Compress-then-analyze vs. analyze-then-compress: Two paradigms for image analysis in visual sensor networks

Alessandro Redondi, Luca Baroffio, Matteo Cesana, Marco Tagliasacchi

Dipartimento di Elettronica, Informazione e Bioingegneria
P.zza Leonardo da Vinci, 32 - 20133 Milano (Italy)
lastname@elet.polimi.it

***Abstract*—We compare two paradigms for image analysis in visual sensor networks (VSN). In the *compress-then-analyze* (CTA) paradigm, images acquired from camera nodes are compressed and sent to a central controller for further analysis. Conversely, in the *analyze-then-compress* (ATC) approach, camera nodes perform visual feature extraction and transmit a compressed version of these features to a central controller. We focus on state-of-the-art binary features which are particularly suitable for resource-constrained VSNs, and we show that the "winning" paradigm depends primarily on the network conditions. Indeed, while the ATC approach might be the only possible way to perform analysis at low available bitrates, the CTA approach reaches the best results when the available bandwidth enables the transmission of high-quality images.**

I. Introduction

In the last few years, Visual Sensor Networks (VSN) have emerged as a potential enabler for a new class of applications [1] in which vision is a key component. Indeed, object recognition, traffic/habitat monitoring, surveillance and many other applications may benefit from the deployment of several battery-operated wireless sensors with an embedded camera, which communicate with a central controller where the visual content (either still images and videos) is analyzed.

The traditional approach to visual analysis is based on a two steps paradigm, as illustrated in Figure 1(a). First, the acquired signals are compressed (e.g., using JPEG or H.264/AVC) in order to be efficiently transmitted over a network. Then, visual analysis is performed. Since the content is compressed, the analysis is based on a lossy representation of the original signals, which might significantly impair efficiency [2][3][4][5][6]. To mitigate this problem, several works focused on developing feature-preserving image compression schemes [7][8].

Although the *compress-then-analyze* (CTA) paradigm is being successfully employed in a number of application scenarios (e.g., video surveillance and smart cameras), recent results obtained experimenting with real world VSNs have demonstrated that it might be infeasible to stream visual content of sufficient quality such as to enable further analysis tasks [9]. Empowering energy-constrained VSNs with visual analysis requires departing from traditional solutions and pursuing a paradigm shift that affects the way visual data is sensed, processed and transmitted.

Stimulated by the recent results in the field of mobile visual search [10], we posit that most visual analysis tasks can be carried out based on a succinct representation based on local features, disregarding the underlying pixel-level representation. This enables to reverse the traditional CTA paradigm, motivating the adoption of an alternative *analyze-then-compress* (ATC) paradigm [11]. In this case, visual features are extracted from the original signal, compressed with a suitable coding scheme, and transmitted to the final destination, as illustrated in Figure 1(b). Visual features extraction may be performed through state-of-the-art algorithms [12][13][14], or by selecting the best features for a particular analysis task through supervised methods [15] or information theory [16].

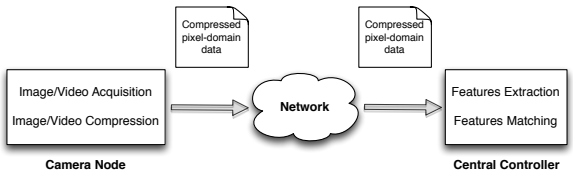
In this work, we compare the two approaches in terms of their rate-accuracy performance for the task of image retrieval. We consider a state-of-the-art retrieval pipeline, and we focus on Binary Robust Invariant Scalable Keypoint (BRISK) [14] visual features, which are tailored for low-power architectures such as VSNs [17][18].

The remainder of this paper is organized as follows. Section II illustrates the background on image retrieval and introduces the evaluation metrics used to compare the two paradigms. Section III summarizes the BRISK algorithm for local features extraction, together with an entropy coding algorithm that can be used to compress the descriptor. Section IV shows the comparison between the two paradigms and Section V concludes the paper.

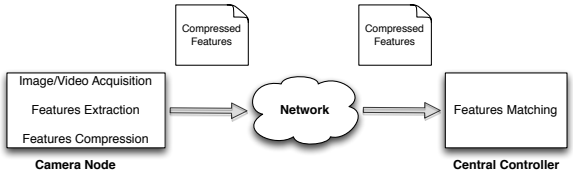
II. Background on image retrieval

In this section we revise the process of image retrieval based on local visual features. That is, given a database of images and a query image as input, we consider the problem of ranking the database so that images similar to the query appear in the first positions. This task can be efficiently carried out by extracting local visual features from the query image, followed

MMSP'13, Sept. 30 - Oct. 2, 2013, Pula (Sardinia), Italy.
978-1-4799-0125-8/13/$31.00



(a) Compress-then-analyze (CTA) paradigm



(b) Analyze-then-compress (ATC) paradigm

Fig. 1. The two different approaches to enable image analysis in visual sensor networks

by a matching step in which these features are compared against those extracted from the images in the database.

A. Local features extraction

The extraction of local features from an image proceeds in a two-steps algorithm. First, the image is processed by means of a *detector*, which identifies a set of salient keypoints. Several algorithms have been proposed in the past that detect keypoints occurring in correspondence of corner-like structures [19] or blob-like structures [12] of an image. Typically, state-of-the-art detectors are scale and rotation invariant, being able to identify keypoints which are stable in the scale-space domain. A keypoint is generally described by a 5-dimensional vector $[x_m, y_m, \sigma_m, \theta_m, t_m]$ indicating the position (x_m, y_m) and the scale σ_m of the detected keypoint, the orientation angle θ_m of the patch of pixels surrounding the keypoints, and the strength t_m of the keypoint. Next, for each detected keypoint, the surrounding patch of pixels is encoded in a D-dimensional *descriptor* vector $\mathbf{d}_m$, which captures the photometric characteristics of the patch. Again, several algorithms have been proposed in the literature to compute descriptors, ranging from approaches based on histograms of local gradients [12], [20], to computationally simpler alternatives such as binary descriptors [21], [14], [22].

B. Local features matching

Visual features extracted from the query image are matched against features (of the same type) extracted from the database of images. Matching consists in pair-wise comparisons of features extracted from, respectively, the query and database image. Depending on the type of feature, either Euclidean or Hamming distance are adopted to compare the corresponding descriptors. Two descriptors are labeled as matching if their distance is below a pre-defined threshold, or if they satisfy the so called ratio-test [12] (i.e., if the ratio between the distance from the nearest neighbour and the second nearest neighbour is less than a pre-defined value). Hence, the images in the database can be ranked according to the number of matches with the query image. Additionally, a geometric consistency check (GCC) step can be applied to filter out outliers. In principle, this process is repeated for all images in the database. In the case of large databases, the process described above is typically applied to re-rank the top-k results obtained using a faster method, e.g., based on bag-of-visual words.

C. Mean Average Precision

Average Precision (AP) is commonly adopted to assess the performance of image retrieval. Given a query q, AP is defined as:

$$AP_q = \frac{\sum_{k=1}^{n} P_q(k) r_q(k)}{R_q}, \tag{1}$$

where $P_q(k)$ is the precision (i.e., the fraction of relevant documents retrieved) considering the top-k results in the ranked list; $r_q(k)$ is and indicator function which is equal to 1 if the item at rank k is relevant for the query, and zero otherwise; R_q is the total number of relevant document for the query q and n is the total number of documents in the list. The Mean Average Precision (MAP) for a set of Q queries is the arithmetic mean of the AP across different queries:

$$MAP = \frac{\sum_{q=1}^{Q} AP_q}{Q} \tag{2}$$

III. BRISK Local Features

Several algorithms are available for extracting local features. In this paper, we consider the state-of-the-art BRISK algorithm [14], which is optimized for fast computation, and thus suitable for low-power and low-complexity hardware, which is often used when deploying visual sensor networks [17][18]. Similarly to other binary descriptors, BRISK has a set of nice properties. First, the descriptor is built by concatenating results of binary tests on smoothed pixel intensities, which are fast to compute. Second, since each descriptor element is a bit (by definition), the size of the descriptor is considerably smaller than that of traditional real-valued descriptors. Third, matching binary descriptors is performed by means of the Hamming distance, which can be executed in a single XOR operation on modern architectures. In the following, we revise the basics of BRISK, focusing our attention on the design of the descriptors. We also illustrate a method introduced in [23] for lossless coding of binary descriptors, in order to further reduce their size.

A. Building the descriptor

BRISK leverages a very simple, yet effective, multi-scale and rotation-invariant corner detector and a binary descriptor algorithm. That is, each descriptor element is a bit, representing the result of a binary test evaluated based on the content of the patch associated to the keypoint. Indeed, each binary test compares the (smoothed) intensity values of a pair of pixels, whose locations within the patch are defined by a concentric sampling pattern.

More formally, let $\mathbf{p}_m^i \in \mathbb{R}^2$, $i = 1, \ldots, N$, denote the position of a sampling point defined in a coordinate system

centered at (x_m, y_m), rotated with an angle θ_m, and properly scaled according to σ_m. Let $\mathcal{I}(\mathbf{p}_m^i, \rho_i)$ denote an intensity value obtained by averaging the pixel values at locations around $\mathbf{p}_m^i$. Although different averaging filters can be used, the publicly available implementation of BRISK adopts a simple box mean filter with floating point boundaries and side length equal to ρ_i. The value of ρ_i depends on the distance from the center of the sampling pattern.

Consider the set $\mathcal{A}$ of all sampling point pairs

$$\mathcal{A} = \{(\mathbf{p}_m^i, \mathbf{p}_m^j) \in \mathbb{R}^2 \times \mathbb{R}^2 | i < N \wedge j < N \wedge i, j \in \mathbb{N}\}. \quad (3)$$

Given a patch corresponding to the detected keypoint, it is possible to compute up to $N(N-1)/2$ binary comparisons, i.e., one for each pair in $\mathcal{A}$. That is,

$$b = \begin{cases} 1, & \mathcal{I}(\mathbf{p}_m^j, \rho_j) > \mathcal{I}(\mathbf{p}_m^i, \rho_i) \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

The total number of possible binary tests depends on the configuration of the sampling pattern. The original BRISK sampling pattern consists of $N = 60$ points, thus $|\mathcal{A}| = 1770$. The construction of the BRISK descriptor proceeds by identifying the subset of short-distance pairs $\mathcal{S}$ and long-distance pairs $\mathcal{L}$:

$$\mathcal{S} = \{(\mathbf{p}_m^i, \mathbf{p}_m^j) \in \mathcal{A} | \|\mathbf{p}_m^j - \mathbf{p}_m^i\| < \delta_{max}\} \quad (5)$$

$$\mathcal{L} = \{(\mathbf{p}_m^i, \mathbf{p}_m^j) \in \mathcal{A} | \|\mathbf{p}_m^j - \mathbf{p}_m^i\| > \delta_{min}\} \quad (6)$$

The long-distance pairs are used in BRISK to estimate the orientation of the patch θ_m. Instead, the descriptor is obtained by concatenating the binary tests corresponding to the short-distance pairs as in (5), such that $(\mathbf{p}_m^i, \mathbf{p}_m^j) \in \mathcal{S}$. Hence, the number of elements D of the descriptor depends on the value of the threshold distance δ_{max}. In [14], δ_{max} was set equal to $13.67\sigma_m$, so as to achieve a descriptor with $D = |\mathcal{S}| = 512$ elements. In our experimental evaluation we vary δ_{max} to test different sizes of the descriptor.

B. Coding the descriptor

The binary tests are not statistically independent. Hence it is possible to model the descriptor as a binary source with memory and perform lossless coding using a number of bits $R \leq D$. Let $H(\pi_n)$, $n = 1, \ldots, D$, denote the entropy of the n-th element of the descriptor, which is computed as

$$H(\pi_n) = -p_n(0) \log_2 p_n(0) - p_n(1) \log_2 p_n(1). \quad (7)$$

In a similar way, it is possible to compute the conditional entropy $H(\pi_{n_1} | \pi_{n_2})$. The statistics used to compute $H(\pi_n)$ and $H(\pi_{n_1} | \pi_{n_2})$ can be obtained by analyzing a training set of descriptors extracted from an image collection. Let $\tilde{\pi}_n$, $n = 1, \ldots, D$, denote a permutation of the D selected pairs, which indicates the sequential order used for encoding the descriptor. The average code length needed to lossless code the descriptor is lower bounded by

$$R = \sum_{n=1}^{D} H(\tilde{\pi}_n | \tilde{\pi}_{n-1}, \ldots, \tilde{\pi}_1) \quad (8)$$

In order to optimize the coding efficiency, it is useful to find the permutation that minimizes the lower bound in (8). In our work, we adopted a greedy strategy, which assumes that the descriptor can be modeled as a Markov source of the first order, i.e., $H(\tilde{\pi}_n | \tilde{\pi}_{n-1}, \ldots, \tilde{\pi}_1) = H(\tilde{\pi}_n | \tilde{\pi}_{n-1})$. Therefore, we propose to reorder the descriptor selecting the elements iteratively. Specifically, the n-th element is chosen as the one that minimizes the conditional entropy with respect to the previously selected element

$$\tilde{\pi}_n = \arg\min_{\pi_n} H(\pi_n | \tilde{\pi}_{n-1}) \quad (9)$$

As for the first element, we opted for selecting the one with the lowest entropy, although we verified that this heuristic choice does not significantly affect the coding rate. After reordering, each descriptor is encoded using a context-based arithmetic encoder, where each bit is encoded based on the previous one in the descriptor.

IV. Experiments

We implemented an image retrieval pipeline following the scheme illustrated in Section II and performed our experiments on two widely used datasets:

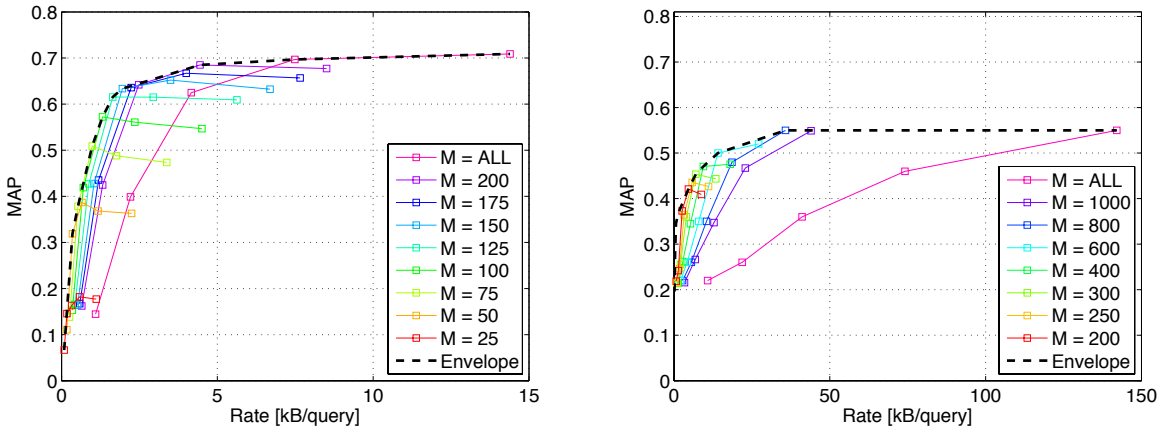
- ZuBuD: Zurich Building Database[1], which contains 1005 images from 201 buildings of the city of Zurich. The dataset also provides 115 QVGA query images (320 $\times$ 240 pixels).
- Oxford: Oxford Building Database[2], which contains images from 16 buildings in Oxford. Only the subsets tagged as GOOD and OK were used, for a total of 568 images. The dataset also provides 55 XGA query images (1024$\times$768 pixels).

The pipeline receives as input a set of visual features for the query image, which differs depending on the adopted paradigm, i.e., *compress-then-analyze* and *analyze-then-compress*

In the compress-then-analyze (CTA) case, image queries are first compressed with JPEG at different rates by varying the JPEG quality factor QF from 1 to 100. The images are then sent to a central controller, which extracts local features from (compressed) query images and matches them against the features extracted from (uncompressed) images in the database. Since the analysis is carried out at the central controller, we considered two options, i.e., either SIFT or BRISK features were extracted from compressed query images. SIFT features are considered as the gold standard in visual analysis, as they typically achieve state-of-the-art performance in most applications. At the same time, extracting and matching SIFT features is costly. Hence, we also consider the option of using BRISK at the central controller, which is a suitable alternative when computational resources are limited.

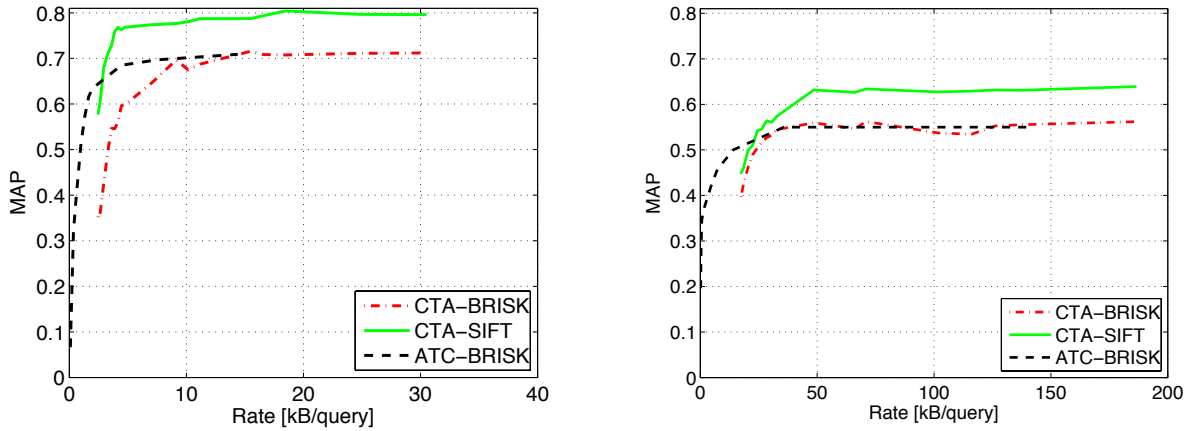
Conversely, in the analyze-then-compress (ATC) case, BRISK visual features are extracted from uncompressed query images, encoded with the algorithm detailed in Section III-B,

[1] http://www.vision.ee.ethz.ch/showroom/zubud/index.en.html
[2] http://www.robots.ox.ac.uk/ vgg/data/oxbuildings/



(a) (b)

Fig. 2. Rate-accuracy curves using the ATC approach and BRISK features: a) ZuBuD dataset; b) Oxford dataset. Each solid colored line represents the rate-accuracy curve for a different number of features M. For a fixed M, the curve is obtained by varying the dimension of each BRISK feature (i.e, the number of computed binary tests). In particular, we tested the following descriptor lengths D: {32, 64, 128, 256, 512}. The curve corresponding to the label ALL is obtained by using all the detected features for matching.



(a) ZuBuD dataset (b) Oxford Building Dataset

Fig. 3. Comparison between the rate-accuracy curves of the *compress-then-analyze* and the *analyze-then-compress* paradigms

and matched with the features extracted from the database images. In this case, the rate is varied by tuning: (i) the dimension D (in bits) of each BRISK descriptor; and (ii) the number of features M to be transmitted to the central controller. To select a subset of features from the whole set, we sorted the features in descending order of their associated keypoint strength t_m, and we selected the top-M features. This allowed us to obtain a family of curves in the rate-accuracy plane, as illustrated in Figure 2. As discussed in [11], we are interested in working on the envelope of such family of curves, which represents the best rate-accuracy tradeoff that can be obtained.

For extracting BRISK features, we used the implementation from the authors[3], setting the detection threshold γ_{det} to 60. For the CTA approach using SIFT, we used the OpenCV v.2.4.3 implementation. For both CTA and ATC approaches, matching features is performed computing either Euclidean or Hamming distances (for SIFT and BRISK, respectively), and filtering matches using the ratio-test and a geometric consistency check with RANSAC.

Figure 3(a) and 3(b) show the experimental results for the ZuBuD and the Oxford datasets, respectively. In each figure, we include the rate-accuracy curves obtained for the following configurations: (i) CTA, when SIFT features are extracted at the central controller; (ii) CTA, when BRISK features are extracted at the central controller; and (iii) ATC, when BRISK features are extracted and compressed at the remote nodes. In this case, the curves correspond to the envelopes in Figure 2.

These results indicate that the choice of the paradigm is dictated by the bandwidth constraints imposed by the network. Indeed, at low bitrates, the analyze-then-compress approach is not only the preferable solution, but also the only one that

[3]http://www.asl.ethz.ch/people/lestefan/personal/BRISK

can be adopted. This is particularly visible in the case of the Oxford dataset, which is characterized by query images at higher spatial resolution, which are more difficult to encode with JPEG. In this case, the minimum rate for the compress-then-analyze approach is equal to 20 kB/query (MAP = 0.45), whereas, in the case of ATC, the rate is as little as 8 kB/query for the same target MAP. Conversely, when the network allows to send high-quality query images at high bitrates, extracting features at the central controller is the best choice. However, note that this is a condition which is seldom met in visual sensor networks. In this case, the analysis relies on a large number of features extracted from the query image. Moreover, if the central controller is not subject to computational constraints, the use of non-binary features like SIFT is to be preferred to BRISK.

V. Conclusions

We compared the CTA and ATC approaches for image analysis in visual sensor networks. Our results indicate that the choice of the approach to use depends on the network conditions, which may impose a constraint on the maximum query size. The ATC paradigm allows to obtain good results especially at low bitrates, while at high bitrates the CTA paradigm leverages the computational resources at the central controller to obtain the best performance.

In the case of the ATC paradigm, it is worth emphasizing the fact that several improvements may be applied to the design of binary descriptors. In this work, we simply varied the size of the BRISK descriptor by changing the δ_{max} threshold. However, as discussed in our previous work [23], a smart selection of the binary tests to use for building the descriptor may significantly increase the performance of the image retrieval task, thus reducing the gap between SIFT and BRISK.

Future research directions include the analysis of the two paradigms when applied to other analysis tasks, e.g., target tracking and scene classification, and the study of optimal resource allocation schemes in the wireless sensor network which adapt the visual analysis paradigm on the network conditions (query rate, available bandwidth, multiple camera nodes).

Acknowledgment

The project GreenEyes acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 296676.

References

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921–960, 2007.

[2] A. Zabala and X. Pons, "Effects of lossy compression on remote sensing image classification of forest areas," *International Journal of Applied Earth Observation and Geoinformation*, vol. 13, no. 1, pp. 43 – 51, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0303243410000693

[3] ——, "Impact of lossy compression on mapping crop areas from remote sensing," *International Journal of Remote Sensing*, vol. 34, no. 8, pp. 2796–2813, 2013. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/01431161.2012.750772

[4] A. Tsifouti, M. M. Nasralla, M. Razaak, J. Cope, J. M. Orwell, M. G. Martini, and K. Sage, "A methodology to evaluate the effect of video compression on the performance of analytics systems," pp. 85 460S–85 460S–15, 2012. [Online]. Available: +http://dx.doi.org/10.1117/12.974618

[5] A. D. Bagdanov, M. Bertini, A. D. Bimbo, and L. Seidenari, "Adaptive video compression for video surveillance applications," in *ISM*. IEEE Computer Society, 2011, pp. 190–197.

[6] 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.

[7] J. Chao and E. G. Steinbach, "Preserving sift features in jpeg-encoded images." in *ICIP*, 2011, pp. 301–304.

[8] O. O. V. Villegas, R. P. Elias, and V. G. C. Sanchez, "Feature preserving image compression: A survey," in *Proceedings of the Electronics, Robotics and Automotive Mechanics Conference - Volume 02*, ser. CERMA '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 35–40. [Online]. Available: http://dx.doi.org/10.1109/CERMA.2006.51

[9] S. Paniga, L. Borsani, A. Redondi, M. Tagliasacchi, and M. Cesana, "Experimental evaluation of a video streaming system for wireless multimedia sensor networks," in *Med-Hoc-Net*, 2011, pp. 165–170.

[10] B. Girod, V. Chandrasekhar, D. Chen, N.-M. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. Tsai, and R. Vedantham, "Mobile visual search," *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 61–76, july 2011.

[11] A. Redondi, M. Cesana, and M. Tagliasacchi, "Rate-accuracy optimization in visual wireless sensor networks," in *International Conference on Image Processing*, oct. 2012, pp. 124 –129.

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

[13] H. Bay, T. Tuytelaars, and L. J. V. Gool, "Surf: Speeded up robust features," in *ECCV (1)*, 2006, pp. 404–417.

[14] S. Leutenegger, M. Chli, and R. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *ICCV*, D. N. Metaxas, L. Quan, A. Sanfeliu, and L. J. V. Gool, Eds. IEEE, 2011, pp. 2548–2555.

[15] E. Kokiopoulou and P. Frossard, "Semantic coding by supervised dimensionality reduction," *Trans. Multi.*, vol. 10, no. 5, pp. 806–818, Aug. 2008. [Online]. Available: http://dx.doi.org/10.1109/TMM.2008.922806

[16] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *CoRR*, vol. physics/0004057, 2000.

[17] A. Canclini, R. Cilla, A. Redondi, J. Ascenso, M. Cesana, and M. Tagliasacchi, "Evaluation of visual feature detectors and descriptors for low-complexity devices," in *Digital Signal Processing Conference*, 2013.

[18] A. Balzarini, M. Cesana, A. Redondi, and M. Tagliasacchi, "A visual sensor network for object recognition: Testbed realization," in *Digital Signal Processing Conference*, 2013.

[19] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 105–119, 2010.

[20] H. Bay, A. Ess, T. Tuytelaars, and L. J. V. Gool, "Speeded-up robust features ( )," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[21] M. Calonder, V. Lepetit, M. Özuysal, T. Trzcinski, C. Strecha, and P. Fua, "Brief: Computing a local binary descriptor very fast," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1281–1298, 2012.

[22] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *CVPR*. IEEE, 2012, pp. 510–517.

[23] A. Redondi, L. Baroffio, A. J., M. Cesana, and M. Tagliasacchi, "Rate-accuracy optimization of binary descriptors," in *submitted to International Conference on Image Processing*, sept. 2013.