

BAMBOO: A FAST DESCRIPTOR BASED ON ASYMMETRIC PAIRWISE BOOSTING

Luca Baroffio, Matteo Cesana, Alessandro Redondi, Marco Tagliasacchi

Politecnico di Milano
Dipartimento di Elettronica e Informazione
Piazza Leonardo da Vinci, 32 - 20133 Milano - Italy

ABSTRACT

A robust hash, or content-based fingerprint, is a succinct representation of the perceptually most relevant parts of a multimedia object. A key requirement of fingerprinting is that elements with perceptually similar content should map to the same fingerprint, even if their bit-level representations are different. In this work we propose BAMBOO (Binary descriptor based on AsymMetric pairwise BOOsting), a binary local descriptor that exploits a combination of content-based fingerprinting techniques and computationally efficient filters (box filters, Haar-like features, etc.) applied to image patches. In particular, we define a possibly large set of filters and iteratively select the most discriminative ones resorting to an asymmetric pairwise boosting technique. The output values of the filtering process are quantized to one bit, leading to a very compact binary descriptor. Results show that such descriptor leads to compelling results, significantly outperforming binary descriptors having comparable complexity (e.g., BRISK), and approaching the discriminative power of state-of-the-art descriptors which are significantly more complex (e.g., SIFT and BinBoost).

Index Terms— Binary descriptors, robust hash, digital fingerprinting, boosting.

1. INTRODUCTION

Visual features provide a compact representation of the content of a given image patch that is robust and invariant to many global and local transformations. Binary descriptors have recently emerged as low-complexity alternatives to state-of-the-art descriptors such as SIFT [1]. The simplest descriptor of this class is BRIEF [2], which provides a binary representation whose entries are the result of different comparisons between pairs of (smoothed) pixel values selected at random within a patch around the keypoint. BRISK [3] refines this process, introducing constraints about the pattern of pixel locations to be used for the comparisons and achieving rotation invariance. FREAK [4] is another example of a binary descriptor, whose sampling pattern of pixels is inspired by the human visual system. More recently, DBRIEF [5] was proposed, whose elements are the result of the binarization of discriminative projections that can be computed fast. Lepetit et al. recently proposed BinBoost [6], a binary descriptor obtained by applying a symmetric boosting algorithm inspired by Similarity Sensitive Coding [7]. The learning phase of BinBoost is based on a nested iterative scheme. In the inner loop, for each bit of the descriptor, BinBoost learns a boosted hash

function based on a set of local gradients. In the outer loop, the most discriminative bits are selected, given the target size of the descriptor. This leads to state-of-the-art results among binary descriptors, in terms of matching time and discriminative power between matching and non-matching patches. At the same time, computing the descriptor is a quite computationally intensive process, with a 50x increase in computing time with respect to BRISK in our experiments based on authors' implementations. For this reason, unlike previously proposed binary descriptors, BinBoost might not be the most suitable choice for implementations running on low-power devices [8][9][10]. In this context, a few works tackled the problems of either improving the efficiency of visual feature extraction algorithms [11] or compressing sets of local features [12][13][14][15].

In this paper we propose BAMBOO (Binary descriptor based on AsymMetric pairwise BOOsting)¹, a novel compact binary descriptor whose construction is computationally efficient. Specifically, our descriptor is inspired by the work by Ke et al. [17], which proposes a music identification system based on content-based fingerprinting, built upon a set of Haar-like filters applied to the signal spectrogram and a pairwise boosting algorithm. The method was later extended in Lee et al. [18] for content-based video retrieval based on spatio-temporal features. Differently from BinBoost [6], our approach adopts an *asymmetric* version of pairwise boosting, in the sense that the weight coefficients assigned to negative training samples are not updated throughout the boosting procedure, allowing for an improved discriminability [19]. In addition, unlike BinBoost, each bit of BAMBOO is the straightforward result of the comparison of average pixel intensities within box-shaped regions, thus inheriting the computationally simple design of fast binary descriptors like BRIEF, BRISK and FREAK, while outperforming them in terms of discriminative power.

The rest of this paper is organized as follows. Section 2 describes BAMBOO, introducing the learning setup and the Asymmetric Pairwise Boosting algorithm. In Section 3 the performance of BAMBOO is evaluated on a test dataset and compared with the one of other popular local descriptors. Conclusions are reported in Section 4.

2. LEARNING AND COMPUTING BAMBOO

In traditional fingerprinting methods, robust binary hashes are obtained starting from intermediate features, applying a set of appropriately chosen quantizers. Fingerprinting can be seen as a classi-

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.

¹A preliminary version of this work was presented as an ongoing work to MMSP'13 [16]. Although it shares with [6] the idea of using boosting, the work was submitted before [6] was published. Note that ongoing works are not archived, so that this paper does not represent a resubmission of previously published material.

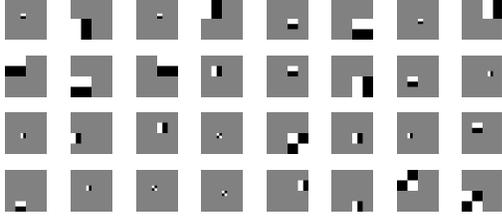


Fig. 1. The first 32 filters selected by the Asymmetric Pairwise Boosting algorithm, with respect to the *Haar* dictionary.

fication problem, since the goal is to assign matching multimedia content to the same class. Hence, different classification algorithms could be employed to obtain discriminative robust hashes. Among all the possible classification algorithms, AdaBoost [20] is a simple yet powerful technique that properly combines several weak classifiers (also known as weak learners) to obtain a single strong classifier whose performance is significantly better than the one of any weak classifier.

The Asymmetric Pairwise Boosting algorithm [17] is quite a natural extension of the Adaboost classifier. It differs from AdaBoost in the fact that samples in the training set are not assigned an individual label. Conversely, labels are assigned to pairs of samples indicating whether they are matching or non-matching pairs. Then, it is the goal of Asymmetric Pairwise Boosting to learn a classifier that determines whether a pair of test samples is matching or not.

Although Pairwise Boosting was originally proposed for audio fingerprinting [17] and later applied to video fingerprinting [18], the underlying nature of the problem is strictly related to the one faced when matching descriptors extracted from local features. More specifically, given an image intensity patch $x_n \in \mathbb{R}^{R \times C}$ we look for a P -dimensional binary descriptor $D(x_n) = [D_1(x_n) \ D_2(x_n) \ \dots \ D_P(x_n)] \in \{-1, +1\}^P$, representing a robust hash for such an image patch. To this end, a training stage is performed resorting to a dataset of image patches extracted from a collection of images. Let $y_n \in \{-1, +1\}$ be a label describing the ground truth relationship between the training pair $(x_{a(n)}, x_{b(n)})$, $n = 1, \dots, N$. Such a label will assume a value equal to $+1$ if the two patches are matching, whereas it will be equal to -1 in the case of two non-matching patches. Moreover, let $\mathcal{H} = \{h_1, h_2, \dots, h_F\}$ define a set of image filters.

For each training image patch x_n , we obtain a vector of F scalar intermediate representations $\{f_1(x_n), f_2(x_n), \dots, f_F(x_n)\}$, where each element $f_i(x_n)$ is the result obtained by filtering the image patch x_n with the filter h_i , i.e., $f_i(x_n) = \langle x_n, h_i \rangle$. Then, such intermediate representations are fed to the Asymmetric Pairwise Boosting algorithm (see Algorithm 1) along with the ground truth relationships between each pair of training patches.

The output of Asymmetric Pairwise Boosting consists of a set of $M < F$ selected filters and the corresponding binarization thresholds. Then, given any image patch x_n , the corresponding descriptor is computed as

$$D_i(x_n) = \begin{cases} +1 & \text{if } f_{i(m)}(x_n) > t_{j(m)} \\ -1 & \text{otherwise} \end{cases}, \quad (1)$$

where $i(m)$, $m = 1, \dots, M$, denotes the index of the filter selected during the m -th iteration, and $j(m)$ the index of the corresponding binarization threshold in the set $\mathcal{T} = \{t_1, \dots, t_T\}$. As an example, Figure 1 shows a set of $M = 32$ filters selected in one of the dictionaries of filters described in Section 3.

Algorithm 1: Asymmetric Pairwise Boosting

Input:

- Ground truth relationships $\langle x_{a(n)}, x_{b(n)}; y_n \rangle$, $n = 1, \dots, N$, $y_n \in \{+1, -1\}$ for each pair of patches.
- A set of filters $\mathcal{H} = \{h_1, h_2, \dots, h_F\}$.
- A set of binarization thresholds $\mathcal{T} = \{t_1, \dots, t_T\}$.

Output:

- A set of $M < F$ filters $[h_{i(1)}, \dots, h_{i(M)}]$
- Corresponding set of thresholds $[t_{j(1)}, \dots, t_{j(M)}]$

```

1 for  $m := 1, \dots, M$  do
2   // For each filter  $h_i$  and for each threshold  $t_p$ , compute
   // the error over the  $N$  pairs of patches:
3   for  $i := 1, \dots, F$  do
4     for  $p := 1, \dots, T$  do
5       for  $n = 1, \dots, N$  do
6          $h_{i,p}(x_{a(n)}, x_{b(n)}) :=$ 
7          $\text{sgn}[(f_i(x_{a(n)}) - t_p)(f_i(x_{b(n)}) - t_p)]$ 
8          $err_{i,p} := \sum_{n=1}^N w_n \cdot \delta(h_{i,p}(x_{a(n)}, x_{b(n)}) \neq y_n)$ 
9   // Pick the best  $f_{i(m)}$  and threshold  $t_{j(m)}$ :
10   $(i^*, p^*) = \arg \min_{i,p} err_{i,p}$ 
11   $err_m := err_{i^*, p^*}$ ,  $f_{i(m)} = f_{i^*}$ ,  $t_{j(m)} = t_{p^*}$ 
12  // Compute the weight for the weak classifier:
13   $c_m := \log(\frac{1 - err_m}{err_m})$ 
14  // Asymmetric weight update (matching pairs only):
15  for  $n := 1, \dots, N$  do
16    if  $y_n = +1$  then
17      if  $h_m(x_{a(n)}, x_{b(n)}) \neq y_n$  then
18         $w_n := w_n \cdot \exp(c_m)$ 
19  for  $n := 1, \dots, N$  do
20    if  $y_n = +1$  then
21       $w_n = w_n / \sum_{n: y_n = +1} w_n$ 

```

The Asymmetric Pairwise Boosting algorithm also assigns a weight c_m , $m = 1, \dots, M$, to each of the selected weak classifiers. Such weights are subsequently exploited to compute a weighted Hamming distance between any two test image patches x_{n_1} and x_{n_2} , according to

$$H(x_{n_1}, x_{n_2}) = \sum_{m=1}^M c_m h_m(x_{n_1}, x_{n_2}), \quad (2)$$

where

$$h_m(x_{n_1}, x_{n_2}) = \text{sgn}[(f_{i(m)}(x_{n_1}) - t_{j(m)})(f_{i(m)}(x_{n_2}) - t_{j(m)})] = \begin{cases} +1 & D_{i(m)}(x_{n_1}) = D_{i(m)}(x_{n_2}) \\ -1 & \text{otherwise} \end{cases} \quad (3)$$

Then, $H(x_{n_1}, x_{n_2})$ is compared to a threshold τ to determine if $\langle x_{n_1}, x_{n_2} \rangle$ is a matching pair. The value of τ is selected based on the desired trade-off between true positives and false positives as illustrated in Section 3.

The fast computation of Hamming distances, thanks to optimized instruction sets and XOR gates, is key to the matching process and it is one of the advantages of binary descriptors. The introduction of a weighted version of the Hamming distance poses

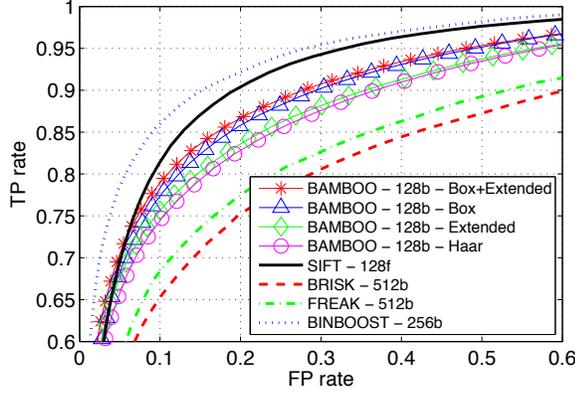


Fig. 2. Receiver Operating Characteristic for *Liberty* dataset and for each filter dictionary. Training: *Notredame 100k patches*, Test: *Liberty 200k*.

some problems in terms of computational efficiency. However, an optimized implementation of weighted Hamming distance has been recently proposed [21], allowing for the usage of such a metric for fast descriptor matching. At the same time, in Section 3 we will show that ignoring these weights by setting $c_m = 1$ does not significantly deteriorate the discriminative power, while enabling the use of simple Hamming distance.

The details of the Asymmetric Pairwise Boosting algorithm are described in the pseudocode in Algorithm 1. Given a set of filters \mathcal{H} and a set of possible binarization thresholds \mathcal{T} , each step of the boosting algorithm selects the filter and the threshold that provide the best possible separation between matching and non-matching patches. That is, it selects the weak classifier that minimizes the weighted error function over the set of all training patches. The assignment of a weight to each pair of patches aims at giving more importance to pairs that are misclassified. Indeed, the more the previously selected weak classifiers are able to correctly classify a pair $\langle x_{a(n)}, x_{b(n)} \rangle$, the lower the corresponding weighting factor w_n . Note that only the weights corresponding to matching training patches are updated throughout the execution of the Asymmetric Pairwise Boosting algorithm, as reported in Algorithm 1. As proved by Sukthankar et al. [22], a symmetric weighting of matching and non-matching training patches would lead to the violation of Adaboost’s weak classifier criterion, resulting in poorer performance.

3. EXPERIMENTS

We conducted several tests to evaluate the performance of BAMBOO, as well as to compare the results with the ones obtained with other popular descriptors. Brown et al. [23] provide a dataset of image patches of size 64×64 pixels, along with the ground truth relationships between pairs of patches, i.e., whether they are matching (referring to the same physical region, possibly from different viewpoints), or not. In particular, the *Notredame* collection is used to train the Asymmetric Pairwise Boosting algorithm described in Section 2, while testing is performed on the *Liberty* and the *Yosemite* collections of patches.

We evaluate the results by means of receiver operating characteristic (ROC) curves, which plot the true positive rate (i.e. the fraction of matching patches for which $H(x_{n_1}, x_{n_2}) < \tau$) at different values of the false positive rate (the fraction of non-matching patches for

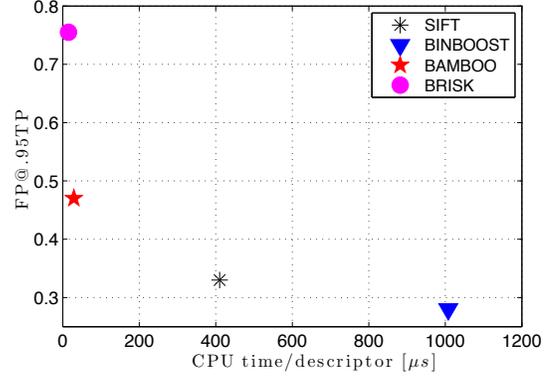


Fig. 3. Comparison of descriptor performance in terms of False Positive rate at 95% True Positive rate as a function of the average CPU time needed to compute a descriptor.

which $H(x_{n_1}, x_{n_2}) < \tau$). The ROC curve is traced by varying τ . In the following we consider the impact of different parameters that determine the construction of the descriptor, namely: i) the choice of the dictionary of filters; ii) the size of the descriptor M ; iii) the impact of weighting when computing Hamming distances. In doing that, we also compare BAMBOO with SIFT and BinBoost, which achieve state-of-the-art discriminative power at the cost of high computational complexity, as well as with BRISK and FREAK, two binary descriptors characterized by low complexity. Finally, we show that the Asymmetric Pairwise Boosting algorithm can be applied to state-of-the-art descriptors like BRISK and FREAK to improve their discriminative power.

Filter dictionaries. We tested different dictionaries of filters. In all cases, we considered dictionaries in which filters assume values in the set $\{-1, 0, +1\}$ and can be efficiently computed resorting to integral images.

- *Haar*: Discrete Wavelet Transform with 5 decomposition levels, excluding LL subband, resulting in a dictionary consisting of 252 different filters.
- *Box*: Pairs of 12×12 boxes (averaging filters), one with a positive weight (+1) and the other one with a negative weight (-1), 14196 filters.
- *Extended Haar*: Set of Haar-like features (edge, line, center surround) as defined by Lienhart and Maydt [24], 1500 filters. Note that each filter can be represented by a combination of up to 4 box filters.
- *Box + Extended Haar*: Combination of *Box* and *Extended Haar* dictionaries, 16212 filters.

Figure 2 shows the results of the test conducted on the *Liberty* dataset in terms of ROC curves, when BAMBOO is constructed retaining the $M = 128$ most discriminant elements by applying the Asymmetric Pairwise Boosting to different filter dictionaries. The selection of a richer dictionary enables to achieve higher performance, with *Box + Extended Haar* achieving the best results.

Comparison with other descriptors. BAMBOO outperforms BRISK and FREAK, despite being four times shorter (128 vs 512 bits), but it is outperformed by SIFT and BinBoost in terms of discriminative power. At the same time, SIFT and BinBoost are significantly more complex than BAMBOO. Figure 3 illustrates the performance of different descriptors in terms of both discriminative power (False Positive (FP) rate when the True Positive (TP) rate is equal

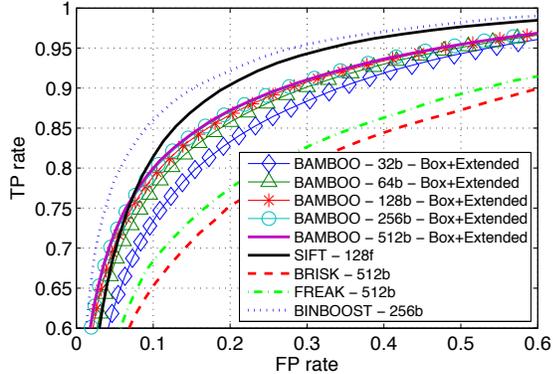


Fig. 4. Receiver Operating Characteristic for *Liberty* dataset and *Box + Extended* dictionary. Training: *Notredame 100k*, Test: *Liberty 200k*.

to 0.95) and complexity (CPU time). The computation of BAMBOO encompasses the application of efficient smoothing filters and the quantization of the resulting values. Such operations are similar to those of BRISK and FREAK. Therefore, the computational complexity of such algorithms is comparable and approximately equal to $20 \mu s$ per descriptor. On the other hand, SIFT and BinBoost result to be very demanding in terms of computational workload, requiring the computation and the pooling of local gradients. Such algorithms might be unsuitable to applications where resources are strictly constrained, such as in the case of visual sensor networks. In our experiments, BAMBOO is 20x faster than SIFT (OpenCV implementation) and 50x faster than BinBoost (author’s implementation). At the same time, it is significantly more discriminative than BRISK ($FP@.95TP$ equal to 0.48 vs. 0.75), thus bridging the gap with SIFT (0.33) and BinBoost (0.28). The results were even more encouraging in the case of the *Yosemite* dataset, since the discriminative power of BAMBOO was similar to that of SIFT. These results are included in an extended technical report made available online [19].

Descriptor length. We investigated the impact of the size of the descriptor on the discriminative power, when adopting the *Box + Extended* dictionary of filters and varying $M = 32, 64, 128, 256, 512$ bits. Figure 4 shows the results in terms of ROC curves, considering the more challenging *Liberty* dataset, when the descriptor size ranges from 32 bits to 512 bits. We observe that increasing the descriptor length improves the discriminative power, although the performance tends to saturate beyond 128 bits. When compared to other descriptors, Figure 5 reports the performance in terms of FP rate at 95% TP rate, as a function of the output bitrate. With BAMBOO, the FP rate decreases from 0.54 to 0.47, saturating when $M \geq 128$. While BinBoost is the best performing descriptor under this aspect, BAMBOO outperforms BRISK and FREAK for every descriptor length.

Weighted Hamming distance. As mentioned in Section 2, the Asymmetric Pairwise Boosting algorithm assigns a weight to each selected filter, which can be used to compute a weighted version of the Hamming distance, as indicated in equation (2). Although the computation of the weighted Hamming distance can be optimized as recently discussed in [21], the unweighted version is still preferable when matching is to be performed in large scale databases. Therefore, we investigated how the performance varies when using BAMBOO, but neglecting the weights when matching descriptors. We observed that the two approaches achieve very similar performance,

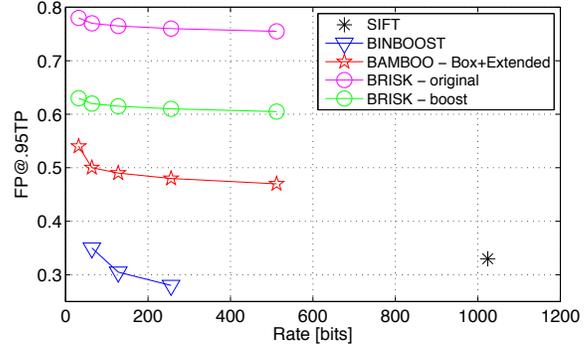


Fig. 5. Comparison of descriptor performance in terms of False Positive rate at 95% True Positive rate for *Liberty* dataset. Training: *Notredame 100k*, Test: *Liberty 200k*.

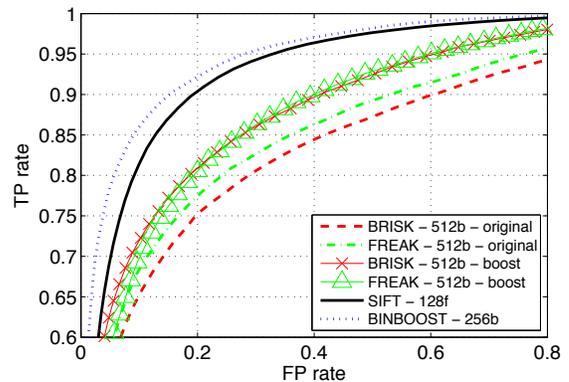


Fig. 6. Receiver Operating Characteristic for *Liberty* dataset for BRISK and FREAK descriptors. Training: *Notredame 100k*, Test: *Liberty 200k*.

with a slight improvement when using weighted Hamming distance. Complete ROC curves are illustrated in a technical report [19].

Boosting of state-of-the-art binary descriptors. In addition to designing the new BAMBOO descriptor, we investigated how to exploit the Asymmetric Pairwise Boosting algorithm to select the most significant components of BRISK [3] and FREAK [4] descriptors. In particular, the BRISK (FREAK) sampling pattern is composed by 60 (43) points, leading to 1770 (903) possible pairwise intensity comparisons. In a previous work we introduced different heuristics to perform feature selection on either BRISK or FREAK [12]. Instead, in this work we employed the Asymmetric Pairwise Boosting algorithm to select the 512 most significant elements. Figure 6 compares the results of such approach to the ones of the original BRISK and FREAK sampling patterns. The boosted versions of the algorithms outperform their original counterparts for the *Liberty* dataset.

4. CONCLUSIONS

We introduce a new method to construct discriminative binary descriptors, inspired by fingerprinting techniques. Tests show that such approach achieves good results in terms of Receiver Operating Characteristics, with as few as 64 or 128 bits, while being computationally efficient. Future work will address a thorough performance evaluation on more general test sets.

5. REFERENCES

- [1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, Nov. 2004.
- [2] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: binary robust independent elementary features," in *Proceedings of the 11th European conference on Computer vision: Part IV*, 2010, ECCV'10.
- [3] S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011.
- [4] R. Ortiz, "FREAK: Fast retina keypoint," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Washington, DC, USA, 2012, CVPR '12.
- [5] T. Trzcinski and V. Lepetit, "Efficient Discriminative Projections for Compact Binary Descriptors," in *Proceedings of the 12th European Conference on Computer Vision*, 2012, ECCV'12.
- [6] V. Lepetit, T. Trzcinski, M. Christoudias and P. Fua, "Boosting Binary Keypoint Descriptors," in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [7] G. Shakhnarovich, "Learning task-specific similarity," in *PhD thesis*, 2006.
- [8] L. Baroffio, M. Cesana, A. Redondi, and M. Tagliasacchi, "Compress-then-analyze vs. analyze-then-compress: Two paradigms for image analysis in visual sensor networks," in *IEEE International Workshop on Multimedia Signal Processing (MMSP) 2013*, Pula, Italy, September 2013.
- [9] A. Canclini, M. Cesana, A. Redondi, M. Tagliasacchi, J. Ascenso, and R. Cilla, "Evaluation of low-complexity visual feature detectors and descriptors," in *IEEE International Conference on Digital Signal Processing 2013*, 2013, pp. 900–903.
- [10] A. Redondi, M. Cesana, and M. Tagliasacchi, "Low bitrate coding schemes for local image descriptors," in *14th IEEE International Workshop on Multimedia Signal Processing, MMSP 2012, Banff, AB, Canada, September 17-19, 2012*, 2012.
- [11] L. Baroffio, A. Canclini, M. Cesana, A. Redondi, and M. Tagliasacchi, "Briskola: Brisk optimized for low-power arm architectures," in *IEEE International Conference on Image Processing 2014*, Paris, France, October 2014.
- [12] A. Redondi, L. Baroffio, M. Cesana, J. Ascenso and, M. Tagliasacchi, "Rate-accuracy optimization of binary descriptors," in *IEEE International Conference on Image Processing 2013*, 2013, pp. 900–903.
- [13] L. Baroffio, A. Redondi, M. Cesana, S. Tubaro, and M. Tagliasacchi, "Coding video sequences of visual features," in *20th IEEE International Conference on Image Processing*, Melbourne, Australia, Sept. 2013.
- [14] L. Baroffio, M. Cesana, A. Redondi, M. Tagliasacchi, and S. Tubaro, "Coding visual features extracted from video sequences," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 2262–2276, 2014.
- [15] L. Baroffio, J. Ascenso, M. Cesana, A. Redondi, and M. Tagliasacchi, "Coding binary local features extracted from video sequences," in *IEEE International Conference on Image Processing 2014*, Paris, France, October 2014.
- [16] L. Baroffio, M. Cesana, A. Redondi, and M. Tagliasacchi, "Binary local descriptors based on robust hashing (on-going work paper)," in *IEEE International Workshop on Multimedia Signal Processing (MMSP) 2013*, Pula, Italy, September 2013.
- [17] Y. Ke, D. Hoiem, and R. Sukthankar, "Computer vision for music identification," in *Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Washington, DC, USA, 2005.
- [18] S. Lee, C. D. Yoo, and T. Kalker, "Robust video fingerprinting based on symmetric pairwise boosting," *IEEE Transactions on Circuits and Systems for Video Technology*, Sept. 2009.
- [19] L. Baroffio, M. Cesana, A. Redondi, and M. Tagliasacchi, "Bamboo: a fast descriptor based on asymmetric pairwise boosting," http://home.deib.polimi.it/baroffio/BAMBOO_techrep.pdf, Feb. 2014, Tech. Rep.
- [20] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proceedings of the Second European Conference on Computational Learning Theory*, London, UK, UK, 1995.
- [21] B. Fan, Q. Kong, X. Yuan, Z. Wang, and C. Pan, "learning weighted hamming distance for binary descriptors," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, Canada, May 26-31, 2013*, 2013.
- [22] R. Sukthankar, Yan Ke, and D. Hoiem, "Semantic learning for audio applications: A computer vision approach," in *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*, 2006, pp. 112–112.
- [23] S. Brown, G. Hua, and S. Winder, "Discriminative learning of local image descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, , no. 1, January 2011.
- [24] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *IEEE ICIP 2002*, 2002, pp. 900–903.