CODING BINARY LOCAL FEATURES EXTRACTED FROM VIDEO SEQUENCES

Luca Baroffio[⋆]*, João Ascenso*[†]*, Matteo Cesana*[⋆]*, Alessandro Redondi*[⋆]*, Marco Tagliasacchi*[⋆]

[⋆] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano
[†] Instituto Superior de Engenharia de Lisboa - Instituto de Telecomunicações, Lisbon

ABSTRACT

Local features represent a powerful tool which is exploited in several applications such as visual search, object recognition and tracking, etc. In this context, binary descriptors provide an efficient alternative to real-valued descriptors, due to low computational complexity, limited memory footprint and fast matching algorithms. The descriptor consists of a binary vector, in which each bit is the result of a pairwise comparison between smoothed pixel intensities. In several cases, visual features need to be transmitted over a bandwidth-limited network. To this end, it is useful to compress the descriptor to reduce the required rate, while attaining a target accuracy for the task at hand. The past literature thoroughly addressed the problem of coding visual features extracted from still images and, only very recently, the problem of coding real-valued features (e.g., SIFT, SURF) extracted from video sequences. In this paper we propose a coding architecture specifically designed for binary local features extracted from video content. We exploit both spatial and temporal redundancy by means of intra-frame and inter-frame coding modes, showing that significant coding gains can be attained for a target level of accuracy of the visual analysis task.

Index Terms— Visual features, binary descriptors, video coding.

1. INTRODUCTION

Visual features are a powerful tool that is being successfully exploited in many 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 local content of a given image patch, while being invariant to many local and global image transformations. The traditional pipeline for visual feature extraction consists of two main components: the keypoint detector, that is responsible for the identification of a set of salient keypoints within an image, and a keypoint descriptor, that computes a discriminative description vector for each identified keypoint, based on the local image content. Traditional keypoint description algorithms such as SIFT [1] and SURF [2] assign a signature to each interest point by means of a set of real-valued elements represented by means of floating point numbers. Instead, binary descriptors recently emerged as a computationally efficient alternative to such approaches [3]. The simplest algorithm belonging to the class of binary descriptors is BRIEF [4], which computes a binary representation whose entries are the results of pairwise comparisons between (smoothed) pixel intensity values randomly sampled from the neighborhood of a given keypoint. Similarly, each descriptor element (dexel) of both BRISK [5] and FREAK [6] is the result of a comparison between the intensity values of a pair of pixels sampled from ad-hoc designed spatial patterns. Furthermore, Differently from BRIEF, both BRISK and FREAK are inherently rotation- and scale-invariant. More recently, BAMBOO [7, 8] exploits a pairwise boosting algorithm in order to learn a discriminative pattern of pairwise pixel intensity comparisons. BinBoost [9], differently from more traditional approaches, proposes a boosted hash function based on a set of local gradients.

Several visual analysis applications, e.g., mobile visual search, distributed monitoring and surveillance, etc., require visual features to be transmitted over a bandwidth-limited network. The evolution of networks towards the "Internet-of-Things", a scenario in which low-power devices are able to collaboratively perform complex visual analysis tasks, might support such applications. In this sense, Visual Wireless Sensor Networks (VWSNs) represent a promising technological platform for distributed visual analysis tasks [10]. The traditional approach to such tasks, denoted hereinafter as "Compress-Then-Analyze" (CTA), consists in the following steps: the visual content is acquired by a sensor node; then, it is compressed (e.g., resorting to JPEG or H.264/AVC coding standards) and efficiently transmitted to a central unit, where visual analysis takes place. The task is thus performed based on a lossy representation of the visual content, possibly resulting in impaired performance. Furthermore, many applications might require only a succinct representation of the acquired visual content, disregarding the pixel-domain representation. In this sense, "Analyze-Then-Compress" (ATC) represents an alternative approach to CTA [11]. According to such a novel paradigm, visual features are extracted and compressed directly at the sensor node; then, they are transmitted to a sink node that performs visual analysis. For this model to be successfully enacted, efficient visual feature coding architectures are needed. A few works tackled the problems of either compressing local features extracted from still images [12] or adapting visual feature extraction algorithm, so that they are more suitable for compression, e.g., CHoG [13]. Moreover, since matching sets of visual features on large scale databases requires a huge amount of computational resources, a large body of works addressed the problem of building more compact, global representation of visual content, e.g., Bag-of-Visual-Words [14], VLAD [15], etc. Although such solutions offer a good compromise between efficiency and accuracy, especially considering retrieval and classification tasks, local features still play a fundamental role, being usually employed to refine the results of such tasks [16]. Furthermore, 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 [17, 18].

In this paper we propose a coding architecture suitable for binary local features extracted from video content. Inspired by traditional

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.

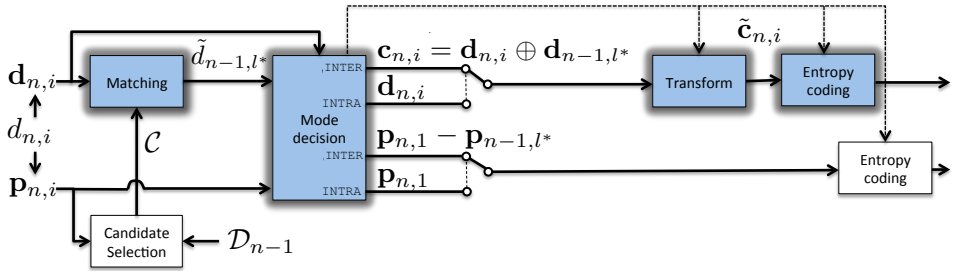

Fig. 1. Block diagram of the proposed coding architecture. The highlighted functional modules needed to be revisited due to the binary nature of the source.

video coding tools, we exploit both spatial and temporal redundancy by means of intra-frame and inter-frame coding. The coding efficiency is evaluated in terms of rate-accuracy curves, clearly demonstrating the advantage of the ATC paradigm, based on binary local features, over the traditional CTA paradigm, based on H.264/AVC video coding and the extraction of SIFT local features at the sink node. This work extends our previous work [19], which tackled the problem of compressing real-valued local features (SIFT and SURF) extracted from video sequences.

The problem of encoding visual features extracted from video is receiving a great deal of attention from the scientific community. In [20, 21], the authors proposed to encode and transmit temporally coherent image patches in the pixel-domain, shifting the computation of the descriptor to the sink node for augmented reality applications. At the same time, a new MPEG ad-hoc group on Compact Descriptors for Video Search (CDViS) [22] has recently started investigating the opportunity of drafting a standard related to the representation and coding of visual features in the context of video retrieval, automotive, tracking, etc.

The rest of this paper is organized as follows. Section 2 states the problem, defining the properties of the features to be coded, whereas Section 3 illustrates the coding architecture. Section 4 is devoted to defining the experimental setup and reporting the results. Finally, conclusions are drawn in Section 5.

2. 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 D-dimensional binary vector $\mathbf{d}_{n,i}$, 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 $K < D$ descriptor elements is actually coded, thus discarding a part of the original data. The latter approach is lossy, since a lossless coding scheme is applied only to a subset of the descriptor elements. 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

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

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 is 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 [16] or weak geometry checking [17]. 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 [23, 19].

Note that the proposed coding architecture, designed to encode visual features extracted from video sequences, can be straightforwardly adapted also to the context of sets of descriptors extracted from multiple cameras observing the same scene.

3. CODING OF LOCAL FEATURES

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 [23, 19]. However, we highlighted the functional modules that needed to be revisited due to the binary nature of the source.

3.1. 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 [24], which is briefly summarized in the following. In binary descriptors, each element represents the binary outcome of a pairwise comparison. Hence, the dexels are potentially statistically dependent, and it is possible to model the descriptor as a binary source with memory.

Let π_j, $j \in [1, D]$ represent the j-th element of a binary descriptor, where D is the dimension of such a descriptor. The entropy

of such a dexel can be computed as

$$H(\pi_j) = -p_j(0)\log_2(p_j(0)) - p_j(1)\log_2(p_j(1)), \quad (2)$$

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

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

with $j_1, j_2 \in [1, D]$. Let $\tilde{\pi}_j$, $j = 1, \ldots, D$, 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

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

In order to maximize the coding efficiency, we aim at finding the permutation of dexels $\tilde{\pi}_1, \ldots, \tilde{\pi}_D$ 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:

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

Note that such optimal ordering is computed offline, thanks to a training phase, and shared between both the encoder and the decoder.

3.2. 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 $d_{n,i}$, $i = 1, \ldots, M_n$, 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 3.1. Similarly, a reordering strategy can be applied also in the case of inter-frame coding, in this case considering the prediction residual $\mathbf{c}_{n,i}$.
- *Entropy coding*: Finally, the sets of local features are entropy coded. In the case of intra-frame coding, 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 images. In particular, for each of the D 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.

3.3. Descriptor element selection

The lossless coding architecture described in the previous section can be used to encode all the D 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 $K < D$ descriptor elements. In our previous work we explored different methods that define how to select the dexels to be retained [24, 7, 8]. In this work, we employed the greedy asymmetric pairwise boosting algorithm described in [8] in order to iteratively select the most discriminative descriptor elements. To this end, we used a training set of image patches [25], along with the ground truth information defining whether two image patches refers 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 $K < D$, it is possible to encode only the first K descriptor elements selected by this algorithm.

4. EXPERIMENTS

For the evaluation process, we extracted BRISK [5] features from a set of six video sequences at CIF resolution (352×288) and 30 fps, namely *Foreman*, *Mobile*, *Hall*, *Paris*, *News* and *Mother*, each with 300 frames [26]. In the training phase, we employed three sequences (*Mother*, *News* and *Paris*), whereas the remaining sequences (*Hall*, *Mobile* and *Foreman*) were employed for testing purposes. The statistics of the symbols to be fed to the entropy coder were learned based on the descriptors extracted from the training sequences. Moreover, the training video sequences were exploited to obtain the optimal coding order of dexels for both intra- and inter-frame coding, as illustrated in Section 3.1. Starting from the original BRISK descriptor consisting in $D = 512$ dexels, we considered a set of target descriptor sizes $K = \{512, 256, 128, 64, 32, 16, 8\}$. For each of such descriptor sizes, we employed the selection algorithm presented in Section 3.3.

As a first test, we evaluated the number of bits necessary to encode each visual features using either intra-frame or inter-frame cod-

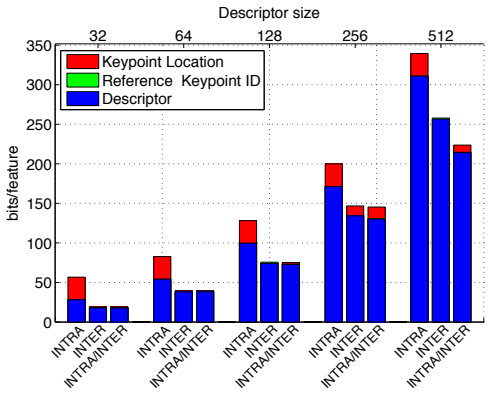

Fig. 2. Bitrate needed to encode each visual feature extracted from the *Foreman* sequence, varying the size of the BRISK descriptor.

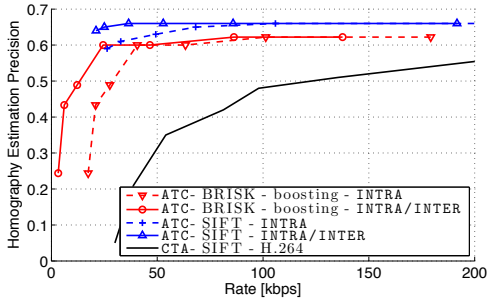

Fig. 3. Rate-accuracy curves obtained for the *Paris - homography* sequence. ATC (either based on BRISK or SIFT) vs. CTA.

ing, when varying the size of the descriptor K. Figure 2 shows the bitrate obtained by coding the BRISK features extracted from the *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. Similar results were obtained for all other test sequences (for supplementary results, refer to the technical report [27]). At high bitrates ($K = 512$), the coding rate is 340 bits/feature in the case of intra-frame coding, and 220 bits/feature in the case of inter-frame coding. At low bitrates ($K = 32$), the rate drops to 57 bits/feature and 19 bits/feature, respectively. Similar results were also obtained for the other test sequences.

As a second test, we evaluated the rate-accuracy performance of a visual analysis task using the coded local features. In this case, we used a publicly available dataset for visual tracking [28], 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 one 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). Given a test sequence, the goal of this experiment was to correctly estimate the homography for each frame. Hence, we measured the fraction of frames for which the homography was correctly estimated (i.e., average error less than 3 pixels) [19], which gives the homography estimation precision. We temporally down-sampled the sequences to 3 fps, in order to have consecutive frames sufficiently different and the task challenging. We compared two paradigms: *Analyze-Then-Compress* (ATC) and *Compress-Then-Analyze* (CTA). As for

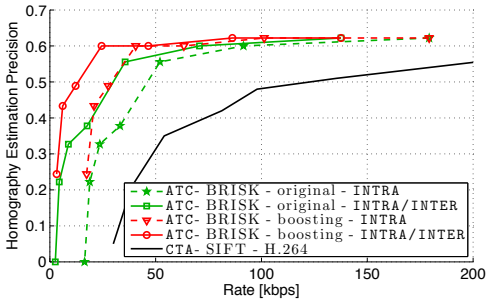

Fig. 4. Rate-accuracy curves obtained for the *Paris - homography* sequence. Comparison of different dexel selection schemes.

the ATC approach, BRISK features were extracted from each frame and encoded at a target descriptor length K. Then, for each frame, the homography was estimated applying the RANSAC [18] algorithm on the compressed local features.

On the other hand, for the CTA approach the video was compressed with the H.264/AVC coding standard (inter-frame coding), using the x264 video coding library and varying the quality factor $Q = \{5, 10, \ldots, 45\}$. Then, SIFT visual features were extracted from each frame by means of the VLFEAT [29] implementation and subsequently fed to RANSAC [18]. Figure 3 compares the results of the two approaches. We also included the results obtained using ATC when SIFT visual features were used [19]. As a reference, when no visual feature compression is used, the bitrate for sending either SIFT or BRISK descriptors in the ATC paradigm would be, respectively, 376 kbps or 220 kbps, attaining a homography estimation precision equal to 0.66 or 0.62. Thus, visual feature compression leads to very large coding gains, since comparable precision levels are achievable at 25 kbps for both SIFT and BRISK (bitrate saving -93% and -89%, respectively). In all cases, ATC outperforms CTA, since higher levels of precision are attained for all target bitrates. Within the tested alternatives of the ATC approach, inter-frame coding significantly improves the coding efficiency, especially at low bitrates, for both SIFT and BRISK. The use of SIFT in ATC allows to achieve a higher accuracy in the homography estimation, but at the cost of a significantly higher complexity to extract the visual features at the sensing node. This is particularly important in visual sensor network applications, in which sensing nodes are critically energy-constrained.

In addition, to evaluate the benefit of using the dexel selection scheme described in Section 3.3, we compare 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 shows that appropriately selecting the dexels significantly improves the task accuracy, which saturates using as few as 32 dexels / descriptors (requiring 25 kbps to be transmitted).

5. CONCLUSIONS

We proposed a coding architecture tailored to binary visual features extracted from video sequences. The efficiency of the proposed solution has been evaluated by means of rate-accuracy curves with respect to a traditional visual analysis task. Considering BRISK [5] descriptors, the proposed coding architecture provides bitrate savings up to 35% (60%) for intra-frame (inter-frame) coding.

6. REFERENCES

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

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

[3] 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.

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

[5] 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. 2011, pp. 2548–2555, IEEE.

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

[7] 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.

[8] L. Baroffio, M. Cesana, A. Redondi, and M. Tagliasacchi, "Bamboo: a fast descriptor based on asymmetric pairwise boosting," in *IEEE International Conference on Image Processing 2014*, Paris, France, October 2014.

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

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

[11] L. Baroffio, M. Cesana, A. Redondi, and M. Tagliasacchi, "Compress-then-analyze vs. analyse-then-compress: Two paradigms for image analysis in visual sensor networks," in *IEEE International Workshop on Multimedia Signal Processing (MMSP) 2013*, Pula, Italy, September 2013.

[12] 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*, Majid Rabbani and Robert L. Stevenson, Eds. 2009, vol. 7257, pp. 725710+, SPIE.

[13] 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.

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

[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.

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

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

[18] 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, June 1981.

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

[20] M. Makar, S.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.

[21] 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.

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

[23] 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, September 2013.

[24] 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, September 2013.

[25] 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.

[26] 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.

[27] L. Baroffio, J. Ascenso, M. Cesana, A. Redondi, S. Tubaro, and M. Tagliasacchi, "Coding binary local features extracted from video sequences," `http://home.deib.polimi.it/baroffio/VideoBRISK_techrep.pdf`, January 2014, Tech. Rep.

[28] 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.

[29] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," `http://www.vlfeat.org/`, 2008.