Clustering based Binary Descriptor Coding for Efficient Transmission in Visual Sensor Networks

Pedro Monteiro* and João Ascenso**
*Instituto Superior Técnico – Instituto de Telecomunicações, Lisbon, Portugal
**Instituto Superior de Engenharia de Lisboa – Instituto de Telecomunicações, Lisbon, Portugal

Abstract — Nowadays, local feature descriptors have emerged as one of the most promising and powerful visual representation solutions. In fact, with a minimal amount of computational effort, the detection and extraction of visual features can provide reliable and a compact image representation that enables a rich set of image analysis tasks. In this paper, a visual sensor network scenario is considered with energy and bandwidth constraints at each sensor node. In this scenario, low-level binary features can be computed with low complexity and efficient coding schemes can reduce the data rate needed to transmit the features. This paper proposes two binary descriptor coding techniques that exploit the correlation between descriptors of the same image, clustering the extracted descriptors with two solutions: divisive-clustering and agglomerative-clustering. While the former starts with one cluster containing all descriptors which is recursively divided, the latter starts with as many clusters as descriptors which are recursively grouped. The disjoint sets of descriptors are differentially coded and the prediction residual is entropy coded. The experimental results show bitrate savings up to 34% without any impact in the accuracy of the final image retrieval task.

I. INTRODUCTION

A visual sensor network (VSN) is a network of low-power camera devices (nodes) that can capture, process and transmit images of a (visual) scene using a distributed and collaborative approach [1]. Typically, camera nodes provide visual information to a central node, according to critical constrains, like energy and bandwidth consumption and communication delay. However, the transmission of medium to high-resolution images and videos require demanding visual compression schemes (such as H.264/AVC) that have significant power and bandwidth requirements considering the limitations of visual (typically wireless) sensor networks. Thus, to meet the bandwidth and energy VSN restrictions, image processing techniques must be first applied to the raw visual data to extract relevant information about the visual scene. However, due to energy efficiency restrictions, VSN platforms are usually designed with limited hardware capabilities [1], which restrict the usage of demanding vision computing algorithms, especially those requiring large databases or complex image analysis. Thus, high-level processing must occur at a central node, where high computational and storage resources are available. In such case, the sensing nodes are responsible to perform low-level tasks such as the computation of visual features and the transmission of this data to the central node. Thus, a succinct and discriminative representation must be found as well as suitable coding and transmission schemes for this new data type. Since a more compact but yet rich representation is made available at the central location, the pixel-level representation is not transmitted, thus extending the lifetime of visual sensor networks while providing rich visual information.

In the literature, there are many visual descriptors available, but for many of them it is not possible to perform efficient streaming over a wireless network with limited resources. In fact, state-of-the-art descriptors such as Scale-Invariant Feature Transform (SIFT) [2] and Speeded Up Robust Features (SURF) [3] have rather high bandwidth and power requirements. In many cases, it is more efficient to transmit the coded pixel representation of the image (e.g. with JPEG or H.264/AVC Intra) when compared to the raw descriptor representation of the image. To address the VSN limitations, low complexity binary descriptors, such as binary robust independent elementary features (BRIEF) [4], binary robust invariant scalable keypoints (BRISK) [5] and fast retina keypoint (FREAK) [6], are more suited. For the computation of binary descriptors, simple intensity difference tests are made over a smoothed image patch, thus requiring not only low energy for extraction but also a more compact representation when compared to non-binary descriptors. However, further improvements can be achieved by exploiting correlation between the binary descriptors that represent patches of the same acquired image. In fact, the compression of binary descriptors allows to obtain savings not only in bitrate, but also in energy since data transmission requires a significant amount of energy resources for a visual sensor network node.

In this paper, two lossless coding techniques are proposed: 1) coding by divisive-clustering (CDC) and 2) coding by agglomerative-clustering (CAC). In both cases, binary descriptors are coded by employing a clustering algorithm that groups similar descriptors and finds an efficient prediction path considering that all descriptors (except the Intra descriptor) in the cluster are differentially Inter coded. The proposed techniques are able to reduce the data rate necessary to represent the binary descriptors by a significant amount while maintaining the same overall quality accuracy.

In the next section, the state-of-the-art in compression of binary descriptors is presented. Section III presents the proposed clustering techniques as well the coding scheme employed and Section IV presents the experimental results. Finally, Section V presents the main conclusions and ideas for future work.

II. REVIEWING STATE-OF-THE-ART

The main target of local descriptors is to provide a meaningful representation of the image. The SIFT and SURF are two vector-based descriptors that have state-of-the-art performance for many visual analysis tasks. Both descriptors rely on a Difference of Gaussians to detect stable (invariant to viewing changes) image locations, i.e. the feature points. However, while SIFT describe the area (patch) around each keypoint with an orientation histogram from the smoothed samples for each region, SURF employs Haar wavelet filters computed over a predefined set of sample points around each keypoint.

However, for these two descriptors, the detection and extraction algorithms have complexity too high for many computationally ‘weak’ devices, such as mobile phones. So, the use of binary descriptors emerged as an interesting alternative that is able to meet the processing and energy capabilities of not only mobile phones but also of VSN nodes. The most popular binary descriptors are BRIEF, BRISK and FREAK. The BRIEF descriptor [4] uses a pre-defined sampling pattern that indicates the point pairs (patch locations) where the intensity tests (descriptor elements) are calculated; in BRIEF, a bi-dimensional Gaussian distribution is used to generate the sampling pattern. Despite its simplicity, BRIEF is not invariant to rotation and scale changes, which makes its performance worse than SIFT and SURF. The BRISK descriptor [5] proposes a new sampling pattern, based on two sets of point pairs, one set to estimate the orientation of each keypoint (to
obtain rotation invariance) and the other to build the descriptor (which is scale invariant). The FREAK descriptor [6] introduces another sampling pattern inspired by the Human Visual System characteristics, i.e. with a high density of keypoints around the center, while selecting the intensity tests with a training procedure that maximizes the descriptor discriminative power.

In the literature, there are several compression techniques of visual descriptors, especially for non-binary descriptors. The popular Compressed Histogram of Gradient (CHoG) [7], extracts gradient histogram descriptors around each keypoint, which are compressed and transmitted to the central server, and matched against a large dataset of images. The histograms are quantized with Huffman or Gagie trees, which are entropy coded. For the compression of binary descriptors, [8] proposes a method that establishes an optimal permutation of the point pairs according to their correlation. In addition, the descriptor elements are selected to maximize the discriminative power (according to bitrate constraints). Each descriptor element is a bit, which represents the result of an intensity test for a certain descriptor. After the reordering of the descriptors elements, a predictive (differential) coding method is applied to each descriptor, i.e. the redundancy between descriptor elements is exploited. Another possible approach is to exploit the redundancy between binary descriptors [9]. In [9], the descriptors are sorted with a Hamming distance metric and predictive coding is performed with a differential pulse coded modulation (DPCM) scheme as well as an adaptive binary arithmetic entropy coding method.

III. CLUSTERING BASED BINARY DESCRIPTOR CODING

In a typical binary descriptor, 512 intensity tests are performed for each detected keypoint, leading to a descriptor length of 512 bits. Although each keypoint can be represented with a lower bitrate, to achieve a complete representation of the image several hundred of descriptors may be needed, requiring a significant amount of bitrate; for example, 4 Mbit/s are needed for a video sequence at 30Hz with 300 descriptors per frame. This bitrate is rather high considering a visual sensor network device with a limited wireless communication channel (e.g. 250kbit/s) and energy resources. So, binary descriptors must be efficiently compressed to save bandwidth and terminal resources, i.e. the bitrate should be as low as possible without affecting the image analysis task accuracy. The architecture for the proposed binary descriptor encoder for visual sensor devices is presented in Figure 1.

![Figure 1. Architecture of the proposed binary descriptor encoder.](image)

First, the descriptors of each feature point are extracted with state-of-the-art binary descriptor extraction methods: BRIEF, BRISK or FREAK. After the descriptor extraction process, a matrix $A_{i,k}$ filled with descriptor elements is obtained; each $i$-th row of $A$ contains several descriptor elements associated to a detected keypoint and each $k$-th column represents an intensity test made according to the descriptor pre-defined sampling pattern. There are two main approaches to compress binary descriptors: 1) exploiting the correlation between the intensity tests ($A$ columns) by coding each $A_{i,k}$ using as reference the $A_{i,k-1}$ descriptor element or 2) exploiting the correlation between the descriptors ($A$ rows) and code each $A_{i,k}$ using as reference the $A_{i-1,k}$ descriptor element. In both cases, the order used for the intensity tests or for descriptor coding along with the Intra or Inter mode is rather relevant to obtain an efficient coding scheme. In this paper, this approach was followed and the coding of the binary descriptors proceeds as:

1. **Cluster Creation**: This module groups all the binary descriptors into $n$ disjoint clusters $C^n$. This grouping can be performed either by the divisive clustering or by the agglomerative clustering methods that are explained in sections III.A and III.B. As shown in Figure 2, each cluster is characterized by one Intra descriptor and a set of Inter coded descriptors and a prediction path that indicates the reference descriptor that must be used to code each Inter descriptor.

2. **Cluster Predictive Coding**: After obtaining the $C^n$ clusters, the descriptors of each cluster are coded with a DPCM-like scheme to exploit the correlation between descriptors, according to the prediction path already established. A first order predictor is used, where a descriptor in the prediction path is coded by calculating the difference (bitwise XOR operator) with respect to its reference descriptor.

3. **Arithmetic Coding**: Finally, the differently coded descriptors are entropy coded with a standard binary adaptive arithmetic codec, generating a bitstream that is ready to be transmitted.

![Figure 2. Descriptors clusters and prediction paths.](image)

A. **Coding by Divisive-Clustering (CDC)**

A cluster consists in a set of descriptors that are correlated in such a way that can be efficiently differentially (Inter) coded with lower bitrate than coding each descriptor independently (Intra); each cluster can contain 1 to $N$ descriptors, where $N$ corresponds to the total number of descriptors extracted. The CDC technique follows a top-down approach starting with one cluster containing all descriptors; then, it recursively divides each cluster until a minimum bitrate solution is found. The descriptors of each cluster form a prediction path (as shown in Figure 2) that classifies each descriptor as Intra/Inter and the coding order of each descriptor. To measure the correlation between two descriptors, the Hamming distance (bitwise XOR operation followed by bit count) is used. The proposed CDC technique architecture is illustrated in Figure 3.

![Figure 3. Architecture of the proposed CDC cluster creation algorithm.](image)

The proposed CDC algorithm reduces the data rate by finding the best partitioning of all descriptors with an iterative $k$-level approach:

1. **Cluster Selection**: The initial cluster at level $k = 0$ contains all extracted descriptors. At this step, one $k$-level cluster is selected for further processing with one restriction: a cluster can only be selected one time. If there are no clusters left to process, the best partitioning $C^n$ was already found and the algorithm stops.

2. **Cluster Split**: Then, the selected cluster $c$ is divided into two new smaller $(k+1)$-level clusters $a$ and $b$ containing all descriptors of cluster $c$. The objective is to find two prediction paths, one for each $(k+1)$-level cluster, as follows:

   a. Select the two descriptors $d_a$ and $d_b$ less correlated by choosing the $d_a$ and $d_b$ pair with the largest Hamming distance. The $d_a$ and $d_b$ descriptors correspond to the last descriptor of the prediction path of clusters $a$ and $b$, respectively.
b. Select the closest descriptor \( d_c \) to \( d_a \) or \( d_b \) between all descriptors of cluster \( c \).
c. Connect the closest descriptor \( d_c \) to \( d_a \) or \( d_b \). The \( d_c \) descriptor becomes the new last descriptor of the respective prediction path.
d. Steps b and c are repeated until all descriptors of the \( k \)-level cluster \( c \) belong to one of the prediction paths.

3. Rate Decision: After computing the two prediction paths defining each cluster, the bitrate to code each one is estimated. In this case, the rate \( R \) of a prediction path is calculated according to:

\[
R_k^u = H(d_1) + \sum_{i=1}^{s} H(d_i \oplus d_{i-1})
\]

where \( s \) corresponds to the total number of descriptors in cluster \( u \) (which can assume \( a \), \( b \), or \( c \) values) of level \( k \), \( d_1 \) stands for the Intra descriptor of the prediction path of cluster \( u \) and \( H \) is the entropy of the descriptor. Then, if \( R_k^{u-1} + R_k^{u+1} < R_k^u \) it is worthwhile to divide the \( k \)-level cluster into two new smaller \((k+1)\)-level clusters; otherwise, the algorithm goes back to step 1 to evaluate the next \( k \)-level cluster. With such approach it is possible to guarantee that the iterative algorithm converges to a solution with minimum bitrate and each split decision lowers the amount of rate necessary to code each cluster.

4. Cluster Storage: If the decision taken in the previous step is to divide the \( k \)-level cluster, the two new \( a \) and \( b \) clusters are stored to be further processed and the algorithm goes back to step 1.

4. **Coding by Agglomerative Clustering (CAC)**

The CAC objective is the same as CDC, i.e. to obtain partition of the descriptors into clusters. Since each cluster is coded by a predictive coding scheme applied to each prediction path, the partitioning found must minimize the compressed descriptor data rate. Initially, the CAC algorithm creates a set of clusters, each one with just one descriptor, which are then grouped together. The architecture of the proposed CAC cluster creation technique is illustrated in Figure 4.

![Figure 4. Architecture of the CDC cluster creation algorithm.](image)

The proposed CAC algorithm also follows an iterative \( k \)-level approach as described in the following:

1. **Correlation Matrix Creation:** Initially, the Hamming distance between all descriptors is computed and stored. The distance values obtained are used in the next step, to decide which descriptors (and clusters) are merged.

2. **Cluster Selection:** At level \( k = 0 \), each extracted descriptor belongs to a cluster with one element. Then, two clusters are selected to be (possibly) merged. As in CDC, each cluster is represented by a prediction path \( i \) and the last descriptor \( d_i \). Thus, the clusters \( a \) and \( b \) selected correspond to the clusters for which their respective last elements \( d_a \) and \( d_b \) can be coded with lower entropy, i.e. with minimum or maximum Hamming distance. This leads to prediction residuals with large or smaller probabilities of zeros, which can be efficiently arithmetic coded. In addition, to create a low entropy connection between clusters, it is also verified if it is worthwhile to exchange the Intra descriptor (first element of the prediction path) with the last descriptor.

3. **Cluster Merge:** The selected clusters are connected by their last elements to create a new cluster \( c \) with one prediction path.

4. **Rate Decision:** After the two closest clusters are merged, it is evaluated if the merging operation is beneficial or not from the bitrate saving point of view. As in CDC, (1) is used to evaluate the rate cost of each cluster \( a \), \( b \), and \( c \). In the previous iteration \( k - 1 \), the rate of the clusters \( a \) and \( b \) was already computed. Thus, the clusters \( a \) and \( b \) should be merged into cluster \( c \) if \( R_k^{a-1} + R_k^{b-1} > R_k^c \); otherwise it is better to keep the separated clusters. If clusters are not merged, the algorithm proceeds to step 2; otherwise it proceeds to the next step.

5. **Cluster Storage:** The merged cluster \( c \) is stored and the clusters \( a \), \( b \) are removed.

The algorithm stops by verifying in step 2 the following stopping criterion: if all possible connections (cluster merge) between clusters were evaluated and none has brought coding efficiency improvements, the final partitioning of the descriptors into clusters was found.

IV. **EXPERIMENTAL RESULTS**

In this section, the binary descriptor coding solution is evaluated with both CDC and CAC clustering creation techniques. The evaluation is done by comparing the rate-accuracy performance for the image retrieval task. Basically, pairwise matching between the decoded descriptors of the query and those already stored in the database is performed. The popular ratio test has been adopted to filter (reject) wrong matches between the query and database descriptors and the Random Sample Consensus algorithm is used to geometrically constraint the filtered matches [9].

A. **Test Conditions**

To perform the rate-accuracy evaluation, two datasets were used: ZuBuD [10] and Torino [11]. These datasets are mainly composed by building, with various viewpoints and different weather and lighting conditions. The ZuBuD dataset contains 1005 images of 201 buildings in the city of Zurich, while the Torino dataset contains 1260 images of 180 buildings in the city of Torino. In respect to the query images, the ZuBuD dataset already has a set of images, while in the Torino dataset, one image of each building was removed from the dataset to create the queries. Both the ZuBuD and Torino datasets have a spatial resolution of 640 × 840; for the Torino dataset, each image was spatially resized to this resolution.

To evaluate the proposed coding solutions, the BRISK and FREAK binary descriptors were tested for both datasets, while the BRIEF was only tested for the Torino dataset, since ZuBuD query images have a different spatial resolution (320 × 240) and the BRIEF descriptor is not scale invariant. To guarantee the same patches for each extractor, the SURF keypoint detector was used in every case. The SURF keypoint detector and the BRIEF, BRISK and FREAK descriptor extractor implementations used can be found in the OpenCV 2.4.5 library. To properly evaluate the rate-accuracy performance, 8 rate-accuracy points were defined. These rate-accuracy points intend to evaluate the accuracy performance of several bitrate budgets by varying the number of descriptor from 15 to 400. To select which descriptors are coded, the keypoint detector extracts a maximum of 400 descriptors which are sorted according to their Hessian response [3], a reliable method used in the past for this purpose. The accuracy is obtained based on the number of correct (positive) matches of all the queries [9]. The bitrate correspond to the average bitrate of all queries, i.e. 115 for the ZuBuD dataset and 180 for the Torino dataset. Note that additional information needs to be sent to the decoder to inform if each descriptor is coded as Intra or Inter, which means 1 bit per extracted descriptor. However, binary arithmetic coding is applied to this overhead information, which significantly reduces the impact in the final bitrate.

B. **Experimental Results**

The Rate-Accuracy performance results for the ZuBuD and Torino datasets are presented on Table I, where positive values of \( \Delta R_{CDC} \) and \( \Delta R_{CAC} \) represent the bitrate savings regarding the PCM rate (raw data representation).
Table I. CDC and CAC Rate-Accuracy Results for the ZuBuD and Torino Datasets.

<table>
<thead>
<tr>
<th>Number of Descriptors</th>
<th>ZuBuD Dataset</th>
<th>Torino Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BRISK</td>
<td>FREAK</td>
</tr>
<tr>
<td></td>
<td>Accuracy [%]</td>
<td>PCM Rate [bits/mb]</td>
</tr>
<tr>
<td>15</td>
<td>35.7</td>
<td>7680</td>
</tr>
<tr>
<td>30</td>
<td>62.8</td>
<td>15380</td>
</tr>
<tr>
<td>50</td>
<td>81.7</td>
<td>25600</td>
</tr>
<tr>
<td>100</td>
<td>87.8</td>
<td>51200</td>
</tr>
<tr>
<td>150</td>
<td>91.3</td>
<td>76800</td>
</tr>
<tr>
<td>200</td>
<td>94.8</td>
<td>102400</td>
</tr>
<tr>
<td>300</td>
<td>95.7</td>
<td>145818</td>
</tr>
<tr>
<td>400</td>
<td>96.5</td>
<td>170465</td>
</tr>
</tbody>
</table>

As expected, the accuracy values are the same for both proposed CAC and CDC solutions, sincelossless coding is employed. From the obtained results, it is possible to conclude that:

- **Impact of the number of descriptors**: The bitrate savings are highly related to the number of descriptors used. In fact, more descriptors lead to more bitrate gains because the distance between descriptors is low, thus creating lower prediction paths. When few descriptors are used, they are not well correlated, so the bitrate gains are lower. However, in such cases, the accuracy of the image analysis task is rather low.

- **Impact of the binary descriptor coding scheme**: The bitrate gains are also highly related to the binary descriptor used. For the CAC, gains of 11.1% to 21.8% are obtained for the BRISK descriptor, while FREAK obtain gains that go from 15.7% to 27.4% and in BRIEF the bitrate savings range from 23.6% to 33.9%.

- **Comparison between cluster creation schemes**: The CAC overcomes the CDC in every case. For example, the CDC technique obtains bitrate gains of 31.9% for the BRIEF descriptor in the Torino dataset (with 400 descriptors), while the CAC technique achieves 33.9%. This behavior occurs because CAC allows the creation of more flexible prediction paths and a more efficient selection of the intra descriptors (first element in each path). However, independently of the chosen binary descriptor both techniques obtain significant and somewhat similar gains with very different final partitions.

Overall, it is possible to conclude that both proposed lossless binary descriptor coding schemes are able to significantly reduce the amount of bitrate necessary for binary descriptor, especially considering that the accuracy is kept the same. Thus, this technique can lower the energy and bandwidth requirements of a visual sensor network.

V. CONCLUSIONS

In this paper, two binary descriptor coding techniques are proposed: Coding by Divisive Clustering and Coding by Agglomerative Clustering. While the CDC considers one cluster with all of the descriptors which is recursively divided into several clusters, the CAC considers each descriptor individually and attempts to merge each one in a cluster. The CDC obtains bitrate gains up to 31.9%, while the CAC achieves gains of 33.9%. As future work, it is proposed to design a binary descriptor coding scheme that exploits the correlation inside the binary descriptors, i.e., the order of the binary descriptors intensity tests.

ACKNOWLEDGEMENT

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