A Joint Compression Scheme for Local Binary Feature Descriptors and their Corresponding Bag-of-Words Representation

Dominik Van Opdenbosch, Martin Oelsch, Adrian Garcea, Eckehard Steinbach

Chair of Media Technology, Technical University of Munich, Germany

dominik.van-opdenbosch@tum.de

***Abstract*—For real-time computer vision tasks, binary feature descriptors are an efficient alternative to their real-valued counterparts. While providing comparable results for many applications, the computational complexity of extracting and processing binary descriptors is significantly lower. In many application scenarios, the local features are transmitted over a channel with limited capacity and processed at a more powerful central processing unit, which requires efficient compression and transmission approaches. In this paper, we present a compression scheme for local binary features, which jointly encodes the descriptors and their respective Bag-of-Words representation using a shared vocabulary between client and server. By sending the visual word index and the entropy-coded residual vector containing the differences between the visual word and the descriptor, we are able to reduce ORB features to 60.62 % of their uncompressed size.**

***Index Terms*—Visual features, binary descriptors, Bag-of-Words, ORB, feature coding, ATC**

I. Introduction

Local visual features are still one of the main tools for visual analysis tasks. However, the application scenarios for visual analysis have shifted. With mobile devices being able to capture images at any corner of the world and mobile robots performing collaborative exploration tasks, one promising idea to reduce the required on-system computational resources is to transfer the information about the content of the images to more powerful processing nodes in the network (e.g. for cloud-based visual analysis). As most of the relevant visual analysis tasks are still quite challenging for low power devices two approaches have been proposed to overcome the on-device processing limitations [1]. The first, often referred to as ”Compress-then-Analyze” (CTA) [2], uses image and video compression to transmit the information to a central processing node where the compressed images are reconstructed and the desired task including feature extraction is performed. While this approach has the benefit of having the image data available for more in-detail analysis, the image quality is limited by the available transmission capacity. The alternative approach is to extract the relevant information for the desired task directly at the mobile device (”Analyze-then-Compress” - ATC) [3]. Usually these are local image features, which are subsequently transmitted to the central computing node for the visual analysis task. Often, these local features are quantized to their Bag-of-Words representation and a lossy representation is sent in the form of visual word indices to the server. However, for many applications it is beneficial to have the unquantized descriptors available at the processing node. This allows for additional outlier removal using for example the matching distance ratio between the best and second best match [4]. In this paper, we propose a feature-coding scheme, which exploits the dependency between feature descriptors and their associated visual words for efficient binary feature compression. To this end, we take a Bag-of-Words vocabulary as shared knowledge, analyze the statistics of this vocabulary and propose to code the features by sending the visual word index and the entropy-coded residual vector containing the difference between the visual word and the descriptor.

The rest of this paper is organized as follows. In Section II, we discuss related work. Section III describes the proposed coding strategy. In Section IV, we provide experimental results and Section V concludes the paper.

II. Related Work

For the CTA approach, there are several choices for designing feature preserving image [5] and video coding schemes [6]. However, while most parts of the image might be irrelevant for the targeted analysis task, we will focus on an ATC-based scheme. Starting with real-valued descriptors like SIFT [4] and SURF [7], Baroffio et al. proposed to code visual features from video sequences using a scheme inspired by hybrid video coding [3]. They propose to use a coding approach that supports intra- and inter coding schemes to exploit both intra-descriptor and inter-frame redundancies. While this work was based on real-valued descriptors, the binary descriptors became a faster alternative suitable for most computer vision tasks. One of the first binary descriptors was BRIEF [8], which uses an ensemble of pixel tests on a fixed pattern around a keypoint location. However, this first approach lacks rotation and scale invariance which was later addressed by ORB [9]. In the following different binary descriptors such as

978-1-5386-0462-5/17/$31.00 2017 IEEE

BRISK [10] and FREAK [11] were introduced. Following this development, Baroffio et al. extended their approach to binary features [12]. In addition, the authors presented an approach to transmit a compressed version of a global image signature, namely the Bag-of-Words representation created from the local features. Redondi et al. [13] proposed a scheme for descriptor element selection for lossy compression of the features.

While previous work mainly focuses on the independent coding of local image descriptors and global image signatures, this work presents a scheme that exploits the dependencies between both image descriptions for improved compression performance.

III. Proposed Compression Scheme

In the following, we describe our joint compression approach in detail. Using the notation of [12], we denote the j-th element of a binary descriptor $\mathbf{d} \in \{0,1\}^P$ with d_j, where $j \in [1, P]$ and P denotes the size of the binary descriptor ($P = 256$ for ORB). Let $\mathbf{C} = \{\mathbf{c}_1 \ldots \mathbf{c}_S\}$ be a visual vocabulary consisting of S visual words. In order to create a Bag-of-Words representation, we assign each binary descriptor to the closest visual word $\mathbf{c}_i$ in terms of Hamming distance within the visual vocabulary such that $\mathbf{c}_i = \text{NN}(\mathbf{d})$, where NN defines the nearest neighbor. We use a hierarchical clustering scheme, where the maximum vocabulary size S can be calculated as $S = k^l$ and the tree can efficiently be traversed even on power-restricted devices. We denote the branching factor with k and the depth of the tree with l. The main motivation for our proposed compression scheme is the observation that, according to our experiments, the Hamming distance of the descriptors $\mathbf{d}$ assigned to a visual word follows a distinctive Gaussian distribution and is minimized when increasing the vocabulary size. In Figure 1, we show the distribution of the Hamming distance for all descriptors being assigned to a visual word as nearest neighbor (blue) and the distance to non-matching visual words (red) for different vocabulary sizes. On the left, we show the results for a vocabulary of size $S = 10^3$ and on the right the statistics for a vocabulary of size $S = 10^5$ obtained from our training database, as explained in Section IV. The larger the vocabulary size, the smaller the average Hamming distance between the visual word and the assigned descriptors. The key idea of our proposed scheme is to transmit the visual word index and the residual vector between the visual word and the descriptor using a common vocabulary as shared knowledge. We compute the residual vector $\mathbf{r}$ between the visual word and the descriptor by using the XOR operation as $\mathbf{r} = \mathbf{c}_i \oplus \mathbf{d}$, which can be inverted at the decoder side using XOR between the residual vector and the visual word. While the residual vector contains mostly zeros, we can use an entropy coding scheme such as arithmetic coding to compress the information.

A. Bag-of-Words Coding

The choice for coding the visual word indices is straightforward. If $k = 2$, then we employ fixed-length coding resulting in $R_{bow} = \log_2(k^l)$ bits. If the vocabulary size should be more flexible, we use arithmetic coding which approaches the lower bound for the visual word index rate given as

$$R_{bow} = -\sum_{i=1}^{S} p_c(i) \log_2(p_c(i)) \tag{1}$$

where $p_c(i)$ denotes the probability of visual word $\mathbf{c}_i$. This probability is either learned on a similar dataset or assumed to be uniformly distributed with $p_c = k^{-l}$.

B. Residual Coding

In order to obtain the probabilities of the residual vector at any position r_j being zero, we can either directly use

$$p_r(r_j = 0) = 1 - \frac{\mu_{inl}}{P} \tag{2}$$

where μ_{inl} denotes the mean inlier distance for a given vocabulary size from the distribution as shown in Figure 1, or we can analyze the data in more detail and use the statistics per descriptor element $p_{r,j}(r_j = 0)$ as illustrated in Figure 2. Although the statistics are averaged over all visual words, there are differences between the residual elements r_j which can be further exploited for coding. Following this approach, we can calculate the lower bound as

$$R_{res} = \sum_{j=1}^{P} H_r(r_j) \tag{3}$$

where $H_r(r_j)$ denotes the entropy of each residual element and is calculated as

$$H_r(r_j) = -p_{r,j}(0) \log_2(p_{r,j}(0)) - p_{r,j}(1) \log_2(p_{r,j}(1)) \tag{4}$$

Due to the binary nature of the residual elements, there are only two terms in (4). It is possible to gather the statistics per visual word and per residual element, but with increasing vocabulary sizes this solution is becoming impracticable due to storage constraints.

C. Keypoint Coding

Besides the feature descriptor, the keypoint position of the feature in pixel coordinates, the scale pyramid level used for ORB extraction and the orientation information is required for most visual analysis tasks. For the keypoints, we follow the findings from [12] and quantize the keypoint positions to quarter pixel accuracy. The orientation information is quantized into 32 bins of size $\pi/16$ and for the scale level the number of bits used is defined by the settings of ORB. Assuming a floating point representation with four bytes each for x,y coordinates of the keypoint, orientation and an additional byte for the scale level, the size of the uncompressed keypoint is given as $R_{kpt,u} = (4 + 4 + 4 + 1) \times 8 = 104$ bits. The size for the compressed keypoint depends on the width w and the height h of the image and the number of scale levels n and is calculated as

$$R_{kpt,c} = \log_2(4 \times w) + \log_2(4 \times h) + \log_2(32) + \log_2(n) \tag{5}$$

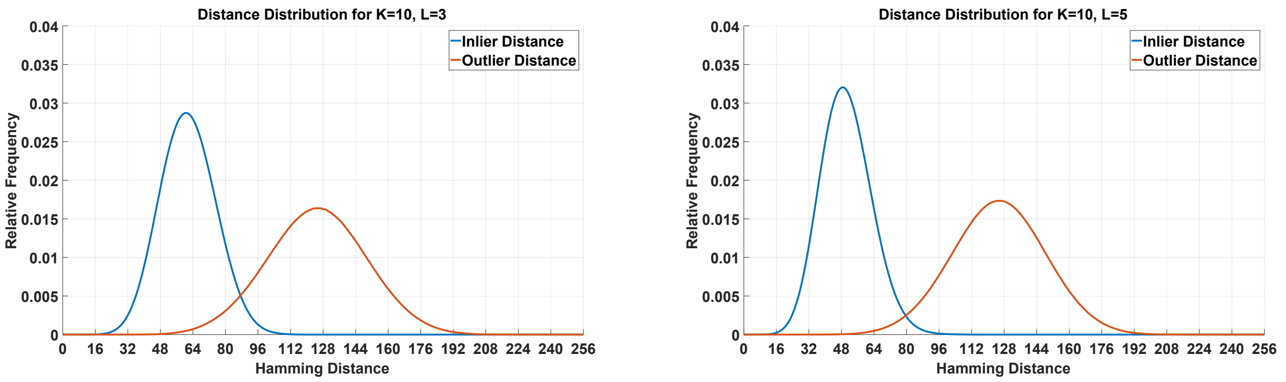

Fig. 1. Distribution of the hamming distance between inlier and outlier descriptors for vocabulary sizes $S = 1000$ (*left*) and $S = 100000$ (*right*). Inlier distance is the distance of a descriptor to its assigned visual word and outlier distance denotes the distance to non-matching visual words.

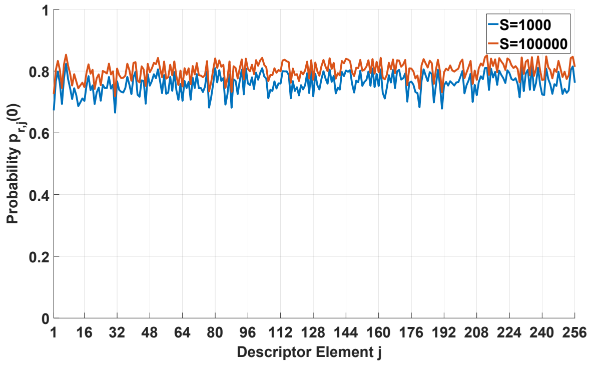

Fig. 2. Probability $p_{r,j}$ of residual element r_j being zero for vocabulary sizes $S = 1000$ and $S = 100000$.

D. Feature Coding

The lower bound for the rate R to encode a local feature is then calculated by combining (1), (3) and (5):

$$R = R_{bow} + R_{res} + R_{kpt,c} \tag{6}$$

IV. Experiments

A. Experimental Setup

In order to evaluate the proposed coding scheme, we extracted ORB features using OpenCV from the MIRFlickr 1M [14] dataset as training data. We trained several vocabularies using an hierarchical implementation of Bag-of-Words for binary descriptors, namely DBoW2 [15], with branching factor $k = 10$ and depths in the range of $l \in \{1, \ldots, 6\}$ using the descriptors from the first 100.000 images of the dataset. We used descriptors from the remaining part of the training dataset to obtain the statistics used for coding. For the visual analysis task, we followed the same approach as [3] which is briefly explained in the following. We used a homography estimation scenario using the public dataset from [16] which provides different video sequences (*brick, building, mission, pairs, sunset and wood*) showing a planar texture where the camera movement is subject to different motion patterns. Ground truth is provided by the authors in form of a homography matrix warping the planar texture into a common reference frame. Each video sequence has an image resolution of 640 × 480 pixels and consists of 500 frames captured at 15 fps. Similar to [3], we downsampled the sequence by a factor of five resulting in 100 frames with significant motion between the individual frames. The evaluation metric is defined as follows: For each frame, ORB features are only extracted within the region of the planar texture defined by the ground truth. These features are then encoded and decoded and a homography is estimated using matched features between the current and the previous frame using a RANSAC scheme. The four corner coordinates of the bounding box of the planar texture from the ground truth data are then warped from the previous image into the current image using the estimated homography and compared with the coordinates provided by the ground truth. If the mean error is larger than 3 pixels, the estimated homography is counted as outlier. The *homography estimation precision* is defined as the ratio between the number of correct estimates and the number of frames.

B. Results

The results of our proposed compression scheme are shown in Figure 3 for different vocabulary sizes for the *sunset* unconstrained motion sequence. As a reference, we show the size of the uncompressed visual feature, which is $m = 256+104 = 360$ bits. In comparison, we show the compression results over different vocabulary sizes. We notice that the keypoint compression is quite effective reducing the required data rate from 104 bits to $H_{kpt,c} = \lceil \log_2(4 \times 640) \rceil + \lceil \log_2(4 \times 480) \rceil + \lceil \log_2(32) \rceil + \lceil \log_2(8) \rceil = 31$ bits using fixed-length coding. For the descriptor part, with growing vocabulary size, we have to spend more bits on signaling the corresponding visual word index but the bits spent increase the efficiency of the residual coding by raising the probability of residual elements being zero. For example, at a visual vocabulary size of $S = 10$, the number of bits used for transmitting the visual word index is 3.32 bits on average, whereas we need 230.1 bits for the residual vectors using arithmetic coding. When changing to $S = 100000$, the average number of bits for the visual word index increases to 16.63 bits, whereas the number

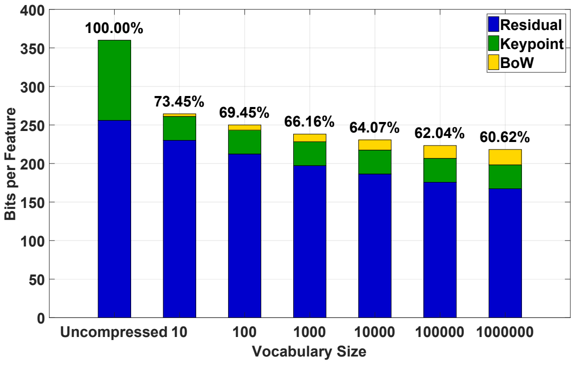

Fig. 3. Comparison of the number of bits required per feature for different vocabulary sizes extracted from the *sunset* unconstrained motion sequence.

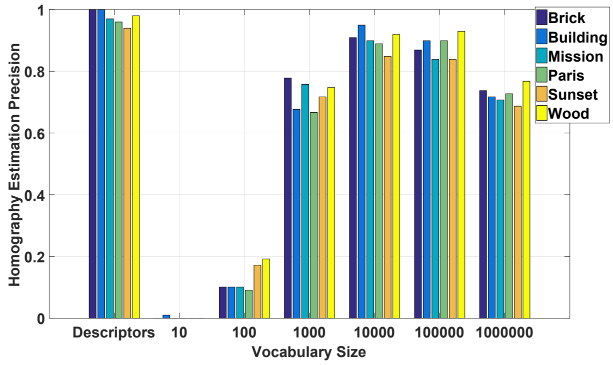

Fig. 4. Homography estimation precision for different vocabulary sizes. On the left, the original descriptors are used for establishing point correspondences. On the right, the corresponding visual words are used for matching.

of bits for the residual decreases to 175.70 bits on average in our experiments. In summary, we are able to encode the local image descriptor including the keypoint and the corresponding visual word index with 218.23 bits for $S = 1000000$ resulting in 60.62 % of the uncompressed size.

In Figure 4 we show the advantage of having the reconstructed descriptor at hand in a visual analysis task. We show the *homography estimation precision* by using descriptor matching and visual word matching. For the descriptors, we use the best match according to minimum Hamming distance and a cross check in order to identify feature matches for the homography estimation. For the Bag-of-Words based approach, we use matching visual word indices to establish the point correspondences. For all experiments, we used the decoded keypoints, which are quantized to quarter pixel resolution. For small vocabulary sizes, the number of false matches is pre-dominant leading the RANSAC-based estimation scheme to fail. Starting with vocabulary sizes of $S = 1000$, the matching allows a reliable estimation of the homography matrix. The performance decreases for larger vocabulary sizes, when similar feature descriptors are assigned to different visual words. Although the Bag-of-Visual words representation provides reasonable performance, it is still outperformed by matching the reconstructed descriptors.

V. Conclusion

In this paper, we propose an efficient method to jointly compress local binary features and their corresponding visual words by coding the residual vector between the feature descriptor and the visual word. The results show a reduction to 60.62 % of the uncompressed visual feature for a vocabulary of size 1000000. Similar to related work, this approach can easily be extended to video sequences and different visual analysis tasks in the future.

Acknowledgment

This work is supported by the space agency of the German Aerospace Center with funds from the Federal Ministry of Economics and Technology on the basis of a resolution of the German Bundestag under the reference 50NA1515.

References

[1] A. Redondi, L. Baroffio, L. Bianchi, M. Cesana, and M. Tagliasacchi, “Compress-then-Analyze vs Analyze-then-Compress: what is best in Visual Sensor Networks?” *IEEE Transactions on Mobile Computing*, vol. 1233, no. c, pp. 1–1, 2016.

[2] G. Gualdi, A. Prati, and R. Cucchiara, “Video streaming for mobile video surveillance,” *IEEE Transactions on Multimedia*, vol. 10, no. 6, pp. 1142–1154, 2008.

[3] 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.

[4] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[5] J. Chao, H. Chen, and E. Steinbach, “On the design of a novel JPEG quantization table for improved feature detection performance,” *International Conference on Image Processing (ICIP)*, 2013.

[6] J. Chao, R. Huitl, E. G. Steinbach, and D. Schroeder, “A Novel Rate Control Framework for SIFT/SURF Feature Preservation in H.264/AVC Video Compression,” *IEEE Trans. Circuits Syst. Video Techn.*, vol. 25, pp. 958–972, 2015.

[7] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-Up Robust Features (SURF),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[8] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “BRIEF: Binary Robust Independent Elementary Features,” in *European Conference on Computer Vision (ECCV)*, 2010.

[9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” *International Conference on Computer Vision (ICCV)*, 2011.

[10] S. Leutenegger, M. Chli, and R. Y. Siegwart, “BRISK: Binary Robust invariant scalable keypoints,” *International Conference on Computer Vision (ICCV)*, pp. 2548–2555, 2011.

[11] A. Alahi, R. Ortiz, and P. Vandergheynst, “FREAK: Fast retina keypoint,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 510–517.

[12] L. Baroffio, A. Canclini, M. Cesana, A. Redondi, M. Tagliasacchi, and S. Tubaro, “Coding Local and Global Binary Visual Features Extracted from Video Sequences,” *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3546–3560, 2015.

[13] A. Redondi, L. Baroffio, J. Ascenso, M. Cesana, and M. Tagliasacchi, “Rate-accuracy optimization of binary descriptors,” *International Conference on Image Processing (ICIP)*, 2013.

[14] M. J. Huiskes and M. S. Lew, “The MIR flickr retrieval evaluation,” *ACM International Conference on Multimedia Information Retrieval*, vol. 4, no. November, pp. 39–43, 2008.

[15] D. Gálvez-López and J. D. Tardós, “Bags of binary words for fast place recognition in image sequences,” *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[16] S. Gauglitz, T. Hoellerer, 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.