# Studying the compression performance of video descriptors

Renam C. da Silva[1], Fernando Pereira[2], Eduardo A. B. da Silva[1]

*Abstract*— **The main objective of this paper is to study the performance of a framework for encoding visual feature descriptors. Local visual feature descriptors are employed in a number of computer vision tasks, e.g. image and video retrieval by visual search, object recognition and automatic annotation. In scenarios strictly constrained in terms of storage capability, memory and network resources such as those observed in visual sensor networks and mobile visual search applications, compression may be imperative. We evaluate coding schemes for the two most used feature descriptors, namely Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF). The coding modes include intra- and inter-frame modes, with and without decorrelating transforms. They are tested in descriptors extracted from video sequences with different content characteristics. A detailed rate-distortion analysis is conducted in order to assess the contribution of each coding mode. Also, is shown that rate-distortion optimization with all coding mode enabled leads to best results.**

*Keywords*— **Visual features, descriptors, compression, SIFT, SURF.**

## I. Introduction

Traditionally visual information has been represented using picture elements (pixels), and one may say that ultimately pixels are the entities that are input to the human visual system. However, there are challenging tasks related to computer vision applications that demand alternative representations. Visual feature descriptors [9][11] are a powerful class of such representations, and have been used to perform tasks such as image/video retrieval, 3D reconstruction, homography estimation and automatic annotation. In addition, developments in areas of communication and networking as well as embedded processing, provide the basis to construct visual sensor network systems (VSN) [1][2]. Such systems aggregate huge amounts of data captured from multiple and distributed visual sensors and perform complex visual analysis [1]. They can be used to provide interesting services such as augmented reality in sport events, behavior analysis in security systems and automotive driver assistance. Targeting such scenarios, the Moving Picture Expert Group (MPEG)[3] has been considering the Analyse then Compress (AtC) paradigm as an alternative to traditional Compress then Analyse (CtA) paradigm to meet transmission, storage and interoperability requirements. In the CtA paradigm a central node gathers a massive amount of visual data (pixel representation) from sensors and performs visual analysis with relaxed requirements. On the other hand, in AtC the visual sensor nodes are empowered with detection, description and coding algorithms in order to send a concise, but effective representation of the captured content, allowing the central node to perform complex visual analysis [2][7]. Also, the AtC paradigm alleviates computational load at the sensing nodes. Efficient data storage and parsimonious usage of network resources are necessary requirements to deploy such systems. Therefore, compression methods specifically designed for visual feature descriptors are necessary to meet such requirements.

Visual descriptors are generated by algorithms that extract succinct information of the captured scene by performing in general two steps: visual feature detection and description. Salient points are detected searching for maxima or minima in some intermediate representation of the image, such as the Difference of Gaussians[10]. Then, a vector describing the image patches around the detected points is constructed. For example, in the case of SIFT and SURF descriptors[9][11] the vector is a histogram of gradient orientations. The problem of compacting visual features extracted from images has been tackled by researchers in several ways: through dimensionality reduction [4], specially designed compressed feature descriptors such as CHoG [6], transform coding [5][7] and binary descriptors [2]. Naturally, the result should preserve desirable properties of the descriptors, and yet being computationally easy to obtain. Recently, the attention has turned to coding visual features extracted from video sequences. In [8], the authors have proposed a visual feature coding framework with various coding modes, including intra- and inter-frame, with and without decorrelating transforms. The objective of this work is to make a comprehensive study of the framework proposed in [8], in order to set up a firm ground for further research in visual descriptors coding. This paper is organized as follows: in Section II a brief overview of the SIFT and SURF descriptors is given. Then, in Section III the coding schemes used to encode visual features extracted from video sequences are described. In Section IV the performance of the coding framework for the SIFT and SURF descriptors is presented and analyzed. Finally, in Section V the conclusions of this work are presented.

## II. Key point detectors and descriptors

SIFT[9][10] is a visual feature detector and descriptor that describes the visual features through 128-dimensional vectors. These vectors capture the gradient information in the region near the interest points, and they are designed to be scale and orientation invariant. In brief, the gradient magnitude and orientation around the detected interest point are computed, and a Gaussian weighting is applied to the magnitude. Then, the

[1]PEE/COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil. [2]Instituto Superior Técnico, Lisboa, Portugal. E-mails: eduardo@smt.ufrj.br, fp@lx.it.pt, renam.silva@smt.ufrj.br

region around the interest point is divided into 4×4 subregions, and for each subregion a histogram with 8 bins is computed, each bin corresponding to one of eight gradient orientations. The gradient orientation of the samples in each subregion is quantized in eight orientations, and the magnitude is averaged. The final descriptor vector is formed concatenating the $4 \times 4$ histograms.

SURF[11] is a visual feature detector and descriptor, and it was built on the strengths of previous works, specially SIFT. After identifying interest points, a vector describing the image region around each detected key point is computed. The descriptor vector is formed by taking the region around the interest point and dividing into 4×4 subregions. For each subregion the Haar wavelet responses in vertical ($dy$) and horizontal ($dx$) directions are computed, and these responses are Gaussian weighted. Then, the values $\{\sum dx, \sum dy, \sum |dx|, \sum |dy|\}$ are computed for each subregion. The final descriptor vector is obtained concatenating these values for all subregions, resulting in a 64-dimensional vector. An alternative description proposed by Bay *et. al.* with an 128-dimensional vector is also widely used.

## III. Compression schemes for feature descriptors

Based on work reported in [5], [7] and [8], we conduct a detailed description of the compression schemes for visual features studied in this work. The coding schemes described can be classified in intra-frame coding schemes and inter-frame coding schemes. However, the best coding scheme from the rate-distortion (RD) optimization point of view is typically one appropriately combining the intra- and inter-frame modes. In intra coding schemes the set of visual features $\mathcal{D}_n$ extracted from the $n$-th frame of a video sequence is encoded independently from those of other frames. On other hand, in inter coding schemes the redundancy between descriptors of neighbor frames can be explored to save bit rate. These schemes are inspired in predictive coding employed in traditional video encoders. The coding framework described in this Section III is essentially the one described in [8], with the following difference: the symbols probabilities used by arithmetic encoder are allowed to be updated while running the encoder. In following subsections III-A, III-B a detailed description of the compression schemes is given. We also describe in subsection III-C the RD optimization procedure which allows the encoder to choose (descriptor-by-descriptor) the best coding strategy between the intra- and inter-frame coding modes.

### A. *Intra coding schemes*

In intra coding schemes the feature set $\mathcal{D}_n$ of each frame is encoded independently. Nevertheless, the correlation between descriptor fields may be explored. Each feature has two components, namely the descriptor vector $\mathbf{d}_{n,i}$ describing the image patch centered at the detected key point and associated information $\mathbf{p}_{n,i} = \begin{bmatrix} x & y & \sigma & \theta \end{bmatrix}$ as location $(x, y)$, scale $\sigma$ and patch dominant orientation $\theta$. Both the associated information $\mathbf{p}_{n,i}$ and the descriptor $\mathbf{d}_{n,i}$ should be encoded. Each element of the associated information is quantized with a quarter of unit precision and entropy encoded. The descriptor vector part $\mathbf{d}_{n,i} \in \mathcal{D}_n$ of each feature is scalar quantized and entropy encoded after an orthonormal transformation. Figure 1 illustrates the general idea of intra coding schemes. Next it is detailed each step of coding process.

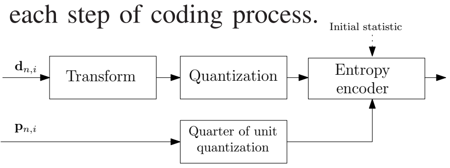


Fig. 1: Intra coding schemes

*1) Transform:* In intra coding mode the encoder applies an orthonormal transform to the descriptor vector before quantization. The simplest transform the encoder can use is the identity matrix, in which case the encoder simply quantizes and entropy encodes the descriptor vector. An alternative option is the Karhunen-Loève (KL) transform, which is known to achieve maximal energy compaction, suitable for compression. Moreover, KL transform was successfully employed in feature coding as reported in [5], [7] and [8]. A collection $\{\mathcal{D}_n\}$ of descriptors extracted from training video sequences is used to estimate covariance matrix $\Sigma_{\mathbf{d}}$ in order to calculate KL transform. Since the descriptor vector used has dimensionality 128, $\mathbf{KL}_{intra}$ is a $128 \times 128$ matrix. After the transform we have $\mathbf{c}_{n,i}^{\text{INTRA}} = \mathbf{T}\mathbf{d}_{n,i}$, where $\mathbf{T} \in \{\mathbf{I}, \mathbf{KL}_{intra}\}$.

*2) Descriptor quantization:* A straightforward scalar quantization is used. Each descriptor vector field (after transformation) is quantized as defined in Equation 1.

$$\tilde{c}_{n,i,j} = \text{round}\left(\frac{c_{n,i,j}}{QP}\right) QP \tag{1}$$

where $c_{n,i,j}$ is $j$-th field of the $\mathbf{c}_{n,i}^{\text{INTRA}}$ (the vector after transformation) and $QP$ is quantization step size, and $\text{round}(x)$ rounds $x$ to the nearest integer.

The quantization of the vector field $c_{n,i,j}$ is the same whatever transform is used, which in the case of our experiments can be either the identity matrix or KL transform.

*3) Entropy coding:* An arithmetic encoder is employed in order to entropy encode the descriptor vector as well as position, scale and patch orientation. An initial statistic is set up to the descriptor symbols and during encoding process the statistic is adaptively updated. The training sequences were encoded to generate the symbols statistics of each descriptor vector field. The key point location, scale and orientation are encoded using a uniform probability model, with cost:

$$\begin{split} R^{\mathbf{p}_{n,i}} = \log_2(4x_{max} + 1() + \log_2(4y_{max} + 1) + \\ \log_2(4\sigma_{max} + 1) + \log_2(4\theta_{max} + 1) \end{split} \tag{2}$$

Considering $x \in [0, 352]$, $y \in [0, 288]$, $\sigma \in [0, \sigma_{max}]$ and $\theta \in [0, 360]$. In case of SURF, for $\sigma_{max} = 120$ it can be verified that the encoder spends approximately 40 bits to encode the position, scale and orientation of each interest point in case of intra modes.

In summary, as described in this subsection III-A, the descriptor vector can be encoded in two ways, namely *Intra* and *Intra-KLT*. The transform applied to descriptor vector is the main difference between the two modes.

### B. *Inter coding schemes*

The inter coding schemes are inspired in traditional video encoders like H.264/AVC and HEVC. A predictive scheme is used to take advantage of the repeatability property of robust local image descriptors like SIFT [9][10] and SURF[11] in addition to the smooth change of the captured scene.

The set of descriptors encoded from last frame $\tilde{\mathcal{D}}_{n-1}$ can be used as a prediction to current frame descriptors $\mathcal{D}_n$. First, a matching for each descriptor vector $\mathbf{d}_{n,i}$ is found, the encoder takes the prediction residue between the current descriptor $\mathbf{d}_{n,i}$ and the matching descriptor vector $\tilde{\mathbf{d}}_{n-1,k*}$. The prediction residue is transformed, followed by quantization and entropy coding steps. A predictive scheme is also adopted to encode the position, scale and orientation. Figure 2 shows a block diagram of the inter coding schemes. In what follows are described the details of each step.

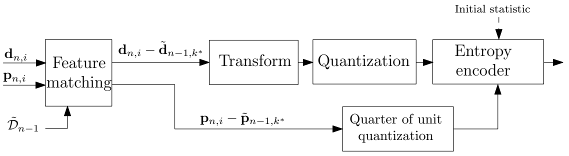

Initial statistic
$\mathbf{d}_{n,i}$
$\mathbf{p}_{n,i}$
Feature matching
$\tilde{\mathcal{D}}_{n-1}$
$\mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n-1,k^*}$
Transform
Quantization
Entropy encoder
$\mathbf{p}_{n,i} - \tilde{\mathbf{p}}_{n-1,k^*}$
Quarter of unit quantization

Fig. 2: Inter coding schemes

*1) Descriptor matching:* The encoder performs a search for a matching descriptor vector encoded from reference set $\tilde{\mathcal{D}}_{n-1}$. The nearest descriptor $\tilde{\mathbf{d}}_{n-1,k*}$ is found using the distance metric:

$$\tilde{\mathbf{d}}_{n-1,k*} = \arg \min_{\tilde{\mathbf{d}}_{n-1,k}} \frac{1}{\sqrt{P}} \|\mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n-1,k}\|_2 \qquad (3)$$

$$\text{subject to} \left\{ \begin{array}{l} x_{n,i} - \tilde{x}_{n-1,k} \leq 30; y_{n,i} - \tilde{y}_{n-1,k} \leq 30 \\ \sigma_{n,i} - \tilde{\sigma}_{n-1,k*} \leq 5 \end{array} \right.$$

where $\tilde{\mathbf{d}}_{n-1,k} \subset \tilde{\mathcal{D}}_{n-1}$, $P$ is vector dimensionality and $\| \cdot \|_2$ refers to the $L^2$-norm.

Interest points in a scene have high probability to be detected repeatedly in a frame sequence with smooth changes in position. Therefore, we constraint the matching to reduce computational complexity. The matching search is performed within a spatial of 30 pixels in horizontal and vertical directions and scale window of 5.

With regard to location, scale and orientation, only the prediction errors and the reference indication are encoded. That is, the differences of position $(x_{n,i} - \tilde{x}_{n-1,k^*}; y_{n,i} - \tilde{y}_{n-1,k*})$, scale $\sigma_{n,i} - \tilde{\sigma}_{n-1,k*}$ and orientation $\theta_{n,i} - \tilde{\theta}_{n-1,k*}$ are quantized and entropy encoded. The feature descriptors of the current frame $\mathcal{D}_n$ are reordered with regard to the matching reference descriptor $\tilde{\mathcal{D}}_{n-1}$ order, and a differential scheme is used to encode the reference indication [8].

*2) Transform:* In inter scheme, the encoder can choose between two transforms, $\mathbf{T} \in \{\mathbf{I}, \mathbf{KL}_{inter}\}$. If the choice is $\mathbf{T} = \mathbf{I}$, the prediction residue $\mathbf{r}_{n,i} = \mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n-1,k^*}$ is simply quantized and entropy encoded. If the choice is $\mathbf{T} = \mathbf{KL}_{inter}$, a KL transform is applied before quantization and entropy coding steps. The procedure to obtain the $\mathbf{KL}_{inter}$ is similar to that to obtain $\mathbf{KL}_{intra}$ as described before. However, is this case, we collect a set of prediction residues in order to obtain the covariance matrix. Only prediction residues which satisfy $\|\mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n-1,k*}\|_2 < \|\mathbf{d}_{n,i}\|_2$ are used. This procedure is done using the training sequences. The transformed feature vector residue $\mathbf{c}^{\text{INTER}}_{n,i} = \mathbf{T}(\mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n-1,k^*})$ is then quantized and entropy encoded as described next.

*3) Descriptor residue quantization:* The quantizer for the inter schemes is equal to that used in intra schemes. But in this case the transformed feature vector residue $\mathbf{c}^{\text{INTER}}_{n,i} = \mathbf{T}(\mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n-1,k^*})$ is quantized, where as described before $\mathbf{T} \in \{\mathbf{I}, \mathbf{KL}_{inter}\}$. The quantization is performed as defined in Equation 4.

$$\tilde{c}_{n,i,j} = \text{round}\left(\frac{c_{n,i,j}}{QP}\right) QP \qquad (4)$$

where $c_{n,i,j}$ is $j$-th field of $\mathbf{c}^{\text{INTER}}_{n,i}$ and $QP$ is the quantization step size.

The position, scale and orientation prediction errors are quantized with a quarter of unit precision.

*4) Entropy coding:* Arithmetic encoder is used to entropy encode the descriptor prediction residue symbols as well as position, scale and orientation prediction error symbols. A training step is conducted in order to collect an initial statistic of the symbols. We run the encoder for the training sequences and store the symbol probabilities. The initial probability is assigned to descriptor residuals as well as to the position, scale and orientation prediction errors. The encoder can update the probability during execution.

In this subsection we have discussed inter coding schemes. In summary, the prediction residue can be encoded in two ways which we call *Inter* mode and the *Inter-KLT* mode, either using or not a KL transform step.

### *C. Rate-distortion optimization*

While subsections III-A and III-B have described the intra and inter coding modes, respectively, the best coding solution is to appropriately combine these coding modes to better exploit the specific correlation associated to each descriptor.

The encoder performs a RD optimization aiming to reach high fidelity with as least as possible rate cost. We have conducted experiments with different combinations of the compression schemes described in subsections III-A and III-B. Depending on which coding modes are enabled the encoder chooses the coding mode which takes minimum Lagrangian cost (see Figure 3). The cost function for intra coding mode is defined as:

$$J_{\text{INTRA}} = \frac{1}{\sqrt{P}} \|\mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n,i}\|_2 + \lambda(R^{\mathbf{p}^{\text{INTRA}}_{n,i}} + R^{\mathbf{d}^{\text{INTRA}}_{n,i}}) \qquad (5)$$

where $R^{\mathbf{p}^{\text{INTRA}}_{n,i}}$ is the cost to encode position, scale and orientation information and $R^{\mathbf{d}^{\text{INTRA}}_{n,i}}$ is the cost to encode the description vector. Note that in case of using the KL transform the rate $R^{\mathbf{d}^{\text{INTRA}}_{n,i}}$ is the rate spent to encode the transformed vector. The cost function for the inter coding mode is defined as:

$$J_{\text{INTER}} = \frac{1}{\sqrt{P}} \|\mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n,i}\|_2 + \lambda(R^{\mathbf{p}^{\text{INTER}}_{n,i}} + R^{\mathbf{d}^{\text{INTER}}_{n,i}}) \qquad (6)$$

where $R^{\mathbf{p}^{\text{INTER}}_{n,i}}$ is the cost to encode the position, scale, orientation prediction errors as well as the reference indication, and $R^{\mathbf{d}^{\text{INTER}}_{n,i}}$ is the cost to encode the descriptor vector with respect to matched reference descriptor. Again, the rate to encode the descriptor depends on which transform was chosen.

The Lagrange multiplier $\lambda$ controls the rate-distortion tradeoff. Experiments were conducted in [8] inspired in [12] to obtain the optimal $\lambda$ value. A rule of thumb is $\lambda(QP) = 1.8 \cdot 10^{-4} QP^2 + 0.1$, where $QP$ is the quantization step size.

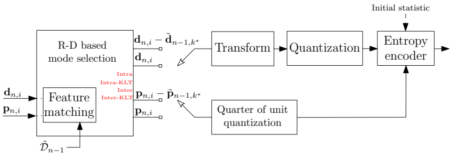


Fig. 3: Rate-distortion optimization based encoder

Besides the rate necessary to encode the description vector and associated information as position, scale and orientation, it is necessary to encode which coding mode was selected in the rate-distortion optimization. Moreover, to know the number of descriptors used for each frame, the encoder also needs to send a frame end flag.

## IV. Results and Discussion

### A. Test conditions and benchmarks

The following sequences were used: *Foreman*, *Mobile*, *Mother*, *News* and *Paris*. All video sequences are in CIF resolution ($352 \times 288$ pixels) with 300 frames and 30 fps. The sequences *Mother*, *News* and *Paris* were used as a training set to estimate the initial statistic for arithmetic encoder, as well as to compute the KL transforms. The encoder's implementation as well as video sequences are available at [16]. The OpenCV's [13] implementations of SIFT and SURF (with 128-dimensional descriptors) were used in subsection IV-B. VLFeat's [14] implementation of SIFT was used in subsection IV-A.

The following coding setups were tested:

- *Intra*: all features are encoded with intra mode, $\mathbf{T} = \mathbf{I}$.
- *Intra-KLT*: all features are encoded with intra coding mode, $\mathbf{T} = \mathbf{KL}_{intra}$.
- *Inter*: the features are encoded with inter coding mode, $\mathbf{T} = \mathbf{I}$. Exceptions are the features for which the matching step was not able to find any reference in the search window, including those features of the first frame. In this case, the features are encoded using *Intra* mode.
- *Inter-KLT*: same as the *Inter* above but $\mathbf{T} = \mathbf{KL}_{inter}$
- *Intra-Inter*: the encoder performs rate-distortion optimization with the *Intra* and *Inter* modes and chooses the mode with lowest cost.
- *4-modes*: the encoder performs rate-distortion optimization with the *Intra*, *Intra-KLT*, *Inter* and *Inter-KLT* modes and chooses the mode with lowest Lagrangian cost.

### B. Rate-distortion performance

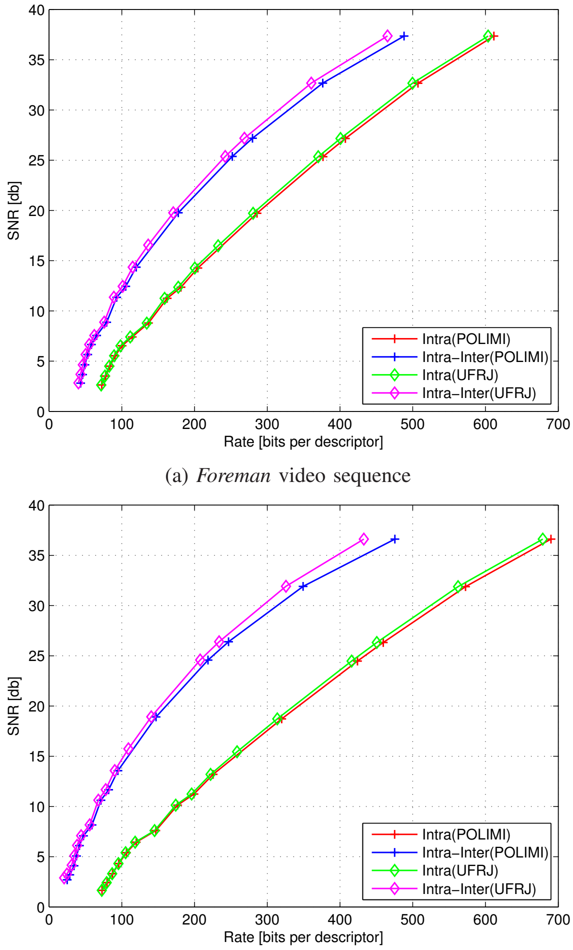
The performance of visual features descriptor compression scheme should be evaluated taking into consideration how much an encoded feature descriptor is effective in a matching visual analysis task. In this sense, rate-distortion results tend to have little meaning. In spite of this, it has been reported in the literature that there is a strong correlation between a descriptor's performance in visual analysis and its rate-distortion results. In fact, it was pointed out in [7] that at 15 dB of SNR the descriptor's rate-accuracy performance saturates. In [8] it is also stated that the matching score saturation is achieved at 15 dB of SNR. Moreover, it was shown in [5] that MSE is good a predictor for the image and feature matching error, and the SURF and SIFT descriptors achieve near-perfect image matching and retrieval below 2 bits/field. Therefore, in this work we decided to conduct the encoder performance evaluation in the rate-distortion sense.

The signal-to-noise ratio (SNR) is measured as:

$$SNR = 10 \log_{10} \frac{\sum_{n=1}^{N} \sum_{i=1}^{M_n} \|\mathbf{d}_{n,i}\|_2^2}{\sum_{n=1}^{N} \sum_{i=1}^{M_n} \|\mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n,i}\|_2^2} \qquad (7)$$

where $N$ is number of frames and $M_n$ is the number of features in the $n$-th frame.

Figure 4 shows a performance comparison between our implementation (labeled UFRJ) of the encoder with results obtained running source code provided by the authors of [8][15] (labeled POLIMI), that has only implementation for SIFT descriptor with *Intra* and *Intra-Inter* modes only. For these modes our implementation of the encoder outperforms the author's encoder. This is due to the adaptive statistic model adopted in the arithmetic encoder. For a fair comparison with authors's encoder, the results presented in Figure 4 do not include the rate to encode the key point orientation. Also, we used VLFeat's implementation of the SIFT as the authors.



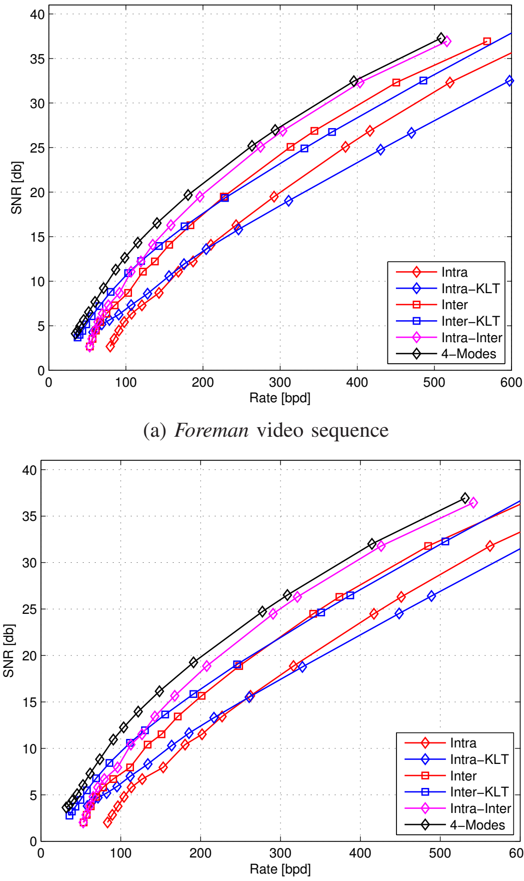
(a) *Foreman* video sequence

(b) *Mobile* video sequence

Fig. 4: Comparative performance for SIFT descriptor compression.

Figures 5 and 6 show encoder performance for SIFT and SURF descriptors, respectively. As expected, when all coding modes are available, the encoder can choose the best coding strategy for each descriptor resulting in better overall performance.

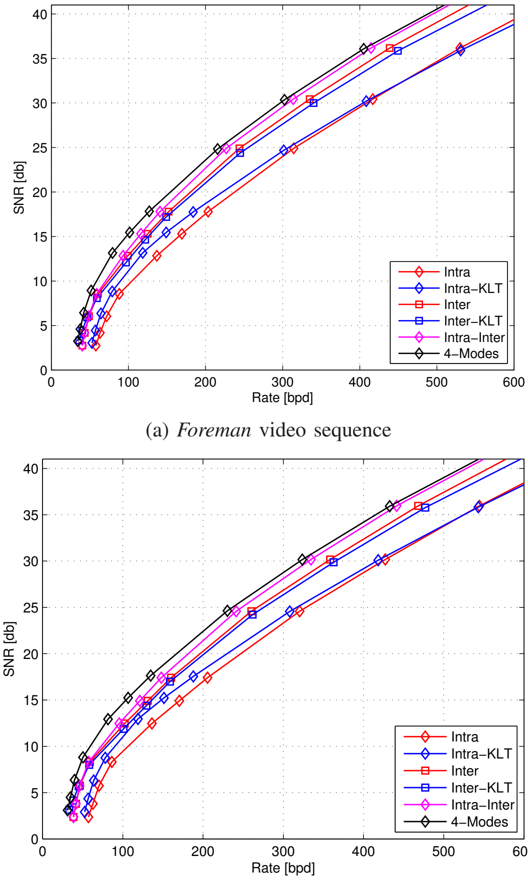
For SIFT descriptors, *Intra-KLT* mode achieves higher coding efficiency than *Intra* only in low bit rates. Similar behavior is observed when comparing inter-frame encoding



(a) *Foreman* video sequence

(b) *Mobile* video sequence

Fig. 5: Comparative performance for SIFT descriptor compression.



(a) *Foreman* video sequence

(b) *Mobile* video sequence

Fig. 6: Comparative performance for SURF descriptor compression.

modes, *Inter-KLT* outperforms *Inter* only in low bit rates. This corroborates the results reported in [5], and is due to the non-Gaussianity of individual descriptor fields.

In case of SURF descriptors, *Intra-KLT* mode outperforms *Intra* coding mode in almost all bit rates. On the other hand, the performance of *Inter-KLT* is worse the one of *Inter*, that is, applying KL transform to descriptor residues is detrimental to coding performance.

Using adaptively intra-frame and inter-frame coding schemes gives better results than intra or inter schemes individually for both feature descriptors.

## V. Conclusion

A comprehensive study of visual features coding schemes was carried out in this paper. Visual features extracted from video sequences were encoded resorting to intra-frame and inter-frame coding schemes and their RD optimized combination. The coding tools were studied with two of the most used feature descriptors, SIFT and SURF. Superior performance in rate-distortion sense was achieved when the four coding modes (*Intra*, *Intra-KLT*, *Inter* and *Inter-KLT*) were available to be chosen by encoder, in which case the best coding strategy is used for each feature.

## Acknowledgements

We would like to thank Fundação de Amparo à Pesquisa do Estado do Amazonas for financial support and Luca Baroffio from Politecnico di Milano for promptly answering our questions.

## References

[1] S. Soro, W. Heinzelman. *A Survey of Visual Sensor Network*. Advances in Multimedia, 2009.

[2] J. Ascenso, F. Pereira. *Lossless Compression of Binary Image Descriptors for Visual Sensor Networks*. 18th Int. Conf. on Digital Signal Processing, 2013, p. 1-8.

[3] The Moving Pictures Experts Group. *Compact Descriptors for Video Analysis (CDVA)*. N14509, Valencia, Spain, April 2014.

[4] Y. Ke, R. Sukthankar. *PCA-SIFT: a more distinctive representation for local image descriptors*. Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004.

[5] V. Chandrasekhar, G. Takacs, D. Chen, S. S. Tsai, J. Singh, B. Girod. *Transform Coding of Image Feature Descriptors*. Proc. SPIE 7257, Visual Communications and Image Processing 2009.

[6] V. Chandrasekhar, G. Takacs, D. Chen, R. Grzeszczuk, B. Girod. *CHoG: Compressed Histogram of Gradients. A low bit-rate feature descriptor*. IEEE Conf. on Computer Vision and Pattern Recognition, 2009.

[7] A. Redondi, M. Cesana, M. Tagliasacchi. *Low bitrate coding schemes for local image descriptors*. IEEE 14th Int. Workshop on Multimedia Signal Processing (MMSP), 2012, p. 124-129.

[8] L. Baroffio, M. Cesana, A. Redondi, M. Tagliasacchi, S. Tubaro. *Coding Visual Features Extracted From Video Sequences*. IEEE Tran. on Image Processing, 2014, vol. 23, issue 5, p. 2262-2276.

[9] D. G. Lowe. *Object Recognition from Local Scale-Invariant Features*. Proc. of IEEE Int. Conf. on Computer Vision, 1999.

[10] D. G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision, 2004, vol. 60, issue 2.

[11] H. Bay, T. Tuytelaars, L. V. Gool. *SURF: Speeded Up Robust Features*. 9th European Conference on Computer Vision, 2006, p. 404-417.

[12] G. Sullivan, T. Wiegand. *Rate-distortion optimization for video compression*. IEEE Signal Processing Magazine, 1998, vol. 15, issue 6.

[13] http://opencv.org.[Accessed in April, 2015].

[14] http://www.vlfeat.org.[Accessed in April, 2015].

[15] http://lucabaroffio.com.[Accessed in April, 2015].

[16] http://www02.smt.ufrj.br/∼renam.silva.