FLEXIBLE RATE ALLOCATION FOR LOCAL BINARY FEATURE COMPRESSION

Dominik Van Opdenbosch, Eckehard Steinbach

Chair of Media Technology, Technical University of Munich

ABSTRACT

Numerous real-time applications in computer vision rely on finding correspondences between local binary features. In many mobile scenarios, the visual information captured at a sensor node needs to be transmitted to a processing server, which is capable of storing the visual information or executing a complex analysis task. However, not necessarily all the visual information need to be transmitted. In this paper, we present a rate allocation scheme that is capable of categorizing features into classes according to their usefulness and select the amount of data spent on each class to maximize the overall performance of a computer vision task. We demonstrate the approach using ORB, BRISK, and FREAK features and show the improvements on a homography estimation task.

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

1. INTRODUCTION

One of the most important requirements for many computer vision applications is the ability to reliably detect correspondences between two images. To this end, local features are often used to describe and match salient parts of images. While this fundamental principle has not changed in the recent past, the application scenarios certainly have. Although mobile robots are conquering the last unknown parts of our planet and mobile devices offer services at any corner of the world, power consumption is still an issue. A feasible solution to allow the mobile application to benefit from the results of complex computer vision tasks is to transmit the visual information to powerful processing nodes, which enable computer vision tasks as cloud-based services. There are two main approaches to transfer visual information from the mobile device to a server [1]. The first one is the concept of "Compress-then-Analyze" (CTA), where an image sequence is compressed using video coding and transmitted to the server [2]. While the visual information is preserved for manual inspection, this approach usually includes redundant information for the targeted computer vision task, thus squandering limited resources. The alternative is to extract the visual information in form of local image descriptions at the mobile device and send a stream of compressed visual features to the processing node. This approach is often referred to as "Analyze-then-Compress" (ATC). In this paper, we extend an existing ATC approach [3] that provides a coarse approximation of binary visual descriptors using a visual vocabulary as shared knowledge. This coarse approximation is refined by sending residual information containing the missing information to reconstruct the original descriptor. Our main contribution is to extend this approach with a novel rate allocation scheme. Our proposed approach consists of three parts: First, we rearrange the descriptor elements according to their entropy in descending order to ensure that we prioritize the most informative residual information to reconstruct the descriptor. Second, for each image we propose to sort the local features into different classes according to their usefulness estimated from the detector response. Third, we use utility functions to determine how many descriptor elements we should reconstruct for each class in order to maximize the overall task performance while staying below a target bitrate.

The rest of this paper is organized as follows: In Section 2, we discuss relevant related work. Section 3 recapitulates the feature coding and introduces our novel rate allocation scheme. In Section 4, we provide experimental results and Section 5 concludes the paper.

2. RELATED WORK

Several works in the context of CTA have addressed the issue of adapting the objective of the rate-distortion optimization of image [4, 5] and video coding schemes [2] to the requirements of machine-based analysis. In this work, we focus on the ATC approach, where related work introduces concepts such as inter-frame coding known from hybrid video coding to feature coding starting with floating point features like SIFT and SURF [6]. Later, these approaches have been extended to binary features [7]. In previous work [3], we presented a joint compression scheme for binary descriptors and their corresponding visual word representation which uses a visual vocabulary as shared knowledge. For descriptor element selection, Redondi et al. [8] compared whether it is more useful to send more informative or less bitrate consuming elements first. With the focus on image retrieval, MPEG standardized an approach for compact descriptors for visual search (MPEG-CDVS) [9] using SIFT-like features. It includes a feature selection optimized for visual search using different

keypoint properties such as detector response, position in the image and keypoint orientation, as proposed in [10]. MPEG also collects proposals for the Compact Descriptors for Video Analysis standard (MPEG-CDVA) [11], which includes exploiting the visual similarities between successive frames using inter-frame coding, and deep learning based features. Visual SLAM is an example for a computationally complex task, where both ATC [12] and CTA-based approaches [13] have been employed. The latter approach sorts local binary features into different classes according to their usefulness and prioritizes more important features. If the channel capacity is reached, the remaining features are skipped, which is not the optimal solution for the rate allocation problem. We propose to improve this mechanism to select for each feature class the amount of information that should be transmitted to maximize the overall task performance. In contrast to the related work [8, 7], we usually have a coarse approximation of the full descriptor in form of the visual word available and propose a much more flexible approach for allocating the bits allowing features from the same image to be approximated at different precision levels. For the evaluation, we have chosen a basic homography estimation scenario, but the concept can be transferred to different computer vision tasks relying on feature matching. In addition, our rate allocation scheme can also be adapted to work with inter-frame and alternative coding schemes [7].

3. FEATURE COMPRESSION

In this section, we first introduce the general binary feature coding framework [3]. Then, we propose to reorder the residual elements to reconstruct the descriptor elements with the highest entropy first. Afterwards, we use the detector response as an indicator how useful a specific feature is and sort the features accordingly into N_g classes. For each class, we obtain a utility function determining how much improvement we expect when sending additional residual elements. We pre-calculate the coding parameters offline to be able to select on the fly the optimal number of residual elements for each class given a target bitrate. We use classes as the signaling required for sending the individual optimal number of residual elements per feature would outweigh the gain.

3.1. Local Binary Feature Coding

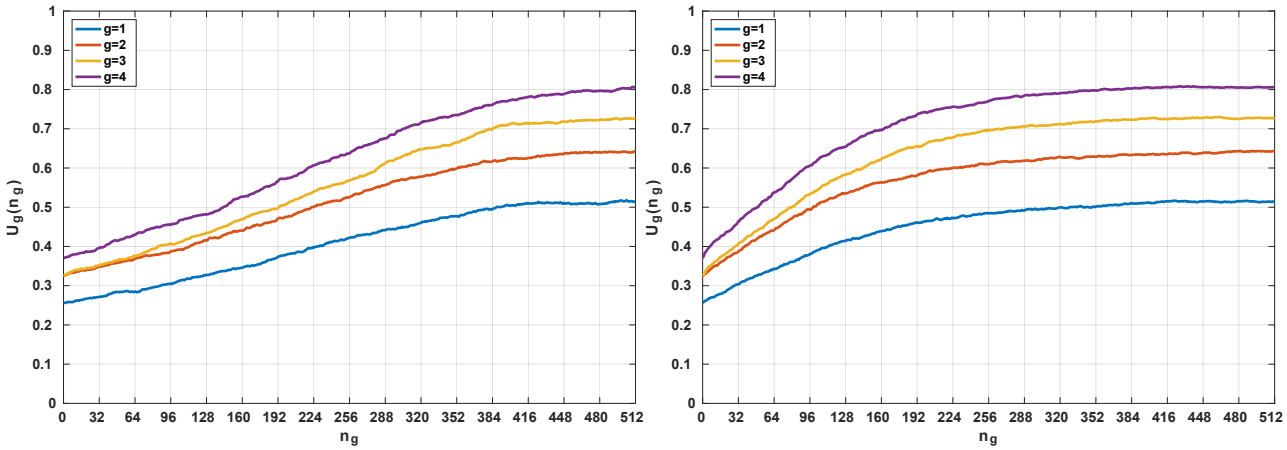
We will start by briefly recapitulating the coding scheme, which is based on the joint compression of binary feature descriptors and their corresponding visual word representation [3]. The motivation for this coding method is that many applications, such as content-based image retrieval with subsequent geometric verification, need both the local features and a global image representation. We follow the existing notation and 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 [14], $P = 512$ for BRISK [15] and FREAK [16]). We use a hierarchical visual vocabulary $\mathbf{C} = \{\mathbf{c}_1 \dots \mathbf{c}_S\}$ with S visual words. Each visual descriptor can be quantized to its visual word representation by nearest neighbor search $\mathbf{c}_i = \text{NN}(\mathbf{d})$. We encode the descriptor information by first sending the Bag-of-Words index using $R_{bow} = \log_2(S)$ bits and then transmitting the residual vector between the visual word and the actual descriptor using the XOR operation as $\mathbf{r} = \mathbf{c}_i \oplus \mathbf{d}$. We approach the lower bound for coding the binary residual vector given by the entropy as

$$R_{res} = \sum_{j=1}^{P} -p_0(j) \cdot \log_2(p_0(j)) - p_1(j) \cdot \log_2(p_1(j)), \quad (1)$$

by using arithmetic coding, where $p_0(j)$ is the probability of a residual element j being zero, and $p_1(j)$ denotes the probability of this element being one. At the decoder side, the original descriptor is reconstructed by applying XOR between the residual vector and the visual word. In order to perform most visual analysis tasks, we need additional information such as the location of the visual feature in the image. Therefore, we send the x and y coordinates quantized to quarter pixel resolution, the orientation of the keypoint quantized into 32 bins and the octave of the extracted feature resulting in $R_{kpt} = \lceil \log_2(4 \times w) \rceil + \lceil \log_2(4 \times h) \rceil + \log_2(32) + \lceil \log_2(n_\sigma) \rceil$ bits per feature using fixed-length coding, where w and h denote the width and height of the image respectively, and n_σ the number of octaves. Using this coding scheme has several advantages. Just sending the visual word index already provides an approximation of the original descriptor. When sending the full residual vector, we can reconstruct the original descriptor but we can also send only parts to approach the original representation depending on the available channel capacity. In the following, we use the concept of residual reordering to send the most useful information first.

3.2. Residual Reordering

In contrast to the coding scheme proposed by Baroffio et al. [7], we do not rearrange the descriptor elements to exploit dependencies among the descriptor entries, but we rather reorder the residual elements to reconstruct the corresponding descriptor elements with the highest entropy first. An evaluation of a related concept is presented in [8]. Our approach is similar to the greedy strategy included in ORB and FREAK. In an offline procedure, we order the descriptor elements according to their entropy in a training set T. Then, we select from T the element t with the highest entropy, add it to the sorted set R and remove it from T. We iteratively repeat this procedure and simultaneously check, if the correlation of descriptor element t with all elements already in R is lower than a threshold, otherwise we select the next best element from T. Intuitively, elements that are correlated with elements already



(a) Utility functions without reordering.

(b) Utility functions with reordering.

Fig. 1: Comparison of the utility functions $U_g(n_g)$ for BRISK features using four classes at a vocabulary size of S=100,000 over varying number of residual elements n_g. Left: Without reordering, the function increases nearly linearly until it reaches saturation. Right: With reordering, the function reaches saturation after fewer elements.

in R would not provide additional information. If no element is found, we increase the threshold and try again. The result is an ordered list of descriptor elements in R sorted according to the entropy with correlated elements placed further towards the end of the list. We use this order to send the corresponding residual elements first. However, because ORB and FREAK use a similar approach, the impact on these descriptors is limited. To ensure generalization, we add this step regardless of the underlying feature descriptor.

3.3. Feature Classification

In order to classify the visual features according to their usefulness, we rely on the detector response, which has been shown to be correlated with correct matching probability [10, 13]. Different from our previous work [13] that uses a fixed quantization scheme, we propose to sort the features according to their detector response values and categorize the features into N_g different classes such that every class $g \in [1, N_g]$ contains a fixed percentage $0 \leq p_g \leq 1$, $\sum_{g=1}^{N_g} p_g = 1$ of the total number of features. Starting with the lowest detector responses being sorted into g_1. The advantages are twofold: First, this splitting is independent of fixed thresholds which are dependent on image content and contrast. Second, fixing the percentage of features per class allows us to pre-calculate the optimal decisions for the number of residual elements for each class in advance.

3.4. Rate Allocation

In order to allocate the optimal number of bits to every class, we use a utility function $U_g(n_g)$ which returns a score depending on the number of residual elements $n_g \in [0, P]$ transmitted. Second, we pre-calculate the number of bits $R(n_g)$ for the residual vector depending on the elements n_g. The final problem can be formulated as

$$\begin{aligned} \max \quad & \sum_{g=1}^{N_g} p_g \cdot U_g(n_g), \\ \text{with} \quad & N \cdot \sum_{g=1}^{N_g} p_g \cdot R(n_g) \leq C. \end{aligned} \tag{2}$$

The number of residual elements should be assigned to each class in such a way that the sum over all utility functions is maximized. As side constraint, the calculated rate to transmit N features for the current image has to be below the available channel capacity C. As feature matching is one of the most fundamental questions for most computer vision tasks, we define the utility function as the percentage of correctly matched features in a homography estimation setup depending on the number of the used residual elements n_g and the class g. We match features across two frames and verify correct matches using available ground truth. We show the utility curves for BRISK and a vocabulary size of S=100,000 for the residual calculation with and without entropy reordering in Figure 1. The first observation is the steeper slope of the utility function when adding the first residual elements in the reordered case. Second, the saturation for high values n_g indicates that at some point it makes more sense to assign further bits to the next class, instead of fully reconstructing the descriptors with highest detector response from class g_4. As the utility function $U_g(n_g)$ and the rate $R(n_g)$ are obtained in an offline process, the optimal n_g for typical values of C can be pre-calculated. If sufficient channel capacity is available, the whole residual vector is transmitted. If the channel capacity is very limited, feature classes can be skipped.

4. EXPERIMENTAL EVALUATION

4.1. Setup

We have selected a similar setup as [7, 3] using a homography estimation task. First, we pre-trained visual vocabularies for ORB, BRISK and FREAK features using the implementations from OpenCV with their default settings. We restricted the maximum number of features per image to $N = 500$ and used the MIRFlickr 1M dataset [17] as training data. We used the DBoW2 [18] implementation of hierarchical Bag-of-Words and trained the vocabulary tree with a branching factor $k = 10$ and a depth of $l = 5$ resulting in a vocabulary size of $S = k^l = 100,000$. We used the public dataset from [19], which provides different video sequences (*brick, building, mission, pairs, sunset* and *wood*) showing a planar texture where the camera movement is subject to unconstrained motion patterns. Ground truth is provided by the authors in form of a homography matrix for each frame. The image resolution of each sequence is 640 $\times$ 480 pixels and is captured at 15 fps. Similar to [6], we downsampled the sequence by a factor of five to increase the motion between the individual frames. We used the *sunset* sequence for obtaining the utility functions. In total we used $N_g = 4$ classes and distributed the features equally ($p_g = 0.25, g \in [1, 4]$) to all classes. For each class, we measured the percentage of correct correspondences by verifying the results with the ground truth data. We evaluated our system using only the remaining sequences. We use the *homography estimation precision* (HEP) [7] as quality assessment, which is defined as follows: For each frame, we use the features extracted within the region of the planar texture given by the ground truth. These features are then encoded and decoded using our proposed method at a fixed target bitrate per frame. A homography is estimated using the matched features between the current and the previous frame using a RANSAC-based scheme. The four corner coordinates defining the bounding box of the planar texture are 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 projection error is larger than 3 pixels, the estimated homography is rejected. The HEP describes the ratio between the number of correct estimates and the total number of frames.

4.2. Results

First, we report the coding properties of the individual features. The costs for keypoint coding R_{kpt} is the same for all features and calculated with 31 bits using fixed length coding. The number of bits for transmitting the Bag-of-Words indices is $R_{bow} = 16.6$ bits. For ORB features, the full residuals require 181.6 bits, for FREAK we have to spend 334.9 bits and for BRISK 383.7 bits. The results of the proposed rate allocation system can be found in Figure 2. We show the HEP over varying target bitrates C for the different features. We added

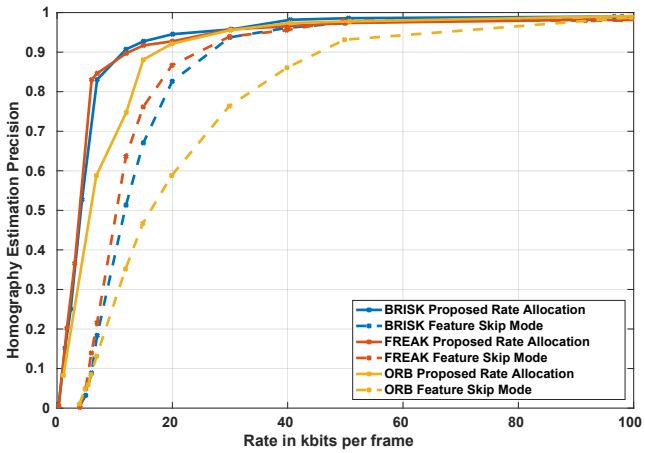


Fig. 2: Homography estimation precision over varying bitrates obtained from *brick, building, mission, pairs* and *wood* sequences. We compare our proposed rate allocation with the feature skipping when reaching the bit budget.

the results from our previously proposed scheme [13] to sort features according to their detector response and skip the remaining features if the bit budget is exhausted. The results indicate for each feature a quite substantial gain in performance at low bitrates. At 7 kbits per frame, we can improve the HEP for BRISK features from 0.18 to 0.83. When operating at low bitrates, it is crucial to allow skipping features, as every visual feature has fixed coding costs for keypoint and visual word information, which makes it impossible to reach very low bitrates without a feature skipping mechanism. In our current approach, we allow skipping all features in a particular class. We have used four classes and distributed the features equally to the classes to keep the signaling overhead reasonable. Depending on the target task and underlying visual feature algorithm, other choices might also be suitable.

5. CONCLUSION

In this paper we propose a novel rate allocation scheme that groups binary local descriptors according to their usefulness into classes and distributes the bits spent on each class using utility functions. To this end, we propose to rearrange the descriptor elements according to their entropy and send the residual information to reconstruct the most informative elements first. We show the substantial improvements of the concept for three well-known binary descriptors. This approach can also be combined with alternative coding algorithms.

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.

6. REFERENCES

[1] Alessandro Redondi, Luca Baroffio, Lucio Bianchi, Matteo Cesana, and Marco 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] Jianshu Chao, Robert Huitl, Eckehard Steinbach, and Damien Schroeder, "A Novel Rate Control Framework for SIFT/SURF Feature Preservation in H.264/AVC Video Compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 6, pp. 958–972, 2015.

[3] Dominik Van Opdenbosch, Martin Oelsch, Adrian Garcea, and Eckehard Steinbach, "A Joint Compression Scheme for Local Binary Feature Descriptors and their Corresponding Bag-of-Words Representation," in *IEEE Conference on Visual Communications and Image Processing (VCIP)*, 2017, pp. 1–4.

[4] Jianshu Chao and Eckehard Steinbach, "Preserving SIFT features in JPEG-encoded images," in *IEEE International Conference on Image Processing (ICIP)*, 2011, pp. 301–304.

[5] Jianshu Chao, Hu Chen, and Eckehard Steinbach, "On the design of a novel JPEG quantization table for improved feature detection performance," in *IEEE International Conference on Image Processing (ICIP)*, 2013, pp. 1675 – 1679.

[6] Luca Baroffio, Matteo Cesana, Alessandro Redondi, Marco Tagliasacchi, and Stefano Tubaro, "Coding visual features extracted from video sequences," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 2262–2276, 2014.

[7] Luca Baroffio, Antonio Canclini, Matteo Cesana, Alessandro Redondi, Marco Tagliasacchi, and Stefano 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.

[8] Alessandro Redondi, Luca Baroffio, Joao Ascenso, Matteo Cesana, and Marco Tagliasacchi, "Rate-accuracy optimization of binary descriptors," in *IEEE International Conference on Image Processing (ICIP)*, 2013, pp. 2910 – 2914.

[9] Ling-yu Duan, Vijay Chandrasekhar, Jie Chen, Jie Lin, Zhe Wang, Tiejun Huang, Bernd Girod, and Wen Gao, "Overview of the MPEG-CDVS Standard," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 179–194, 2016.

[10] Gianluca Francini, Skjalg Lepsoy, and Massimo Balestri, "Selection of local features for visual search," *Signal Processing: Image Communication*, vol. 28, no. 4, pp. 311–322, 2013.

[11] Ling-Yu Duan, Vijay Chandrasekhar, Shiqi Wang, Yihang Lou, Jie Lin, Yan Bai, Tiejun Huang, Alex Chichung Kot, and Wen Gao, "Compact Descriptors for Video Analysis: the Emerging MPEG Standard," *arXiv:1704.08141*, 2017.

[12] José Martínez-Carranza, Francisco Marquez, Esteban O. Garcia, Angélica Muñoz-Meléndez, and Walterio Mayol-Cuevas, "On combining wearable sensors and visual SLAM for remote controlling of low-cost micro aerial vehicles," in *Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, 2015, pp. 232–240.

[13] Dominik Van Opdenbosch, Martin Oelsch, Adrian Garcea, and Eckehard Steinbach, "Selection and Compression of Local Binary Features for Remote Visual SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 7270–7277.

[14] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski, "ORB: An efficient alternative to SIFT or SURF," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2564–2571.

[15] Stefan Leutenegger, Margarita Chli, and Roland Siegwart, "BRISK: Binary Robust invariant scalable keypoints," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2548–2555.

[16] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst, "FREAK: Fast retina keypoint," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 510–517.

[17] Mark J Huiskes and Michael S Lew, "The MIR flickr retrieval evaluation," in *ACM International Conference on Multimedia Information Retrieval*, 2008, vol. 4, pp. 39–43.

[18] Dorian Gálvez-López and Juan 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.

[19] Steffen Gauglitz, Tobias Hoellerer, and Matthew 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.