Coding Visual Features Extracted From Video Sequences

Luca Baroffio, Matteo Cesana, Alessandro Redondi, Marco Tagliasacchi, and Stefano Tubaro

I. Introduction

VISUAL features provide a succinct, yet efficient, representation of the underlying visual content, which is robust and invariant to many global and local transformations. They are effectively employed in many tasks, ranging from image/video retrieval, object recognition, object tracking, image registration, structure-from-motion, etc. Visual feature extraction algorithms consist of two main components: the detector, which identifies salient keypoints within an image; and the descriptor, which provides a concise representation of the image patch surrounding each keypoint. Although several different descriptors have been proposed in recent years, they all share a similar processing pipeline. That is, a feature vector is computed following three main processing steps, namely pre-smoothing, transformation and spatial pooling [3]. For example, the state-of-the-art SIFT descriptor [4] is obtained performing Gaussian smoothing, followed by the computation of local gradients, which are then pooled together to build a histogram.

Several visual analysis applications, such as object recognition, traffic/habitat/environmental monitoring, surveillance, etc., might benefit from the technological evolution of networks towards the "Internet-of-Things", where low-power battery-operated nodes are equipped with sensing capabilities and are able to carry out computational tasks and collaborate over a network. In particular, Visual Wireless Sensor Networks (VWSNs) are a promising technology for distributed visual analysis tasks [5], [6]. The traditional approach to such scenarios, which will be denoted hereinafter as *"Compress-Then-Analyze"* (CTA), is based on the following steps: the signal of interest (i.e., a still image or a video sequence) is acquired by a sensor node, then it is compressed (e.g., resorting to JPEG or H.264/AVC coding standards) in order to be efficiently transmitted over a network. Finally, visual analysis is performed at a sink node [7]–[10]. Since the signal is acquired and subsequently compressed, visual analysis is based on a lossy representation of the visual content, possibly resulting in impaired performance [11], [12]. Although such paradigm has been efficiently employed in a number of applications (e.g., video surveillance, smart cameras, etc.), several analysis tasks might require streaming high quality visual content. This might be infeasible even with state-of-the-art VWSN technology [13] due to the limited network bandwidth. A possible solution consists in driving the encoding process so as to optimize visual analysis, rather than perceptual quality, at the receiver side. For example, JPEG coding can be tuned so as to preserve SIFT features in decoded images [14].

At the same time, an alternative *"Analyze-Then-Compress"* (ATC) approach, in a sense orthogonal to CTA, is gaining popularity in the research community. The ATC paradigm relies on the fact that some tasks can be performed resorting to a succinct representation based on local features, disregarding the actual pixel-level content. According to ATC, local features are extracted from a signal directly by the sensing node. Then, they are compressed to be efficiently dispatched over the

Manuscript received August 4, 2013; revised December 23, 2013 and March 3, 2014; accepted March 10, 2014. Date of publication March 19, 2014; date of current version April 11, 2014. This work was supported by the Future and Emerging Technologies Programme within the Seventh Framework Programme for Research of the European Commission under FET-Open Grant 296676 through the GreenEyes Project. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Bulent Sankur.

The authors are with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan 20133, Italy (e-mail: luca.baroffio@polimi.it; cesana@elet.polimi.it; redondi@elet.polimi.it; tagliasa@elet.polimi.it; stefano.tubaro@polimi.it).

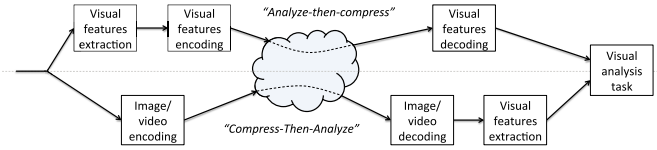

Fig. 1. Pipelines for the *"Analyze-Then-Compress"* and *"Compress-Then-Analyze"* paradigms.

network. As illustrated in Fig. 1, *"Analyze-Then-Compress"* and *"Compress-Then-Analyze"* represent concurrent paradigms that can be employed to address the problem of analyzing content capture from distributed cameras. Compression of visual features is key to the successful deployoment of the ATC scheme, since VWSNs typically pose strict constraints regarding the available bandwidth. Several works tackled this problem for the case of features extracted from still images, proposing methods to efficiently encode state-of-the-art visual features [15]–[18] or to modify the design of local feature extraction algorithms, so that the representation of the underlying visual content is more suitable for compression [19]–[22]. In this context, an ad-hoc MPEG group on Compact Descriptors for Visual Search (CDVS) is currently working towards the definition of a standard tailored to this scenario [23].

Several visual analysis tasks (e.g., motion estimation and tracking, structure-from-motion, 3D reconstruction, event detection, etc.) require to process sets of visual features extracted from *video sequences*. Such tasks require the usage of a representation in terms of a set of local features for each frame to be analyzed, therefore resulting in a necessarily large volume of data. Hence, the design of efficient coding mechanisms for video sequences of visual features plays a fundamental role. Although a large body of research deals with the extraction and compression of visual features obtained from still images, the execution of similar operations starting from video sequences has not received a similar attention so far. Therefore, the main contribution of this work is the proposal of a novel coding architecture tailored to compress sets of visual features, including both the location of the keypoints and the corresponding descriptors, extracted from video sequences. Inspired by traditional video coding architectures, we exploit both spatial and temporal redundancy by means of intra-frame and inter-frame coding schemes. To this end, we propose a coding mode decision algorithm based on rate-distortion optimization that adaptively selects either intra- or inter-frame coding for each feature. This goes beyond previously proposed visual feature coding schemes, which exploit redundancy either within the same descriptor [15], [24], [25] or descriptors extracted from the same image [24], [26], [27], [28].

Our experiments are based on SIFT [4], which is known to deliver state-of-the-art performance in several visual analysis tasks. However, the proposed coding architecture is not bound to a specific kind of local features, since it can be immediately applied to any kind of real-valued descriptor. To demonstrate such flexibility, we also provide results for SURF [29], a popular descriptor partially inspired by SIFT, which provides slightly worse performance in visual analysis, but at a lower computational complexity.

The proposed coding architecture achieves high coding efficiency, which is necessary to enable the ATC paradigm in those scenarios that require the analysis of video content. As a further contribution of our work, we thoroughly compared the ATC approach, based on the proposed visual feature coding scheme, with the CTA approach, based on traditional video coding (i.e., H.264/AVC). As illustrated in Fig. 1, a direct comparison between the two approaches cannot be performed based on traditional rate-distortion analysis, since different and incomparable signals are reconstructed at the receiver side. Conversely, the comparison is based on evaluation metrics that capture the quality of the analysis task. To this end, we adopted metrics that are routinely used to quantify the performance of visual features in the context of, e.g., content-based retrieval, object recognition and tracking. Experimental results demonstrate that, thanks to the significant coding gain achieved by the proposed coding scheme, ATC outperforms CTA with respect to all evaluation metrics. Of course, CTA remains a valuable option when one needs to have access to the full pixel-domain representation of the video sequence at the decoder, e.g., to store a copy of the sequences for future use (e.g., in surveillance), or when different kinds of analysis are necessary, possibly requiring different forms of visual features. In some cases, though, the bandwidth constraints imposed by the network are so stringent, that only ATC can operate under these conditions [30].

In our work we do not focus on the problem of matching sets of visual features extracted from different images/video sequences. Indeed, matching is known to represent an expensive operation in terms of computational resources. For this reason, there are several works in the field of content-based image and video retrieval that propose fast matching schemes that scale with the size of the database. These schemes are based on the idea of building a very compact representation mapping a set of descriptors extracted from an image or a video frame into a fixed-dimensional feature vector, so that multidimensional indices can be employed. The most popular approach is known as *Bag-of-Visual-Words* (BoVW) [31], which offers reduced computational complexity for the matching process while trading off some precision in the execution of the task. Other global, fixed-dimensional, representations based on local features were recently proposed [32]. However, having access to the decoded set of local features offers unique advantages. First, encoding local features does not preclude the opportunity of using a BoVW approach at the decoder. Indeed, the received features can be mapped to the corresponding visual words and used to perform fast matching based on a fixed-dimensional BoVW representation. At the same time, the availability of local features at the decoder enables the possibility of re-ranking the top-matching items obtained by means of global descriptors (e.g., by enforcing spatial verification [33]), thus leading to improved precision. This is in contrast to those approaches that compute and compress global descriptors based on visual words at the sensor node [34], for which re-ranking cannot be performed. Second, the underlying spatial configuration of local features is completely retained,

i.e., the coordinates of the detected keypoints are encoded together with the corresponding descriptors. This is not the case when using a BoVW approach, in which the quantization process used to map a descriptor to the nearest visual word discards the coordinates of its keypoint. In some applications, e.g., in object tracking and structure from motion, it is necessary to reconstruct the spatial configuration of keypoints at the decoder, since the spatial evolution of local features needs to be tracked by matching them along the temporal dimension. This is demonstrated in the homography estimation scenario evaluated in the experiments in Section IV, which explicitly requires the availability of local features, rather than global features.

The question addressed by the ATC paradigm bears some similarity with problems discussed previously in the literature, e.g., in the context of semantic coding [35] (representing multimedia information under a compressed form that permits efficient classification) and model-based coding [36] (computing face models that can be efficiently encoded and transmitted). More formally, the problem is cast under the theoretical framework of the information bottleneck method in [37], in which the source to be encoded is abstracted by means of a random variable, and an auxiliary representation that achieves the best trade-off between accuracy and compression is sought. Differently, in this paper we focus on analysis tasks that can be performed by means of a representation based on local features.

The rest of the paper is organized as follows. Section II introduces the problem of coding visual features extracted from video sequences. Section III illustrates the details about the coding schemes for both intra- and inter-frame approaches and discusses rate-distortion optimization. Section IV is devoted to a comprehensive experimental study, introducing the processing pipelines implementing different visual analysis tasks, defining the evaluation metrics and comparing the results obtained by both the ATC and CTA paradigms. Finally, section V draws the conclusions and discusses future work.

II. Coding Video Sequences of Local Features: Problem Statement

Let $\mathcal{I}_n$ denote the n-th frame of a video sequence of size $N_x \times N_y$, which is processed to extract a set of local features $\mathcal{D}_n$. First, a detector (possibly scale-invariant) is applied, to identify stable keypoints in the scale-space domain. The use of a scale-invariant detector [38] (e.g., Laplacian-of-Gaussian, Difference-of-Gaussians, Harris-Laplace, etc.) allows the extraction of keypoints associated with image patches of different physical sizes. The number of detected keypoints $M_n = |\mathcal{D}_n|$ depends on both the image content, the detector type and the settings of the detector parameters (e.g., number of scales, detection threshold, etc.). Then, the patches around the detected keypoints are processed further to compute the corresponding descriptors. Several descriptors are rotation-invariant, i.e., they take into account the orientation of the detected keypoint and compensate for that while building the descriptor vector. To this end, the main direction of the keypoint is estimated, typically based on the analysis of local gradients. An oriented patch surrounding the keypoint is extracted and subsequently processed to compute a concise representation. The descriptor is built having the keypoint orientation as a reference coordinate system, resulting in an orientation-invariant descriptor.

Although the proposed coding architecture can be used to encode different kinds of descriptors, in our experiments we focus on two popular choices, namely SIFT [4] and SURF [29]. For each detected keypoint, both descriptors consider a patch centered at the keypoint, rotated by θ, whose size is proportional to σ. In the case of SIFT, the local patch is divided into 16 sub-regions, and an orientation histogram with 8 bins is created based on the gradients computed from the (smoothed) samples for each region. The descriptor is then obtained by concatenating these 16 histograms, leading to a descriptor with 128 elements. The descriptor is finally normalized to unit length to achieve robustness against illumination changes. In the case of SURF, the local patch is split in a grid of 4×4 sub-regions. For each sub-region a 4-dimensional feature vector is defined as:

$$[\sum g_x, \sum g_y, \sum |g_x|, \sum |g_y|] \,, \tag{1}$$

where g_x and g_y represent the result of convolving the pixel values of the local patch with two Haar wavelets along orthogonal directions, and the sums are computed over a predefined set of sample points in the respective sub-region. The final descriptor is obtained by concatenating the feature vectors of all the sub-regions, obtaining a vector with 64 elements.

Each element $d_{n,i} \in \mathcal{D}_n$ is a visual feature, which consists of two components: i) a 4-dimensional vector $\mathbf{p}_{n,i} = [x, y, \sigma, \theta]^T$, indicating the position (x, y), the scale σ of the detected keypoint, and the orientation angle θ of the image patch; ii) a P-dimensional vector $\mathbf{d}_{n,i}$, which represents the descriptor associated to the feature $d_{n,i}$.

As mentioned in Section I, we introduce a coding architecture which aims at efficiently coding the sequence $\{\mathcal{D}_n\}_{n=1}^{N}$ of sets of descriptors, where N denotes the number of frames. In particular, a lossy coding technique is proposed, which enables to reconstruct, at the decoder, an approximation $\tilde{\mathcal{D}}_n$ of the local features extracted from $\mathcal{I}_n$. Each reconstructed descriptor can be written as $\tilde{d}_{i,n} = \{\tilde{\mathbf{p}}_{n,i}, \tilde{\mathbf{d}}_{n,i}\}$. The number of bits necessary to encode the visual features of frame $\mathcal{I}_n$ is equal to

$$R_n = \sum_{i=1}^{M_n} (R^c_{n,i} + R^d_{n,i}), \tag{2}$$

where $R^c_{n,i}$ is the rate used to represent the location component $\mathbf{p}_{n,i}$, $R^d_{n,i}$ is the number of bits used to encode the descriptor component $d_{n,i}$ and $M_n = |\mathcal{D}_n|$ is the number of features extracted from frame $\mathcal{I}_n$. Distortion is measured in terms of the mean square error between the original and the decoded descriptor, averaged over the descriptors extracted from $\mathcal{I}_n$:

$$D_n = \frac{1}{M_n P} \sum_{i=1}^{M_n} \|\tilde{\mathbf{d}}_{i,n} - \mathbf{d}_{i,n}\|_2^2, \tag{3}$$

where $\|\cdot\|_2$ denotes the l-2 norm. As for the component $\tilde{\mathbf{p}}_{n,i}$, we decided to encode the coordinates of the keypoint and its scale, i.e., $\tilde{\mathbf{p}}_{n,i} = [\tilde{x}, \tilde{y}, \tilde{\sigma}]^T$. At the decoder, the information

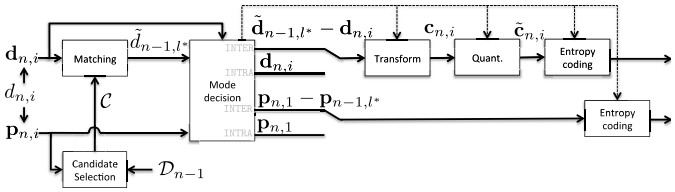

Fig. 2. Block diagram of the proposed coding architecture for visual features extracted from video sequences.

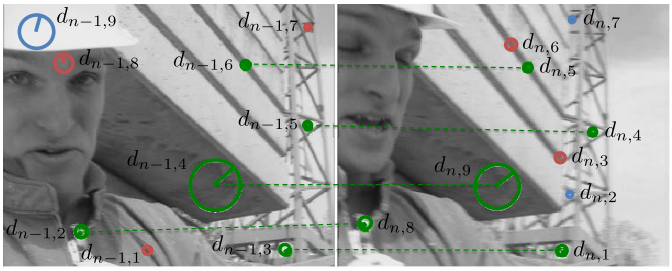

Fig. 3. **[Best viewed in color]** The inter-frame coding scheme matches sets of features extracted from consecutive frames. The key idea is that features corresponding to the same physical entity (green circles) would also obtain similar description vectors. On the other hand, intra-frame scheme addresses either local features that are occluded or not visible in one of the two frames (blue circles) or non-repeatible detections (red circles).

regarding the location (x, y) is necessary in several visual analysis tasks: i) when the matching score between image pairs relies on a geometric consistency check based on spatial verification [33], e.g., using RANSAC [39]; ii) when correspondences among local features need to be found over time, e.g., in the case of object tracking, or across different views, e.g., in the case of stereo matching. Although most detectors produce as output coordinates represented in floating point precision thanks to a sub-pixel interpolation step, we decided to round the coordinates to quarter-pixel precision, which is typically sufficient for most analysis tasks. The scale parameter is also quantized with a step size equal to 0.25. In Section IV we will show experimentally that this choice does not impair the repeatability of the detector. The encoded vector $\tilde{\mathbf{p}}_{n,i}$ does not contain the orientation of the keypoint θ, as it is typically not employed to match descriptors. Note that this piece of information might be necessary when using alternative spatial verification schemes, e.g., when weak geometry checking [40] is enforced.

The main contribution of this paper is the investigation of an intra- and inter-frame coding scheme, which aims at exploiting the spatio-temporal redundancy in sets of local features $\{\mathcal{D}_n\}_{n=1}^{N}$ extracted from consecutive video frames. In Section III we provide the details of the proposed coding architecture, which leverages some of the coding tools that are successfully employed in state-of-the-art video coding to achieve high coding efficiency. Note that the same coding architecture can be adapted in a straightforward manner to encode sets of descriptors acquired from multiple cameras observing the same scene.

III. Coding Video Sequences of Local Features: Algorithms

Similarly to video coding, the sequence of descriptors is organized according to a GOP (Group of Pictures) structure. In this work, we consider two kinds of frames, namely I-frames and P-frames and a simple IPPP GOP structure, with an I-frame every G frames. Descriptors in I-frames are encoded exploiting an intra-frame coding mode, which is described in Section III-A. Conversely, descriptors in P-frames are encoded exploiting an inter-frame coding mode, illustrated in Section III-B. The principles illustrated in this section can be generalized to more complex GOP structures, including, e.g., B-frames and hierarchical B-frames, which are commonly used in state-of-the-art video coding architectures.

A. Intra-Frame Coding

The intra-frame coding scheme is based on a frame-by-frame processing, in which the local features extracted from frame $\mathcal{I}_n$ are encoded independently from those extracted from other frames. Such approach is in a sense equivalent to coding local features extracted from still images, which has being widely investigated in the past literature as discussed in Section I.

Considering a baseline architecture, each dexel (descriptor element) of $\mathbf{d}_{n,i}$ is encoded by applying scalar quantization with step size Δ_j and a central deadzone. That is,

$$\tilde{d}_{n,i,j} = \mathrm{sgn}(d_{n,i,j}) \left\lfloor \frac{|d_{n,i,j}|}{\Delta_j} \right\rfloor . \quad (4)$$

We fix the same quantization step size for all dexels, i.e., $\Delta_j = \Delta$, $j = 1, \ldots, P$. Then, the quantization symbols are compressed by means of entropy coding using $R_{n,i}^{d,\mathtt{INTRA}}$ bits, whereas the location component of the keypoint is encoded using $R_{n,i}^{c,\mathtt{INTRA}}$ bits, as discussed in Section III-C.

In previous works it was observed that dexels of the same descriptors are somehow correlated [24], [27], thus leading to intra-frame coding schemes that exploit the inherent intra-descriptor redundancy. Hence, we consider an intra-descriptor coding scheme that applies the Karhunen-Loève Transform matrix $\mathbf{T} \in \mathbb{R}^{P \times P}$ to each descriptor $\mathbf{d}_{n,i}$. The matrix $\mathbf{T}$ is determined based on the descriptors collected from a large set of training images, as detailed in Section IV. Although the Karhunen-Loève Transform can also be used to reduce the number of elements of the descriptor [22], in this work it is adopted to address the correlation among descriptor elements. Then, let $\mathbf{c}_{n,i} = \mathbf{T}\mathbf{d}_{n,i} \in \mathbb{R}^P$ denote the descriptor in the transform domain, and $\tilde{\mathbf{c}}_{n,i}$ the result of scalar quantization. Similarly to the case above, the output symbols of the quantizer are entropy coded.

In the literature, it is also observed that descriptors extracted from the same image are correlated (e.g., because of the presence of recurring patterns), suggesting the adoption of inter-descriptor coding schemes [26], i.e., approaches that exploit redundancy among local features belonging to the same set $\mathcal{D}_n$. However, in our previous work [27], we show that inter-descriptor coding does not bring significant coding gains, once intra-descriptor redundancy is addressed. Hence, we do

not consider inter-descriptor coding within the same frame in this work.

B. Inter-Frame Coding

In the case of inter-frame coding, each set of features $\mathcal{D}_n$ is encoded resorting to a reference set of features. In this work, we consider as reference the set of local features extracted from the previous frame, i.e., $\mathcal{D}_{n-1}$, thus mimicking P-frames in traditional video coding. Such approach can be straightforwardly extended by adapting the ideas behind state-of-the-art video coding tools, e.g., introducing the possibility to use multiple sets of features as a reference, or bi-directional predictive coding schemes similar to the ones of B-frames.

The key intuition behind inter-frame coding is that the keypoints detected in neighbouring frames correspond to the same physical entities, provided that the underlying visual content does not change abruptly. As such, the image patches around two matching keypoints are similar, leading to correlated descriptors. This is illustrated in Fig. 3, for the case of feature pairs $\langle d_{n-1,3}, d_{n,1}\rangle$, $\langle d_{n-1,5}, d_{n,4}\rangle$, $\langle d_{n-1,2}, d_{n,8}\rangle$ and $\langle d_{n-1,4}, d_{n,9}\rangle$. Of course, not all the keypoints in $\mathcal{D}_n$ have a matching keypoint in $\mathcal{D}_{n-1}$. This is due different reasons: i) objects are covered/uncovered, so that the image region corresponding to a keypoint in frame $\mathcal{I}_n$ does not appear in image $\mathcal{I}_{n-1}$ (for example, $d_{n,2}$, $d_{n,7}$ and $d_{n-1,9}$ in Fig. 3); ii) the non-ideal behaviour of the detector, which might not necessarily detect the same keypoint across different frames (for example, $d_{n,3}$, $d_{n,6}$, $d_{n-1,1}$, $d_{n-1,7}$ and $d_{n-1,8}$ in Fig. 3).

Therefore, we consider a coding architecture that is able to adaptively switch between inter-frame and intra-frame coding. Specifically, considering each descriptor $d_{n,i} \in \mathcal{D}_n$, $i = 1, \ldots, M_n$, encoding proceeds as follows (see Fig. 2):

- *Descriptor matching*: Compute the best matching descriptor in the reference frame, i.e.,

$$\tilde{d}_{n-1,l^*} = \arg \min_{\tilde{d}_{n-1,l} \in \mathcal{C}} J^{\texttt{INTER}}(d_{n,i}, \tilde{d}_{n-1,l}), \quad (5)$$

  where

$$J^{\texttt{INTER}}(d_{n,i}, \tilde{d}_{n-1,l}) = \frac{1}{\sqrt{P}} \|\mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n-1,l}\|_2 + \lambda R^{c,\texttt{INTER}}_{n,i}(l), \quad (6)$$

  and $\mathcal{C} \subseteq \mathcal{D}_{n-1}$ represents the set of all possible matching candidates. In this work, $\mathcal{C}$ is populated with the local features whose coordinates are in a neighborhood of $d_{n,i}$, within a search window of $(\pm\Delta x, \pm\Delta y)$ and whose scale is in a range of $\pm\Delta\sigma$. Such initial filtering stage has two main motivations: i) enforcing stable matches between keypoints that possibly represent the same physical entity; ii) reducing the computational complexity by evaluating only a set of candidates, thus avoiding a complete scan of the set $\mathcal{D}_{n-1}$ for each local feature to be matched. In the cost function in (6), the first term represents the Root Mean Square Error (RMSE) between two matching candidate descriptors, whereas the second is a penalty term $R^{c,\texttt{INTER}}_{n,i}(l)$ that takes into account the rate needed to encode the position of the keypoint associated to $d_{n,i}$, relative to $\tilde{d}_{n-1,l}$. Such term takes into account: i) the number of bits needed to encode the identifier of the reference keypoint; ii) the bits used to entropy code the differences $\tilde{\mathbf{p}}_{n,i} - \tilde{\mathbf{p}}_{n-1,l}$, i.e., the equivalent of motion vectors in traditional video coding, as discussed in detail in Section III-C. The cost function $J^{\texttt{INTER}}$ is the result of a Lagrangian relaxation of a rate-distortion optimization problem. Section III-D aims at investigating the value to be assigned to the Lagrangian multiplier λ.
- *Intra-descriptor transform*: Compute the output of the intra-descriptor transform

$$\mathbf{c}^{\texttt{INTRA}}_{n,i} = \mathbf{T}^{\texttt{INTRA}} \mathbf{d}_{n,i}, \quad (7)$$

$$\mathbf{c}^{\texttt{INTER}}_{n,i} = \mathbf{T}^{\texttt{INTER}} (\mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n-1,l^*}), \quad (8)$$

  where $\mathbf{T}^{\texttt{INTRA}}$ and $\mathbf{T}^{\texttt{INTER}}$ are two different Karhunen-Loève Transform matrices that are trained as detailed in Section IV based on, respectively, descriptors and prediction residuals. Note that $\mathbf{c}_{n,i}, \mathbf{d}_{n,i} \in \mathbb{R}^P$, that is, the KLT transform is applied in order to decorrelate descriptor elements, thus increasing the coding efficiency, rather than reducing the dimensioality. When no transform is used, $\mathbf{T}^{\texttt{INTRA}} = \mathbf{T}^{\texttt{INTER}} = \mathbf{I}$.
- *Coding mode decision*: For each local feature, the coding mode that leads to the highest coding efficiency is selected. For the sake of clarity, we initially describe how the coding mode is selected when no transform is used. In this case, two coding modes are available: inter-frame coding and intra-frame coding. Therefore, for each feature, we compare the cost of inter-frame coding, i.e.,

$$J^{\texttt{INTER}}(d_{n,i}, \tilde{d}_{n-1,l^*}) = \frac{1}{\sqrt{P}} \|\mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n,i}\|_2 + \ldots + \lambda (R^{c,\texttt{INTER}}_{n,i}(l^*) + R^{d,\texttt{INTER}}_{n,i}(l^*)), \quad (9)$$

  with that of intra-frame coding, i.e.,

$$J^{\texttt{INTRA}}(d_{n,i}) = \frac{1}{\sqrt{P}} \|\mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n,i}\|_2 + \lambda (R^{c,\texttt{INTRA}}_{n,i} + R^{d,\texttt{INTRA}}_{n,i}), \quad (10)$$

  where $R^{c,\cdot}_{n,i}$ and $R^{d,\cdot}_{n,i}$ indicate, respectively, the number of bits to encode the location component and the descriptor component of $d_{n,i}$, as detailed in Section III-C[1], while l^* represents the index of the reference feature identified by the *Descriptor Matching* phase. To compute $R^{d,\texttt{INTER}}_{n,i}(l^*)$, the prediction residuals $\mathbf{c}^{\texttt{INTER}}_{n,i}$ are quantized and entropy coded, counting the number of bits of the corresponding bit-stream. Similarly, $R^{d,\texttt{INTRA}}_{n,i}$ is obtained by quantizing and entropy coding $\mathbf{c}^{\texttt{INTRA}}_{n,i}$.
  A mode decision is made comparing $J^{\texttt{INTER}}(d_{n,i}, \tilde{d}_{n-1,l^*})$ and $J^{\texttt{INTRA}}(d_{n,i})$. Specifically, if $J^{\texttt{INTER}}(d_{n,i}, \tilde{d}_{n-1,l^*}) < J^{\texttt{INTRA}}(d_{n,i})$, then we select inter-frame coding. Otherwise, the descriptor is encoded in intra-frame mode.

[1]Since $\mathbf{T}^{\texttt{INTRA}}$ and $\mathbf{T}^{\texttt{INTER}}$ are orthonormal transforms, $\|\mathbf{d}_{n,i}\|_2 = \|\mathbf{c}^{\texttt{INTRA}}_{n,i}\|_2$ and $\|\mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n-1,l^*}\|_2 = \|\mathbf{c}^{\texttt{INTER}}_{n,i}\|_2$.

Note that the two cost functions, corresponding to inter-frame coding used during descriptor matching (6) and coding mode decision (9), are slightly different. In particular, (9) takes into account also the number of bits needed to encode the descriptor prediction residuals $R_{n,i}^{d,\texttt{INTER}}$, which is not included in (6). In accordance with a best practice in conventional video coding, this is done to reduce the computational complexity of evaluating (6), which is repeated for all possible matching candidates, since computing $R_{n,i}^{d,\texttt{INTER}}$ requires to quantize the prediction residuals corresponding to each matching candidate, and compute the size of the bitstream generated after entropy coding the quantization symbols. Although this can be done by computing the codeword lengths based on the probabilities of the symbols (i.e., without explicitly generating the codewords), it would nevertheless add complexity to the encoder. Furthermore, our experiments showed that the coding gain attained taking into account $R_{n,i}^{d,\texttt{INTER}}$ in addition to $R_{n,i}^{c,\texttt{INTER}}$ in the cost function (6) was negligible, namely lower than 0.3% on average. In addition, in (6) the prediction residuals are computed based on the reconstructed descriptor $\tilde{d}_{n-1,l^*}$.

When the intra-descriptor transform is enabled, the coding mode decision selects the best out of four different coding modes: i) inter-frame coding, no transform; ii) inter-frame coding, with transform; iii) intra-frame coding, no transform; iv) intra-frame coding, with transform. Case i) and ii) use the same cost function as in (9), with the difference that the term $R_{n,i}^{d,\texttt{INTER}}(l^*)$ accounts for the rate needed to encode the prediction residuals, or the transformed prediction residuals, respectively. Case iii) and iv) use the same cost function as in (10), with the difference that the term $R_{n,i}^{d,\texttt{INTRA}}$ accounts for the rate needed to encode the descriptor in its original domain, or the transformed descriptor, respectively.

- *Quantization*: Scalar quantization with step size Δ_j is applied to the P elements of $\mathbf{c}_{n,i}$. In the case of $\mathbf{c}_{n,i}^{\texttt{INTRA}}$, we use the deadzone quantizer in (4). In the case of $\mathbf{c}_{n,i}^{\texttt{INTER}}$, we adopt a scalar uniform quantizer without deadzone:

$$\tilde{c}_{n,i,j} = \Delta_j \cdot \text{round}(c_{n,i,j}/\Delta_j) \tag{11}$$

  Our experiments revealed that the adoption of a deadzone does not bring any coding gain in the case of quantizing the prediction residuals $\mathbf{c}_{n,i}^{\texttt{INTER}}$, due to the fact that the statistics differ from those of the original dexels. We fix the same quantization step size for all dexels, i.e., $\Delta_j = \Delta,\ j = 1,\ldots,P$.
- *Entropy coding*: Entropy coding proceeds as detailed in Section III-C.

C. Entropy Coding

Entropy coding takes care of exploiting the statistical redundancy for both the location and the descriptor component of $d_{n,i}$. As for the descriptor component, the output symbols of the quantizer $\tilde{\mathbf{c}}_{n,i}$ are entropy coded using arithmetic coding, resulting in $R_{n,i}^{d}$ bits. Depending on the coding mode, either $\tilde{\mathbf{c}}_{n,i}^{\texttt{INTRA}}$ or $\tilde{\mathbf{c}}_{n,i}^{\texttt{INTER}}$ is encoded. The probabilities of the symbols used by the entropy coder are learned from descriptors extracted from a training set of frames.

- *Intra-frame coding*: The statistics are collected by quantizing a large set of descriptors at different quantization step sizes. Then, for a given value of Δ, and for each of the P dexels, we estimate the probability of the symbols counting the number of occurrences in the training set of each of the possible reconstruction levels of the quantizer. Different statistics are computed depending on whether the transform $\mathbf{T}^{\texttt{INTRA}}$ is used.
- *Inter-frame coding*: The probabilities of the symbols used for entropy coding are learned from a training set of video frames. To this end, we considered only descriptors for which a good match was found, i.e., $\|\mathbf{d}_{n,i} - \mathbf{d}_{n-1,l^*}\|_2 < \|\mathbf{d}_{n,i}\|_2$. For each possible value of the quantization step size Δ, we computed the quantized prediction residuals $\tilde{\mathbf{c}}_{n,i}$, possibly after an intra-descriptor transform $\mathbf{T}^{\texttt{INTER}}$, and obtained the statistics as in the case of intra-frame coding.

To encode the location component of $d_{n,i}$, we proceed as follows.

- *Intra-frame coding*: The coordinates of each keypoint (at quarter-pel accuracy) are encoded using $R_n^{c,\texttt{INTRA}} = M_n \cdot (\log_2 4N_x + \log_2 4N_y + S)$ bits, where S is the number of bits use to encode the scale parameter. Higher coding efficiency is achievable implementing ad-hoc lossless coding schemes to compress the coordinates of the keypoints [41], [42].
- *Inter-frame coding*: In this case the location component of the keypoint is encoded with respect to the one of the matching keypoint d_{n-1,l^*}, which requires $R_{n,i}^{c,\texttt{INTER}}(l^*)$ bits. The motivation behind this choice is that encoding the displacement $\tilde{\mathbf{p}}_{n,i} - \tilde{\mathbf{p}}_{n,l^*}$ requires fewer bits than encoding $\tilde{\mathbf{p}}_{n,i}$ directly, due to the temporal redundancy between matching features belonging to contiguous frames. To this end, it is necessary to encode: i) the identifier of the matching keypoint in the reference frame; ii) the position and the scale of the keypoint with respect to the matching keypoint.

  We devised a predictive strategy for encoding the reference keypoint identifiers. Let $\mathcal{M}_{n\rightarrow n-1}$ denote a mapping between each inter-frame encoded feature belonging to the set of visual features $\tilde{\mathcal{D}}_n$ extracted from the frame $\mathcal{I}_n$ and the corresponding reference feature belonging to the set $\tilde{\mathcal{D}}_{n-1}$, as illustrated in Fig. 4(a). Based on this, we reorder the visual features in $\tilde{\mathcal{D}}_n$ by ascending keypoint reference identifier, such that the resulting mapping vector $\mathcal{M}'_{n\rightarrow n-1}$ contains monotonically increasing identifiers, as depicted in Fig. 4(b). Then, we compute the vector $\mathcal{O}_{n\rightarrow n-1}$, which contains the offsets between each pair of contiguous identifiers, by performing a simple cell-by-cell subtraction. We observe that the probabilities of observing different offset values are not uniformly distributed. Conversely, values with smaller modulus are much more likely to occur.

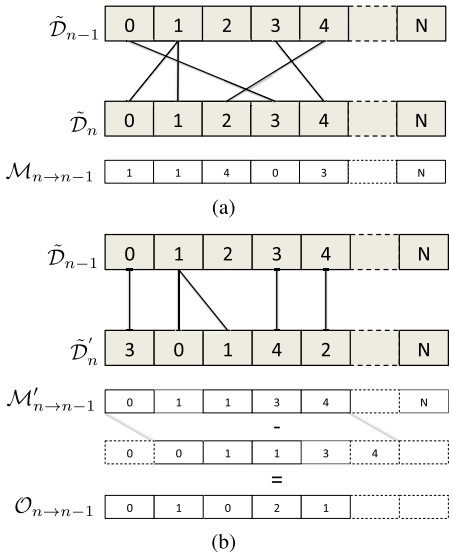

Fig. 4. The set $\tilde{\mathcal{D}}_n$ of visual features belonging to the current frame (a) is reordered by ascending keypoint reference identifiers $\mathcal{M}'_{n\to n-1}$ obtaining $\tilde{\mathcal{D}}'_n$; (b) The identifier offset vector $\mathcal{O}_{n\to n-1}$ is computed by subtracting to each element of $\mathcal{M}'_{n\to n-1}$ the previous one. Note that some features belonging to the reference frame (e.g. feature 2) might possibly be unused.

Therefore, we learn such statistics and store them in a look-up table, so that they can be used to drive the arithmetic coder. Note that the reordering of the features that leads from $\mathcal{M}_{n\to n-1}$ to $\mathcal{M}'_{n\to n-1}$ need not to be communicated explicitly to the decoder, since it is implicitly obtained by the ordering in which the features are encoded and written to the bitstream.

Once the identifier of the reference matching keypoint is determined, we need to encode the position and the scale of the keypoint, relative to its matching keypoint. That is, we encode $\tilde{\mathbf{p}}_{n,i} - \tilde{\mathbf{p}}_{n-1,l^*}$, which is similar to the notion of motion vector in the case of video coding. Given a large set of features extracted from training video sequences, we learn the statistics of $\tilde{\mathbf{p}}_{n,i} - \tilde{\mathbf{p}}_{n-1,l^*}$, so that they can be used by the arithmetic coder. For further details, refer to the technical report [2].

D. Rate-Distortion Optimization

The coding mode decision described in Section III-B follows the rate-distortion optimization (RDO) approach commonly employed in state-of-the-art video coding. Rate-distortion optimization aims at minimizing distortion, subject to a constraint on the available bit budget. Such constrained optimization problem can be solved resorting to Lagrangian relaxation, in which an unconstrained problem is formulated, whose objective function is obtained combining the distortion introduced by lossy coding with the rate needed to encode the visual features. Such a trade-off is controlled by means of the Lagrange multiplier λ in (9) and (10).

To find the optimal value of λ we proceeded adapting the approach presented in [14] and [43] to the case of visual features. First, we sampled a set of possible values, i.e., $\lambda \in \{0, 0.1, 0.2, 0.3, \ldots 10\}$. For each value of λ, we processed

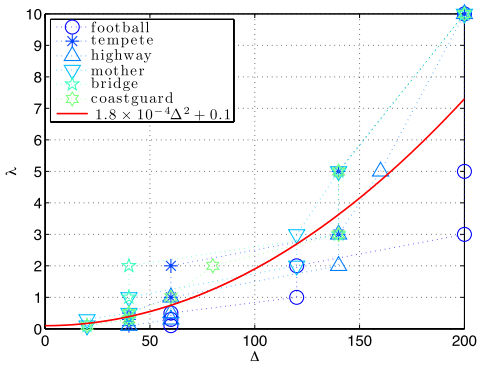

Fig. 5. Empirical relation between the target quantization step size Δ and the optimal Lagrangian parameter λ, depicted along with the function $\lambda(\Delta) = 1.8 \times 10^{-4}\Delta^2 + 0.1$

all descriptors in a set of video sequences. Each visual feature was encoded using each of the four coding modes described in Section III-B, by varying the quantization step size $\Delta \in \{200, 180, 160, ..., 2\}$. For each visual feature, we collected the set of values attained by the four cost functions, for each of the tested quantization step sizes Δ. Then, we searched for the coding mode and the value of Δ that minimized such cost functions. With this, we obtained an empirical distribution of the quantization step size Δ, which indicates the fraction of times a given value of Δ led to the minimum value of the cost function, when using a specific value of λ.

Finally, for all the video sequences, we selected the modes of such empirical distributions so as to obtain a relationship between the quantization step size Δ and the Lagrange multiplier λ, as illustrated in Fig. 5. Then, we fit a model $\lambda(\Delta)$ by means of least squares considering the curves obtained for all video sequences, in order to obtain a relationship which is content-independent.

Note that, unlike in traditional video coding, in this case the distributions are relatively flat. Therefore the estimated location of the mode of the distribution cannot always be determined exactly, as demonstrated by the noisy values of λ reported in Fig. 5.

IV. Experimental Study

We experimentally validated the coding efficiency of the proposed video coding architecture for visual features according to two evaluation methods. In the first case, we used conventional rate-distortion curves, whereby distortion is expressed in terms of the signal-to-noise ratio (SNR) between the original and reconstructed features in the dexel domain. Second, and more interestingly, we compared the two paradigms, namely *"Analyze-Then-Compress"* and *"Compress-Then-Analyze"* in terms of rate-efficiency curves, whereby efficiency expresses a quantitative metrics related to the performance of fulfilling a visual analysis task. To this end, two main tasks were investigated:

- *Content Based Video Retrieval (CBVR)*. Given an input query in the form of some kind of visual content, the goal is to retrieve the most relevant frames in a database of video sequences according to a similarity criterion.

Visual features play a fundamental role in Content Based Video Retrieval as they are key for the summarization and indexing of large databases of video sequences.

- *Homography estimation*. Several computer vision tasks, including camera calibration, 3D reconstruction, scene understanding, structure-from-motion, object tracking, etc., may require the estimation of a homography describing the geometric deformation between two frames of the same video sequence. Visual features have been successfully employed for this kind of task.

Due to the page limit constraint, we included a selection of the experimental results in this manuscript. Additional results are available as supplementary material in the extended technical report [2].

A. Data Sets

The proposed coding architecture requires an initial training phase to determine the intra-descriptor transform, the statistics of the symbols to be entropy coded, and the relationship between λ and the selected quantization step size Δ. The training set is composed by three video sequences at CIF resolution (352 $\times$ 288) and 30 fps, namely *Paris*, *News* and *Mother*, each with 300 frames.

Two different test sets were used. The rate-distortion analysis and the rate-efficiency analysis related to the CBVR scenario were based on eight video sequences at CIF resolution (352 $\times$ 288) and 30 fps, namely *Hall*, *Mobile*, *Foreman*, *Football*, *Coastguard*, *Bus*, *Bridge* and *Container*. Each test video sequence consists of 300 frames. For the homography estimation scenario, a publicly available dataset for visual tracking was employed [44]. Each video sequence consists in a planar texture subject to a given motion path. A total of six different rectangular textures (*Bricks*, *Building*, *Mission*, *Paris*, *Sunset*, *Wood*) and several motion paths (unconstrained, panning, rotation, perspective distortion, zoom, motion blur, static lighting, dynamic lighting) are combined. For each frame of each sequence, the homography that warps such frame to the reference one is provided as ground truth. In details, the ground truth homography represents the transformation that projects the coordinates of the four corners of the texture in the current frame to their own coordinates in the canonical frame. In our experiments we employed the six sequences corresponding to unconstrained motion due to their generality. Each video sequence has a resolution of 640 $\times$ 480 pixel at 15 fps and a length of 500 frames (33.3 seconds). We temporally down-sampled the sequences to 3 fps, in order to increase the differences between consecutive frames, so as to make the homography estimation task sufficiently arduous.

B. Methods

In the *"Analyze-Then-Compress"*paradigm, the input video sequences were analyzed to produce a compressed representation in the form of a set of visual features for each frame. Two algorithms, namely SIFT [4] and SURF [29] were considered for visual feature extraction. In particular, we adopted the VLFEAT [45] and OpenSURF [46] implementations for SIFT and SURF, respectively.

A training phase was necessary to learn the intra-descriptor transforms and the statistics used by the entropy coder. In the case of intra-frame coding, the transform $\mathbf{T}^{\texttt{INTRA}}$ was estimated using the KLT, considering all the descriptors extracted from the training set after subtracting the average of each element. The statistics of the quantization symbols were determined as described in Section III-C. In the case of inter-frame coding, we estimated the transform $\mathbf{T}^{\texttt{INTER}}$ using the KLT, the statistics of the differences between the keypoint locations and feature identifiers, and the statistics of the prediction residuals, as detailed in Section III-C. Such statistics were stored in look-up tables both at the encoder and at the decoder.

Each video sequence in the test set was processed as follow. For each frame $\mathcal{I}_n$, a set of visual features $\mathcal{D}_n$ was computed. Then, fixing a value Δ for the quantization step size and resorting to the information learned in the training phase, the features were encoded and decoded following the procedure described in Section III. Then, the sets of reconstructed visual features $\tilde{\mathcal{D}}_{n,\Delta}$ were given as input to the specific visual analysis task.

Within the `ATC` paradigm, we distinguish between several different coding schemes:

- `INTRA`: all visual features were encoded resorting to an intra-frame coding scheme. No intra-descriptor transform was used.
- `INTRA` - KLT: all visual features were encoded resorting to an intra-frame coding scheme. The intra-descriptor transform was applied to all descriptors.
- `INTER`: all visual features were encoded resorting to an inter-frame coding scheme. No intra-descriptor transform was used.
- `INTER` - KLT: all visual features were encoded resorting to an inter-frame coding scheme. The intra-descriptor transform was applied to all prediction residuals.
- `INTRA-INTER`: for each visual feature, a 2-way coding mode decision module selects the best coding mode between `INTRA` and `INTER`. No intra-descriptor transform was used.
- `INTRA-INTER` - KLT : for each visual feature, a 4-way coding mode decision module selects the best coding mode between `INTRA`, `INTER`, `INTRA` - KLT, `INTER` - KLT.

Note that for `INTER` (`INTER` - KLT), all the features are coded resorting to inter-frame coding, except for the ones for which it is not possible to find a reference keypoint within the spatial search window. Such features are coded resorting to intra-frame coding mode. Nonetheless, tests show that more than 98% of the features are encoded resorting to `INTER` (`INTER` - KLT) coding mode.

For the CBVR and homography estimation scenarios, we also report the results obtained according to a traditional *"Compress-Then-Analyze"* paradigm. The input video sequences were compressed with the H.264/AVC coding standard, using the x264 video coding library. The library enables to specify a quality factor Q which is mapped to a target bitrate. We used $Q \in \{5, 10, \ldots, 45\}$ in our experiments. Then, either SIFT or SURF visual features were extracted from each

frame $\tilde{\mathcal{I}}_n$ of the compressed test sequence, providing sets of visual features $\tilde{\mathcal{D}}_{n,Q}$. Such sets of visual features were given as input to the specific visual analysis task.

C. Parameter Settings

In both SIFT and SURF, a threshold determines the number of detected keypoints, which is content-dependent. We set the thresholds so as to obtain the average number of features reported in Table I and II. The other parameters were left equal to their default values. Each element of the SIFT descriptor is represented by an 8-bit integer, whereas the SURF algorithm provides descriptors in the form of vectors of 32-bit floating point variables. To make the two descriptors comparable, we quantized each 32-bit SURF dexel to an 8-bit signed integer.

The parameters x, y, σ, representing the location and the scale of each keypoint, were rounded to the nearest quarter of unit, as mentioned in Section III.

For the tests to be as fair as possible, the video coding scheme and the visual feature coding scheme should operate under comparable conditions. In particular, the following settings were employed with the x264 library, by adopting coding tools that are supported by the H.264/AVC baseline profile, which is tailored for wireless communications:

- number of reference frames: 1 (`–ref 1`)
- B-frames disabled (`–bframes 0`)
- subpixel motion estimation complexity: quarter of pixel (`–subme 4`)
- Trellis quantization disabled (`–trellis 0`).
- Context-Adaptive Binary Arithmetic Coding (CABAC) disabled (`–no-cabac`).

The *Constant Rate Factor* parameter (`-crf <integer>`) was employed to control the output bitrate. It is important to emphasize that the H.264/AVC standard is the result of many years of optimization, while coding of visual features has only been recently explored. Therefore, some of the coding tools successfully adopted in H.264/AVC (e.g., B-frame, multiple reference frames, etc.), might also be integrated into our coding architecture. This is left to future investigation.

D. Evaluation Metrics

We evaluated the proposed coding architecture according to different testing pipelines and related metrics:

- *Rate-Distortion Analysis*. Considering a test sequence, for each frame $\mathcal{I}_n$ the set of features $\mathcal{D}_n$ was computed. Then, for each possible value Δ of the quantization step size, the reconstructed sets of features $\tilde{\mathcal{D}}_{n,\Delta}$ were obtained following the `ATC` paradigm as explained in Section IV-B. Finally, the tradeoff between the rate needed to encode the visual features and the distortion introduced by the quantization was investigated. The rate includes the number of bits needed to encode both the locations of the keypoints and the descriptors (expressed in bits/feature). The distortion is measured in terms of the signal-to-noise ratio (SNR), which is defined as

$$SNR = 10 \log_{10} \frac{\sum_{n=1}^{N} \sum_{i=1}^{M_n} \|\mathbf{d}_{n,i}\|_2^2}{\sum_{n=1}^{N} \sum_{i=1}^{M_n} \|\mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n,i}\|_2^2} \quad (12)$$

- *Content Based Video Retrieval*. In the `ATC` case, the sets of visual features $\mathcal{D}_n$ were encoded and decoded to $\tilde{\mathcal{D}}_{n,\Delta}$ as described above. Instead, in the `CTA` case, the sets of features $\tilde{\mathcal{D}}_{n,Q}$ were obtained as described in Section IV-B. The efficiency of `ATC` and `CTA` was evaluated in terms of *repeatability* and *matching score*. These metrics are used to assess the influence of lossy coding on, respectively, the accuracy of the detector and of the descriptor.

  The accuracy of the detector was evaluated according to the *repeatability* score between the set of original features $\mathcal{D}_n$ and the set of reconstructed ones $\tilde{\mathcal{D}}_n$ (i.e., either $\tilde{\mathcal{D}}_{n,\Delta}$ or $\tilde{\mathcal{D}}_{n,Q}$). Such a metric measures the average number of corresponding regions detected in the two sets [47]. Specifically, for each visual feature d, it is possible to define a spatial region R_d whose area is proportional to the scale of the detected keypoint. Moreover, it is possible to define the *overlap error* between two regions R_{d_a} and R_{d_b} as

$$\mathcal{E}_{a,b} = 1 - \frac{R_{d_a} \cap R_{d_b}}{R_{d_a} \cup R_{d_b}} \quad (13)$$

  Two regions are deemed to correspond if their *overlap error* is lower than $\epsilon = 0.5$. The *repeatability* for a given pair of sets of visual features is computed as the ratio between the number of region-to-region correspondences and the smallest between the number of features in the two sets, i.e., $\min\{|\mathcal{D}_n|, |\tilde{\mathcal{D}}_n|\}$ in our case. To evaluate the performance we computed the *repeatability* score between the set of original features $\mathcal{D}_n$ and the set of reconstructed features, i.e., $\tilde{\mathcal{D}}_{n,\Delta}$ or $\tilde{\mathcal{D}}_{n,Q}$, in the `ATC` or `CTA` case. The final *repeatability* value was obtained by averaging the scores of all the frames belonging to the test video sequence.

  The accuracy of the descriptor was measured in terms of *matching score* [47]. The correspondences estimated during the computation of the *repeatability* provide a ground truth for the computation of such a metrics. For each correspondence between a pair of keypoints, a match is deemed correct if the two features are also the nearest neighbours (in terms of Euclidean distance) in the descriptor space. The rationale is that two descriptors corresponding to matching patches should be close to each other in the descriptor space, and possibly far from descriptors associated to other patches. If this were not the case, incorrect correspondences between patches would be determined, thus undermining the retrieval process. The *matching score* is defined as the ratio between the number of correct matches and the smallest between the number of detected features in the two sets. Once again, the final value is obtained by averaging the *matching scores* of all frames.
- *Homography Estimation*. In the case of `ATC`, the sets of features $\mathcal{D}_n$ were extracted starting from the test sequences. Such sets were filtered, removing the keypoints that did not belong to the planar texture identified by the available ground truth. For each value of the quantization step size Δ, the sets $\tilde{\mathcal{D}}_{n,\Delta}$ were obtained

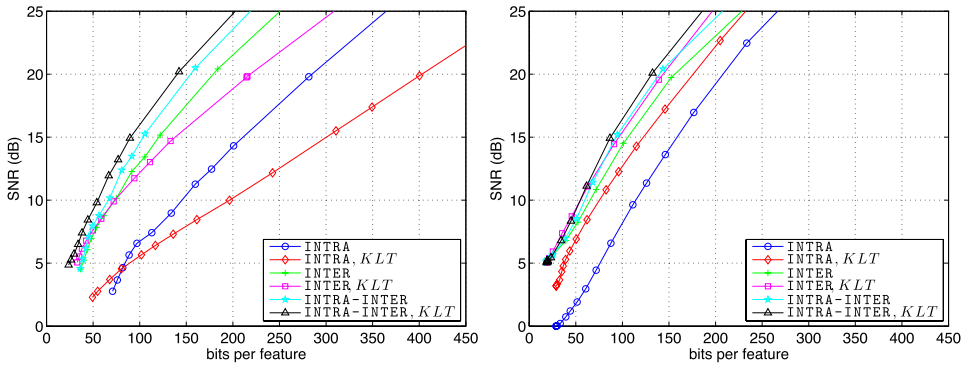

Fig. 6. Rate-distortion curves obtained with the ATC coding architecture for the *Foreman* sequence. (a) *SIFT*, (b) *SURF*.

following the ATC paradigm. For each pair of consecutive frames $\mathcal{I}_n$ and $\mathcal{I}_m$, a homography $\tilde{H}_{nm,\mathrm{ATC},\Delta}$ was estimated based on $\tilde{\mathcal{D}}_{n,\Delta}$ and $\tilde{\mathcal{D}}_{m,\Delta}$. To this end, the matches between the two sets of features were identified and given as input to the RANSAC algorithm [39].

As for CTA, the test sequences were encoded with each one of the quality factors $Q = \{5, 10, \ldots, 45\}$. For each frame $\mathcal{I}_n$ of the encoded sequence the sets of features $\tilde{\mathcal{D}}_{n,Q}$ were extracted. Similarly to the ATC case, the sets of visual features were filtered and for each pair of consecutive frames $\mathcal{I}_n$ and $\mathcal{I}_m$, a homography $\tilde{H}_{nm,\mathrm{CTA},Q}$ was estimated resorting to $\tilde{\mathcal{D}}_{n,Q}$ and $\tilde{\mathcal{D}}_{m,Q}$.

The performance of ATC and CTA was evaluated in terms of rate-efficiency curves. For the task at hand, efficiency was measured computing the *homography estimation precision*. Specifically, let $\tilde{H}_{nm}$ denote the homography estimated according to the procedure presented above, following either the ATC or the CTA approach. The coordinates of the four corners of the texture $c_{1,n}$, $c_{2,n}$, $c_{3,n}$, $c_{4,n}$ in frame $\mathcal{I}_n$ are provided as ground truth. Applying the homography $\tilde{H}_{nm}$ to such points, it was possible to obtain the estimated coordinates $\tilde{c}_{1,m}$, $\tilde{c}_{2,m}$, $\tilde{c}_{3,m}$, $\tilde{c}_{4,m}$ in frame $\mathcal{I}_m$ and compare them with the real coordinates of the corners $c_{1,m}$, $c_{2,m}$, $c_{3,m}$, $c_{4,m}$, also available as ground truth. The *backprojection error* for the frame m is defined as $\mathcal{E}_{bp}(m) = \frac{1}{4}\sum_{p=1}^{4} |\tilde{c}_{p,m} - c_{p,m}|$. An estimated homography was deemed correct if the relative backprojection error was lower than $\epsilon_{bp} = 3$ pixels. Finally, the *homography estimation precision* is defined as the ratio between the number of correctly estimated homographies and the total number of frames.

E. Results

- *Rate-distortion analysis*. The rate-distortion curves obtained for the *Foreman* video sequence are reported in Fig. 6, for both SIFT (a) and SURF (b) (for supplementary results, refer to the technical report [2]). A comparison between the INTRA and INTER coding strategies reveals that the inter-frame coding scheme leads to significant coding gains with respect to intra-frame coding at all bitrates, regardless of the encoded content.

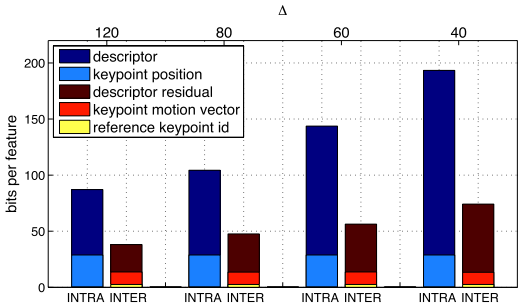

Fig. 7. Bit budget allocation between keypoint coordinates and descriptor elements, averaged over the test sequences.

Larger gains are observed for SIFT than for SURF. Indeed, it is possible to note that, in the case of intra-frame coding, SURF requires a lower amount of bits than SIFT to achieve the same distortion. This is due to the intrinsic nature of the SIFT and SURF descriptors, consisting of 128 and 64 dexels, respectively. However, in the case in inter-frame coding, the 128-elements prediction residuals obtained using SIFT can be more efficiently compressed, thus outperforming the results achieved by inter-frame coding SURF. For example, considering the *Foreman* test sequence, 1.13 bits (2.11 bits) are needed to encode each dexel of SIFT (SURF), in order to attain a distortion equal to 20dB SNR. This can be attributed to the higher repeatability of the detector used by SIFT. Due to this, the patches around the corresponding keypoints in the current and reference frames are more similar in SIFT than in SURF, thus leading to descriptors that are more temporally correlated.

In this context, we also investigated the allocation of the bit budget between keypoint coordinates and descriptor elements. Fig. 7 shows such allocation with respect to the SIFT algorithm and for different values of Δ, which correspond to a subset of the points used to generate the operational rate-distortion curves. In the case of inter-frame coding, the cost of encoding the keypoint coordinates includes the bits to represent the reference keypoint identifiers, as well as the displacement between keypoints. We observe that inter-frame coding reduces the number of bits necessary to represent both the keypoint coordinates and the descriptor elements.

TABLE I

MINIMUM BITRATE TO ACHIEVE PERFORMANCE SATURATION - CONTENT-BASED VIDEO RETRIEVAL

SIFT (15dB SNR)	*foreman*	*hall*	*mobile*	*football*	*coastguard*	*bus*	*bridge*	*container*	ave. rate reduction
ave. number of features	177	117	185	126	179	190	162	163	
Uncompressed (kbps)	5434	3592	5680	3715	5496	5833	4974	5004	-
INTRA (kbps)	1037	788	1280	617	944	1108	806	864	5:1
INTER (kbps)	544	154	379	602	570	580	320	382	11:1
INTRA/INTER (kbps)	477	102	216	596	519	514	299	358	12:1
INTRA, KLT (kbps)	1555	1045	1734	978	1520	960	1620	1024	4:1
INTER, KLT (kbps)	726	188	461	697	750	751	480	475	9:1
INTRA/INTER, KLT (kbps)	**452**	**98**	**202**	**586**	**470**	**497**	**275**	**337**	**14:1**
SURF (15dB SNR)									
ave. number of features	182	156	178	182	158	153	117	137	
Uncompressed (kbps)	2858	2449	2795	2120	2481	2386	1837	2151	-
INTRA (kbps)	880	708	714	972	1035	1002	832	983	2:1
INTER (kbps)	517	347	469	498	362	481	192	236	4:1
INTRA/INTER (kbps)	480	288	391	474	338	423	161	193	5:1
INTRA, KLT (kbps)	720	548	574	950	998	980	804	931	3:1
INTER, KLT (kbps)	512	343	469	492	352	478	190	230	4:1
INTRA/INTER, KLT (kbps)	**472**	**285**	**384**	**468**	**316**	**405**	**154**	**178**	**5:1**

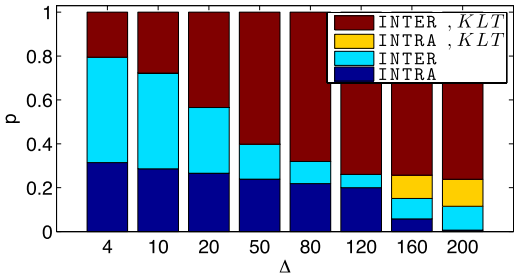

Fig. 8. Proportion of SIFT keypoints encoded with any of the four possible coding modes for *Foreman* test sequence.

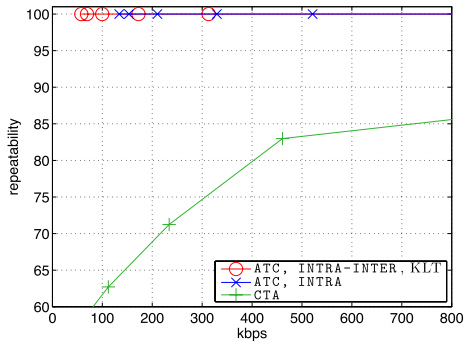

Fig. 9. Rate-repeatability curves for the Content Based Video Retrieval test, *Foreman* sequence, SIFT.

Further coding gains can be achieved by adaptively selecting either intra-frame or inter-frame coding on a feature-by-feature basis, according to the RDO-based mode decision detailed in Section III-B, as shown by the INTRA − INTER curve in Fig. 6.

The adoption of the KLT to decorrelate the descriptor elements leads to substantial gains for SURF, while it provides worse coding efficiency for SIFT, apart at very low bitrates. A similar observation is reported by Chandrasekhar et. al. [24]. The KarhunenLove theorem guarantees optimal energy compaction in the case of a Gaussian source. Indeed, Feng et. al. [48] showed that KLT is not optimal when other distributions are considered. In this case, the distribution of SIFT descriptor elements is bi-modal, and cannot be well approximated by means of a Gaussian [1], [24]. Similar considerations can be made when using the KLT on the prediction residuals, in the case of inter-frame coding. Coding gains are achieved for SURF, and for SIFT at low bitrates.

In order to achieve the highest coding efficiency it is necessary to optimally switch among different coding modes, on a feature-by-feature basis. Indeed, a coding scheme that performs a 4-way mode decision (see INTRA − INTER, KLT curves) achieves the best results at all bitrates. Note that, in this case, the KLT transform is applied only when deemed useful in rate-distortion sense. In particular, Fig. 8 shows the fraction of SIFT features that are encoded resorting to each possible coding mode. It is possible to note that the KLT seems to be an advantageous option especially at low bitrates, and it is mostly applied to inter-frame coded descriptors.

- *Content-Based Video Retrieval*. Fig. 9 shows that the repeatability values in the case of ATC are not influenced by the quarter of unit approximation of the keypoint location elements, thus resulting in 100% repeatability at all working points. Conversely, in the CTA case, lossy coding introduces distortion in the pixel domain, thus affecting the output of the detector. In particular, only a subset of the keypoints extracted from the original uncompressed sequence are obtained from the decoded video sequence. For example, at 200 kbps, only 70% (80%) of the original SIFT (SURF) keypoints are detected. Similar results were obtained for all other tested video sequences.

  Fig. 10 shows that the ATC scheme outperforms CTA in terms of matching score, saturating to 100% at high rates. The results are presented for the *Foreman* sequence, although similar outcomes were obtained for all other test sequences (for supplementary results, refer to the technical report [2]). The use of inter-frame coding of visual features enables to reduce the bitrate with respect to intra-frame coding, while attaining the same performance in terms of matching score. In particular, the SIFT features resulted to be particularly robust to lossy coding. Indeed, as illustrated in Fig. 10, the Content-Based Video

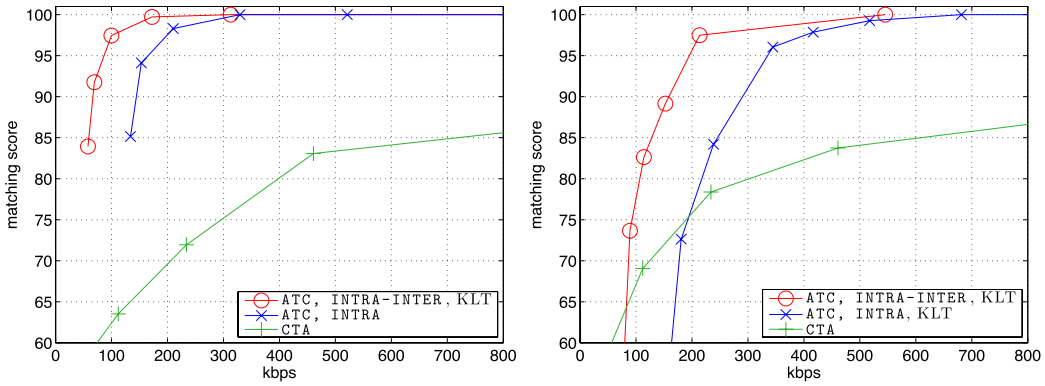

Fig. 10. Rate-matching score curves for the Content Based Video Retrieval test, *Foreman* sequence. (a) SIFT; (b) SURF.

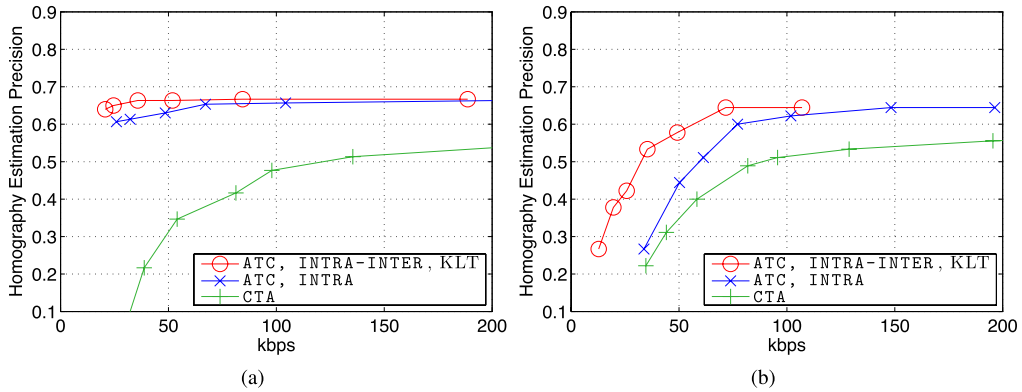

Fig. 11. Rate-accuracy curves for the Homography Estimation test, *Building* sequence. (a) *SIFT*, (b) *SURF*.

Retrieval task does not significantly suffer a loss of performance, even when considering the working point corresponding to the highest possible value of the quantization step (i.e., when SIFT dexels are represented as binary values). For *Foreman*, the loss is up to 15%, whereas it is as little as 5% for *Hall* and *Mobile*. In this respect, a different behaviour was observed for SURF. In fact, at very low bitrates, the matching score drops significantly, also in the ATC case.

It is interesting to consider the operating point corresponding to the minimum bitrate at which saturation of matching score is achieved. For both SIFT and SURF, this corresponds to the case of visual features encoded at approximately 15dB. Table I summarizes the results obtained for all test video sequences, considering the six different coding schemes considered in the rate-distortion analysis.

Finally, in the case of CTA, we observe that the curves representing repeatability and matching score are very similar to each other for all tested sequences. Indeed, only a subset of the keypoints extracted from an original uncompressed video sequence were obtained from the analysis of the corresponding reconstructed sequence at the decoder. However, for the detected keypoints, the descriptors were correctly matched, despite being extracted from the decoded sequences. Hence, it is possible to conclude that video coding impairs the performance of the detector more than that of the descriptor.

- *Homography Estimation*. The results obtained with both the ATC and CTA cases are reported in Fig. 11 (for supplementary results, refer to the technical report [2]). In all cases, ATC outperforms CTA by a large margin. In the ATC case, the gain achieved when adopting inter-frame coding instead of intra-frame coding is narrower than in the CBVR scenario. This is due to the fact that sequences were down-sampled to 3 fps, and temporal redundancy is weaker.

  In the case of SIFT, performance rapidly saturates when using ATC. Indeed, we are able to reconstruct (with the same precision obtained using uncompressed SIFT features extracted from uncompressed sequences) 3 homographies per second, with an available bitrate equal to 50 kbps. The working points corresponding to performance saturation correspond to a distortion in the SNR range 12-17dB.

  On the other hand, in the case of SURF, ATC still outperforms CTA but the gap between the two approaches is smaller than in the case of SIFT. Similarly to the results of the CBVR test, SURF is more sensitive than SIFT when descriptors are quantized. Moreover, saturation is slower and it is reached on average at 67 kbps, which corresponds to a target distortion in the SNR range 18-25dB.

TABLE II

MINIMUM BITRATE TO ACHIEVE PERFORMANCE SATURATION - HOMOGRAPHY ESTIMATION TEST

SIFT (15dB SNR)	*Building*	*Paris*	*Wood*	*sunset*	*mission*	ave. rate reduction
ave. number of features	110	124	125	110	98	
Uncompressed (kbps)	338	381	384	338	301	-
INTRA (kbps)	65	67	71	56	54	5:1
INTRA/INTER, KLT (kbps)	51	52	60	47	40	7:1
SURF (23dB SNR)						
ave. number of features	127	185	132	114	106	
Uncompressed (kbps)	199	290	208	179	166	-
INTRA, KLT (kbps)	75	108	65	89	72	3:1
INTRA/INTER, KLT (kbps)	52	54	51	52	45	4:1

As a summary, Table II reports the minimum bitrate to achieve performance saturation for both SIFT and SURF, with respect to all the test video sequences. Finally, considering the *"Analyze-Then-Compress"* approach, note that SIFT outperforms SURF with respect to all the test video sequences. Conversely, considering the *"Compress-Then-Analyze"* approach, SIFT generally outperforms SURF (for *building*, *sunset*, *mission* test sequences), whereas SURF is the best option for the *Paris* test sequence. On the remaining sequences the two visual features achieve comparable results.

V. CONCLUSION

In this paper we considered the problem of encoding sets of visual features extracted from video sequences. This is an extremely promising direction that enables the *"Analyze-Then-Compress"* paradigm in application scenarios involving video content including, e.g., content-based video retrieval, object tracking, etc. The proposed coding architecture is general, and it can be used to compress any kind of real-valued feature. In our experiments, we showed that large coding gains can be achieved with both SIFT and SURF. In those cases in which the content need not to be reconstructed in the pixel domain, our results demonstrate that the ATC paradigm outperforms CTA.

At the same time, extracting visual features is a computationally intensive task. This issue might be particularly critical when dealing with video content. To address computational concerns, the design of binary descriptors (e.g. BRISK [49], FREAK [50], D-BRIEF [51], etc., and their optimization [52]) is receiving a great deal of attention in the research community. In the case of still images, it was recently shown by the authors that a ATC paradigm based on an optimized version of BRISK outperforms a CTA paradigm based on SIFT [30]. This will stimulate future investigations, which will address the problem of coding binary descriptors extracted from video sequences.

REFERENCES

[1] L. Baroffio, A. Redondi, M. Cesana, S. Tubaro, and M. Tagliasacchi, "Coding video sequences of visual features," in *Proc. 20th IEEE Int. Conf. Image Process.*, Melbourne, Australia, Sep. 2013, pp. 1–5.

[2] L. Baroffio, M. Cesana, A. Redondi, S. Tubaro, and M. Tagliasacchi. (2014, Mar.). *Coding Visual Features Extracted from Video*. Dept. di Elettron., Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy, Tech. Rep. DEIB-2014-1 [Online]. Available: http://home.deib.polimi.it/tagliasa/greeneyes/technote.pdf

[3] M. Brown, G. Hua, and S. A. J. Winder, "Discriminative learning of local image descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 43–57, Jan. 2011.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[5] A. Yang, S. Maji, C. Christoudias, T. Darrell, J. Malik, and S. Sastry, "Multiple-view object recognition in band-limited distributed camera networks," in *Proc. 3rd ACM/IEEE Int. Conf. Distrib. Smart Cameras*, Como, Italy, 2009, pp. 1–8.

[6] A. Redondi, M. Cesana, and M. Tagliasacchi, "Rate-accuracy optimization in visual wireless sensor networks," in *Proc. 19th IEEE Int. Conf. Image Process.*, Orlando, FL, USA, Oct. 2012, pp. 124–129.

[7] B. Tavli, K. Bicakci, R. Zilan, and J. Barcelo-Ordinas, "A survey of visual sensor network platforms," *Multimedia Tools Appl.*, vol. 60, no. 3, pp. 689–726, 2012.

[8] K. Obraczka, R. Manduchi, and J. J. Garcia-Luna-Aveces, "Managing the information flow in visual sensor networks," in *Proc. 5th Int. Symp. Wireless Pers. Multimedia Commun.*, vol. 3. Honolulu, HI, USA, Oct. 2002, pp. 1177–1181.

[9] S. Lee, S. Lee, and A. Bovik, "Optimal image transmission over visual sensor networks," in *Proc. 18th IEEE Int. Conf. Image Process.*, Brussels, Belgium, Sep. 2011, pp. 161–164.

[10] A. Canclini, L. Baroffio, M. Cesana, A. Redondi, and M. Tagliasacchi, "Comparison of two paradigms for image analysis in visual sensor networks," in *Proc. 11th ACM Conf. Embedded Netw. Sensor Syst.*, New York, NY, USA, 2013, pp. 62:1–62:2.

[11] A. Zabala and X. Pons, "Effects of lossy compression on remote sensing image classification of forest areas," *Int. J. Appl. Earth Observat. Geoinf.*, vol. 13, no. 1, pp. 43–51, 2011.

[12] A. Zabala and X. Pons, "Impact of lossy compression on mapping crop areas from remote sensing," *Int. J. Remote Sens.*, vol. 34, no. 8, pp. 2796–2813, Apr. 2013.

[13] S. Paniga, L. Borsani, A. Redondi, M. Tagliasacchi, and M. Cesana, "Experimental evaluation of a video streaming system for wireless multimedia sensor networks," in *Proc. 10th IEEE IFIP Ann. Medit. Ad Hoc Netw. Workshop*, Favignana Island, Italy, Jun. 2011, pp. 165–170.

[14] J. Chao and E. Steinbach, "Preserving SIFT features in JPEG-encoded images," in *Proc. 18th IEEE Int. Conf. Image Process.*, Brussels, Belgium, Sep. 2011, pp. 301–304.

[15] V. Chandrasekhar, G. Takacs, D. Chen, S. S. Tsai, J. Singh, and B. Girod, "Transform coding of image feature descriptors," in *Visual Communications and Image Processing*, vol. 7257, M. Rabbani and R. L. Stevenson, Eds. Bellingham, WA, USA: SPIE, 2009, pp. 725–710.

[16] M. Johnson, "Generalized descriptor compression for storage and matching," in *Proc. 21st Brit. Mach. Vis. Conf.*, 2010, pp. 23.1–23.11.

[17] M. Stommel, "Binarising SIFT-descriptors to reduce the curse of dimensionality in histogram-based object recognition," *Int. J. Signal Image Process.*, vol. 3, no. 1, pp. 25–36, Mar. 2010.

[18] C. Yeo, P. Ahammad, and K. Ramchandran, "Rate-efficient visual correspondences using random projections," in *Proc. 15th IEEE Int. Conf. Image Process.*, San Diego, CA, USA, Oct. 2008, pp. 217–220.

[19] V. Chandrasekhar *et al.*, "Compressed histogram of gradients: A low-bitrate descriptor," *Int. J. Comput. Vis.*, vol. 96, no. 3, pp. 384–399, Feb. 2012.

[20] G. Zhao, L. Chen, G. Chen, and J. Yuan, "KPB-SIFT: A compact local feature descriptor," in *Proc. Int. Conf. Multimedia*, Firenze, Italy, 2010, pp. 1175–1178.

[21] L. Ledwich and S. Williams, "Reduced sift features for image retrieval and indoor localisation," in *Proc. Austral. Conf. Robot. Autom.*, Camberra, Australia, Dec. 2004, pp. 1–3.

[22] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Washington, DC, USA, Jun. 2004, pp. II–506–II–513.

[23] MPEG, Denver, DC, USA. *Compact Descriptors for Visual Search* [Online]. Available: http://mpeg.chiariglione.org/standards/mpeg-7/compact-descriptors-visual-search

[24] V. Chandrasekhar *et al.*, "Survey of SIFT compression schemes," in *Proc. 2nd Int. Workshop Mobile Multimedia Process.*, 2010, pp. 1–8.

[25] M. Diephuis, S. Voloshynovskiy, O. Koval, and F. Beekhof, "Statistical analysis of binarized SIFT descriptors," in *Proc. 7th Int. Symp. Image Signal Process. Anal.*, Dubrovnik, Croatia, Sep. 2011, pp. 460–465.

[26] J. Chen, L. Duan, R. Ji, H. Yao, and W. Gao, "Sorting local descriptors for lowbit rate mobile visual search," in *Proc. 36th IEEE Int. Conf. Acoust., Speech, Signal Process.*, Prague, Czech Republic, May 2011, pp. 1029–1032.

[27] A. Redondi, M. Cesana, and M. Tagliasacchi, "Low bitrate coding schemes for local image descriptors," in *Proc. 14th IEEE Int. Workshop Multimedia Signal Process.*, Banff, Canada, Sep. 2012, pp. 124–129.

[28] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Comput. Society Conf. Comput. Vis. Pattern Recognit.*, vol. 2. New York, NY, USA, 2006, pp. 2161–2168.

[29] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[30] A. Redondi, L. Baroffio, M. Cesana, and M. Tagliasacchi, "Compress-then-analyze vs. analyze-then-compress: Two paradigms for image analysis in visual sensor networks," in *Proc. 15th IEEE Int. Workshop Multimedia Signal Process.*, Pula, Italy, Sep. 2013, pp. 278–282.

[31] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Washington, DC, USA, Oct. 2003, pp. 1470–1477.

[32] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.

[33] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. CVPR*, Jun. 2007, pp. 1–8.

[34] Y. Wu, S. Lu, T. Mei, J. Zhang, and S. Li, "Local visual words coding for low bit rate mobile visual search," in *Proc. 20th ACM Int. Conf. Multimedia*, Nara, Japan, Oct. 2012, pp. 989–992.

[35] E. Kokiopoulou and P. Frossard, "Semantic coding by supervised dimensionality reduction," *IEEE Trans. Multimedia*, vol. 10, no. 5, pp. 806–818, Aug. 2008.

[36] B. Moghaddam and A. Pentland, "An automatic system for model-based coding of faces," in *Proc. Data Compress. Conf.*, Snowbird, UT, USA, Mar. 1995, pp. 362–370.

[37] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *Comput. Res. Repository*, Apr. 2000, arXiv:physics/0004057.

[38] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *Int. J. Comput. Vis.*, vol. 60, no. 1, pp. 63–86, Oct. 2004.

[39] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communi. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[40] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. 10th Eur. Conf. Comput. Vis.*, Marseille, France, Oct. 2008, pp. 304–317.

[41] S. S. Tsai, D. Chen, G. Takacs, V. Chandrasekhar, J. P. Singh, and B. Girod, "Location coding for mobile image retrieval," in *Proc. 5th Int. ICST Mobile Multimedia Commun. Conf.*, London, U.K., 2009, pp. 8:1–8:7.

[42] S. S. Tsai *et al.*, "Improved coding for image feature location information," *Proc. SPIE*, vol. 8499, pp. 1–10, Oct. 2012.

[43] G. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

[44] S. Gauglitz, T. Höllerer, and M. Turk, "Evaluation of interest point detectors and feature descriptors for visual tracking," *Int. J. Comput. Vis.*, vol. 94, no. 3, pp. 335–360, 2011.

[45] A. Vedaldi and B. Fulkerson. (2008). *VLFeat: An Open and Portable Library of Computer Vision Algorithms* [Online]. Available: http://www.vlfeat.org/

[46] C. Evans, "Notes on the opensurf library," Dept. Comput. Sci., Univ. Bristol, Bristol, U.K., Tech. Rep. CSTR-09-001, Jan. 2009.

[47] K. Mikolajczyk *et al.*, "A comparison of affine region detectors," *Int. J. Comput. Vis.*, vol. 65, nos. 1–2, pp. 43–72, 2005.

[48] H. Feng and M. Effros, "On the rate-distortion performance and computational efficiency of the Karhunen-Loeve transform for lossy data compression," *IEEE Trans. Image Process.*, vol. 11, no. 2, pp. 113–122, Feb. 2002.

[49] S. Leutenegger, M. Chli, and R. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Proc. 13th IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 2548–2555.

[50] R. Ortiz, "Freak: Fast retina keypoint," in *Proc. 25th IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 510–517.

[51] T. Trzcinski and V. Lepetit, "Efficient discriminative projections for compact binary descriptors," in *Proc. 12th Eur. Conf. Comput. Vis.*, Firenze, Italy, Oct. 2012, pp. 228–242.

[52] A. Redondi, L. Baroffio, J. Ascenso, M. Cesana, and M. Tagliasacchi, "Rate-accuracy optimization of binary descriptors," in *Proc. 20th IEEE Int. Conf. Image Process.*, Melbourne, Australia, Sep. 2013, pp. 2910–2914.