Coding local and global binary visual features extracted from video sequences

Luca Baroffio, Antonio Canclini, Matteo Cesana, Alessandro Redondi, Marco Tagliasacchi, Stefano Tubaro

***Abstract*—Binary local features represent an effective alternative to real-valued descriptors, leading to comparable results for many visual analysis tasks, while being characterized by significantly lower computational complexity and memory requirements. When dealing with large collections, a more compact representation based on global features is often preferred, which can be obtained from local features by means of, e.g., the Bag-of-Visual Word (BoVW) model. Several applications, including for example visual sensor networks and mobile augmented reality, require visual features to be transmitted over a bandwidth-limited network, thus calling for coding techniques that aim at reducing the required bit budget, while attaining a target level of efficiency. In this paper we investigate a coding scheme tailored to both local and global binary features, which aims at exploiting both spatial and temporal redundancy by means of intra- and inter-frame coding. In this respect, the proposed coding scheme can be conveniently adopted to support the *"Analyze-Then-Compress"* (`ATC`) paradigm. That is, visual features are extracted from the acquired content, encoded at remote nodes, and finally transmitted to a central controller that performs visual analysis. This is in contrast with the traditional approach, in which visual content is acquired at a node, compressed and then sent to a central unit for further processing, according to the *"Compress-Then-Analyze"* (`CTA`) paradigm. In this paper we experimentally compare `ATC` and `CTA` by means of rate-efficiency curves in the context of two different visual analysis tasks: homography estimation and content-based retrieval. Our results show that the novel `ATC` paradigm based on the proposed coding primitives can be competitive with `CTA`, especially in bandwidth limited scenarios.**

***Index Terms*—Visual features, binary descriptors, BRISK, Bag-of-Words, video coding.**

I. Introduction

Visual analysis is often performed extracting a feature-based representation from the raw pixel domain. Indeed, visual features are being successfully exploited in a broad range of visual analysis tasks, ranging from image/video retrieval and classification, to object tracking and image registration. They provide a succinct, yet effective, representation of the visual content, while being invariant to many transformations.

Several visual analysis applications (e.g., distributed monitoring and surveillance in visual sensor networks, mobile visual search and augmented reality, etc.) require visual content to be transmitted over a bandwidth-limited network. The traditional approach, denoted hereinafter as *"Compress-Then-Analyze"* (`CTA`), consists in the following steps: the visual content is acquired by a sensor node in the form of still images or video sequences; then, it is encoded and efficiently transmitted to a central unit where visual feature extraction and analysis takes place. The central unit relies on a lossy representation of the acquired content, potentially leading to impaired performance. Furthermore, such a paradigm might lead to an inefficient management of bandwidth and storage resources, since a complete pixel-level representation might be unnecessary.

In this respect, *"Analyze-Then-Compress"* (`ATC`) represents an alternative approach to visual analysis in a networked scenario. Such a paradigm aims at moving part of the analysis from the central unit directly to sensing nodes. In particular, nodes process visual content in order to extract relevant information in the form of visual features. Then, such information is compressed and sent to a central unit, where visual analysis takes place. The key tenet is that the rate necessary to encode visual features in `ATC` might be less than the rate needed for the original visual content in `CTA`, when targeting the same level of efficiency in the visual analysis. This is particularly relevant in those applications in which visual analysis requires access to video sequences. Therefore, in order to maximize the rate saving, it is necessary to carefully select suitable visual features and design efficient coding schemes.

In this paper we consider the problem of encoding both local and global binary features extracted from video sequences. The choice of this class of visual features is well motivated from different standpoints [2]. First, binary features are significantly faster to compute than real-valued features such as SIFT [3] or SURF [4], thus being suitable whenever energy resources are an issue, such as in the case of low-power devices, where they constitute the only available option. Second, binary features have been recently shown to deliver performance close to state-of-the-art real-valued features. Third, they can be compactly represented and coded with just a few bits [5]. Forth, binary features are faster to match, thus being suitable when dealing with large scale collections.

The processing pipeline for the extraction of local features comprises: i) a keypoint detector, which is responsible for the identification of a set of salient keypoints within an image, and ii) a keypoint descriptor, which assigns a description vector to each identified keypoint, based on the local image content. Within the class of local binary descriptors, BRIEF [6] computes the descriptor elements as the result of pairwise comparisons between (smoothed) pixel intensity values that

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy, email: name.surname@polimi.it

The project GreenEyes acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 296676.

The material in this paper has been partially presented in [1].

are randomly sampled from the neighborhood of a keypoint. BRISK [7], FREAK [8] and ORB [9] are inspired by BRIEF, and similarly to their predecessor, are also based on pairwise pixel intensity comparisons. They differ from each other in the way pixel pairs are spatially sampled in the image patch surrounding a given keypoint In particular, they introduce ad-hoc spatial patterns that define the location of the pixels to be compared. Furthermore, differently from BRIEF, they are designed so that the generated binary descriptors are scale- and rotation-invariant. More recently, in order to bridge the gap between binary and real-valued descriptors, BAMBOO [10][11] adopts a richer dictionary of pixel intensity comparisons, and selects the most discriminative ones by means of a boosting algorithm. This leads to a matching accuracy similar to SIFT, while being 50x faster to compute. A similar idea is also exploited by BinBoost [12], which proposes a boosted binary descriptor based on a set of local gradients. BinBoost is shown to deliver state-of-the-art matching accuracy, at the cost of a computational complexity comparable to that of real-valued descriptors such as SIFT or SURF.

On the other hand, global features represent a suitable alternative to local features when considering scenarios in which very large amounts of data have to be processed, stored and matched. Global features computed by summarizing local features into a fixed-dimensional feature vector have been effectively employed in the context of large scale image and video retrieval [13]. Global features can be computed based on the Bag-of-Visual-Words (BoVW) [14] model, which is inspired by traditional text-based retrieval. VLAD [15] and Fisher Vectors [16] represent more sophisticated approaches that achieve improved compactness and matching performance. More recently, the problem of building global features starting from sets of binary features was addressed in [17] and [18], extending, respectively, the BoVW and VLAD model to the case of local binary features. Solutions based on global image descriptors offer a good compromise between efficiency and accuracy, especially considering large scale image retrieval and classification. Nonetheless, local features still play a fundamental role, being usually employed to refine the results of such tasks [19] [14]. Furthermore, the approaches based on global features disregard the spatial configuration of the keypoints, preventing the use of spatial verification mechanism and thus being unsuitable to tracking and structure-from-motion scenarios [20], [21].

This paper proposes a number of novel contributions:

1) We consider the problem of coding local binary features extracted from video sequences, by exploiting both intra- and inter-frame coding. In this respect, we adopt the general architecture of our previous work [22], which targeted real-valued features, and propose coding tools specifically devised for binary features.
2) For the first time, we consider the problem of coding global binary features extracted from video sequences, obtained by summarizing local features according to the BoVW model, exploiting both intra- and inter-frame coding.
3) We evaluate the proposed coding scheme in terms of rate-efficiency curves for two different visual analysis tasks: homography estimation and content-based retrieval. We show the impact of the main configuration parameters, namely, the number of keypoints, descriptor elements and visual words. Unlike our previous work, content-based retrieval is evaluated by means of a complete image retrieval pipeline, in which a video is used to query an image database.
4) We compare the overall performance of `ATC` vs. `CTA` for both analysis tasks. In the case of homography estimation, we show that `ATC` based on local features always outperforms `CTA` by a large margin. In the case of content-based retrieval, we show the `ATC` achieves a significantly lower bitrate than `CTA` when using global features, while it is on a par with `CTA` when using local features.

In the context of local visual features, several past works tackled the problem of compressing both real-valued and binary local features extracted from still images. As for real-valued local features, architectures based on closed-loop predictive coding [23], transform coding [24][25] and hashing [26] were proposed. In this context, an ad-hoc MPEG group on Compact Descriptors for Visual Search (CDVS) has been working towards the definition of a standard [27] that relies on SIFT features. As for binary local features, predictive coding architectures aimed at exploiting either inter-descriptor correlation [28] or intra-descriptor redundancy [29] were proposed. Furthermore, Monteiro et al. proposed a clustering based coding architecture tailored to the context of binary descriptors [30]. Moreover, some works aimed at modifying traditional extraction algorithms, so that the output data is more compact or more suitable for compression. In this context, CHOG [31] is a gradient-based descriptor that offers performance comparable to that of SIFT at a much lower bitrate. As an alternative approach, Chao et al. [32] studied how to adjust the JPEG quantization matrix in order to preserve local features extracted from decoded images.

The problem of encoding visual features extracted from video content has been addressed only very recently. Makar et al. [33], [34] propose to encode and transmit temporally coherent image patches in the pixel-domain, for augmented reality and image recognition applications. Thus, the detector is applied at the transmitter side, while the descriptors are extracted from decoded patches at the receiver. The encoding of local features (both keypoint locations and descriptors) extracted from video sequences was addressed for the first time in [35] for the case of real-valued features (SIFT and SURF) and later extended in [22]. To the best of the authors' knowledge, the encoding of streams of binary features has not been addressed in the previous literature. Furthermore, the interest of the scientific community in this kind of problem is witnessed by the creation of a new MPEG ad-hoc group, namely Compact Descriptors for Video Analysis (CDVA), which has recently started its activities [36]. CDVA targets the standardization of the extraction and coding of visual features in application scenarios ranging from video retrieval, automotive, surveillance, industrial monitoring, etc., in which video, rather than images, plays a key role.

The rest of this paper is organized as follows. Section II

states the problem of coding sets of local binary descriptors, defining the properties of the features to be coded, whereas Section III illustrates the coding architecture. Section IV introduces the problem of coding Bag-of-Visual-Words extracted from a video sequence and Section V defines the coding algorithms. Section VI is devoted to defining the experimental setup and reporting the results. Finally, conclusions are drawn in Section VII.

II. Coding local features: problem statement

Let $\mathcal{I}_n$ denote the n-th frame of a video sequence, which is processed to extract a set of local features $\mathcal{D}_n$. First, a keypoint detector is applied to identify a set of interest points. Then, a descriptor is applied on the (rotated) patches surrounding each keypoint. Hence, each element of $d_{n,i} \in \mathcal{D}_n$ 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 *binary* vector $\mathbf{d}_{n,i} \in \{0,1\}^P$, which represents the descriptor associated to the keypoint $\mathbf{p}_{n,i}$.

We propose a coding architecture which aims at efficiently coding the sequence $\{\mathcal{D}_n\}_{n=1}^N$ of sets of local features. In particular, we consider both lossless and lossy coding schemes: in the former, the binary description vectors are preserved throughout the coding process, whereas in the latter only a subset of $K < P$ descriptor elements is lossless coded, thus discarding a part of the original data. Each decoded descriptor can be written as $\tilde{d}_{n,i} = \{\tilde{\mathbf{p}}_{n,i}, \tilde{\mathbf{d}}_{n,i}\}$. The number of bits necessary to encode the M_n visual features extracted from frame $\mathcal{I}_n$ is equal to

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

That is, we consider the rate used to represent both the location of the keypoint, $R^p_{n,i}$, and the descriptor itself, $R^d_{n,i}$. For both the lossless and the lossy approach, no distortion is introduced during the coding process in the received descriptor elements. Nonetheless, since in the lossy case part of the descriptor elements are discarded, the accuracy of the visual analysis task might be affected.

As for the component $\tilde{\mathbf{p}}_{n,i}$, we decided to encode the coordinates of the keypoint, the scale and the local orientation i.e., $\tilde{\mathbf{p}}_{n,i} = [\tilde{x}, \tilde{y}, \tilde{\sigma}, \tilde{\theta}]^T$. Although some visual analysis tasks might not require this information, it could be used to refine the final results. For example, it is necessary when the matching score between image pairs is computed based on the number of matches that pass the spatial verification step using, e.g., RANSAC [19] or weak geometry checking [20]. Most of the detectors produce floating point values as keypoint coordinates, scale and orientation, thanks to interpolation mechanisms. Nonetheless, we decided to round such values with a quantization step size equal to 1/4 for the coordinates and the scale, and $\pi/16$ for the orientation, which has been found to be sufficient for typical applications [35], [22].

III. Coding local features: algorithms

Figure 1 illustrates a block diagram of the proposed coding architecture. The scheme is similar to the one we recently proposed for encoding real-valued visual features [35], [22]. However, we highlighted the functional modules that needed to be revisited due to the binary nature of the source.

A. Intra-frame coding

In the case of intra-frame coding, local features are extracted and encoded separately for each frame. In our previous work we proposed an intra-frame coding approach tailored to binary descriptors extracted from still images [5], which is briefly summarized in the following. In binary descriptors, each element represents the binary outcome of a pairwise comparison. The descriptor elements (dexels) are statistically dependent, and it is possible to model the descriptor as a binary source with memory.

Let π_j, $j \in [1, P]$ represent the j-th element of a binary descriptor $\mathbf{d} \in \{0,1\}^P$. The entropy of such a dexel can be computed as

$$H(\pi_j) = -p_j(0)\log_2(p_j(0)) - p_j(1)\log_2(p_j(1)), \tag{2}$$

where $p_j(0)$ and $p_j(1)$ are the probability of $\pi_j = 0$ and $\pi_j = 1$, respectively. Similarly, the conditional entropy of dexel π_{j_1} given dexel π_{j_2} can be computed as

$$H(\pi_{j_1}|\pi_{j_2}) = \sum_{x\in\{0,1\}, y\in\{0,1\}} p_{j_1,j_2}(x,y)\log_2 \frac{p_{j_2}(y)}{p_{j_1,j_2}(x,y)}, \tag{3}$$

with $j_1, j_2 \in [1, P]$. Let $\tilde{\pi}_j$, $j = 1, \ldots, P$, denote a permutation of the dexels, indicating the sequential order used to encode a descriptor. The average code length needed to encode a descriptor is lower bounded by

$$R = \sum_{j=1}^{P} H(\tilde{\pi}_j|\tilde{\pi}_{j-1}, \ldots, \tilde{\pi}_1). \tag{4}$$

In order to maximize the coding efficiency, we aim at finding the permutation of dexels $\tilde{\pi}_1, \ldots, \tilde{\pi}_P$ that minimizes such a lower bound. For the sake of simplicity, we model the source as a first-order Markov source. That is, we impose $H(\tilde{\pi}_j|\tilde{\pi}_{j-1}, \ldots \tilde{\pi}_1) = H(\tilde{\pi}_j|\tilde{\pi}_{j-1})$. Then, we adopt the following greedy strategy to reorder the dexels:

$$\tilde{\pi}_j = \begin{cases} \arg\min_{\pi_j} H(\pi_j) & j = 1 \\ \arg\min_{\pi_j} H(\pi_j|\tilde{\pi}_{j-1}) & j \in [2, P] \end{cases} \tag{5}$$

The reordering of the dexel is described by means of a permutation matrix $\mathbf{T}^{\text{INTRA}}$, such that $\tilde{\mathbf{c}}_{n,i} = \mathbf{T}^{\text{INTRA}}\mathbf{d}_{n,i}$. Note that such optimal ordering is computed offline, thanks to a training phase, and shared between both the encoder and the decoder. As such, this does not require additional bitrate.

B. Inter-frame coding

As for inter-frame coding, each set of local features $\mathcal{D}_n$ is coded resorting to a reference set of features. In this work we consider as a reference the set of features extracted from the previous frame, i.e., $\mathcal{D}_{n-1}$. Considering a descriptor $d_{n,i}$,

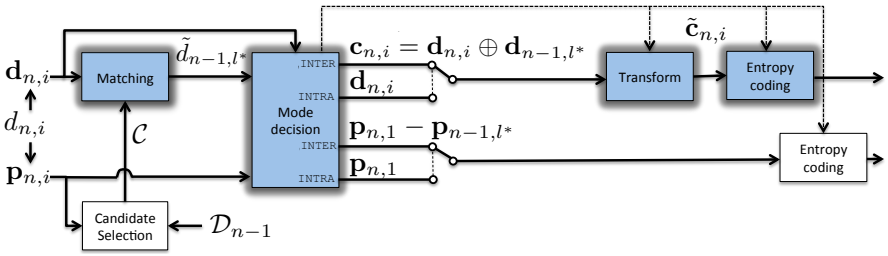


Fig. 1. Block diagram of the proposed coding architecture. The highlighted functional modules needed to be revisited due to the binary nature of the source.

$i = 1, \ldots, M_n$, the encoding process consists in the following steps:

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

$$\mathbf{d}_{n-1,l^*} = \arg\min_{l \in \mathcal{C}} D(\mathbf{d}_{n,i}, \mathbf{d}_{n-1,l}) + \lambda R_{n,i}^{p,\text{INTER}}(l), \quad (6)$$

  where $D(\mathbf{d}_{n,i}, \mathbf{d}_{n-1,l}) = \|\mathbf{d}_{n,i} - \mathbf{d}_{n-1,l}\|_0$ is the Hamming distance between the descriptors $\mathbf{d}_{n,i}$ and $\mathbf{d}_{n-1,l}$, $R_{n,i}^{p,\text{INTER}}(l)$ is the rate needed to encode the keypoint motion vector and l^* is the index of the selected reference feature used in the next steps. We limit the search for a reference feature within a given set $\mathcal{C}$ of candidate features, i.e., the ones whose coordinates and scales are in the neighborhood of $d_{n,i}$, in a range of $(\pm\Delta x, \pm\Delta y, \pm\Delta\sigma)$. The prediction residual is computed as $\mathbf{c}_{n,i} = \mathbf{d}_{n,i} \oplus \mathbf{d}_{n-1,l^*}$, that is, the bitwise XOR between $\mathbf{d}_{n,i}$ and $\mathbf{d}_{n-1,l^*}$.
- *Coding mode decision*: Compare the cost of inter-frame coding with that of intra-frame coding, which can be expressed as

$$J^{\text{INTRA}}(d_{n,i}) = R_{n,i}^{p,\text{INTRA}} + R_{n,i}^{d,\text{INTRA}}, \quad (7)$$

$$J^{\text{INTER}}(d_{n,i}, \tilde{d}_{n-1,l^*}) = R_{n,i}^{p,\text{INTER}}(l^*) + R_{n,i}^{d,\text{INTER}}(l^*), \quad (8)$$

  where $R_{n,i}^{p}$ and $R_{n,i}^{d}$ represent the bitrate needed to encode the location component (either the location itself or location displacement) and the one needed to encode the descriptor component (either the descriptor itself or the prediction residual), respectively. If $J^{\text{INTER}}(\mathbf{d}_{n,i}, \mathbf{d}_{n-1,l^*}) < J^{\text{INTRA}}(\mathbf{d}_{n,i})$, then inter-frame coding is the selected mode. Otherwise, proceed with intra-frame coding.
- *Intra-descriptor transform*: This step aims at exploiting the spatial correlation between the dexels. If intra-frame is the selected coding mode, then the dexels of $\mathbf{d}_{n,i}$ are reordered according to the permutation algorithm presented in Section III-A. Similarly, a reordering strategy can be applied also in the case of inter-frame coding, in this case considering the prediction residual $\mathbf{c}_{n,i}$, that is, $\tilde{\mathbf{c}}_{n,i} = \mathbf{T}^{\text{INTER}}\mathbf{c}_{n,i}$. Note that, in general, $\mathbf{T}^{\text{INTER}} \neq \mathbf{T}^{\text{INTRA}}$
- *Entropy coding*: Finally, the sets of local features are entropy coded. In the case of intra-frame coding, for each local feature, it is necessary to encode the reordered descriptor and the quantized location component. Otherwise, for inter-frame coding, it is necessary to encode: i) the identifier of the matching keypoint in the reference frame and the displacement in terms of position, scale and orientation of the keypoint with respect to the reference, which require $R_{n,i}^{p,\text{INTER}}(l^*)$ bits; ii) the reordered prediction residual $\tilde{\mathbf{c}}_{n,i}$.
  For both intra-frame and inter-frame coding, the probabilities of the symbols (respectively, descriptor elements or prediction residuals) used for entropy coding are learned from a training set of frames. In particular, for each of the P dexels, we estimated the conditional probability of each symbol, given the previous one defined by the optimal permutation. The estimated probabilities are then exploited to entropy code the features.

C. Descriptor element selection

The lossless coding architecture described in the previous section can be used to encode all the P elements of the original binary descriptor. However, in order to operate at lower bitrates, it is possible to decide to code only a subset of $K < P$ descriptor elements. In our previous work we explored different methods that define how to select the dexels to be retained [5], [10], [11]. In this work, we employed the greedy asymmetric pairwise boosting algorithm described in [11] in order to iteratively select the most discriminative descriptor elements. To this end, we used a training set of image patches [37], along with the ground truth information defining whether two image patches refer to the same physical entity. At each step, the asymmetric pairwise boosting algorithm selects the dexel that minimizes a cost function, which captures the error resulting from the wrong classification of matching and non-matching patches. The output of this procedure is a set of dexels, ordered according to their discriminability. Hence, given a target descriptor size $K < P$, it is possible to encode only the first K descriptor elements selected by this algorithm.

IV. Global descriptors based on binary visual features

Let $\mathcal{I}_n$ denote the n-th frame of a video sequence, which is processed to extract a set of local features $\mathcal{D}_n$. A global representation for the frame $\mathcal{I}_n$ can be computed, starting from such set of local image descriptors. The key idea behind the Bag-of-Visual-Words (BoVW) approach is to quantize each local feature into one visual word. To this end, a vocabulary $\mathcal{V} = \{\mathbf{w}_1, \ldots, \mathbf{w}_V\}$ composed of V distinct visual words has to be computed. Traditional approaches to the creation of BoVW models are based on real-valued local descriptors

such as SIFT [3] or SURF [4]. In this context, a large set of descriptors $\mathbf{d} \in \mathbb{R}^K$ is exploited for learning the vocabulary, along with a clustering algorithm such as k-means (with $k = V$) based on Euclidean distance [14], Gaussian Mixture Model [38], etc.

More recently, the problem of constructing a BoVW model starting from sets of binary local descriptors was addressed in [39]. Analogously to the case of real-valued descriptors, a dictionary is learned starting from a large set of descriptors $\mathbf{d} \in \{0,1\}^K$. To this end, a naive approach would consist in k-means clustering paired with Euclidean distance [40]. Besides, clustering techniques tailored to the peculiar nature of the signal at hand have been introduced. In particular, k-medoids and k-medians algorithms, paired with Hamming distance, have been successfully exploited for creating the dictionary [39].

Then, given a vocabulary that consists of V of visual words $\mathcal{V} = \{\mathbf{w}_1, \ldots, \mathbf{w}_V\}$ learned offline and a set of visual features $\mathcal{D}_n$ extracted from the frame $\mathcal{I}_n$, a global descriptor is obtained by mapping such a set of features to a fixed-dimensional vector $\mathbf{v}_n \in \mathbb{R}^V$. The simplest strategy is to apply hard quantization, which assigns each feature $\mathbf{d} \in \mathcal{D}_n$ to the nearest visual word's centroid $\mathbf{w}_j \in \mathcal{V}$. The resulting global descriptor $\mathbf{v}_n = [v_1, \ldots, v_V]^T$ is a histogram, where v_j represents the number of local features in $\mathcal{D}_n$ having the dictionary word $\mathbf{w}_j$ as nearest neighbor. Soft quantization represents a more sophisticated alternative to hard quantization, mapping each local feature to multiple visual words. Finally, several techniques for normalizing the global descriptor $\mathbf{v}_n$ have been proposed, aimed at improving matching performance. A widely accepted approach consists in adopting the tf-idf weighting scheme, followed by L_2 normalization [41]. The former gives more emphasis to rare visual words and less importance to common ones, whereas the latter avoids short vectors, i.e. BoVWs built starting from few local features, to be penalized during the matching stage.

V. Coding global features

For each frame $\mathcal{I}_n$ of an input video sequence, a set of binary local features $\mathcal{D}_n$ is extracted and mapped to a V-dimensional global descriptor $\mathbf{v}_n = [v_1, \ldots, v_V]^T$ by applying the procedure described in Section IV. We propose a coding architecture which aims at effectively encoding the sequence $\{\mathbf{v}_n\}_{n=1}^N$ of global image descriptors. In particular, such a lossy coding architecture enables the decoder to reconstruct an approximation $\{\tilde{\mathbf{v}}_n\}_{n=1}^N$ of the original sequence of global descriptors.

Differently from the case of local descriptors, the coordinates of the keypoints are disregarded during the construction of the BoVW and they are not encoded. Hence, the number of bits needed to encode a Bag-of-Visual-Words $\mathbf{v}_n$ extracted from frame $\mathcal{I}_n$ is equal to $R_n = \sum_{j=1}^V R_{n,j}^{\mathbf{v}}$, i.e., the sum of the number of bits needed to encode each component of the vector $\mathbf{v}_n$.

A. Intra-frame coding

The Intra-frame coding approach is based on a frame-by-frame processing scheme, in which the global descriptor extracted from the frame $\mathcal{I}_n$ is encoded independently from the ones extracted from other frames. Considering a baseline architecture, uniform scalar quantization with step size Δ_j is applied to each element $v_{n,j}, j = 1, \ldots, V$ of the global descriptor $\mathbf{v}_n$, that is

$$\tilde{q}_{n,j} = \left\lfloor \frac{v_{n,j}}{\Delta_j} \right\rfloor. \tag{9}$$

Since the vectors are normalized according to a tf-idf weighting scheme, the same quantization step size $\Delta_j = \Delta,\ j = 1, \ldots, V$, is fixed for each visual word.

The quantization index $\tilde{q}_{n,j}$ is considered as the outcome of a discrete memoryless source and entropy coded. To this end, the probabilities of the quantization symbols are estimated offline and fed to an arithmetic coder, so that the corresponding rate is equal to $R_{n,j}^{\mathbf{v}} = -\log_2(p(\tilde{q}_{n,j}))$.

B. Inter-frame coding

In the case of inter-frame coding, local features are extracted on a frame-by-frame basis and quantized into BoVWs in order to obtain a sequence of global descriptors $\{\mathbf{v}_n\}_{n=1}^N$. Then, differently from intra-frame coding, temporal redundancy is exploited in the coding phase: the global descriptor $\mathbf{v}_n$ extracted from frame $\mathcal{I}_n$ is encoded using $\mathbf{v}_{n-1}$ as reference. In particular, each descriptor element $v_{n,j}$ is encoded having $v_{n-1,j}$ as context.

To this end, considering a quantization step size Δ_j, the quantization symbols $\tilde{q}_{n,j}$ are obtained according to Equation (9), and then entropy coded using R_n bits. Similarly to the case of intra-frame coding, the statistics of the quantization symbols are estimated at training time. In particular, given a sufficiently large sequence of training global descriptors, a training phase aims at estimating the probabilities $p(q_{n,j} = X_p | q_{n-1,j} = X_q)$, i.e. the probabilities of the j-th dexel at frame $\mathcal{I}_n$ assuming value X_p, given the j-th dexel at frame $\mathcal{I}_{n-1}$ having value X_q. An arithmetic coder is used to entropy code the quantization symbols, with an expected number of bits that amounts to

$$R_n = \sum_{j=1}^{M} R_{n,j}^{\mathbf{v}} = \sum_{j=1}^{M} \log_2(p(q_{n,j} = \tilde{q}_{n,j} | q_{n-1,j} = \tilde{q}_{n-1,j})). \tag{10}$$

VI. Experiments

We validated the effectiveness of the feature coding architectures and compared the two different paradigms, namely "Analyze-Then-Compress" (ATC) and "Compress-Then-Analyze" (CTA), on two traditional visual analysis tasks:

- *Homography estimation*. Several high- and low-level visual analysis tasks, including camera calibration, 3D reconstruction, structure-from-motion, tracking, etc. might require the estimation of the homography defining the geometrical relationship between two frames with homogeneous visual content. In this scenario, local features can be conveniently used to find correspondences between pixel locations in different frames or views. Conversely,

global features based on BoVW do not represent a viable option, since they do not include any geometrical information about the visual content.

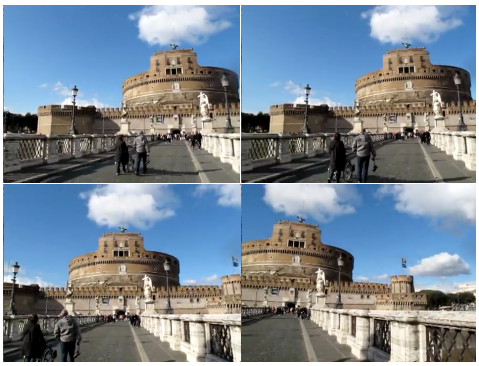
- *Content Based Retrieval* (CBR). Content Based Retrieval is a traditional, yet challenging, task within the computer vision community. Given an input query in the form of some kind of visual content, the goal is to retrieve the relevant multimedia documents within a large database. Accuracy and computational efficiency are key tenets to be considered when implementing algorithms for CBR, which typically target large scale scenarios. Our test considers an input query in the form of a video clip, with the goal of retrieving the most relevant database images. In this scenario, both global and local features are considers, in order to explore a trade-off between accuracy and computational efficiency.

A. Data sets

1) Training data sets: The methods discussed for encoding binary local descriptors require the knowledge of the probabilities of the symbols to operate the entropy coder, which were estimated from training sequences. To this end, we employed three video sequences, namely *Mother*, *News* and *Paris* [42]. The training video sequences were also exploited to obtain the optimal coding order of dexels for both intra- and inter-frame coding, as illustrated in Section III. Furthermore, a dataset of patches [37] was exploited along with an asymmetric pairwise boosting algorithm [11] [10] in order to identify the K most discriminative dexels according to the method presented in Section III-C.

In the case of BoVW-based global descriptors, the visual word dictionary was estimated exploiting a large database of images, namely *VOC2010* [43], whereas the statistics of the coding symbols for both intra- and inter-frame coding architectures were estimated offline, resorting to a sufficiently long video sequence, namely *Rome in a nutshell*, which consists of 15375 frames.

2) Test data sets: First, the coding architecture was evaluated on three video sequences, namely *Hall*, *Mobile* and *Foreman*, to investigate the bitrate saving which can be obtained by properly encoding the binary features. Then, for the *Homography Estimation* test, we used a publicly available dataset for visual tracking [44], consisting in a set of video sequences, each containing a planar texture subject to a given motion path. For each frame of each sequence, the homography that warps such frame to the reference is provided as ground truth. The sequences have a resolution of 640 $\times$ 480 pixels at 15 fps and a length of 500 frames (33.3 seconds). Finally, for the *Content Based Retrieval* (CBR) test, a set of 10 query video sequences was used, each capturing a different landmark in the city of Rome with a camera embedded in a mobile device [45]. The frame rate of such sequences is equal to 24fps, whereas the resolution ranges from 480x360 pixels (4:3) to 640x360 pixels (16:9). The first 50 frames of each video were used as query. On average, each query video corresponds to 9 relevant images representing the same physical object under different conditions and with heterogeneous qualities and resolutions. Then, distractor images randomly sampled from the *MIRFLICKR-1M* dataset, so that the final database contains 10k images. As an example, Figure 2 shows some frames of a query sequence, along with a relevant image to be retrieved. The dataset is publicly available for download at XX.



(a)

(b)

Fig. 2. a) Four frames sampled from one of the query videos employed for the retrieval task. b) A matching database image.

B. Methods

1) ATC-Training: The proposed coding architecture can be applied to any kind of local binary feature. Hence, in our experiments we evaluated the use of two different state-of-the-art binary features, namely BRISK [7] and BINBOOST [12]. The detection threshold was set equal to 70 for both BRISK and BINBOOST. All other parameters were left equal to their default values. In both cases, the parameters x, y and σ, representing the location and the scale of each keypoint, were rounded to the nearest quarter of unit. Descriptors consisting of $P = 512$ and $P = 256$ dexels for BRISK and BINBOOST, respectively, were extracted from the training video sequences, using the original implementations of the feature extraction algorithms provided by authors.

As for BRISK, we considered a set of target descriptor sizes $K = \{512, 256, 128, 64, 32, 16, 8\}$. For each size, we employed the dexel selection algorithm presented in Section III-C in order to identify the elements to be retained. Then, in the case of intra-frame coding and for each descriptor length, the optimal coding order and the corresponding coding probabilities were estimated according to the procedure

introduced in Section III-A. Instead, in the case of inter-frame coding, for each descriptor a match was found within the features extracted from the previous frame, according to the method presented in Section III-B. Similarly to the case of intra-frame coding, a coding-wise optimal permutation of the elements of the binary prediction residual was computed, and the corresponding coding probabilities were estimated. As to BINBOOST, we considered a set of target descriptor sizes $K = \{256, 128, 64, 32, 16, 8\}$. For each size, the first K dexels of the BINBOOST descriptor were retained. Then, similarly to the case of BRISK, coding-wise optimal dexel permutations for intra- and inter-frame coding were computed, along with the probabilities of the coding symbols.

In the case of BoVW-based global descriptors, we fixed a set of target sizes for the dictionary of visual words $M = \{1024, 4096, 16384\}$. Then, for each possible dictionary size, BRISK or BINBOOST descriptors were extracted from the training set of images and vector quantization was applied in order to identify M visual words. Both k-means and k-medians algorithms have been tested for the dictionary construction stage, yielding similar results in terms of rate-accuracy performance. Furthermore, global descriptors based on BRISK and BINBOOST local features achieve very similar results. In the following, we refer to the best performing setup, that is, k-means clustering applied to BRISK descriptors, initialized according to the k-means++ [46] algorithm, and Euclidean distance. The output of this first stage is a dictionary composed of M visual words each represented by a P-dimensional vector, where $P = 512$ ($P = 256$) is the size of BRISK descriptors (BINBOOST descriptors). Then, a training video sequence was adopted to compute the coding probabilities. For each frame, local features were extracted and the global descriptor was computed by hard assigning each feature to its nearest neighbor within the dictionary, according to the procedure presented in Section IV. Then, for each target quantization step size $\Delta = \{0.01, 0.05, 0.1, 0.2\}$, global descriptors were quantized and the coding probabilities for both intra- and inter-frame were computed according to the algorithms introduced in Section V.

Concerning the CBR test, a representation of each database image had to be computed, in the form of both a set of local features and a global descriptor. To this purpose, for each image a set of local features was extracted and stored. Furthermore, such a set was also exploited to compute a BoVW-based global descriptor that was stored, too.

2) CTA-Training: The *Compress-Then-Analyze* paradigm relies on traditional video compression, paired with state-of-the-art visual features extraction algorithms. In the case of local features, no training was needed. Instead, in the case of global features, a dictionary of visual words had to be learned in order to compute the BoVW representation of an image. The dimensionality of the dictionary was fixed to $M = 16384$ visual words and, similarly to the case of ATC paradigm, SIFT local features were extracted from the *VOC2010* dataset and clustered into M visual words, once again resorting to k-means (k-means++ initialization).

3) ATC-Testing: within the ATC paradigm, we distinguished between several different schemes:

- `BRISK/BINBOOST - INTRA`: all binary local features (either BRISK or BINBOOST) were encoded resorting to an intra-frame coding scheme.
- `BRISK/BINBOOST - INTER`: all binary local features were encoded resorting to an inter-frame coding scheme.
- `BRISK/BINBOOST - INTRA/INTER`: for each binary local feature, a 2-way coding mode decision module was used to select the best coding mode between INTRA and INTER.
- `BoVW - INTRA`: all global features were encoded resorting to an intra-frame coding scheme.
- `BoVW - INTER`: all global features were encoded resorting to an inter-frame coding scheme.

4) CTA-Testing: within the CTA paradigm, we distinguished between two different schemes:

- `SIFT - INTER`: visual content was encoded resorting to H.264/AVC coder, SIFT features were employed.
- `BoVW - INTER`: visual content was encoded resorting to H.264/AVC coder, BoVW-based global features were employed.

For the tests to be as fair as possible, the video coding scheme and the visual feature coding scheme were configured to 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.

C. Experiments and evaluation metrics

Each visual analysis task was evaluated according to an ad-hoc metric:

1) 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 following the `ATC` paradigm. For each pair of consecutive frames $\mathcal{I}_n$ and $\mathcal{I}_m$, a homography $\tilde{H}_{nm,\texttt{ATC},\Delta}$ 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 [21].

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 $\tilde{H}_{nm,\texttt{CTA},Q}$ 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*, which was adopted in our previous work [22] and briefly summarized here for completeness. 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 $\mathcal{I}_n$ were provided as ground truth. Applying the homography $\tilde{H}_{nm}$ to such points, it was possible to estimate the 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 $\mathcal{I}_m$ 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.

2) Content Based Retrieval: Considering ATC and given a query video sequence, a set of local features was extracted from each frame of the clip and mapped to a BoVW-based global descriptor, according to the procedure described in Section IV. The goal of the task is the retrieval of relevant images within a database consisting of Z elements. Considering traditional applications of CBR, database dimensionality Z ranges from thousands to millions. Hence, matching based on sets of local features might represent an inefficient, or even unfeasible, approach. On the other hand, global image descriptors represent an effective yet computationally efficient solution. Indeed, a two-step approach was proposed [14], which consist in i) retrieving the top-k relevant results within the database exploiting global descriptors and ii) refine the results of the previous step exploiting local features. Such an approach represents a good tradeoff between task accuracy and computational efficiency, since fast matching based on global features is exploited in order to identify a subset of possibly relevant documents, whereas an accurate re-ranking is performed on such a small subset of data, resorting to local visual features.

Considering the first stage of the pipeline, i.e. the retrieval of top-k relevant items, the global descriptor extracted from each frame of each test video sequence was matched against the global descriptors of all the database images. Due to the adoption of the weighting and normalization procedure described in Section IV, Euclidean distance was employed to compare pairs of global descriptors. Then, database images were ranked according to their distance with respect to the query, in increasing order. The top-k elements of the ranking are the matching candidates for the query at hand. For such a test, we fixed $k = 200$, so that re-ranking is performed only on 2% of database images. We evaluated the performance in terms of rate-efficiency curves. In particular, the accuracy of the task was evaluated according to the *Mean Average Precision* (MAP). Given an input query sequence q, for each frame $\mathcal{I}_{q,n}$ it is possible to define the *Average Precision* as

$$AP_{q,n} = \frac{\sum_{k=1}^{Z} P_{q,n}(k) r_{q,n}(k)}{R_{q,n}}, \tag{11}$$

where $P_{q,n}(k)$ is the precision (i.e., the fraction of relevant documents retrieved) considering the top-k results in the ranked list of database images; $r_{q,n}(k)$ is an indicator function, which is equal to 1 if the item at rank k is relevant for the query, and zero otherwise; $R_{q,n}$ is the total number of relevant document for frame $\mathcal{I}_{q,n}$ of the query sequence q and Z is the total number of documents in the list. The overall accuracy for the query sequence q is evaluated according to

$$AP_q = \frac{\sum_{n=1}^{N} AP_{q,n}}{N}, \tag{12}$$

where N is the total number of frames of the query video q.

Finally, the *Mean Average Precision* for the CBR task is obtained as

$$MAP = \frac{\sum_{q=1}^{Q} AP_q}{Q}, \tag{13}$$

that is, the mean of the MAP_q measure over all the query sequences.

We also considered an alternative way of aggregating the results of a video query q, resorting to *Median Rank Aggregation* (MRA). To this end, considering a test sequence of N frames, the retrieval pipeline is executed on each frame $\mathcal{I}_{q,n}$ leading to N ranked lists of retrieved documents $\mathcal{R}_{q,n}$, $n = 1, \ldots, N$. Each database image D_k, $k = 1, \ldots, Z$, can be assigned with a ranking value $\mathcal{P}_{q,n,k}$, equal to its position in the list $\mathcal{R}_{q,n}$. Then, for a database image D_l, it is possible to define a relevance score $\mathcal{P}_{q,k}$ to the query q by aggregating the ranking values $\mathcal{P}_{q,n,k}$ obtained for each query frame $\mathcal{I}_n$. In details, $\mathcal{P}_{q,k}$ is equal to the median value within the set of ranking values $\mathcal{P}_{q,n,k}$, $n = 1, \ldots, N$. Finally, an overall ranking of database images is obtained for a given test sequence, by sorting such documents according to their scores $\mathcal{P}_{q,k}$, in ascending order. Starting from such a ranking, it is possible to compute the *Average Precision* for the query q as

$$AP_{q,MRA} = \frac{\sum_{k=1}^{Z} P_{q,MRA}(k) r_{q,MRA}(k)}{R_q}, \tag{14}$$

where $P_{q,MRA}(k)$ is the precision (i.e., the fraction of relevant documents retrieved) considering the top-k results in the ranked list of database images obtained exploiting *Median Rank Aggregation*; $r_{q,MRA}(k)$ is an indicator function, which is equal to 1 if the item at rank k is relevant for the query, and zero otherwise; R_q is the total number of relevant document for the query at hand. Finally, the overall MAP is computed as the mean of $AP_{q,MRA}$ over all the query video sequences q.

With respect to the second stage of the CBR task, considering a query video sequence q, a set of visual features was extracted from each frame $\mathcal{I}_{q,n},\ n = 1, \ldots, N$. Then, such a set of local features was matched against the sets corresponding to the top-k candidate database images identified by means of the procedure detailed above resorting to global features. First, each feature extracted from the query frame was matched with its nearest neighbor in the test set, resorting to Hamming distance or Euclidean distance in the case of ATC - BRISK or CTA - SIFT, respectively. Second, matches were filtered resorting to the ratio test [3], with ratio parameter set to 0.7. Then, database images were ranked according to the number of matches with the query frame that passed the ratio test. Finally, we computed the MAP metric based on the ranking induced by the number of matches, resorting to the procedure adopted for global descriptors. Similarly, we also obtained results when *Median Rank Aggregation* was used.

As a further experiment, we evaluate the effect of temporal subsampling on the overall efficiency of the retrieval pipeline. We tested several different values for the GOP size parameter. When the GOP size is equal to f frames, a global descriptor (set of local descriptors) is sent every f frames. With respect to global descriptors, we tested different approaches:

- BoVW-SKIP: considering a Group Of Pictures, a global feature is extracted considering only the first frame of such GOP.
- BoVW-GOP: considering a Group of Pictures, global features are extracted from each frame of the GOP, then, the median global descriptor vector is computed and used for the retrieval.

D. Results

1) Homography estimation: First, we evaluated the number of bits necessary to encode each visual feature using either intra-frame or inter-frame coding, when varying the size of the descriptor K. Figure 3 shows the bitrate obtained by coding the BRISK features extracted from *Foreman* video sequence, indicating separately the number of bits used for encoding the keypoint location, the reference keypoint identifier (inter-frame only), and the descriptor elements. At high bitrates ($K = 256$), the coding rate is equal to 200 bits/feature and 222 bits/feature in the case of intra-frame coding, 156 bits/feature and 178 bits/feature in the case of intra-frame coding for BRISK and BINBOOST, respectively. At low bitrates ($K = 32$), the rate drops to approximately 55 bits/feature and 40 bits/feature for intra- and inter-frame coding, respectively. Similar results were also obtained for the other test sequences.

Figure 4 compares the results of ATC and CTA. As a benchmark, we also included the results obtained using ATC when SIFT visual features were used [22]. As a reference, when no visual feature compression is used, the bitrate for sending SIFT, BINBOOST or BRISK descriptors in the ATC paradigm would be, respectively, 376 kbps, 107 kpps and 220 kbps, attaining a homography estimation precision equal to 0.66, 0.66 and 0.62. Thus, visual feature compression leads to very large coding gains, since comparable precision levels are achievable with at approximately 25 kbps for SIFT, BINBOOST and BRISK (bitrate saving -93%, -77% and -89%, respectively). In all cases, ATC outperformed CTA, since higher levels of precision are attained for all target bitrates. With respect to the ATC approach, inter-frame coding significantly improves the coding efficiency, especially at low bitrates.

TABLE I
MEAN AVERAGE PRECISION (MAP) FOR THE RETRIEVAL TASK, AS A FUNCTION OF THE SIZE OF THE NUMBER OF VISUAL WORDS M COMPOSING THE DICTIONARY AND THE BRISK DETECTION THRESHOLD.

		BRISK threshold			
		30	50	70	90
# words	1k	.15	**.21**	.18	.15
	4k	.23	**.31**	.28	.21
	16k	.30	**.46**	.44	.35

In addition, to evaluate the benefit of using the dexel selection scheme described in Section III-C, we compared our results with a baseline in which the original selection scheme embedded in the BRISK descriptor was used. The latter simply chooses the elements corresponding to smallest spatial distance between the pattern points whose intensities are to be compared. Figure 4(b) shows that appropriately selecting the dexels significantly improves the task accuracy, which saturates using as few as 64 dexels / descriptors (requiring approximately 25 kbps to be transmitted).

2) Content-based retrieval task: Given a query video sequence, the task consists in retrieving the relevant images within a database composed of $Z = 10000$ images using global features and, possibly, refine the result using local features.

Since global features are computed from local features, we evaluated first the impact of the BRISK detection threshold, which determines the number of local features extracted from each query frame. A high threshold value leads to a low number of local features and, consequently, to sparser BoVW global descriptors. This allows for more efficient encoding, at the cost of less discriminating, and thus less accurate, global descriptors. In contrast, a sufficiently low threshold (high number of local features) allows unstable descriptors to be detected and leads to noisy global descriptors. Table I shows the impact of both dictionary size and BRISK detection threshold on the *Mean Average Precision* measure. A BRISK threshold value set to a value of 50 leads to the best results for all the possible dictionary sizes.

Then, we considered the impact of coding global features in ATC, by tracing the rate-MAP curves obtained for different dictionary sizes. For example, Figure 5(a) and 5(b) show the rate-MAP curves obtained with dictionary of size $M = 4096$ and $M = 16384$, respectively. Each curve was obtained by varying the quantization step size Δ. A larger dictionary allows for improved accuracy. In particular, MAP saturates at approximately 0.34 and 0.49 when the dictionary has size $M = 4096$ and $M = 16384$, respectively. On the other hand, a larger dictionary leads to larger descriptors and, thus, a higher number of bits is required for each query. In details, the value of MAP saturates when using approximately 160 (180) and 350 (360) Bytes/query for $M = 4096$ and $M = 16384$, respectively, when inter-frame (intra-frame) coding is the selected

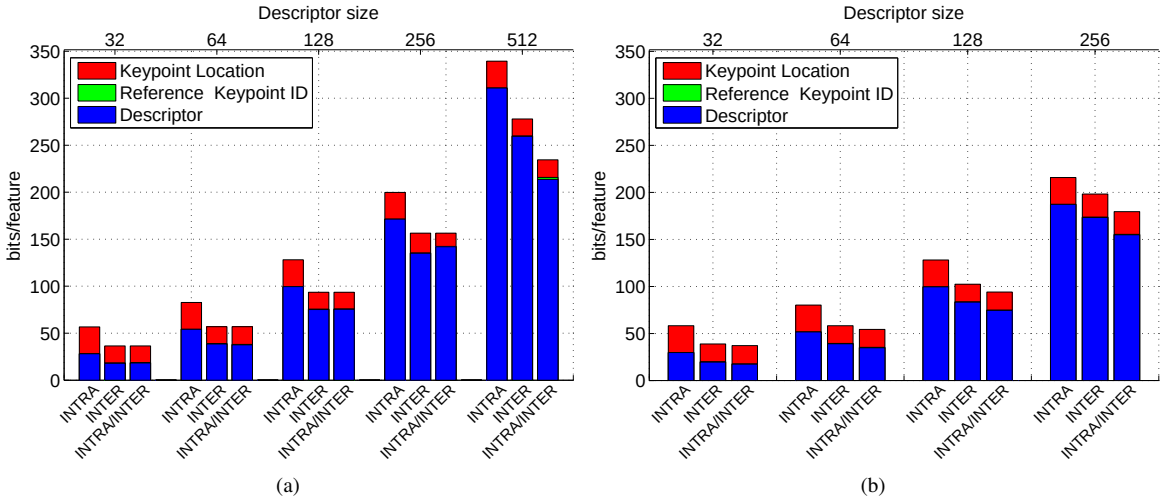


Fig. 3. Bitrate needed to encode each visual feature extracted from the *Foreman* sequence, varying the size of the binary descriptor, for a) BRISK; b) BINBOOST.

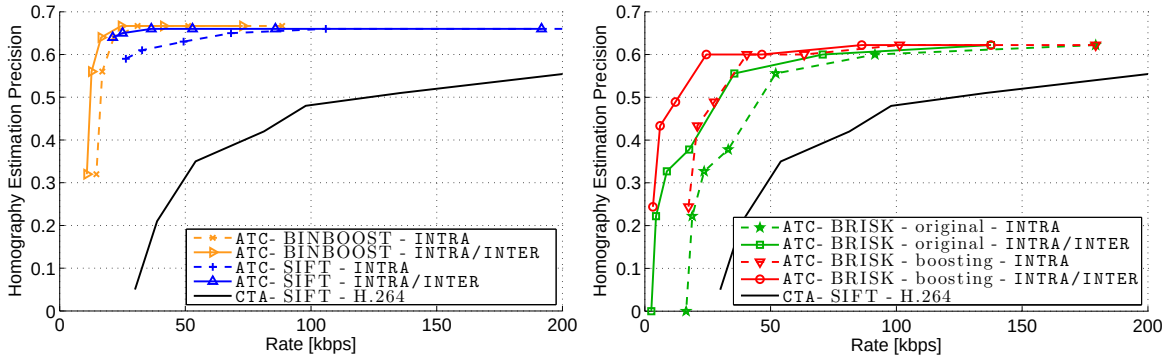


Fig. 4. Rate-accuracy curves obtained for the *Paris - homography* sequence. a) ATC (either based on SIFT or BINBOOST) vs. CTA; b) BAMBOO boosted dexel selection scheme vs. BRISK original dexel selection scheme within the ATC approach.

method. Large dictionaries lead to quantizing similar features of consecutive frames to different visual words, thus reducing the amount of temporal redundancy and preventing inter-frame coding to achieve significant coding gains. Regardless of the dictionary size, the usage of *Median Rank Aggregation* leads to an improvement of about 5% in terms of MAP. Figure 6 summarizes the best rate-MAP curve for each dictionary size in the same chart, including also the case $M = 1024$. By inspecting the envelope of the rate-MAP curves, it is possible to observe that the dictionary size should be adjusted based on the target bitrate, namely, $M = 1024$ when using less than 50 Bytes/query, $M = 16384$ when using more than 200 Bytes/query, and $M = 4096$ in all other cases.

As a further experiment, we fixed the dictionary size to $M = 16384$ to achieve the highest MAP, and we investigated how to reduce the rate by sending only one global descriptor per GOP, when the GOP size was varied in the set $\{1, 2, 5, 10, 20, 50\}$. In Figure 7 we observe that when using the BoVW-SKIP approach, the MAP slightly decreases when increasing the GOP size, while achieving a significant bitrate saving. This is due to the fact that fewer query frames were used for the same video query, thus reducing the bitrate but also the diversity in the query content. To overcome this issue, BoVW-GOP aggregates the global descriptors extracted from all frames of a GOP into a single descriptor. This leads to a significantly higher MAP (+8%), while achieving the same

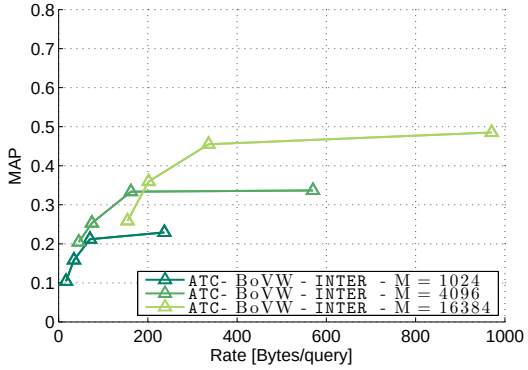


Fig. 6. Envelope of rate-MAP curves for the content-based retrieval task, when matching is performed resorting to Bag-of-Words based on BRISK local features. The curves are obtained by varying both the dictionary size M and the quantization step size Δ, when "Median Rank Aggregation (MRA)" is employed.

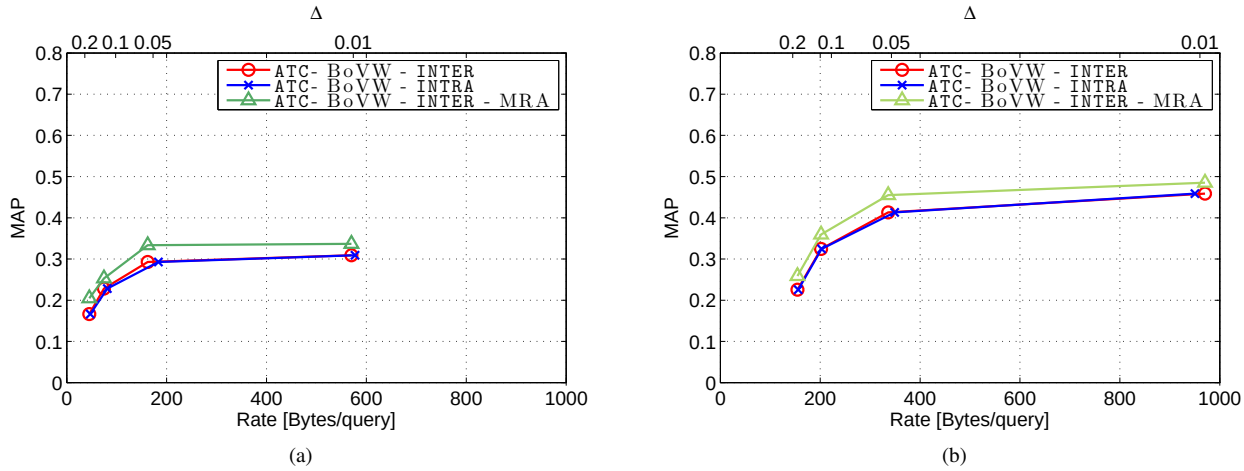


Fig. 5. Rate-MAP curves for the retrieval task, when matching is performed resorting to Bag-of-Words based on BRISK local features, considering a dictionary of a) $M = 4096$ visual words; b) $M = 16384$ visual words.

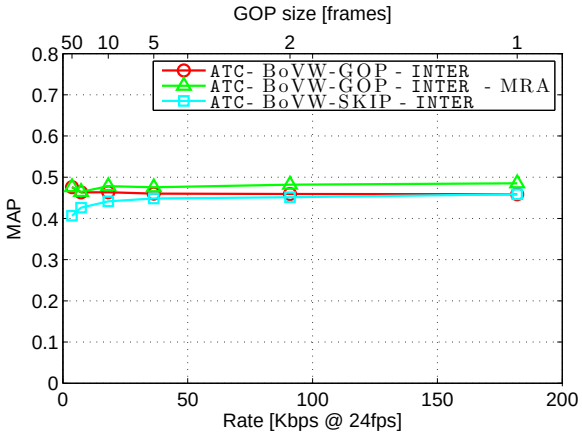


Fig. 7. Rate-MAP curves for the content-based retrieval task, when matching is performed using Bag-of-Words based on BRISK local features, considering a dictionary of $M = 16386$ visual words.

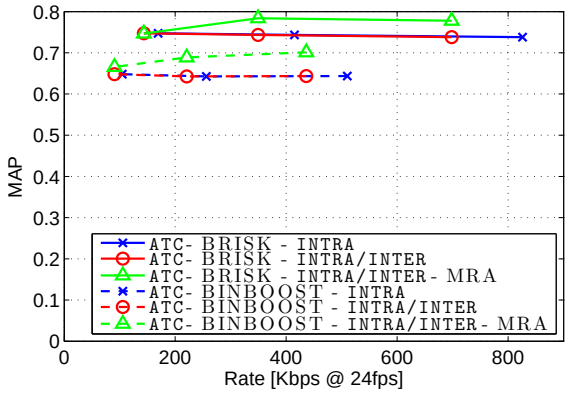


Fig. 8. Rate-MAP curves for the content-based retrieval task, when a re-ranking step is performed on the top-200 candidates, resorting to either BRISK or BINBOOST, as a function of the GOP size.

bitrate saving. In addition, *Median Rank Aggregation* can also be used at the receiver side to further improve the MAP. This is useful especially when considering small GOP sizes, i.e., when aggregation is performed resorting to a higher number of frames with a high temporal correlation. Although Figure 7 might suggest that additional coding gains can be achieved by increasing the GOP size beyond 25 frames, in real application scenarios there are other requirements that typically constrain the largest GOP size allowed, namely the maximum tolerable delay, or the dynamic nature of the underlying video sequence.

In a typical content-based retrieval pipeline, local features are often used to re-rank the result obtained using global features. Figure 8 shows the rate-MAP curves when either BRISK or BINBOOST descriptors were used in the re-ranking step. Similarly to the case of global descriptors, we investigated the impact of temporal subsampling on the overall accuracy. Considering a Group Of Pictures (GOP), a set of visual features is extracted from the first frame of such GOP and used in order to refine the results provided by the retrieval pipeline based on global descriptors. Each curve is traced by varying the GOP size in the set $\{5, 10, 25\}$ and using the largest descriptor size ($K = 512$ for BRISK and $K = 256$ for BINBOOST). With respect to the retrieval based on global features only, MAP was boosted from 0.49 to 0.78 (BRISK) and 0.69 (BINBOOST). Note that, unlike for the homography estimation task, BRISK outperforms BINBOOST for this task. At the same time, this comes at an additional cost in terms of bitrate, which is increased by approximately an order of magnitude. For example, when the GOP size is equal to 25, the bitrate increases from 8 kbps (global features) to 150 kbps for BRISK and 95 kbps for BINBOOST. Figure 8 also shows that inter-frame coding reduces the bitrate with respect to intra-frame coding between 5% and 15%, depending on the GOP size. Similarly to the case of global features, Median Rank Aggregation brings significant advantages in terms of MAP, when a sufficiently small GOP size is employed.

Finally, we compared the results obtained resorting to either ATC or CTA in Figure 9 (note that the curve $\texttt{ATC} - BoVW$ corresponds to the operating points in the MAP-rate curve in Figure 7 corresponding to a GOP size equal to either 25, 10 or

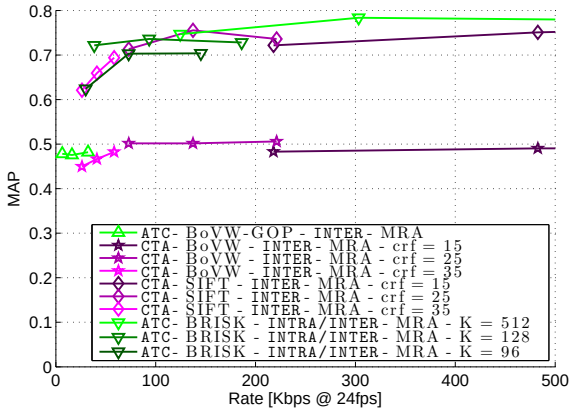


Fig. 9. Rate-accuracy curves comparing ATC and CTA approaches.

5 frames). When using global features only, ATC outperforms CTA by a large margin. Indeed, at very low bitrate, ATC based on global features is the only viable option, since at least 30kbps are needed to transmit a pixel-level representation of the visual content and thus, to enact the CTA paradigm.

When setting the GOP size to 10 frames (i.e., corresponding to the operating point in the middle of each curve), ATC requires as few as 18 kbps to achieve a MAP equal to 0.48. In contrast, CTA requires 40 kbps (MAP = 0.46), 140 kbps (MAP = 0.50) and 480 kbps (MAP = 0.49), when changing the Constant Rate Factor parameter *crf* of H.264/AVC.

When considering re-ranking based on local features, CTA is able to significantly improve MAP at no extra cost in terms of bitrate. The best performance achieved by CTA at $crf = 25$ (for both global and local features) can be attributed to the mild smoothing operated by lossy coding at this bitrate, which reduces noise and allows detecting more stable keypoints. Conversely, ATC requires sending additional bits to be able to encode the local features. Figure 9 shows different curves obtained by varying the number of dexels K. In particular, descriptors with size equal to 512, 128 or 96 dexels were tested. Smaller descriptor lengths lead to a significant loss in terms of accuracy. This is due to the inefficiency of a very short BRISK descriptor. In the case of local descriptors, ATC performs on a par with CTA, and what is the best paradigm is determined by the target bitrate. For example, at 40 kbps, MAP is equal to 0.72 for ATC and 0.65 for CTA. Conversely, at 30 kbps, MAP is equal to approximately 0.63 for both ATC and CTA.

VII. Conclusions

We proposed two coding architectures tailored to either local binary features (tested on BRISK and BINBOOST) or global features (based on Bag-of-Visual-Words), extracted from video sequences. The efficiency of the proposed solution was evaluated by means of rate-efficiency curves with respect to traditional visual analysis tasks. In the case of homography estimation the ATC paradigm always outperforms CTA by a large margin, achieving the same task efficiency that can be obtained using uncompressed sequences with as few as 20 kbps. In the case of content-based retrieval, the ATC paradigm always outperforms CTA when using global features, operating at 8 kbps and achieving the same MAP obtained using uncompressed sequences. When using local features, ATC and CTA perform on a par, calling for the investigation of more compact descriptors and more sophisticated coding tools (e.g., filtering the keypoints to be encoded based on the temporal coherence). Future work will address the use of recently proposed global descriptors extracted from binary features, e.g. BVLAD [18], and hybrid CTA − ATC coding schemes.

References

[1] L. Baroffio, J. Ascenso, M. Cesana, A. Redondi, and M. Tagliasacchi, "Coding binary local features extracted from video sequences," in *IEEE International Conference on Image Processing (ICIP)*, Paris, France, 2014.

[2] A. Canclini, M. Cesana, A. Redondi, M. Tagliasacchi, J. Ascenso, and R. Cilla, "Evaluation of low-complexity visual feature detectors and descriptors," in *Digital Signal Processing (DSP), 2013 18th International Conference on*, July 2013, pp. 1–7.

[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[4] H. Bay, T. Tuytelaars, and L. J. Van Gool, "Surf: Speeded up robust features," in *ECCV (1)*, 2006, pp. 404–417.

[5] A. Redondi, L. Baroffio, J. Ascenso, M. Cesana, and M. Tagliasacchi, "Rate-accuracy optimization of binary descriptors," in *20th IEEE International Conference on Image Processing*, Melbourne, Australia, Sep. 2013.

[6] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *ECCV (4)*, 2010, pp. 778–792.

[7] S. Leutenegger, M. Chli, and R. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *ICCV*, D. N. Metaxas, L. Quan, A. Sanfeliu, and L. J. Van Gool, Eds. IEEE, 2011, pp. 2548–2555.

[8] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *CVPR*. IEEE, 2012, pp. 510–517.

[9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov 2011, pp. 2564–2571.

[10] L. Baroffio, M. Cesana, A. Redondi, and M. Tagliasacchi, "Binary local descriptors based on robust hashing," in *IEEE International Workshop on Multimedia Signal Processing (MMSP) 2013*, Pula, Italy, September 2013.

[11] ——, "Bamboo: a fast descriptor based on asymmetric pairwise boosting," in *IEEE International Conference on Image Processing 2014*, Paris, France, October 2014.

[12] T. Trzcinski, M. Christoudias, V. Lepetit, and P. Fua, "Boosting Binary Keypoint Descriptors," in *Computer Vision and Pattern Recognition*, 2013.

[13] H. Jégou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *Int. J. Comput. Vision*, vol. 87, no. 3, pp. 316–336, May 2010. [Online]. Available: http://dx.doi.org/10.1007/s11263-009-0285-2

[14] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV*. IEEE Computer Society, 2003, pp. 1470–1477.

[15] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in *IEEE Conference on Computer Vision & Pattern Recognition*, jun 2010, pp. 3304–3311. [Online]. Available: http://lear.inrialpes.fr/pubs/2010/JDSP10

[16] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, June 2007, pp. 1–8.

[17] D. Galvez-Lopez and J. Tardos, "Real-time loop detection with bags of binary words," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, Sept 2011, pp. 51–58.

[18] D. Van Opdenbosch, G. Schroth, R. Huitl, S. Hilsenbeck, A. Garcea, and E. Steinbach, "Camera-based indoor positioning using scalable streaming of compressed binary image signatures," in *IEEE International Conference on Image Processing*, October 2014.

[19] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *CVPR*. IEEE Computer Society, 2007.

[20] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *ECCV (1)*, ser. Lecture Notes in Computer Science, D. A. Forsyth, P. H. S. Torr, and A. Zisserman, Eds., vol. 5302. Springer, 2008, pp. 304–317.

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

[22] L. Baroffio, M. Cesana, A. Redondi, M. Tagliasacchi, and S. Tubaro, "Coding visual features extracted from video sequences," *Image Processing, IEEE Transactions on*, vol. 23, no. 5, pp. 2262–2276, May 2014.

[23] J. Chen, L. Duan, R. Ji, H. Yao, and W. Gao, "Sorting local descriptors for low bit rate mobile visual search," in *36th IEEE International Conference on Acoustic, Speech, and Signal Processing*, Prague, Czech Republic, May 2011.

[24] 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*, M. Rabbani and R. L. Stevenson, Eds., vol. 7257, no. 1. SPIE, 2009, pp. 725 710+.

[25] A. Redondi, M. Cesana, and M. Tagliasacchi, "Low bitrate coding schemes for local image descriptors," in *International Workshop on Multimedia Signal Processing*, sept. 2012, pp. 124 –129.

[26] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–8.

[27] MPEG, "Compact descriptors for visual search," http://mpeg.chiariglione.org/standards/mpeg-7/compact-descriptors-visual-search.

[28] J. Ascenso and F. Pereira, "Lossless compression of binary image descriptors for visual sensor networks," in *Digital Signal Processing (DSP), 2013 18th International Conference on*, July 2013, pp. 1–8.

[29] A. Redondi, M. Cesana, and M. Tagliasacchi, "Rate-accuracy optimization in visual wireless sensor networks," in *International Conference on Image Processing*, oct. 2012.

[30] P. Monteiro and J. Ascenso, "Clustering based binary descriptor coding for efficient transmission in visual sensor networks," in *Picture Coding Symposium (PCS), 2013*, Dec 2013, pp. 25–28.

[31] V. Chandrasekhar, G. Takacs, D. Chen, S. S. Tsai, R. Grzeszczuk, and B. Girod, "Chog: Compressed histogram of gradients a low bit-rate feature descriptor," in *CVPR*, 2009, pp. 2504–2511.

[32] J. Chao and E. Steinbach, "Preserving sift features in jpeg-encoded images," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, Sept 2011, pp. 301–304.

[33] M. Makar, S. Tsai, V. Chandrasekhar, D. Chen, and B. Girod, "Interframe coding of canonical patches for mobile augmented reality," in *Multimedia (ISM), 2012 IEEE International Symposium on*, 2012, pp. 50–57.

[34] M. Makar, S. S. Tsai, V. Chandrasekhar, D. Chen, and B. Girod, "Interframe coding of canonical patches for low bit-rate mobile augmented reality." *International Journal of Semantic Computing*, vol. 7, no. 1, pp. 5–24, 2013.

[35] L. Baroffio, A. Redondi, M. Cesana, S. Tubaro, and M. Tagliasacchi, "Coding video sequences of visual features," in *20th IEEE International Conference on Image Processing*, Melbourne, Australia, Sep. 2013.

[36] MPEG, "Compact descriptors for video analysis (cdva)," http://mpeg.chiariglione.org/standards/exploration/compact-descriptors-video-analysis/n14509-mpeg-vision-compact-descriptors.

[37] S. Brown, G. Hua, and S. Winder, "Discriminative learning of local image descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 1, January 2011.

[38] B. Fern, E. Fromont, D. Muselet, and M. Sebban, "Supervised learning of gaussian mixture models for visual vocabulary generation," 2011.

[39] D. Galvez-Lopez and J. Tardos, "Real-time loop detection with bags of binary words," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, Sept 2011, pp. 51–58.

[40] J. Paratte, "Sparse binary features for image classification," Master's thesis, cole polytechnique fdrale de Lausanne, 2013.

[41] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, June 2007, pp. 1–8.

[42] F. De Simone, M. Tagliasacchi, M. Naccari, S. Tubaro, and T. Ebrahimi, "A H.264/AVC video database for the evaluation of quality metrics," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, march 2010, pp. 2430 –2433.

[43] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results," http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

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

[45] L. Baroffio, A. Canclini, M. Cesana, A. Redondi, M. Tagliasacchi, and S. Tubaro, "Rome landmark dataset," http://home.deib.polimi.it/baroffio/romelandmark.

[46] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '07. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035. [Online]. Available: http://dl.acm.org/citation.cfm?id=1283383.1283494