However, we have no access to $\mathbf{z}_i$'s, but only to their quantized versions $\tilde{\mathbf{z}}_i$. The second quantization can be seen as a form of noise. Hence, we estimate Δ_a by using the algorithm in [36], which is a generalized Euclid's algorithm for noisy measurements. Given Δ_a, to recover vector distances, we first note that

$$|\tilde{\rho}_j - \tilde{\delta}_j| = |\|\tilde{\mathbf{z}}_j\|_2 - \|\tilde{\mathbf{x}}_j\|_2| \leq \frac{1}{2}\sqrt{N}\Delta_b\,. \tag{24}$$

The quantization error on the length of a vector can be as large as half of the diagonal of the quantization cell of the second quantizer. In case of lengths of vector differences $|\tilde{\rho}_{j_1,j_2} - \tilde{\delta}_{j_1,j_2}|$, the error can be twice as that, since both vectors are quantized.

If the quantization error induced by the second quantizer is sufficiently small, it is possible to recover the exact values of $\tilde{\delta}_j$ and $\tilde{\delta}_{j_1,j_2}$ from the observed vectors. Indeed, we exploit the fact that $\tilde{\delta}_j^2$ is a multiple of Δ_a^2 (a similar argument holds for $\tilde{\delta}_{j_1,j_2}^2$). To this end, we compute an estimate $\hat{\delta}_j$ of $\tilde{\delta}_j$ as follows:

$$\hat{\delta}_j = \sqrt{\mathcal{Q}_{\Delta_a^2}(\tilde{\rho}_j^2)}\,. \tag{25}$$

Note that any value of $\tilde{\rho}_j$ in the interval $[l_j, u_j]$:

$$\left[\sqrt{\tilde{\delta}_j^2 - \frac{\Delta_a^2}{2}}, \sqrt{\tilde{\delta}_j^2 + \frac{\Delta_a^2}{2}}\right] \tag{26}$$

is quantized to $\tilde{\delta}_j$. Hence, if $|\tilde{\rho}_j - \tilde{\delta}_j| < \min\{u_j - \tilde{\delta}_j, \tilde{\delta}_j - l_j\} = u_j - \tilde{\delta}_j$, it is possible to guarantee that $\hat{\delta}_j = \tilde{\delta}_j$.

Figure 4 illustrates an example in which two N-dimensional (unobserved) vectors $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2\}$ are processed by a second transform coder. The coordinate axes are aligned with the basis functions of the known transform $\mathbf{W}_b$. Hence, we display $\{\tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_2\}$. Figure 5 shows that the corresponding vector lengths, i.e., $\tilde{\rho}_1$ and $\tilde{\rho}_2$, and the distance between vectors, i.e., $\tilde{\rho}_{1,2}$, can be effectively de-quantized so that $\tilde{\delta}_1$, $\tilde{\delta}_2$ and $\tilde{\rho}_{1,2}$ can be recovered exactly.

We conclude that $\hat{\delta}_j = \tilde{\delta}_j$ whenever the following sufficient condition is satisfied

$$\frac{1}{2}\sqrt{N}\Delta_b < u_j - \tilde{\delta}_j = \sqrt{\tilde{\delta}_j^2 + \frac{\Delta_a^2}{2}} - \tilde{\delta}_j = \tau(\tilde{\delta}_j; \Delta_a). \tag{27}$$

B. Denoising observed vectors

Given the de-quantized values of vector lengths and inter-vector distances, it is possible to write the following set of constraints in the unknown vectors $\{\hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_P\}$

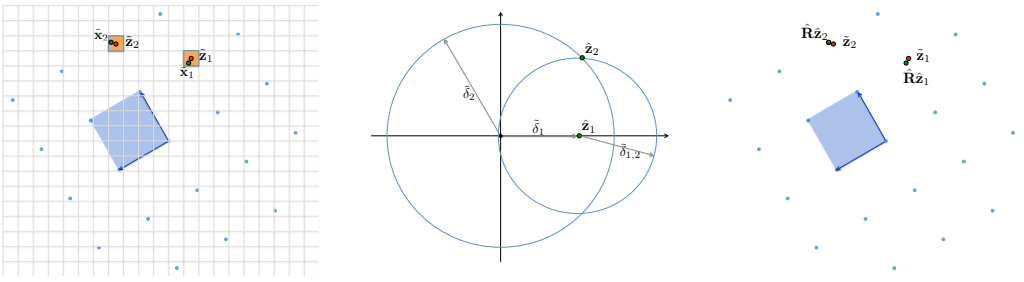
$$\|\hat{\mathbf{z}}_j\|_2 = \sqrt{\mathcal{Q}_{\Delta_a^2}(\tilde{\rho}_j^2)} = \tilde{\delta}_j \quad j \in \mathcal{O}, \tag{28}$$

$$\|\hat{\mathbf{z}}_{j_1} - \hat{\mathbf{z}}_{j_2}\|_2 = \sqrt{\mathcal{Q}_{\Delta_a^2}(\tilde{\rho}_{j_1,j_2}^2)} = \tilde{\delta}_{j_1,j_2} \quad (j_1, j_2) \in \mathcal{D}, \tag{29}$$

where $\mathcal{O}$ denotes the set of indexes of the vectors whose length can be recovered exactly, i.e., those that satisfy (27).

$\mathcal{D}$ is similarly defined, denoting the set of indexes of pairs of vectors whose distance can be recovered exactly.

Consider a subset of the unknown vectors $\{\hat{\mathbf{z}}_{l_1}, \ldots, \hat{\mathbf{z}}_{l_D}\}$ for which the distances from the origin are known, i.e., $l_j \in \mathcal{O}$, and the distances between all pairs are also known, i.e., $(l_{j_1}, l_{j_2}) \in \mathcal{D}$. If $D \geq N$, the position of the vectors can be determined exactly, apart from an ambiguity that can be represented by means of an arbitrary orthonormal transform, which accounts for the rotation around the origin and mirroring with respect to the coordinate



(a) (b) (c)

Fig. 4. A toy example with two unobserved vectors $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2\}$ and the corresponding observed vectors $\{\tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_2\}$, in the coordinate system of the known transform $\mathbf{W}_b$.

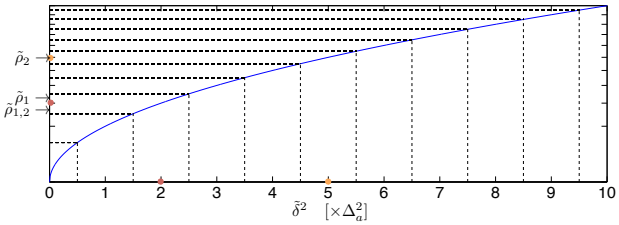


Fig. 5. Square distance values $\tilde{\delta}^2$ are constrained to be integer multiples of Δ_a^2. On the y-axis, it is illustrated the corresponding quantizer applied to de-quantized observed distances $\tilde{\rho}$.

axes. Therefore, we proceed in two steps. First, we seek an arbitrary feasible solution $\{\hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_P\}$ of (28). It can be shown that a feasible solution can be found as follows

- Initialize the solution by setting $\hat{\mathbf{z}}_{l_1} = [\tilde{\delta}_{l_1}, 0, \ldots, 0]^T$
- The remaining components are iteratively estimated, by setting $\hat{\mathbf{z}}_{l_j} = [\hat{z}_{l_j,1}, \hat{z}_{l_j,2}, \ldots, \hat{z}_{l_j,j}, \mathbf{0}^T]^T$, $j = 2, \ldots, N$, and finding a solution in a j-dimensional subspace of the following system of equations

$$\begin{aligned} \hat{\mathbf{z}}_{l_j}^T \hat{\mathbf{z}}_{l_j} &= \tilde{\delta}_{l_j}^2 \\ (\hat{\mathbf{z}}_{l_j} - \hat{\mathbf{z}}_{l_1})^T (\hat{\mathbf{z}}_{l_j} - \hat{\mathbf{z}}_{l_1}) &= \tilde{\delta}_{l_1,l_j}^2 \\ &\ldots \\ (\hat{\mathbf{z}}_{l_j} - \hat{\mathbf{z}}_{l_{j-1}})^T (\hat{\mathbf{z}}_{l_j} - \hat{\mathbf{z}}_{l_{j-1}}) &= \tilde{\delta}_{l_{j-1},l_j}^2 \end{aligned} \quad . \tag{30}$$

It is possible to show that (30) has a straightforward geometric interpretation, since it represents the intersection of a line in N-dimensional space with a hypersphere centered in the origin with radius $\tilde{\delta}_j$. The solution is not unique. However, for the problem at hand, it suffices to select arbitrarily a feasible solution. Figure 4(b) illustrates a feasible solution corresponding to the example in Figure 4(a).

C. Resolving the reference system ambiguity

The resulting vectors $\{\hat{\mathbf{z}}_{l_1}, \ldots, \hat{\mathbf{z}}_{l_D}\}$ lie exactly on a lattice whose basis can be expressed as $\mathbf{B}_{\hat{z}} = \mathbf{R}\mathbf{W}_b\mathbf{B}_x$. The orthonormal matrix $\mathbf{R}$ represents the ambiguity introduced by the arbitrary choice of the reference system, as well as the arbitrary choice when selecting the feasible solution. In order to solve such ambiguity, we seek an estimate $\hat{\mathbf{R}}$ of the matrix $\mathbf{R}$ by matching the positions of the vectors $\{\hat{\mathbf{z}}_{l_1}, \ldots, \hat{\mathbf{z}}_{l_D}\}$ to those of the observed vectors $\{\tilde{\mathbf{z}}_{l_1}, \ldots, \tilde{\mathbf{z}}_{l_D}\}$. This can be formulated as the following orthogonal Procrustes problem

$$\hat{\mathbf{R}} = \arg\min_{\mathbf{R}} \|\mathbf{R}\hat{\mathbf{Z}} - \tilde{\mathbf{Z}}\|_F \quad s.t. \quad \mathbf{R}^T\mathbf{R} = \mathbf{I}, \tag{31}$$

where $\hat{\mathbf{Z}} = [\hat{\mathbf{z}}_{l_1}, \ldots, \hat{\mathbf{z}}_{l_D}]$ and $\tilde{\mathbf{Z}} = [\tilde{\mathbf{z}}_{l_1}, \ldots, \tilde{\mathbf{z}}_{l_D}]$ and $\|\cdot\|_F$ denotes the Frobenius norm.

D. Determining the transform basis functions

Finally, we obtain an estimate of the unobserved vectors as $\hat{\mathbf{x}}_j = \mathbf{W}_b^{-1}\hat{\mathbf{R}}^{-1}\hat{\mathbf{z}}_{l_j}$, $j = 1, \ldots, D$. The set of vectors $\{\hat{\mathbf{x}}_{l_1}, \ldots, \hat{\mathbf{x}}_{l_D}\}$ is guaranteed to lie on a lattice $\hat{\mathbf{B}}_x$, which represents an estimate of the lattice induced by the first transform coder $\mathbf{B}_x$. This can be achieved with the method described in Section IV, provided that a sufficiently large number of vectors is available to converge to the correct lattice defined by $\hat{\mathbf{B}}_x$, rather than to one of its sub-lattices. The basis functions (row vectors) of the transform $\mathbf{W}_a$ can be obtained as $\hat{\mathbf{w}}_{a,i} = \hat{\mathbf{b}}_{x,i}/\|\hat{\mathbf{b}}_{x,i}\|$, exploiting the orthonormality of the transform.

E. Iterative estimation

Given a set of P observed vectors $\{\tilde{\mathbf{z}}_1, \ldots, \tilde{\mathbf{z}}_P\}$, the number $D \leq P$ of vectors that can be effectively de-quantized depends on different factors: i) the statistical distribution of the source from which the original vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_P\}$ are sampled; ii) the quantization step sizes used in the first and second transform coder. A

careful analysis of the bound in equation (27) reveals that it is easier to denoise vectors whose length (expressed in Δ_a units) is short, and when $\Delta_b \ll \Delta_a$. In addition, when the dimensionality N increases: i) the average length of vectors increases; ii) the bound in (27) is more stringent, thus enabling to denoise shorter vectors. Therefore, in some cases, it might not be possible to denoise at least $N+n$ vectors at once, since the sufficient condition in (27) may not be satisfied for a large enough number of vectors. Hence, we propose to modify the algorithm in such a way that the solution is sought incrementally. The key idea is to start from the largest set of $D^{(0)}$ short vectors that can be efficiently denoised. These are used to estimate a subset of the basis functions of $\mathbf{W}_a$, i.e., $\hat{\mathbf{W}}_a^{(0)} \in \mathbb{R}^{M^{(0)} \times N}$, where $M^{(0)}$ is the dimensionality of the span of the denoised vectors. Then, the remaining vectors are projected in the null-space of $(\hat{\mathbf{W}}_a^{(0)})^T$, and the denosing procedure is applied to the result. The iterative procedure terminates when either $M^{(k)} = N$, or when all vectors are denoised.

VI. Handling outliers

In this section we discuss how to handle outlier observations. This is useful to model cases in which some of the observed vectors do not represent the output of a processing chain of one or two transform coders. For example, this might capture two important application scenarios:

- *Local tampering*: The signal produced as output of a chain of transform coders is locally manipulated in the original domain. For example, an image is tampered with the aim of altering its content. As a result, the observed vectors corresponding to the modified region do not lie on a (noisy) lattice.
- *Predictive coding*: In some applications, transform coding is combined with predictive coding. For example, in the case of video, transform coding is applied to motion compensated prediction residuals. Prediction residuals can be recovered from the decoded samples by applying the same motion vectors used at the encoder. Despite using the same motion estimation algorithm (which can be detected, e.g., using the method in [37]), some motion vectors might differ from those used at the encoder and, consequently, the prediction residuals of the corresponding blocks do not lie on a (noisy) lattice.

Based on the theory and algorithms presented in Section IV and V, we propose the following model to address the aforementioned scenarios. Given a set of observed vectors $\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_P\}$, a subset of O vectors are outliers, in the sense that they do not represent the output of a transform coder chain. In order to apply our techniques, first we need to identify the outliers and to remove them from the set of observed vectors. To do so, we rely on the constraints imposed on the vector lengths by the transform coding chain, detecting as outliers those vectors that do not fulfil such constraints.

First, let us consider the case in which the remaining $P-O$ vectors are observed at the output of a single transform coder. The lengths of such genuine vectors are constrained to be multiples of Δ_a^2, as indicated in (22). Therefore, it is possible to estimate Δ_a by using the algorithm in [36], and then identify as outliers those observed vectors whose squared lengths do not fulfil the constraint. The transform coder can be identified by providing the remaining $P-O$ as input to the algorithm described in Section IV, provided that $P-O-N>0$.

Second, let us consider the more challenging case in which the remaining $P-O$ are the output of a chain of two transform coders. In this case, the lengths of genuine vectors are constrained to satisfy (24). Although it is possible for an outlier to satisfy such constraint, we can quantify the probability α that this occurs. This allows one to discard a fraction of $1-\alpha$ outliers from the set of observed vectors. To compute α, one needs to know the statistical distribution of the outlier vectors. For example, if the elements of the outlier vectors are modelled as an i.i.d. Gaussian distribution $N(0, \sigma_x^2)$, the distribution of $\tilde{\delta}^2$ is chi-square with N degrees of freedom. Figure 6 illustrates such example, in which the shaded area represents the values of $\tilde{\delta}^2$ that are compatible with (24) for the case where $N=8$, $\Delta_a = 1$, $\Delta_b = 0.01$, $\sigma_x = 1$. The area under the curve corresponding to the shaded area is equal to α. In this example, 85% of the outliers were successfully detected. In Section VII we evaluate α as a function of the main operational parameters.

Although this method does not eliminate all outliers, on average only $\bar{O} = \text{round}\{\alpha O\}$ remain. The algorithm is then executed multiple times, in a RANSAC-like fashion, each time with a subset of $\bar{N} = N + k$ vectors (with $k = 7$, as determined in Section VII). There are $\binom{P-(1-\alpha)O}{\bar{N}}$ such subsets on average, but the algorithm does not need to visit all subsets. Indeed, when a subset does not contain any outlier, the algorithm converges to the correct solution. Convergence can be verified by checking the coverage of points outside the subset used by the algorithm. The probability that a subset does not contain any outlier is equal to $(1-q)^{\bar{N}}$ (illustrated in Figure 7), where $q = O/P$ is the fraction of outliers. Note that q decreases to $q = \alpha O/P$ when applying the method described above. Continuing with the example above, if $N = 16$ and $q = 0.1$, then the probability of finding a subset without outliers grows from 20% to 78%, thus significantly reducing the number of subsets that need to be visited before reaching convergence. In Section VII we evaluate the number of visited subsets as a function of the main operational parameters.

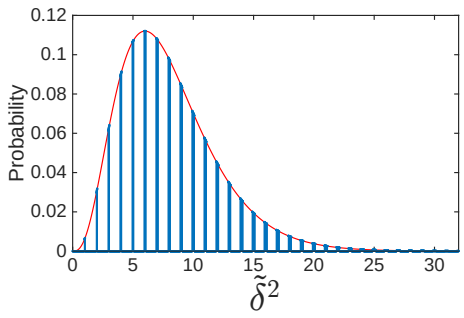


Fig. 6. Distribution of square vector lengths. The shaded area under the curve is equal to α.

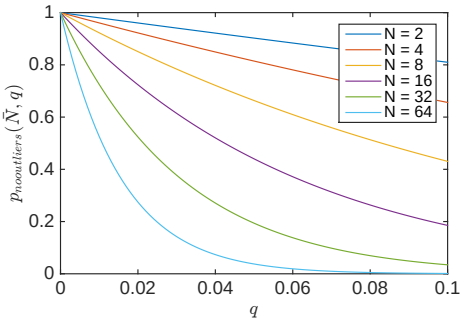


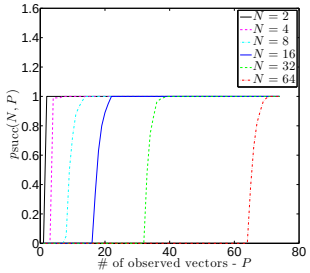
Fig. 7. Probability that a subset of $\bar{N}$ points does not contain any outlier, as a function of the fraction of outliers q.

VII. Experiments

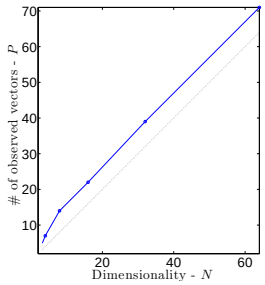
We experimentally evaluated the algorithms discussed in Section IV and Section V. First, for the case of a single transform coder, we evaluate in Section VII-A the number of observed vectors needed to successfully identify the transform and the number of recursive steps needed to compute the sought solution, which provides an insight on the complexity of the algorithm. Then, for the case of a chain of two transform coders, we evaluate under which conditions it is possible to de-quantize the observed vectors, and show how it is possible to recover the transform used to compress an image.

A. Transform coder identification

We generated data sets of N-dimensional vectors, whose elements are sampled from a Gaussian random variable $\mathcal{N}(0, \sigma^2)$. We considered the adverse case in which the elements are independent and identically distributed. Therefore, the distribution of the vectors is isotropic and no clue could be obtained from a statistical analysis of the distribution. Without loss of generality, we set $\sigma = 2$, $\mathbf{W} = \mathbf{I}$ and $\Delta_i = 1$, $i = 1, \ldots, N$. The same results were obtained using different transform matrices and quantization step sizes.



(a)



(b)

Fig. 8. (a) Empirical probability of success of Algorithm 1 in identifying the transform and the quantization step sizes as a function of the number of observed vectors P and the dimensionality of the embedding vector space N. (b) Number of observed vectors P needed to achieve $p_{\text{succ}}(N, P) > 1 - \epsilon$, with $\epsilon = 10^{-15}$.

Figure 8(a) shows the empirical probability of success when $N = 2, 4, 8, 16, 32, 64$, and the number of observed vectors P is varied, averaged over 100 realizations. As expected $p_{\text{succ}}(N, P) = 0$ when the number of vectors P does not exceed the dimensionality of the embedding vector space, i.e., $P \leq N$. Then, as soon as $P > N$, $p_{\text{succ}}(N, P)$ grows rapidly to one. More specifically, Figure 8(b) illustrates the number of observed vectors P needed to achieve $p_{\text{succ}}(N, P) > 1 - \epsilon$, where ϵ was set equal to 10^{-15}. We note that when $N > 2$, the number of observed vectors needs to exceed by 6-7 units the dimensionality, and such a difference is independent from N, as expected based on the analysis in Section IV-B. Note that the results shown in Figure 8 are completely oblivious of the specific implementation of Algorithm 2.

At the same time, it is interesting to evaluate the complexity of Algorithm 2. Figure 9 shows the total number of recursive calls needed to converge to the solution of (13). Note that when a large enough number P of vectors is observed, the algorithm converges to the correct lattice $\mathcal{L}_x$. Thus, visiting additional vectors does not increase the number of recursive calls, since the base step of the recursion is always met. Figure 9 shows two cases, that differ in the way the set of observed vectors is visited, i.e., randomly, or sorted in ascending order of distance from the origin of the vector space. In both cases, the number of recursive calls grows linearly with

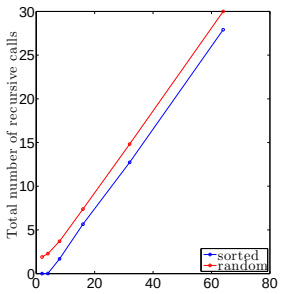


Fig. 9. Total number of recursive calls to `recurseTI` as a function of the dimensionality of the space N and the strategy adopted to visit the observed vectors.

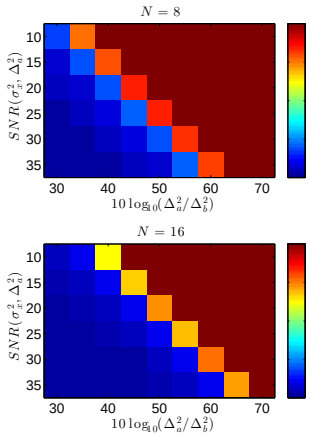


Fig. 10. Number of de-quantized vectors D. Blue: 0. Dark red: $P = 40$.

N. This is aligned with the analysis in Section IV-B2, which shows that convergence proceeds at a rate such that the number of recursive steps is upper bounded by $\lceil \log_2 |\mathcal{L}(\mathbf{B}^{(0)})|/|\mathcal{L}_x| \rceil$. A (loose) bound on the lattice determinant is given by

$$|\mathcal{L}(\mathbf{B}^{(0)})| = |\det(\mathbf{B}^{(0)})| \leq \|\mathbf{b}_1^{(0)}\|_2 \|\mathbf{b}_2^{(0)}\|_2 \cdot \|\mathbf{b}_N^{(0)}\|_2 \leq \|\mathbf{b}_{\max}^{(0)}\|_2^N, \quad (32)$$

where the first inequality stems from Hadamard inequality and $\mathbf{b}_{\max}^{(0)}$ is the column of $\mathbf{B}^{(0)}$ with the largest norm. Therefore,

$$\lceil \log_2 |\mathcal{L}(\mathbf{B}^{(0)})|/|\mathcal{L}_x| \rceil \leq \lceil N \log_2(\|\mathbf{b}_{\max}^{(0)}\|_2)/|\mathcal{L}_x| \rceil \quad (33)$$

This explain the dependency on N, as well as the fact that sorting the vectors so as to initialize $\mathbf{B}^{(0)}$ with shorter vectors reduces the number of recursive calls.

B. Handling a chain of two transform coders

First, we tested our method on synthetic data sampled from a N-dimensional i.i.d. Gaussian distribution with variance equal to σ_x^2. For a fixed dimensionality (either $N = 8$ or $N = 16$), we sampled $P > N$ vectors and fed them into the processing chain depicted in Figure 1. We varied the signal-to-noise (SNR) ratio due to the quantization of the first quantizer, $SNR \simeq 10 \log_{10} \sigma_x^2/(\Delta_a^2/12)$, as well as the ratio between the quantization step sizes of the first and second transform coder. The goal of the experiment is to evaluate in which conditions it is possible to de-quantize a sufficient number of input vectors. Figure 10 illustrates the number D of vectors which are effectively de-quantized, when $P = 40$ vectors were observed. It is possible to notice a "cliff" effect, such that the configurations in the top-right part of the figure correspond to cases in which the proposed algorithm is able to find a solution, since $D \geq N + n$. When varying N, we observed that, for a given value of SNR, a smaller value of Δ_b is required when N increases.

Then, we tested the proposed method on real images. In this case, we considered the 1338 images from the UCID dataset. Each image was compressed with a JPEG-like transform coder. That is, the DCT transform was applied to non-overlapping 8×8 blocks ($N = 64$). Transform coefficients were quantized with a step size in the set $\Delta_a \in \{20, 30, 40, 50, 60\}$, which led to an average PSNR equal to, respectively, $\{36.2, 33.6, 31.8, 30.5, 29.5\}$. Then, the inverse transform was applied to each block, and the result was rounded to the nearest integer in the pixel-domain. Thus, $\mathbf{W}_b = \mathbf{I}$ and $\Delta_b = 1$. In this case, it was possible to successfully recover the transform by adopting the iterative version of the algorithm illustrated in Section V-E, although it was not possible to find at least $D > 64$ vectors that could be de-quantized at once. Figure 11(a) shows for each basis function $\mathbf{w}_{a,j}$ associated to one of the DCT coefficients, the quantity $E[\hat{\mathbf{w}}_{a,j}^T \mathbf{w}_{a,j}]$, which indicates the average cosine of the angle between true and estimated basis functions. We observe that when the PSNR is less than 30dB, all basis functions can be estimated with very high accuracy. At higher values of the PSNR, the error increases as more basis functions are estimated. This is due to the fact that, at each iteration, we project in the null-space of the estimated basis functions $(\hat{\mathbf{W}}_a^{(k)})^T$. Hence, the noise in the observed data is due not only to quantization, but also to the non-ideal projection. In order to evaluate the algorithm with different transforms, we tested the use of the Hadamard transform 11(b), as well as a content-dependent KLT transform 11(c), estimated independently for each image.

As an illustrative example, we show in Figure 11(d) and Figure 11(e) the estimated basis functions (represented by each column of the matrix), when the first image in the UCID dataset was coded using the DCT and $\Delta_a = 20, 30$, respectively. This result can be compared with the DCT basis functions illustrated in Figure 11(f). We observe that most of the basis functions

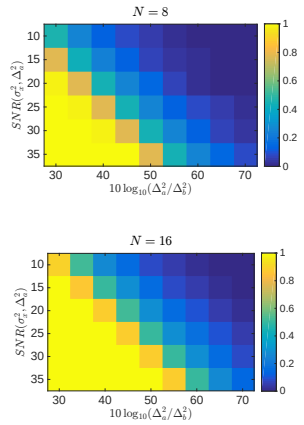


Fig. 12. Fraction of outlier reduction α.

were correctly recovered when $\Delta_a = 30$.

In order to show that our proposed algorithm does not require the step size to be constant across coefficients, we repeated the experiment in the case each coefficient uses a different quantization step size. While in the theory in Section IV we estimate a single quantization step in order to simplify notation, we note that if different quantization steps are used and share a greatest common divisor, this is identified by our algorithm. From the estimated g.c.d., the different steps can be correctly recovered. Since the steps considered are not irrational numbers, they will always share a g.c.d., which can be recovered by our algorithm. We test this by first selecting a base quantization step size $\Delta_a \in \{2, 5, 10, 20, 30, 40, 50, 60\}$. Then, each coefficient was assigned a quantization step size that is an integer multiple of Δ_a, selected at random. The results shown in Figure 13 demonstrate that the algorithm is able to cope with non-uniform quantization step sizes.

C. Handling outliers

We tested our method on a synthetic data sample as in Section VII-B. The goal of the experiment is to assess the fraction α of outliers that cannot be detected by enforcing the constraint on vector lengths, as described in Section VI. Figure 12 illustrates how α varies as a function of the signal-to-noise (SNR) ratio due to the quantization of the first quantizer as well as the ratio between the quantization step sizes of the first and second transform coder. It is possible to observe that: i) for a given value of SNR, α decreases when the ratio Δ_a/Δ_b increases; ii) for a given SNR and Δ_a/Δ_b, α increases with N; iii) when comparing Figure 12 to Figure 10, a significant reduction on the number of outliers is observed for those combinations of parameters for which the algorithm converges.

TABLE I. NUMBER OF VISITED SUBSETS OF VECTORS.

N	Δ_b	α	5th%tile	median	95th%tile
8	0.01	0.15	1	1	3
		1	1	3	9.5
16	0.005	0.16	1	1	3
		1	1	10	32
32	0.0025	0.16	1	2	9
		1	10	89	375
64	0.00125	0.15	1	4	13.5
		1	2620	24800	>100000

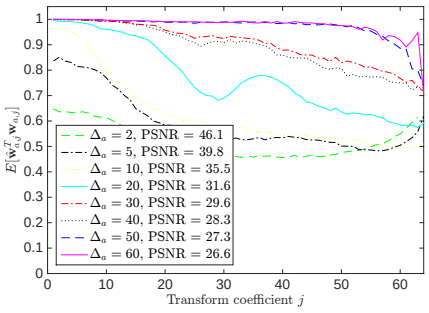
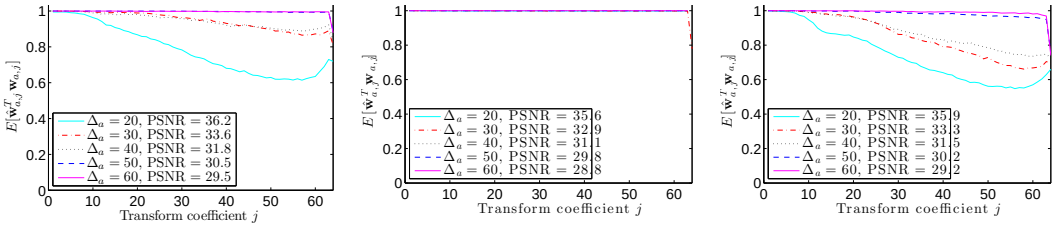


Fig. 13. Transform estimation example in the case of non-uniform quantization step sizes.

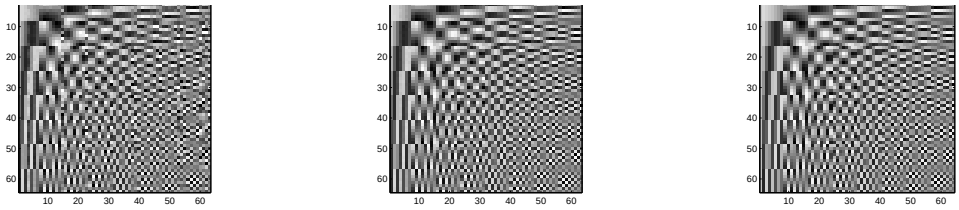
Second, we evaluated the number of subsets of observed vectors of size $\bar{N}$ that need to be visited before finding at least one set without outliers. The results are shown in Table I, where $\sigma_x = 1$, $\Delta_a = 1$, $P = 200$ and $q = 0.1$. For different values of N, Δ_b is adjusted so as to obtain a value of $\alpha \sim 0.16$. The rows with $\alpha = 1$ indicate the case in which no outliers are discarded by the method described in Section VI. It can be observed that: i) even with a relatively small number of observed vectors ($P = 200$), the algorithm converges after visiting a small number of candidate subsets; ii) detecting and discarding outliers is key to dramatically improve convergence, especially when N increases.

VIII. CONCLUSIONS

In this paper we proposed a method which is able to identify the parameters of a transform coder from a set of P transform decoded vectors embedded in a N-dimensional space. We proved that it is possible to successfully identify the transform and the quantization step sizes when $P > N$ and this despite of the huge number of potential quantization bins, which grows exponentially with N for a target bitrate. In addition, we proved that the probability of failure decreases exponentially to zero when $P - N$ increases. In our experiments we found that an excess of approximately 6-7 observed vectors beyond the dimension N of the space is generally sufficient to ensure successful convergence.



(a) DCT (b) Hadamard (c) KLT



(d) Estimated. $\Delta_a = 20$ (e) Estimated. $\Delta_a = 30$ (f) DCT

Fig. 11. Top row: Estimation accuracy. Bottom row: Transform estimation example for the first image in the UCID dataset.

We also studied the challenging problem of estimating both the transform and the quantizer of the first transform coder in a chain of two transform coders. We showed for the first time that, under specific conditions, the effect of the second transform coder can be removed, so that the problem can be addressed using the algorithm used for a single transform coder. Then, we presented an iterative method that successfully solves the problem even when the conditions are not entirely satisfied. Results on real data show the potential of this new approach which might be useful also in image forensics and video processing.

APPENDIX

of Lemma 4.1: If $\mathcal{S} \subset \mathcal{L}(\mathbf{B}^{(r)})$, then $\mathbf{B}^{(r+1)} = \mathbf{B}^{(r)}$ and the recursion terminates. Otherwise, let $\mathbf{z} \in \mathcal{S} \setminus \mathcal{L}(\mathbf{B}^{(r)})$ be any of the points which does not belong to the lattice defined by $\mathbf{B}^{(r)}$, $\mathbf{v}$ any of the vertices of $\mathcal{P}_{\mathbf{z}}(\mathbf{B}^{(r)})$ and $\mathbf{d} = \mathbf{z} - \mathbf{v}$. The vector $\mathbf{d}$ can be expressed in terms of the basis $\mathbf{B}^{(r)}$ as $\mathbf{d} = \mathbf{B}^{(r)}\boldsymbol{\theta}$. By definition, the vector $\mathbf{z}$ belongs to $\mathcal{P}_{\mathbf{z}}(\mathbf{B}^{(r)})$, hence $-1 \leq \theta_i \leq 1$. Since $\mathbf{z} \notin \mathcal{L}(\mathbf{B}^{(r)})$, $\mathbf{z}$ does not belong to the vertices of $\mathcal{P}_{\mathbf{z}}(\mathbf{B}^{(r)})$. It follows that there is at least one coefficient θ_l in the basis expansion of $\mathbf{d}$, such that $0 < |\theta_l| < 1$.

The vector $\mathbf{d}$ replaces the i-th column of $\mathbf{B}^{(r)}$ to obtain $\mathbf{B}_i^{(r)}$. From Cramer's rule, $\det(\mathbf{B}_i^{(r)}) = \theta_i \det(\mathbf{B}^{(r)})$. Therefore, if we select l, such that $0 < |\theta_l| < 1$,

$$\begin{aligned} |\mathcal{L}(\mathbf{B}^{(r+1)})| &= |\det(\mathbf{B}^{(r+1)})| = |\det(\mathbf{B}_l^{(r)})| \\ &= |\theta_l||\det(\mathbf{B}^{(r)})| < |\det(\mathbf{B}^{(r)})| = |\mathcal{L}(\mathbf{B}^{(r)})| \end{aligned} \tag{34}$$

Note that there must be at least one such an index l, as indicated above. ■

of Theorem 4.2: Let $\mathcal{L}^*$ denote the solution of (13), i.e., the lattice with maximum volume that includes all observed vectors $\mathcal{S}$. We need to prove that $\mathcal{L}(\mathbf{B}^{(R)}) = \mathcal{L}^*$.

First, we prove that $|\mathcal{L}(\mathbf{B}^{(R)})|$ cannot decrease beyond $|\mathcal{L}^*|$, i.e., $|\mathcal{L}^*| \leq |\mathcal{L}(\mathbf{B}^{(R)})|$. To this end, let $\mathcal{L}(\mathbf{B}^{(R-1)})$ denote the lattice obtained at the iteration just before convergence. Hence, there is at least one observed vector $\tilde{\mathbf{x}} \in \mathcal{L}^*$ such that $\tilde{\mathbf{x}} \notin \mathcal{L}(\mathbf{B}^{(R-1)})$. Lemma 4.1 establishes that $|\mathcal{L}(\mathbf{B}^{(R)})| < |\mathcal{L}(\mathbf{B}^{(R-1)})|$.

Let $\mathbf{d}$ denote the difference vector as in line 7 of Algorithm 2. By construction, $\mathbf{d} \in \mathcal{L}^*$. Let $\mathbf{B}^*$ denote a basis for $\mathcal{L}^*$. Then, it is possible to write $\mathbf{d} = \mathbf{B}^*\boldsymbol{\theta}^*$, $\theta_i^* \in \mathbb{Z}$. $\mathcal{L}(\mathbf{B}^{(R-1)})$ is a sublattice of $\mathcal{L}^*$. Hence, $\mathbf{B}^{(R-1)} = \mathbf{B}^*\mathbf{A}$, where $\mathbf{A}$ is a matrix of integer elements such that $\det(\mathbf{A}) = m$, with $m \in \mathbb{Z} \setminus \{0\}$, and $|\mathcal{L}(\mathbf{B}^{(R-1)})|/|\mathcal{L}^*| = m$.

It is possible to express $\mathbf{d}$ in the basis expansion of $\mathbf{B}^{(R-1)}$. That is,

$$\begin{aligned} \boldsymbol{\theta} &= (\mathbf{B}^{(R-1)})^{-1}\mathbf{d} = (\mathbf{B}^*\mathbf{A})^{-1}\mathbf{B}^*\boldsymbol{\theta}^* \\ &= \mathbf{A}^{-1}\boldsymbol{\theta}^* = \frac{1}{\det(\mathbf{A})}\text{cofactor}(\mathbf{A})\boldsymbol{\theta}^*. \end{aligned} \tag{35}$$

Note that both the cofactor matrix $\text{cofactor}(\mathbf{A})$ and $\boldsymbol{\theta}^*$ have integer elements. Hence, the vector $\text{cofactor}(\mathbf{A})\boldsymbol{\theta}^*$ has integer elements. Any nonzero element of $\boldsymbol{\theta}$ is an integer multiple of $1/\det(\mathbf{A}) = 1/m$. Therefore, if $\theta_i \neq 0$, $|\theta_i| \geq 1/m$.

From the proof of Lemma 4.1, we know that

$$|\mathcal{L}(\mathbf{B}^{(R)})| = |\theta_l||\mathcal{L}(\mathbf{B}^{(R-1)})| \geq \frac{1}{m}|\mathcal{L}(\mathbf{B}^{(R-1)})| = |\mathcal{L}^*|, \tag{36}$$

where θ_l is one of the nonzero elements of $\boldsymbol{\theta}$.

To prove that $|\mathcal{L}(\mathbf{B}^{(R)})| = |\mathcal{L}^*|$, it remains to be shown that cannot be $|\mathcal{L}(\mathbf{B}^{(R)})| > |\mathcal{L}^*|$. Indeed, if this

were the case, $\mathcal{L}(\mathbf{B}^{(R)})$ would be the optimal solution of (13), since it includes all observed points $\mathcal{S}$ and has volume larger than $|\mathcal{L}^*|$.

In order to prove that convergence requires a finite number of steps, we construct the sequence of integer numbers $s_r = |\mathcal{L}(\mathbf{B}^{(r)})|, \quad r = 0, 1, \ldots, R$.

Let R denote the smallest integer such that $|\mathcal{L}(\mathbf{B}^{(R)})| = |\mathcal{L}(\mathbf{B}^{(R+1)})|$. That is, R is the number of steps needed to achieve convergence. Note that $R < \infty$. Indeed, $\{s_r\}$ is a sequence of integer values. The sequence is monotonically decreasing due to Lemma 4.1, until convergence is achieved and $\mathcal{S} \subset \mathcal{L}(\mathbf{B}^{(R)})$. In addition, it is bounded from below by $|\mathcal{L}_x|$. Therefore, convergence is achieved in up to $|\mathcal{L}(\mathbf{B}^{(0)})|/|\mathcal{L}_x|$ number of steps.

■

of Theorem 4.3: Since $\mathcal{S} \not\subset \mathcal{L}(\mathbf{B}^{(r)})$ the recursion is not terminated. Consider the vector $\mathbf{d} = \mathbf{z} - \mathbf{v}$, which can be expressed in the basis $\mathbf{B}^{(r)}$ as $\mathbf{d} = \mathbf{B}^{(r)}\boldsymbol{\theta}$. Dropping the superscript $^{(r)}$, it is possible to write

$$\boldsymbol{\theta} = \mathbf{B}^{-1}\mathbf{d} = \mathbf{B}^{-1}(\mathbf{z} - \mathbf{v}) \tag{37}$$

$$= \mathbf{B}^{-1}\mathbf{z} - \mathbf{B}^{-1}(\mathbf{B} \cdot \text{round}(\mathbf{B}^{-1}\mathbf{z})) \tag{38}$$

$$= \mathbf{B}^{-1}\mathbf{z} - \text{round}(\mathbf{B}^{-1}\mathbf{z}) = \mathbf{a} - \text{round}(\mathbf{a}), \tag{39}$$

where we set $\mathbf{a} = \mathbf{B}^{-1}\mathbf{z}$. Due to the properties of rounding, $-1/2 \leq \theta_i < 1/2$. Thus, replacing any of the columns of $\mathbf{B}^{(r)}$ such that $\theta_l \neq 0$, we obtain, using Cramer's rule,

$$\frac{|\mathcal{L}(\mathbf{B}^{(r+1)})|}{|\mathcal{L}(\mathbf{B}^{(r)})|} = |\theta_l| < \frac{1}{2} \tag{40}$$

■

of Lemma 4.4: It is possible to derive both an upper and a lower bound on the number of sub-lattices that are independent from the prime factorisation of m starting from (17). Since for all cases of interest $N > 1$, we have:

$$\frac{p_i^{N+j-1} - 1}{p_i^j - 1} > \frac{p_i^{N+j-1}}{p_i^j}. \tag{41}$$

Substituting in (17), we have a function $\underline{f_N}(m)$ that is guaranteed to yield values below $f_N(m)$:

$$\underline{f_N}(m) = \prod_{i=1}^{q} \prod_{j=1}^{t_i} \frac{p_i^{N+j-1}}{p_i^j} = \prod_{i=1}^{q} p_i^{t_i(N-1)}. \tag{42}$$

This is equivalent to the $(N-1)^{\text{th}}$ power of the product of the prime factors of m. That is, the lower bound of $f_N(m)$ can be expressed as $\underline{f_N}(m) = m^{N-1}$.
In terms of the upper bound of $f_N(m)$, we proceed similarly by starting with the observation that:

$$(p_i^{N+j-1} - 1)/(p_i^j - 1) < (p_i^{N+j})(p_i^j). \tag{43}$$

By substituting back into (17), we can observe that:

$$\prod_{i=1}^{q} \prod_{j=1}^{t_i} \frac{p_i^{N+j}}{p_i^j} = m\underline{f_N}(m). \tag{44}$$

Hence, it is easy to see that the upper bound on $f_N(m)$ can be expressed as $\overline{f_N}(m) = m^N$.
Therefore, since $\underline{f_N}(m) < f_N(m) < \overline{f_N}(m)$, we have $m^{N-1} < f_N(m) < m^N$.

■

REFERENCES

[1] M. Tagliasacchi, M. Visentini Scarzanella, P. L. Dragotti, and S. Tubaro, "Transform coder identification," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2013.

[2] M. Tagliasacchi, M. Visentini Scarzanella, P. L. Dragotti, and S. Tubaro, "Transform coder identification with double quantized data," in *IEEE International Conference on Image Processing (ICIP)*, September 2013.

[3] G.K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, feb 1992.

[4] G.J. Sullivan and T. Wiegand, "Video compression - from concepts to the H.264/AVC standard," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 18–31, jan 2005.

[5] M. Winken, P. Helle, D. Marpe, H. Schwarz, and T. Wiegand, "Transform coding in the HEVC test model," in *IEEE International Conference on Image Processing (ICIP)*, 2011, pp. 3693–3696.

[6] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, June 2006.

[7] M. Chen, J. J. Fridrich, M. Goljan, and J. Lukás, "Determining image origin and integrity using sensor noise," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 74–90, 2008.

[8] T. Bianchi and A. Piva, "Image forgery localization via block-grained analysis of JPEG artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1003–1017, June 2012.

[9] G. Valenzise, S. Magni, M. Tagliasacchi, and S. Tubaro, "No-reference pixel video quality monitoring of channel-induced distortion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 4, pp. 605–618, 2012.

[10] M. Naccari, M. Tagliasacchi, and S. Tubaro, "No-reference video quality monitoring for H.264/AVC coded video," *IEEE Transactions on Multimedia*, vol. 11, no. 5, pp. 932–946, 2009.

[11] M.R. Banham and A.K. Katsaggelos, "Digital image restoration," *IEEE Signal Processing Magazine*, vol. 14, no. 2, pp. 24–41, 1997.

[12] Z. Fan and R. L. de Queiroz, "Identification of bitmap compression history: JPEG detection and quantizer estimation," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 230–235, 2003.

[13] J. Lukás and J. Fridrich, "Estimation of primary quantization matrix in double compressed JPEG images," in *Proc. of DFRWS*, 2003.

[14] Y. Chen, K. S. Challapali, and M. Balakrishnan, "Extracting coding parameters from pre-coded MPEG-2 video," in *IEEE International Conference on Image Processing (ICIP)*, 1998, pp. 360–364.

[15] H. Li and S. Forchhammer, "MPEG2 video parameter and no reference PSNR estimation," in *Picture Coding Symposium (PCS)*, 2009, pp. 1–4.

[16] W. Luo, M. Wu, and J. Huang, "MPEG recompression detection based on block artifacts," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 2008, vol. 6819 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*.

[17] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting double quantization," in *Proceedings of the 11th ACM workshop on Multimedia and security*, New York, NY, USA, 2009, MM&Sec, pp. 39–48, ACM.

[18] M. Tagliasacchi and S. Tubaro, "Blind estimation of the QP parameter in H.264/AVC decoded video," in *International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), 2010*, 2010, pp. 1–4.

[19] P. Bestagini, A. Allam, S. Milani, M. Tagliasacchi, and S. Tubaro, "Video codec identification," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, March 2012, pp. 2257–2260.

[20] R. M. Gray and D.L. Neuhoff, "Quantization," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325–2383, oct 1998.

[21] R. Zamir, S. Shamai, and U. Erez, "Nested linear/lattice coes for structured multiterminal binning," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1250–1276, june 2002.

[22] D. Wübben, D. Seethaler, J. Jaldén, and G. Matz, "Lattice reduction: A survey with applications in wireless communications," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 70–91, May 2011.

[23] R. Neelamani, R. de Queiroz, Z. Fan, S. Dash, and R.G. Baraniuk, "JPEG compression history estimation for color images," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1365–1378, June 2006.

[24] P. Comesaña, F. Pérez-González, and N. Liste, "Quantization lattice estimation for multimedia forensics," in *IEEE International Conference on Image Processing (ICIP)*, September 2013.

[25] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, pp. 515–534, 1982, 10.1007/BF01457454.

[26] M. Pohst, "A modification of the {LLL} reduction algorithm," *Journal of Symbolic Computation*, vol. 4, no. 1, pp. 123 – 127, 1987.

[27] Johannes Buchmann and Michael Pohst, "Computing a lattice basis from a system of generating vectors," in *European Conference on Computer Algebra (EUROCAL)*. June 1987, vol. 378 of *Lecture Notes in Computer Science*, pp. 54–63, Springer Berlin Heidelberg.

[28] B. Hemkemeier and F. Vallentin, "On the decomposition of lattices," Tech. Rep. 52, Electronic Colloquium on Computational Complexity, August 1998.

[29] Henri Cohen, *A Course in Computational Algebraic Number Theory*, Graduate Texts in Mathematics. Springer, 1993.

[30] F. Vallentin and B. Hemkemeier, "Incremental algorithms for lattice problems," Tech. Rep. 52-1, Electronic Colloquium on Computational Complexity, September 2006.

[31] M. Visentini Scarzanella, M. Tagliasacchi, and P. L. Dragotti, "Quantisation invariants for transform parameter estimation in coding chains," in *Data Compression Conference*, March 2013.

[32] J. H. Conway, N. J. A. Sloane, and E. Bannai, *Sphere-packings, lattices, and groups*, Springer-Verlag New York, Inc., New York, NY, USA, 1987.

[33] H. Cohen, *A Course in Computational Algebraic Number Theory*, vol. 138 of *Graduate Texts in Mathematics*, Springer, 1993.

[34] B. Gruber, "Alternative formulae for the number of sublattices," *Acta Crystallographica Section A*, vol. 53, pp. 807–808, 1997.

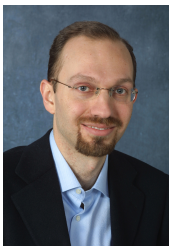
[35] J. Stillwell, *Elements of number theory / John Stillwell*, Springer, New York :, 2003.

[36] S.D. Casey and B.M. Sadler, "Modifications of the euclidean algorithm for isolating periodicities from a sparse set of noisy measurements," *IEEE Transactions on Signal Processing*, vol. 44, no. 9, pp. 2260–2272, sep 1996.

[37] S. Milani, M. Tagliasacchi, and S. Tubaro, "Identification of the motion estimation strategy using eigenalgorithms," in *IEEE International Conference on Image Processing (ICIP)*, September 2013, pp. 4477–4481.

Marco Tagliasacchi is currently Associate Professor at the "Dipartimento di Elettronica e Informazione - Politecnico di Milano", Italy. He received the "Laurea" degree (2002, cum Laude) in Computer Engineering and the Ph.D. in Electrical Engineering and Computer Science (2006), both from Politecnico di Milano. He was visiting academic at the Imperial College London (2012) and visiting scholar at the University of California, Berkeley (2004). His research interests include multimedia forensics, multimedia communications (visual sensor networks, coding, quality assessment) and information retrieval. Dr. Tagliasacchi co-authored more than 120 papers in international journals and conferences, including award winning papers at MMSP 2013, MMSP2012, ICIP 2011, MMSP 2009 and QoMex 2009. He has been actively involved in several EU-funded research projects. He is currently co-coordinating two ICT-FP7 FET-Open projects (GreenEyes - www.greeneyesproject.eu, REWIND - www.rewindproject.eu). Dr. Tagliasacchi is an elected member of the IEEE Information Forensics and Security Technical committee for the term 2014-2016, and served as member of the IEEE MMSP Technical Committee for the term 2009-2012. He is currently Associate Editor for the IEEE Transactions on Circuits and Systems for Video Technologies (2011 best AE award) and APSIPA Transactions on Signal and Information Processing. Dr. Tagliasacchi was General co-Chair of IEEE Workshop on Multimedia Signal Processing (MMSP 2013, Pula, Italy) and he will be Technical Program Coordinator of IEEE International Conference on Multimedia&Expo (ICME 2015, Turin, Italy).

Marco Visentini Scarzanella Marco Visentini-Scarzanella (M'12) received the MEng degree (Hons.) in Information Systems Engineering and the Ph.D. degree in Medical Image Computing from Imperial College London, UK in 2007 and 2012 respectively. He subsequently worked as a Research Associate in the Communications and Signal Processing Group at Imperial College London, UK from 2011 to 2014, and was a Visiting Researcher at the Politecnico di Milano, Italy in 2013 for the REWIND project. He was awarded a Japanese Society for the Promotion of Science Fellowship in 2014, and worked as a Research Fellow in Kagoshima University, Japan until 2015. He is the recipient of the Toshiba Research Fellowship and is currently a Toshiba Fellow at the Toshiba R&D Center in Kawasaki, Japan. His research interests include 3D reconstruction, Shape-from-X and image-based endoscopic navigation.



Pier Luigi Dragotti Pier Luigi Dragotti (M'02–SM'11) is Professor of Signal Processing with the Electrical and Electronic Engineering Department at Imperial College London. He received the Laurea degree (summa cum laude) in electrical and electronic engineering from the University of Naples Federico II, Naples, Italy, in 1997; the Master's degree in communications systems from the Swiss Federal Institute of Technology of Lausanne (EPFL), Lausanne, Switzerland, in 1998, and the Ph.D. degree from EPFL in April 2002. He has held several visiting positions at different universities and research centers. He was a Visiting Student with Stanford University, Stanford, CA, USA, in 1996, a Researcher with the Mathematics of Communications Department, Bell Labs, Lucent Technologies, Murray Hill, NJ, USA, in 2000 and a Visiting Scientist with the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2011. He was Technical Co-Chair for the European Signal Processing Conference in 2012, an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING from 2006 to 2009 and an Elected Member of the IEEE Image, Video and Multidimensional Signal Processing Technical Committee (2008–2013). He is a recipient of the ERC Starting Investigator Award for the project RecoSamp. His work includes sampling theory, wavelet theory and its applications, image and video compression, image-based rendering, and image super-resolution.

Stefano Tubaro was born in Novara in 1957. He completed his studies in Electronic Engineering at the Politecnico di Milano, Italy, in 1982. He then joined the Dipartimento di Elettronica e Informazione of the Politecnico di Milano, first as a researcher of the National Research Council, and then (in November 1991) as an Associate Professor. Since December 2004 he has been appointed as Full Professor of Telecommunication at the Politecnico di Milano. His current research interests are on advanced algorithms for video and sound processing. Stefano Tubaro authored over 150 scientific publications on international journals and congresses. In the past few years he has focused his interest on the development of innovative techniques for image and video tampering detection and, in general, for the blind recovery of the "processing history" of multimedia objects. Stefano Tubaro coordinates the research activities of the Image and Sound Processing Group (ISPG) at the Dipartimento di Elettronica, Informazione e Bioingegneria of the Politecnico di Milano. He had the role of Project Coordinator of the European Project ORIGAMI: A new paradigm for high-quality mixing of real and virtual. From May 2011 he is the Coordinator of the research project ICT-FET-OPEN REWIND: REVerse engineering of audio-VIsual coNtent Data. This project is aimed at synergistically combining principles of signal processing, machine learning and information theory to answer relevant questions on the past history of such objects. Stefano Tubaro is a member the IEEE Multimedia Signal Processing Technical Committee and of the IEEE SPS Image Video and Multidimensional Signal Technical Commitee. He was in the organization committee of a number of international conferences: IEEE-MMSP-2004/2013, IEEE-ICIP-2005, IEEE-AVSS-2005/2009, IEEE-ICDSC-2009, IEEE-MMSP-2013 to mention a few. From May 2012 he is an Associate Editor of the IEEE Transactions on Image Processing.