Lossless Compression of Binary Image Descriptors for Visual Sensor Networks

João Ascenso
Instituto Superior de Engenharia de Lisboa – Instituto de Telecomunicações, Lisboa, Portugal
joao.ascenso@lx.it.pt

Fernando Pereira
Instituto Superior Técnico – Instituto de Telecomunicações, Lisboa, Portugal
fp@lx.it.pt

Abstract — Nowadays, visual sensor networks have emerged as an important research area for distributed signal processing, with unique challenges in terms of performance, complexity, and resource allocation. In visual sensor networks, the energy consumption must be kept low to extend the lifetime of each battery-operated camera node. Thus, considering the large amount of data that visual sensors can generate, all the sensing, processing, and transmission operations must be optimized considering strict energy constraints. In this paper, camera nodes sense the visual scene but instead of transmitting the pixel coded representation, which demands high computation and bandwidth, a compact but yet rich visual representation is created and transmitted. This representation consists of discriminative visual features offering tremendous potential for several image analysis tasks. From all low-level image features available, the novel class of binary features, very fast to compute and match, are well suited for visual sensor networks. In this paper, lossless compression of binary image features is proposed to further lower the energy and bandwidth requirements. The coding solution exploits the redundancy between descriptors of an image by sorting the descriptors and applying DPCM and arithmetic coding. Experimental results show improvements up to 32% in terms of bitrate savings without any impact in the final image retrieval task accuracy.

Keywords—visual sensor networks, feature coding, binary features, image retrieval

I. INTRODUCTION

A visual sensor network is a network of low-power camera nodes providing visual information from a monitored location and performing distributed and collaborative processing of the data acquired at each camera node [1]. The deployment of a large number of low-power camera nodes allows the development of novel vision-based applications (e.g. smart moitoring) and the enhancement of the captured events and decisions reliability. The target is to provide high-level information about the acquired visual data by performing distributed and collaborative processing tasks. However, visual sensor networks are critically resource-constrained in terms of battery life, local node processing and bandwidth, demanding real-time operation, efficient data storage and camera node collaboration.

In this scenario, the first approach could be to transmit the large amount of video data produced by all cameras to a central node, taking into account an error prone network (typically wireless) and node resource constraints. At the central node, all visual data is analyzed and decisions can be made about the events captured by the camera nodes. Thus, camera nodes are responsible to acquire, compress and transmit the visual information to the central node. However, since each camera node has severe energy constraints, an alternative approach is to reverse this conventional compress-then-analyze paradigm. In this case, image features are collected by the camera nodes, coded and delivered to the final destination(s) to enable higher level visual analysis tasks by means of either centralized or distributed classifiers. Thus, it is not necessary to transmit the underlying pixel-level representation since most visual analysis tasks can be performed based on a succinct description of the image, including both global and local features. By following this approach, only the description of an image is coded and transmitted; this description is made of features or descriptors that only represent an excerpt of the image or patch.

However, many visual features proposed in the literature [2,3] were not originally designed to be efficiently transmitted, and some of them may even require more data than the coded pixel representation of an image, if not appropriately coded. For example, the bandwidth and computational requirements of the state-of-the-art SIFT [2] and SURF [3] descriptors is rather high for mobile devices with limited computational resources [4]. Thus, it is necessary to adopt visual features that can fulfill the critical visual sensor networks requirements, such as low feature extraction complexity (energy constraint) and low rate representation (bandwidth constraint). Naturally, both these requirements must be fulfilled without a significant impact in terms of the image analysis task efficiency. In this context, the GreenEyes (Networked Energy-Aware Visual Analysis) European project aims to find a feature representation suitable for visual sensor networks and advance previous state-of-the-art with a new set of methodologies, algorithms and protocols for sensor networks with vision capabilities [5].

Thus, novel schemes to code visual features that can fulfill the energy, bandwidth and efficiency requirements of visual sensor networks are needed. In the literature, a few low bit-rate descriptors using compression techniques tailored for this type
of data have been proposed. An early example of a visual feature coding scheme aiming to achieve a low bandwidth representation and high efficiency for image analysis tasks is the Compressed Histogram of Gradients (CHoG) [6]. In this case, a mobile wireless device with camera extracts CHoG features from pictures and transmits the compressed descriptors to a central server, which matches the query against a large dataset of images; the most similar images can then be returned to the user with some additional metadata information. In such case, the uplink channel from the wireless device has limited bandwidth and thus the query data must be efficiently compressed. In visual sensor networks, due to the even more limited camera node resources, the requirements are even more demanding; the complexity must be kept low since the battery operated sensor nodes must have an extended life. However, the complexity to compute the CHoG descriptors [6] is not suited for visual sensors with strict energy constraint requirements, despite its lower complexity for matching.

Recently, efficient binary feature point descriptors have become increasingly popular in computer vision since they can be computed with very low complexity. The Binary Robust Independent Elementary Features (BRIEF) [7] descriptor is highly discriminative and can be computed using simple intensity difference tests. BRIEF is rather memory efficient, fast to compute and match, thus, well suited for the visual sensor network scenario. However, BRIEF is not scale invariant and does not take into account keypoint orientation and scale, thus having lower performance for several image analysis tasks, especially when compared to the popular (non-binary) visual descriptors such as SIFT and SURF [8]. To overcome some of these limitations and further improve the quality of the representation provided by these descriptors, several binary descriptors were recently proposed, such as ORB [9], BRISK [10] and FREAK [11].

This paper proposes a method to lossless compress binary descriptions by exploiting the redundancy between descriptors that were extracted from the same image. Since each binary descriptor is computed from a patch around a detected keypoint, it is expected that neighboring descriptors or descriptors representing repetitive patterns that occur often within the image are well correlated. Thus, methods exploiting inter-descriptor redundancy by predicting a descriptor value from other descriptors should provide compression benefits. To minimize the bandwidth requirements, this paper proposes to appropriately sort the descriptors and perform predictive coding with a differential pulse coded modulation (DPCM) scheme and an adaptive binary arithmetic entropy codec. Only descriptors that meet the complexity requirements of visual sensor networks are considered here, thus, the proposed coding method targets the BRIEF, ORB, BRISK and FREAK binary descriptors. Experimental results show that rate savings up to 32% may be achieved by using the proposed coding framework to exploit the inter-descriptor redundancy.

II. VISUAL DESCRIPTORS WITH BINARY REPRESENTATION

Local image feature descriptors computed from patches have been vastly used in many computer vision applications, such as image retrieval, camera registration, 3D reconstruction and object identification. The main objective of local descriptors is to obtain a meaningful representation of an image while being invariant to many transformations, such as scale, rotation and viewpoint changes. However, the search for a robust representation (description) with local features is still pursued, especially for devices with a scarce amount of resources in terms of computation power, communication resources and energy capacity.

To address this challenge, several binary descriptors computed directly on image patches such as BRIEF, ORB, BRISK and FREAK, have been recently proposed. These descriptors were designed to address the low power requirements of mobile (mostly wireless) devices. Therefore, binary descriptors have characteristics that are well suited for visual sensor networks, such as very low complexity in the extraction and a rather small memory footprint, especially when compared to vector-based descriptors such as SIFT [2] and SURF [3]. Binary descriptors also enable faster matching, since they typically just require the computation of Hamming distances (bitwise XOR operation followed by a bit count), instead of the more expensive L2 norm. Also, to further reduce the complexity, some of these descriptors, such as BRISK and ORB, have also proposed a method to detect keypoints with limited computational resources.

A. BRIEF

The BRIEF descriptor is calculated for an image patch around a detected keypoint and corresponds to a binary vector where each bit is obtained by performing an intensity test between two pixels of the patch. To reduce noise, the patches are previously smoothed with a Gaussian kernel. The BRIEF feature is calculated as:

\[
f(p_1, p_2) = \begin{cases} 
1 & \text{if } I(p_1) > I(p_2) \\
0 & \text{otherwise}
\end{cases} \tag{1}
\]

where \(p_1\) and \(p_2\) are two locations (sampling points) inside the patch, \(I(p_1)\) and \(I(p_2)\) their corresponding pixel values. To efficiently represent an image patch, a certain number of tests must be performed (according to the descriptor length) with the descriptor corresponding to the concatenation of the bits produced by such tests. The set of pairs of positions \(p_1\) and \(p_2\) must be selected before the descriptor computation and is usually referred as the sampling pattern. For the BRIEF descriptor, these positions follow a Gaussian distribution whose variance depends on the size of the patch that is calculated before the descriptor is calculated. Due to their simplicity, BRIEF descriptors are very fast to compute and well suited for the visual sensor network scenario. However, BRIEF is not invariant to rotation and scale changes and its performance is worst compared to non-binary descriptors for many application scenarios [8].

B. BRISK

The BRISK descriptor is invariant to rotation and scale and proposes the usage of a symmetric sampling pattern with locations equally spaced on concentric circles around the keypoint. To achieve rotation invariance, the direction of each keypoint is estimated by averaging the local gradients obtained with several long-distance sampling points; thus, all BRISK descriptors are normalized according to their orientation. Before the descriptor is calculated, Gaussian pre-smoothing is
applied but with a standard deviation proportional to the distance between each sampling point and its concentric circle; thus, pixels of the patch cannot contribute to more than one point in the sampling pattern. Then, the descriptor is calculated by using the same intensity tests as the BRIEF descriptor but with a new deterministic sampling pattern that is rotated according to the estimated orientation and scaled following the keypoint scale (obtained by the keypoint detector). For the calculation of the BRISK descriptor, only short distance intensity tests of sampling points are used, leading to a descriptor with 512 bits in length.

C. ORB

The Orientated FAST [12] and Rotated BRIEF (ORB) descriptor includes a keypoint detector (based on FAST) and computes the orientation of each keypoint. ORB calculates the orientation as the angle of the vector from the keypoint center (detected by FAST) to the intensity centroid. In such case, it is assumed that these two points are not always the same and that the centroid of a patch is a reliable and stable estimation for the orientation. As in BRISK, the sampling pattern is rotated before the extraction of the descriptor to obtain a rotation normalized representation (if the centroid location is consistent between similar images).

ORB also assumes that intensity tests must be uncorrelated to obtain good matching performance; thus, ORB proposes a learning algorithm to select a subset of intensity tests that are uncorrelated and lead to high variance, among all possible intensity tests inside the patch. With this design for the sampling pattern, more discriminative binary features are obtained with just 256 bits of descriptor length.

D. FREAK

The Fast Retina Keypoint (FREAK) descriptor does not include any keypoint detector as in BRISK/ORB, but proposes a new sampling pattern that is inspired by the topology of the human eye retina. This sampling pattern is circular as for BRISK, but with the difference of having higher density of points near the center. In addition, Gaussian pre-smoothing is applied as in BRISK but with an exponential change in size (larger kernels are applied far from the center) and overlapping Gaussian kernels, i.e. there are pixels inside the patch that contribute to more than one position of the sampling pattern. For the orientation computation, a similar approach to BRISK is followed but with fewer position pairs that are carefully selected to be symmetric with respect to the center. In the creation of the FREAK descriptor, the same approach as for ORB was employed, notably using a learning algorithm to obtain the best intensity tests, restricted to the biological inspired FREAK sampling pattern. For maximum performance, the number of intensity tests is 512, which results in a binary descriptor of 512 bits in length.

III. COMPRESSION OF BINARY FEATURES

As described in the previous section, for many descriptor schemes, the descriptor length is 512 bits, i.e. 512 intensity tests are performed in each image patch. Typically, to provide a robust and complete representation of an image and good performance in most visual analysis tasks, 300-400 descriptors are needed [13]. In case no further processing is performed, each of the image descriptors must be represented with its descriptor length, which means that an entire image binary feature representation can demand up to 153600 bits. Assuming that binary features are sent for each frame (with 30 frames per second), up to 4 Mbit/s are needed just for one camera node. However, it is not possible to efficiently transmit this amount of data in visual sensor networks since: 1) most sensor devices have wireless technologies with a low data rate (and low power consumption) such as data links conforming to the IEEE 802.15.4 standard [14]; in such case, the maximum data rate is 250 kbit/s, which is not enough to meet the data rate requirements of raw binary features; 2) since the wireless medium is shared among neighboring sensor nodes, the bandwidth allocated for each sensor can be significantly lower than the maximum bandwidth limit; 3) to extend battery life, most sensor nodes require to sleep their communication during some period [14], which would not be possible if it is necessary to continuously transmit data. Thus, it is essential to reduce the amount of bitrate required to stream binary features in a visual sensor network scenario. Only by proposing new techniques capable to efficiently code binary features the bandwidth requirements can be met. To obtain a low-data representation that satisfies the requirements of camera nodes in a visual sensor network, the framework shown in Figure 1 is proposed.

![Figure 1. Proposed framework for coding and matching visual features.](image-url)
To compress binary (and even non-binary) features, two different approaches can be followed:

1. **Intra coding descriptor schemes**: each descriptor is independently encoded from the others. In this case, only the correlation inside the descriptor elements can be exploited. The coding efficiency depends essentially on the order the binary descriptors intensity tests are made and the correlation between the results of such tests.

2. **Inter coding descriptor schemes**: each descriptor is differentially encoded from others in the same image. In this case, the correlation between the whole set of descriptors representing an image is exploited. The order by which the binary descriptors are coded is rather important.

Experiments have suggested that the coding efficiency of Intra coding schemes is low mainly because the intensity tests were designed to have a rather unique descriptor per patch, i.e. intensity tests are uncorrelated if no sorting is made. However, coding schemes that prioritize the importance of each intensity tests and consider rate constraints could lead to compression improvements [15]. On the other hand, the spatial correlation between neighboring descriptors or the correlation between descriptors that describe a repetitive pattern of the image, i.e. descriptors of patches with similar appearance, can also be exploited with an efficient Inter coding scheme. This work uses a new inter coding descriptor scheme for binary descriptors that can be applied to any binary descriptor, such as BRIEF, BRISK, ORB and FREAK.

**IV. LOSSLESS CODING OF BINARY FEATURES**

This paper proposes a lossless predictive coding scheme for binary features, employing tools well-suited for this coding problem. Note that available works [6,13,16] have only addressed the coding of non-binary descriptors, such as SIFT and SURF. By performing lossless coding, the binary features representing the image at the sender side are the same at the receiver side, thus, no penalty in the image retrieval task accuracy occurs.

**A. Architecture**

In Figure 2, the proposed binary descriptor coding (BDC) architecture is shown. Let \( I \) denote an image with \( n \) descriptors \( d_i \) with \( i = 1, \ldots, n \). Assume that \( n \) keypoints were already detected with a state-of-the-art technique, such as SURF, ORB or BRISK detectors. Thus, the first step corresponds to the extraction of the binary descriptors, \( d_i \), from a patch surrounding each keypoint on the corresponding input image. Then, the set of descriptors are sorted according to a certain criteria and residual coded, the major novelty of the proposed solution. Finally, the residual is compressed with an adaptive binary arithmetic codec to exploit the statistical redundancy in the residual signal.

**B. Binary Descriptor Coding: Sender**

The idea of the proposed framework is to sort descriptors in such a way that predictive coding methods can reduce the amount of data that is necessary to transmit. The following BDC framework techniques are used at the sender side:

1. **Binary descriptor extraction**: First, descriptors \( d_i \) are extracted with any of the state-of-the-art binary descriptor methods described in Section II; other binary descriptors representations can also be used, e.g. a binarized SIFT descriptor [17].

2. **Descriptor sorting**: Since the order by which the descriptors are sent does not have any impact on the image retrieval task efficiency, descriptors can be reordered in any arbitrary way. Thus, descriptors are sorted to maximize the correlation between adjacent descriptors, i.e. each \( d_i \) descriptor is selected to minimize the distance to its \( d_{i-1} \) previous descriptor. In this case, the Hamming distance is adopted to define the similarity between binary descriptors. The following sorting algorithm is proposed:

   a) Define the first descriptor that should correspond to the most stable keypoint. In this case, the descriptor with the highest Hessian response [3] (i.e. value calculated by the keypoint detector) is selected. This selection was made because low values of Hessian response correspond to less discriminative descriptors. The selected descriptor is sent uncoded to the receiver side.

   b) Next, consider two sets of descriptors: 1) the set \( S_j \) of sorted descriptors only including the first descriptor selected in the previous step; 2) the set \( U_j \) of unsorted descriptors including all the other extracted descriptors in any arbitrary order, except the descriptor selected in the previous step. Then, choose from the set \( U_j \) the most similar descriptor to the previous selected descriptor (i.e. the last descriptor included in set \( S_j \)) according to the Hamming distance criterion, i.e. the number of different bits between two binary descriptors. The selected descriptor is removed from set \( U_j \) and added to the set \( S_j \) of sorted descriptors. This step is repeated until all extracted descriptors are in the set \( S_j \).

**3. Residual coding**

Then, previously coded descriptors that are known at both sender and receiver can be used to predict the current descriptor. To exploit the order of the descriptors in the set \( S_j \), it is possible to employ a first order predictor, which results in the following DPCM based coding scheme:
\[ \tilde{d}_i = [d_0, d_0 \bigoplus d_1, d_1 \bigoplus d_2, \ldots, d_{n-2} \bigoplus d_{n-1}] \]  

(2)

where \( \tilde{d}_i \) corresponds to the residual descriptor and \( \bigoplus \) corresponds to the bitwise XOR operator.

4. **Arithmetic coding**: Then, the residual descriptors are arithmetic coded and transmitted, except for the first descriptor for which no prediction is available. In this case, a binary adaptive arithmetic coder is used with a simple model that stores the probability of each bit being 1 or 0. This probability model is initialized at the beginning of the coding process of each residual descriptor \( \tilde{d}_i \), with equal probabilities for the symbols 1 and 0; these probabilities are updated along the coding process. In addition, by using quantized probability ranges and states, only integer instructions are used and a low amount of memory is required.

C. **Binary Descriptor Coding**: Receiver

The BDC coding scheme is evaluated in the context with a state-of-the-art image retrieval scheme. At the receiver side, all the binary descriptors are decoded and matched to a database of images, obtaining a ranked list of the most similar images (and associated information). There are several methods to perform this matching. While for some of the methods, the keypoint locations where the binary descriptors were extracted are not necessary [18], other methods enforce geometric consistency constraints [19], and thus, the keypoint locations are needed. In the latter case, an improvement of the retrieval performance is normally obtained; however, the sender must code and transmit the keypoint locations with a state-of-the-art performance is normally obtained; however, the sender must code and transmit the keypoint locations with a state-of-the-art performance.

1. **Descriptor decoding**: By applying (2) to the residual descriptors, \( \tilde{d}_i \), the descriptors \( d_i \) are obtained without any mismatch or error. With the proposed framework, the order of the descriptors is not explicitly transmitted to the receiver.

2. **Descriptor matching**: The next step is to perform pairwise matching between the decoded descriptors of the query and those already stored in the database. The correspondences between each descriptor of the query image and the descriptor of a certain database image are found by assigning each query descriptor to its nearest neighbor (using the Hamming distance metric). The process of finding correspondences is performed for each image in the database. To enable robust matching (i.e. only 'true' matches between patches are allowed), the correspondences must be filtered. In this case, two techniques were used:

   a) The popular ratio test [2, 8] was adopted. Basically, this test compares the ratio of distances between the two top matches for a given keypoint and rejects the top match if the ratio is above a threshold of 0.7. This type of test has proven to be effective to remove wrong matches between keypoints of different images, thus increasing the reliability of the matching procedure.

   b) The Random Sample Consensus (RANSAC) algorithm [21] has been adopted to force a geometric constraint in the matches obtained between the query and database descriptors. This involves the generation of multiple hypotheses using a minimal number of correspondences and evaluating the goodness of each hypothesis as the number of inliers, i.e. correspondences that follow the estimated rigid geometric model (for this case, a perspective homography model is used). After several iterations, the final model corresponds to the model that has more inliers over all computed models.

After establishing the homography with RANSAC between the two sets of keypoints coordinates (from query and database images), all correspondences are tested. If a correspondence fits well the estimated model, it is considered an inlier. The filtered correspondences (after the ratio test and RANSAC) between the query descriptors and every image in the database are found. Then, the database images are ranked according to the number of inliers.

V. **Performance Evaluation**

In this section, an experimental evaluation of the BDC compression scheme proposed in Section IV is made in terms of rate-accuracy for the task of image retrieval; having a more compact descriptor representation may allow the transmission of a higher number of descriptors for the same available rate, thus increasing the rate-accuracy behavior.

A. **Test Conditions**

Two public available datasets, including outdoor scenes (mainly buildings) are used, namely: i) ZuBuD dataset [22], and ii) Torino dataset [23]. These datasets represent varying weather conditions, different cameras and random viewpoints and different lighting conditions of the same outdoor scene. While the ZuBuD dataset contains 1005 images of 201 buildings of the city of Zurich, the Torino dataset contains 1260 images of 180 buildings of the city of Torino. For the ZuBuD dataset, a separate archive with query images is available and used in this evaluation. For the Torino dataset, one image for each building was selected as query, removing it from the matching dataset. In addition, all Torino dataset images were resized to the 640 \( \times \) 480 spatial resolution, while for the ZuBuD dataset, the original spatial resolution 320 \( \times \) 240 was kept unchanged.

To understand the coding gains obtained with the BDC framework, the SURF keypoint detector is always used. Using the same detector guarantees that the same patches are described for all binary descriptors (described in Section II) and BDC compression efficiency is evaluated independently of the keypoint detector (i.e. where keypoints are located). Note, however, that comparable results were obtained with other detectors, such as the ORB detector. For the two datasets, the BRISK, ORB and FREAK binary descriptors are evaluated, while the BRIEF descriptor is only evaluated for the Torino dataset. For the ZuBuD dataset, BRIEF is not able to perform matching since the query and database images have different resolutions and BRIEF is not scale invariant. The OpenCV 2.4 implementation of the SURF detector and BRIEF, BRISK, ORB and FREAK descriptors was used.
To perform a rate-accuracy evaluation, a simple rate control technique was implemented. The keypoint detector extracts a maximum of 400 descriptors, then the keypoints are sorted according to their reliability; for the SURF detector employed here, the Hessian response [3] was used as sorting criterion. Thus, the descriptors corresponding to the most stable keypoints are sent first, which brings improvements in terms of accuracy. Here, the Hessian response [3] was used as sorting criterion. In this experiment, the best RA performance was obtained when the top ranked descriptors of the query image are transmitted. In these experiments, $N$ assumes the following values: 15, 30, 50, 100, 150, 200, 250, 300, 350 and 400 binary descriptors per image. To characterize the performance of the BDC system, the accuracy metric is used along with the bitrate necessary to represent the $N$ transmitted binary descriptors. In this experiment, only the bitrate of the descriptors was considered but in a realistic scenario, the keypoint bitrate is also necessary. The accuracy corresponds to the number of correct queries over the total number of queries made to the system. A query is considered correct when the top ranked image (i.e. the database image with more inliers) corresponds to the same building as the query image. Note that the accuracy does not suffer any change when a certain number of descriptors are compressed with the lossless BDC framework. The bitrate corresponds to the average bitrate calculated over all the queries made, i.e. 115 and 180 queries for the ZuBuD and Torino datasets, respectively.

### B. Experimental Results

The rate-accuracy (RA) of the BDC framework is shown in Table I and Table II for the ZuBuD and Torino datasets, respectively.

#### TABLE I. BDC RATE-ACCURACY RESULTS FOR THE TORINO DATASET.

<table>
<thead>
<tr>
<th></th>
<th>BRIEF</th>
<th>BRISK</th>
<th>ORB</th>
<th>FREAK</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rate</strong></td>
<td>Accuracy</td>
<td>ΔR(%)</td>
<td>Accuracy</td>
<td>ΔR(%)</td>
</tr>
<tr>
<td>6897</td>
<td>0.357</td>
<td>10</td>
<td>3508</td>
<td>0.148</td>
</tr>
<tr>
<td>13460</td>
<td>0.626</td>
<td>12</td>
<td>6862</td>
<td>0.374</td>
</tr>
<tr>
<td>22123</td>
<td>0.817</td>
<td>14</td>
<td>11266</td>
<td>0.609</td>
</tr>
<tr>
<td>43415</td>
<td>0.878</td>
<td>15</td>
<td>22171</td>
<td>0.795</td>
</tr>
<tr>
<td>64475</td>
<td>0.922</td>
<td>16</td>
<td>30259</td>
<td>0.943</td>
</tr>
<tr>
<td>85026</td>
<td>0.948</td>
<td>17</td>
<td>43790</td>
<td>0.826</td>
</tr>
<tr>
<td>103855</td>
<td>0.957</td>
<td>17</td>
<td>54139</td>
<td>0.843</td>
</tr>
<tr>
<td>120311</td>
<td>0.948</td>
<td>17</td>
<td>63471</td>
<td>0.904</td>
</tr>
<tr>
<td>132838</td>
<td>0.965</td>
<td>18</td>
<td>71525</td>
<td>0.896</td>
</tr>
<tr>
<td>139992</td>
<td>0.965</td>
<td>18</td>
<td>77374</td>
<td>0.904</td>
</tr>
</tbody>
</table>

#### TABLE II. BDC RATE-ACCURACY RESULTS FOR THE ZUBUG DATASET.

<table>
<thead>
<tr>
<th></th>
<th>BRIEF</th>
<th>BRISK</th>
<th>ORB</th>
<th>FREAK</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rate</strong></td>
<td>Accuracy</td>
<td>ΔR(%)</td>
<td>Accuracy</td>
<td>ΔR(%)</td>
</tr>
<tr>
<td>3037</td>
<td>0.156</td>
<td>21</td>
<td>6754</td>
<td>0.533</td>
</tr>
<tr>
<td>5759</td>
<td>0.272</td>
<td>25</td>
<td>13100</td>
<td>0.756</td>
</tr>
<tr>
<td>9304</td>
<td>0.439</td>
<td>27</td>
<td>21429</td>
<td>0.883</td>
</tr>
<tr>
<td>18109</td>
<td>0.617</td>
<td>29</td>
<td>41929</td>
<td>0.943</td>
</tr>
<tr>
<td>26847</td>
<td>0.700</td>
<td>30</td>
<td>62215</td>
<td>0.944</td>
</tr>
<tr>
<td>35548</td>
<td>0.722</td>
<td>31</td>
<td>82398</td>
<td>0.961</td>
</tr>
<tr>
<td>44200</td>
<td>0.756</td>
<td>31</td>
<td>102474</td>
<td>0.956</td>
</tr>
<tr>
<td>52776</td>
<td>0.761</td>
<td>31</td>
<td>122490</td>
<td>0.967</td>
</tr>
<tr>
<td>61321</td>
<td>0.783</td>
<td>32</td>
<td>142322</td>
<td>0.967</td>
</tr>
<tr>
<td>69798</td>
<td>0.800</td>
<td>32</td>
<td>162096</td>
<td>0.961</td>
</tr>
</tbody>
</table>

The accuracy obtained for the four binary descriptors under evaluation is shown along with the query bitrate (in bits per query image) of the binary compressed representation. The bitrate gains $\Delta R$ obtained for the proposed BDC scheme when compared to the raw (PCM) representation are also shown. From the experimental results, the following conclusions can be taken:

1. The compression gains associated to the proposed BDC framework are highly dependent on the binary descriptor, notably up to 32% (minimum of 14%) of bitrate saving for the FREAK descriptor, and up to 18% (minimum of 9%) bitrate savings for the ORB descriptor. Thus, it is possible to conclude that the sampling patterns of each descriptor have a significant influence on the rate-accuracy performance. In general, FREAK and BRIEF are the descriptors with more inter-descriptor redundancy and the ORB descriptor with less inter-descriptor redundancy.

2. Naturally, the BCS compression efficiency improves when the number of descriptors transmitted increases. This is rather expectable since when more descriptors are available, the easier it is to find good predictors with the proposed BDC coding scheme, thus leading to less bitrate. When only a few descriptors are sent, these 'isolated' descriptors can represent parts of the image that are not well correlated.

3. In general, for the ZuBuD dataset, the compression ratio is lower than for the Torino dataset. Since the images of the Torino dataset have higher resolution, the patches do not change so significantly, spatially, (especially after Gaussian smoothing) when compared to the ZuBuD dataset. Typically, higher spatial resolutions lead to a better rate-accuracy when inter-descriptor correlation is exploited with the BDC framework.

From the results obtained, it is also possible to identify the binary descriptors for which higher accuracies are achieved. The BRISK and FREAK descriptors are the best, for the ZuBuD and Torino datasets, while the BRIEF descriptor shows the worst performance (for the Torino dataset) since it provides a representation that is not invariant to rotation or scale changes. Moreover, by plotting the rate against the accuracy, the charts of Figure 3 and Figure 4 are obtained for the proposed BDC scheme and for the PCM representation. As shown, the best RA performance was obtained when the proposed BDC scheme was applied to the PCM descriptor. This can be explained by the high matching performance that this descriptor provides, which leads to high accuracy values.
For the Torino dataset, the ORB descriptor length with a lower bitrate for the PCM representation (only 256 binary tests are made) comes second, while for the ZuBuD dataset, its lower accuracy for low bitrates deteriorates its RA performance.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, a visual sensor network scenario is targeted where camera nodes acquire visual data, extract representative features, and finally code and transmit these features for further analysis. To meet the bandwidth and energy constraints of the camera nodes, a framework is proposed which extracts binary descriptors and compresses the binary raw representation by exploiting the inter-descriptor redundancy. For an image retrieval task, the experimental results show that the proposed coding framework can lower the data rate associated to the set of binary descriptors extracted for each image (up to 32%), while preserving the retrieval accuracy.

ACKNOWLEDGMENTS

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

REFERENCES

