MULTI-VIEW CODING OF LOCAL FEATURES IN VISUAL SENSOR NETWORKS

Luca Bondi, Luca Baroffio, Matteo Cesana, Alessandro Redondi, Marco Tagliasacchi

Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano
Email:{name.surname@polimi.it}

ABSTRACT

Local visual features extracted from multiple camera views are employed nowadays in several application scenarios, such as object recognition, disparity matching, image stitching and many others. In several cases, local features need to be transmitted or stored on resource-limited devices, thus calling for efficient coding techniques. While recent works have addressed the problem of efficiently compressing local features extracted from still images or video sequences, in this paper we propose and evaluate an architecture for coding features extracted from multiple, overlapping views. The proposed Multi-View Feature Coding architecture can be applied to either real-valued or binary features, and allows to obtain bitrate reductions in the order of 10-20% with respect to simulcast coding.

Index Terms— local visual features, multi-view coding

1. INTRODUCTION

In the last few years, local visual features have become a popular tool in the image processing and computer vision communities. Being able to efficiently summarize the salient parts of an image content, local visual features are nowadays used to perform an extremely wide set of tasks in several application scenarios, ranging from object recognition and image retrieval, to forgery detection for forensic applications. There exist several different algorithms for extracting local features from an image, all following a two-steps approach: first, a detector is applied for identifying salient *keypoints* in an image. Then, for each keypoint, the photometric properties of the pixel area around that keypoint are encoded in a *descriptor*. The most known and used descriptor is SIFT [1], which produces real-valued descriptors by relying on local gradient information. Recently, a new class of descriptors has been proposed, based on pixel intensity comparisons, rather then on gradient information. Such *binary* descriptors represent a much cheaper alternative to their real-valued counterparts, thus they are especially suited for low-power applications such as mobile visual search and Visual Sensor Networks (VSNs).

Especially for the latter application, local binary features constitute a very promising tool for enabling visual analysis in resource-limited scenarios. Instead of relying on a traditional paradigm where compressed images or videos are transmitted to a server for further analysis, camera sensors may extract, compress and transmit local features from the acquired content. Since such a feature-based representation is generally more compact than the traditional pixel-based representation, this paradigm shift constitutes an efficient, yet powerful way to implement visual analysis in energy-scarce, bandwidth-limited scenarios. The benefits of transmitting features instead of images are even clearer when considering recent advanced feature coding algorithms which exploit the redundancy between elements of the same descriptor, or the temporal correlation between descriptors belonging to adjacent video frames [2].

Often, multiple camera sensors are deployed in the same area. This is typically done to increase the accuracy of monitoring or automatic visual analysis, e.g., by avoiding occlusions. In such a scenario, it is likely that the fields of view of the cameras overlap and exploiting the inter-view redundancy between similar views may be beneficial for encoding local features. In this paper, we propose an architecture for compressing local visual features extracted from multiple, overlapping views. The Multi-View Feature Coding (MVFC) scheme is inspired by the practices used in the field of multi-view video coding, and may be applied to either real-valued or binary features. The performance of the proposed coding scheme is evaluated in details, considering several factors that might affect a real-case scenario, such as the amount of inter-camera displacement or the effect of non ideal inter-camera temporal sampling.

2. RELATED WORK

Efficient coding of local visual features is of paramount importance. Since many applications leveraging such kind of data are run on battery-operated devices in bandwidth constrained scenarios (e.g., smartphones, Visual Sensor Networks), compression is needed to minimize: (i) the amount of data transmitted from the camera sensors and (ii) the energy spent in the transmission process.

The problem of efficiently coding local visual features has been recently addressed by many works in the literature. For what concerns the compression of features extracted from still images, coding schemes that exploit the correlation between the elements of each feature (intra-descriptor coding) [3], or the redundancy between similar features (inter-descriptor coding) [4] are available. The work in [5] compares different lossy coding schemes for real-valued SURF features, reporting that bitrates as low as 200 bits per features may be achieved without notable performance decrease. Moreover, the experiments show that intra-descriptor coding generally provides better performance than inter-descriptor coding. As for binary features, both intra- [6] and inter-descriptor [7] coding schemes have been proposed very recently, as well. A completely different approach to local features encoding is the one pursued by the so-called Bag-of-Words (BoW) representation [8]. In this approach, local features are vector quantized into visual words. The occurrences of each visual word are then counted so as to form a histogram, which plays the role of an unique image signature. As a consequence, the final BoW histogram can be represented with very few bits compared to the full local visual feature representation, at the cost of decreased application accuracy.

Several applications (e.g., augmented reality, tracking) rely on features extracted from video sequences rather than still images. Recently, architectures for coding local feature extracted from video have been proposed for the case of real-valued [2] and binary descriptors [9]. The key tenet is to exploit both spatial and temporal redundancy by means of intra-frame and inter-frame coding, respectively. Mimicking the techniques routinely used for blocks of pixels in traditional video coding, a mode decision algorithm decides whether each descriptor should be encoded on its own (intra-frame coding) or predicted using a descriptor in the previous video frame (inter-frame coding). Experimental results show that temporal redundancy allows to reach up to 85% of bit rate reduction with respect to intra-coding only [2].

Besides temporal redundancy, inter-view redundancy is of particular interest when multiple views of the same content are available. In the field of video coding, the exploitation of inter-view redundancy has been studied for more than 20 years and recently standardized as an amendment of the H.264/AVC standard [10]. Motivated mainly by the need of supporting 3D video applications, multiview video coding (MVC) provides a compact representation of multiple synchronized views of a video scene. The coding tools used in MVC are actually the same as those traditionally used for exploiting temporal redundancy, with a reference view, called the *base view*, used to predict another correlated view. It has been shown that coding multiview video with interview prediction may significantly outperform independent coding of the single views (i.e., *simulcast* coding): an average bit-rate reduction of 20-30$\%$ is reported in several studies [10]. The results obtained in the area of MVC are of particular interest for the field of Visual Sensor Networks, where multiple battery-operated cameras, possibly with overlapping fields of view, are deployed for monitoring tasks. Since such applications are generally bandwidth constrained, MVC is recognized as a promising tool for overcoming such limitation, and has been applied to VSNs in several occasions [11]. Moreover, when such camera networks acquire images with a very low frame rate, temporal redundancy may be insufficient to provide considerable bitrate reduction, and inter-view redundancy may be the only viable option.

However, to the knowledge of the authors, very limited attention has been given to the problem of encoding local visual features extracted from multiple views, whereas some works analyzing the problem of multi-view encoding of BoWs are present. In [12], an unsupervised feature selection algorithm is proposed to avoid the transmission of redundant information. The algorithm uses an offline training phase to learn a statistical model of the dependency between BoWs of different views. Inspired to distributed source coding [13], the statistical model is used to drive a joint decoding of the multiple views. Experiments on a two-views scenario report compression ratios as high as 100:1 in the second view, with respect to independent coding. In [14], a joint sparsity model is used to encode BoWs extracted from multiple views. Since such histograms are (i) sparse, (ii) non-negative and (iii) correlated, a distributed encoding scheme based on compressed sensing can be applied to efficiently encode the multiple views, by projecting the original histograms on a random subspace of much lower dimensionality.

In contrast to such approaches based on BoWs, in this paper we focus explicitly on the problem of coding local visual features extracted from multiple views. Using local features instead of BoWs may be required to achieve excellent performance (e.g., in the case of object recognition), or may be an application-driven requisite (e.g., as in the case of stereo matching, depth estimation or disparity map computation [15]). We propose an architecture tailored to the encoding of both non-binary and binary local features extracted from multiple views and we thoroughly evaluate the performance of such an architecture on different, publicly available multi-view datasets, focusing in particular on the impact of the inter-camera geometry and on the case of non-synchronized cameras.

3. MULTI-VIEW FEATURE CODING

We consider a scenario where multiple camera nodes with overlapping fields of view extract local visual features from an input image. Let $\mathcal{F}_\text{b}$ be the set of features extracted from the *base* view, and $\mathcal{F}_\text{s}$ the set of features extracted from a second view, partially overlapping with the base view. Each element f belonging to $\mathcal{F}_\text{b}$ or $\mathcal{F}_\text{s}$ is a visual feature, consisting of a keypoint $\mathbf{k}$ and a descriptor $\mathbf{d}$, i.e., $f = \{\mathbf{k}, \mathbf{d}\}$. Let $\mathbf{k} = [x, y, \sigma, \theta]$ contains the position (x, y), scale σ and ori-

entation θ of the keypoint, respectively. As for the descriptor, let $\mathbf{d} \in \mathbb{R}^P$ for the case of real valued descriptors and $\mathbf{d} \in \{0,1\}^P$ for the case of binary descriptors, where P is the length of each descriptor. The proposed multi-view coding architecture aims at efficiently encoding the second view set of descriptors $\mathcal{F}_\mathrm{s}$ given the knowledge of the base view set $\mathcal{F}_\mathrm{b}$.

We denote the number of bits needed to encode the visual features set $\mathcal{F}_\mathrm{s}$ as:

$$R_\mathrm{s} = \sum_{i=1}^{M} R_i^\mathrm{k} + R_i^\mathrm{d}, \tag{1}$$

where M is the number of features extracted from the second view, and R_i^k and R_i^d are the number of bits necessary to represent the keypoint and descriptor of each feature, respectively. Similarly, we may define R_b as the number of bits needed to encode the features contained in $\mathcal{F}_\mathrm{b}$.

Figure 1 illustrates the general encoding scheme proposed in this work. The key idea is to use the set of features in $\mathcal{F}_\mathrm{b}$ to predict the second view set of features $\mathcal{F}_\mathrm{s}$. A matching step selects potential candidates for prediction. In the case of inter-view coding, the residual between the input descriptor and its best match in the base view is encoded. Mimicking the practices used in recent video encoders, a mode decision algorithm decides whether each feature in $\mathcal{F}_\mathrm{s}$ should be encoded with respect to a local feature in $\mathcal{F}_\mathrm{b}$ (i.e., in an inter-view coding fashion, exploiting the spatial redundancy), or intra-view coded (i.e., solely exploiting the correlation between the elements of its descriptor). In the following we give details on each building block of the coding architecture, highlighting the implementation differences that have to be adopted to deal with either real-valued or binary descriptors.

3.1. Intra-view coding

In case of intra-view coding, the approach is specific to the type of features to be encoded.

3.1.1. Real-valued descriptors

In the case of real-valued descriptors (e.g., SIFT, SURF), the proposed implementation performs lossy coding following the transform-coding scheme based on the KLT transform, as proposed in [2]. First, the descriptors are projected in the transform domain to decorrelate their elements. Then, scalar quantization is applied to each individual descriptor element with the same quantization step. The output symbols of the quantizer are entropy coded using arithmetic coding, producing a rate of R_i^d bits. As usually done in related works [2], the coordinates of the i-th keypoint are encoded at quarter-pixel precision, using $R_i^\mathrm{k} = (\log_2 4N_x + \log_2 4N_y + S)$, where $N_x \times N_y$ is the input image size, and S is the number of bits to encode the scale parameters.

3.1.2. Binary descriptors

In the case of binary descriptors, lossy coding is not applicable as descriptor elements are already represented with one bit. Instead, we rely on a lossless coding scheme which aims at reordering the descriptor elements to maximize the efficiency of arithmetic coding, eploiting the correlation between adjacent symbols [6]. The optimal order of the descriptor elements is learned offline during a training phase, and shared between both the encoder and the decoder. As for coding of keypoints, the same logic used in the case of real-valued descriptors is adopted.

3.2. Inter-view coding

The inter-view coding process consists in the following steps:

3.2.1. Candidate matching and residuals computation

For each descriptor in the second view, a set of candidate descriptors $\mathcal{C}$ in the base view is computed. The candidate set $\mathcal{C}$ can be either the full set of descriptors $\mathcal{F}_\mathrm{b}$, or a subset of it. As an example, when the geometric relationship between the two views is available, $\mathcal{C}$ can be computed through epipolar geometry by projecting the location of the second view on the base view and searching in the neighbourhood. Then, the best matching descriptor in the candidate set is computed, i.e.:

$$\mathbf{d}_{\mathrm{b},l^*} = \arg\min_{\mathbf{d}_{\mathrm{b},l} \in \mathcal{C}} D(\mathbf{d}_\mathrm{s}, \mathbf{d}_{\mathrm{b},l}) + \lambda R^{\mathrm{k,INTER}}(l), \tag{2}$$

where $R^{\mathrm{k,INTER}}(l)$ is the rate needed to encode the keypoint motion vector and l^* is the index of the selected reference feature. The function $D(\mathbf{d}_\mathrm{s}, \mathbf{d}_{\mathrm{b},l})$ is the distance between the input descriptor and a descriptor in the candidate set. For real-valued descriptors, the Euclidean distance is used, whereas the Hamming distance is used for binary descriptors. Having identified the best matching descriptor in the base view, the prediction residuals $\mathbf{c}$ are computed: for real-valued descriptors, the difference between the two descriptors is used (i.e. $\mathbf{c} = \mathbf{d}_\mathrm{s} - \mathbf{d}_{\mathrm{b},l^*}$). For binary descriptors, residuals are computed using the bitwise XOR (i.e. $\mathbf{c} = \mathbf{d}_\mathrm{s} \oplus \mathbf{d}_{\mathrm{b},l^*}$).

3.2.2. Coding mode decision

The mode decision algorithm computes and compares the cost of intra-view coding J^{INTRA} with that of inter-view coding J^{INTER}, defined as:

$$J^{\mathrm{INTRA}}(\mathbf{d}_i) = D(\mathbf{d}_i, \tilde{\mathbf{d}}_\mathbf{i}) + \lambda(R_i^{\mathrm{k,INTRA}} + R_i^{\mathrm{d,INTRA}}) \tag{3}$$

$$J^{\mathrm{INTER}}(\mathbf{d}_i, \mathbf{d}_{l^*}) = D(\mathbf{d}_i, \tilde{\mathbf{d}}_\mathbf{i}) + \lambda(R_i^{\mathrm{k,INTER}}(l^*) + R_i^{\mathrm{d,INTER}}(l^*)) \tag{4}$$

In the previous equations, $\mathbf{d}_{l^*}$ is the selected reference descriptor in the base view which is used to predict the descriptor $\mathbf{d}_i$ in the second view, and $D(\mathbf{d}_i, \tilde{\mathbf{d}}_\mathbf{i})$ is the distance between the original and reconstructed descriptor. Note that

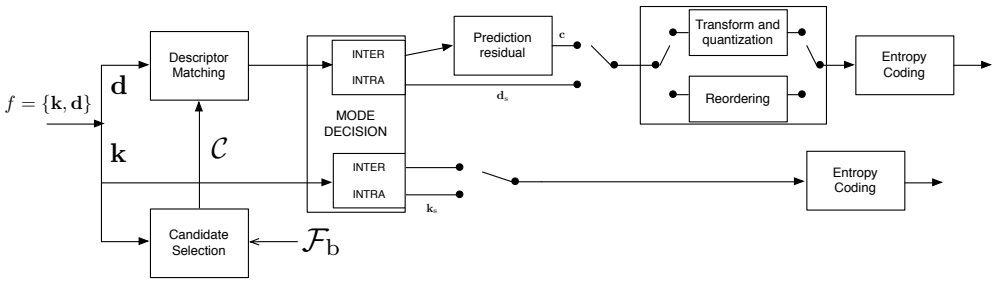


Fig. 1. The proposed multi-view features coding architecture. Solid lines represent the work flow for real-valued features, while dotted lines represent the flow for binary features.

$D(\mathbf{d}_i, \tilde{\mathbf{d}}_\mathbf{i}) = 0$ in case of binary descriptors (lossless coding). Depending on the coding mode, R_i^d represents the bitrate needed to encode the descriptor components (in case of intra-view coding), or the prediction residuals (in case of inter-view coding). If $J^\mathrm{INTER}(\mathbf{d}_i, \mathbf{d}_{l^*}) < J^\mathrm{INTRA}(\mathbf{d}_i)$, the feature is inter-view coded. Otherwise it is intra-view coded (see Section 3.1).

3.2.3. Descriptors coding

Similarly to intra-view coding, the steps for encoding the prediction residuals are different for real-valued or binary features. In the first case, the prediction residuals are transformed into the KLT domain (using a different transform matrix than the one used for intra-view coding) and quantized with a fixed step size. In the case of binary descriptors, an optimal reordering strategy can be learned also in the case of binary residuals. In both cases, entropy coding is applied at the end of the process. Differently from the intra-view coding case, here it is necessary to encode: (i) the residuals output symbols; (ii) the identifier of the matching descriptor in the base view, needed to reconstruct the descriptor at the decoder, and which requires $R_i^{\mathrm{k,INTER}}(l^*)$ bits. For both intra-view and inter-view coding, the KLT transform matrices and the probabilities of the quantized symbols (respectively, descriptor elements or prediction residuals) used for entropy coding are learned from a training set of images.

4. EXPERIMENTS

The experimental evaluation is based on the two publicly available multi-view sequences *Kendo* and *Balloons*[1]. For each sequence, seven views are available, obtained with a linear array of cameras with 5cm spacing. In all tests, the sequences have been resampled to CIF resolution and the results are averaged over 100 frames. For real-valued descriptors, SIFT features have been extracted using the VLFEAT[2] software implementation, whereas BRISK features have been extracted using the original implementation from the authors, for the case of binary descriptors.

As a measure of performance, we consider the bitrate reduction that can be obtained using the proposed encoding scheme, with respect to independently encoding the two views (i.e., *simulcast* coding). That is:

$$\text{Bit rate reduction} = \frac{R_\mathrm{s}^\mathrm{INTRA} - R_\mathrm{s}^\mathrm{INTER}}{R_\mathrm{s}^\mathrm{INTRA} + R_\mathrm{b}^\mathrm{INTRA}} \tag{5}$$

For those tests regarding non-binary features, we measure the distortion introduced by quantization using the signal-to-noise ratio (SNR), which is defined as:

$$SNR = 10 \log 10 \frac{\sum_{i=1}^{M} ||\mathbf{d}_i||_2^2}{\sum_{i=1}^{M} ||\mathbf{d}_i - \tilde{\mathbf{d}}_i||_2^2}, \tag{6}$$

where $\tilde{\mathbf{d}}_\mathbf{i}$ is the decoded i-th descriptor.

In all experiments, λ in (2),(3) and (4) is set equal to 1. We expect improved performance when adjusting the value of λ depending on the target bitrate, but we leave such analysis to future investigations. Similarly, we leave to future works evaluating the impact of using features from multiple views on the accuracy of automatic analysis tasks such as object recognition.

4.1. Impact of camera displacement

As a first test, we evaluate the bitrate reduction achievable at different values of the displacement between the two views. Figures 2(a) and 2(b) report the results obtained for the Balloons and Kendo sequence respectively. As expected, the achievable bitrate reduction increases as (i) the inter-camera spacing decreases and (ii) the distortion increases (i.e., the quantization step gets larger). It is particularly interesting to focus on the gain achievable at 15dB of SNR, a distortion level beyond which the performance of analysis tasks such as object recognition typically saturates [6]. In this case, the bitrate reduction is as high as 25% when the cameras are 5cm apart, and drops to about 10-15$\%$ for a displacement of 30cm.

[1] Available at http://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/

[2] Available at http://www.vlfeat.org/

The same test was repeated for binary descriptors, this time considering different descriptor sizes instead of different distortion levels. As a rule of thumb, the bigger the descriptor size, the better its performance in a visual analysis task. Figure 3(a) and 3(b) report the results obtained for the Balloons and Kendo sequence, respectively. As one can see, the overall trend is the same as for the non-binary scenario, but a smaller bitrate reduction, limited to about 5-10% is achievable in this case.

4.2. Impact of non-ideal synchronization

One basic assumption in the field of multi-view video coding is that the cameras are perfectly synchronized. This is not a problem when the two views are acquired simultaneously, as it happens in 3D acquisition systems. However, in some cases the cameras are driven by independent clocks and such an assumption fails. Visual Sensor Networks are an example of a distributed system where cameras are not perfectly synchronized by a central clock. Synchronization may be still achieved with ad-hoc protocols, at the cost of increasing energy consumption [16]. As a consequence, complex synchronization protocols are typically avoided in VSNs. Thus, it is interesting to evaluate the performance of the proposed encoding scheme when the views are non-synchronized. In Figure 4(a) and 4(b) we report the achievable bitrate reductions obtained by varying the delay in the acquisition of two frames by two cameras with a spacing of 5cm, for the Balloons and Kendo sequence, respectively. Clearly, the performance decreases as the delay increases. For BRISK, the performance decrease is limited to 5 percentage points in the worst case (512 bits descriptor - 1 second of delay), whereas for SIFT the performance decrease can be as high as 15%.

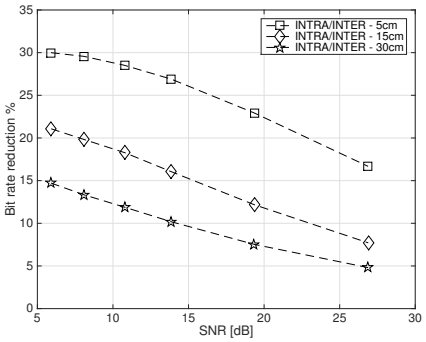
5. CONCLUSIONS

We have proposed an architecture for coding local visual features extracted from multiple views. The proposed method can be applied to either real-valued or binary local features. Experiments on publicly available datasets show that bitrate savings in the order of 20% can be achieved for real-valued features. For binary features, the gain is limited to 10%.

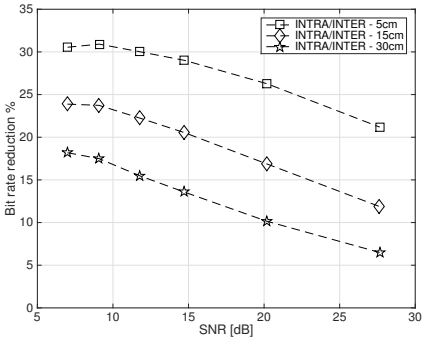
6. REFERENCES

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

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



(a) Balloons



(b) Kendo

Fig. 2. Rate-distortion curve for SIFT features

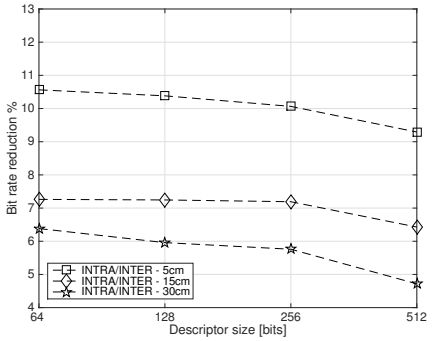
[3] Vijay Chandrasekhar, Gabriel Takacs, David Chen, Sam S. Tsai, Jatinder Singh, and Bernd Girod, "Transform coding of image feature descriptors," 2009, vol. 7257.

[4] Jie Chen, Ling-Yu Duan, Rongrong Ji, Hongxun Yao, and Wen Gao, "Sorting local descriptors for lowbit rate mobile visual search," in *Proceedings of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing, ICASSP, May 22-27, 2011, Prague*, 2011, pp. 1029–1032.

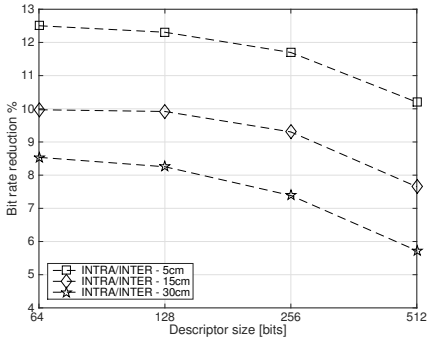
[5] Redondi A., Cesana M., and Tagliasacchi M., "Low bitrate coding schemes for local image descriptors," in *14th IEEE Intl. Workshop on Multimedia Signal Processing, MMSP 2012, Banff, AB, Canada, September 17-19, 2012*, 2012, pp. 124–129.

[6] Redondi A., Baroffio L., Ascenso J., Cesana M., and Tagliasacchi M., "Rate-accuracy optimization of binary descriptors," in *IEEE Intl. Conf. on Image Processing, ICIP 2013, Melbourne, Australia, September 15-18, 2013*, 2013, pp. 2910–2914.

[7] Pedro Monteiro and João Ascenso, "Coding mode decision algorithm for binary descriptor coding," in *22nd*

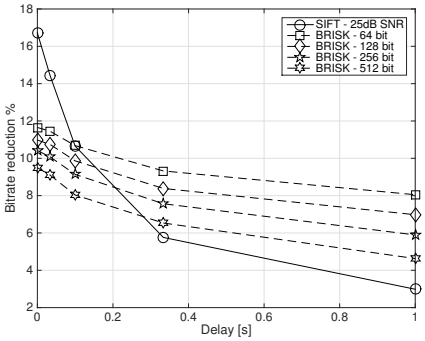


(a) Balloons

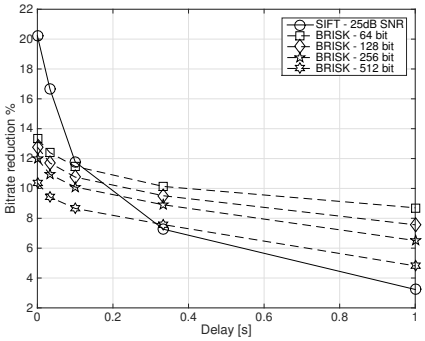


(b) Kendo

Fig. 3. Bitrate reduction achievable at different binary descriptors size



(a) Balloons



(b) Kendo

Fig. 4. Impact of non-ideal synchronization on the achievable bitrate reduction

European Signal Processing Conf., EUSIPCO 2014, Lisbon, Portugal, September 1-5, 2014, 2014, pp. 541–545.

[8] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conf. on*, June 2010, pp. 3304–3311.

[9] L. Baroffio, J. Ascenso, M. Cesana, A. Redondi, and M. Tagliasacchi, "Coding binary local features extracted from video sequences," in *Image Processing (ICIP), 2014 IEEE Intl. Conf. on*, Oct 2014, pp. 2794–2798.

[10] Anthony Vetro, Thomas Wiegand, and Gary J. Sullivan, "Overview of the stereo and multiview video coding extensions of the h.264/mpeg-4 avc standard.," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 626–642, 2011.

[11] M. Flierl and B. Girod, "Coding of multi-view image sequences with video sensors," in *Image Processing, 2006 IEEE Intl. Conf. on*, Oct 2006, pp. 609–612.

[12] C. Mario Christoudias, Raquel Urtasun, and Trevor Darrell, "Unsupervised feature selection via distributed coding for multi-view object recognition," in *2008 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*. 2008, IEEE Computer Society.

[13] Zixiang Xiong, Javier Garcia-Frias, and Bernd Girod, "Distributed source coding," *Signal Processing*, vol. 86, no. 11, pp. 3095, 2006.

[14] Nikhil Naikal, Allen Y. Yang, and S. Shankar Sastry, "Towards an efficient distributed object recognition system in wireless smart camera networks," in *13th Conf. on Information Fusion, FUSION 2010, Edinburgh, UK, July 26-29, 2010*. 2010, pp. 1–8, IEEE.

[15] Engin Tola, V. Lepetit, and P. Fua, "A fast local descriptor for dense matching," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conf. on*, June 2008, pp. 1–8.

[16] Bharath Sundararaman, Ugo Buy, and Ajay D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281 – 323, 2005.