SIP (2016), vol. 5, e13, page 1 of 22 © The Authors, 2016.

doi:10.1017/ATSIP.2016.13

OVERVIEW PAPER

A survey on compact features for visual content analysis

LUCA BAROFFIO, ALESSANDRO E. C. REDONDI, MARCO TAGLIASACCHI AND STEFANO TUBARO

Visual features constitute compact yet effective representations of visual content, and are being exploited in a large number of heterogeneous applications, including augmented reality, image registration, content-based retrieval, and classification. Several visual content analysis applications are distributed over a network and require the transmission of visual data, either in the pixel or in the feature domain, to a central unit that performs the task at hand. Furthermore, large-scale applications need to store a database composed of up to billions of features and perform matching with low latency. In this context, several different implementations of feature extraction algorithms have been proposed over the last few years, with the aim of reducing computational complexity and memory footprint, while maintaining an adequate level of accuracy. Besides extraction, a large body of research addressed the problem of ad-hoc feature encoding methods, and a number of networking and transmission protocols enabling distributed visual content analysis have been proposed. In this survey, we present an overview of state-of-the-art methods for the extraction, encoding, and transmission of compact features for visual content analysis, thoroughly addressing each step of the pipeline and highlighting the peculiarities of the proposed methods.

Keywords: Visual features, Keypoint, Detector, Descriptor, Extraction, Compression, Networking, Encoding, SIFT, Mobile visual search, Visual sensor networks

Received 8 March 2016; Accepted 17 May 2016

I. INTRODUCTION

Throughout our lifetime we seamlessly perform simple actions such as detecting and recognizing faces, identifying objects and events, and reading handwritten text on a daily basis. The human visual system is a powerful yet very efficient apparatus that is able to detect visible light and process it to extract and store a semantic representation of the environment. It acquires data thanks to light receptive sensors, i.e. the eyes, and generates electro-chemical impulses that are transmitted up to the visual cortex through neural pathways. While comprising a number of complex operations, such a process is very efficient and requires very few resources to be performed.

Man-made systems for image acquisition and processing, such as digital cameras, mimic a simplified version of the visual system. Images are acquired by sampling and quantizing the continuous light field on a lattice of pixels. Then, images are compressed in order to be efficiently stored or transmitted. Besides image acquisition and encoding, a large body of research addressed the problem of extracting semantic information from visual content. The first contributions to computer vision date back to the early 1960s [1], mainly devoted to a statistical characterization of visual patterns. Thereafter, computer vision emerged as a research community, addressing a large number of problems, e.g. character recognition, event and object detection, and image classification.

In the last two decades, visual features have been proposed and used as a powerful tool that enables a broad range of visual content analysis tasks. Visual features can be categorized in two main classes: local features that capture the visual characteristics of specific regions of interest within an image, and global features that condense the characteristic of a whole image in a single, compact signature. Due to their ability to concisely summarize the semantic content of an image, visual features are a cornerstone for many complex visual analysis pipelines, including object detection, tracking and recognition, image classification, image calibration, and many others.

Recently, several efforts have been made to integrate image acquisition, analysis and storage on low-power and distributed devices [2]. Smartphones, visual sensor nodes, and smart cameras, are able to carry out complex tasks in a distributed fashion or interacting over a network. Besides acquiring and storing images and videos, they are able to recognize objects, people, landmarks and buildings, automatically detect hazardous events, and stitch shots so as to generate a unique, panoramic photograph. The

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy

Corresponding author:
A.E.C. Redondi
Email: alessandroenrico.redondi@polimi.it

traditional approach to distributed visual content analysis, hereafter denoted as *"Compress-Then-Analyze"* (CTA), heavily depends on a central processing node. According to such approach, a device acquires visual content in the form of still images or video sequences, compresses it resorting to either image or video coding techniques and transmits it to a central node, where a given analysis task is executed. Finally, the central unit transmits the results of the task back to the peripheral node. Although being successfully implemented in a number of applications, CTA has some limitations. The sink node relies on a lossy representation of the original signal, due to image or video compression, which contains coding artifacts that could possibly impair the results of the analysis. Furthermore, most visual content analysis tasks require only a succinct representation of the acquired visual content in order to be performed. Hence, sending pixel-level representations of the content might not be the most rate- and energy-efficient solution, yielding a possibly large transmission overhead [3].

In recent years, with the advent of more and more powerful computing architectures and efficient computer vision algorithms, a novel approach is gaining popularity within both the scientific community [2] and the industry [4]. Such an approach, hereafter denoted as *"Analyze-Then-Compress"* (ATC), moves part of the analysis directly on sensing nodes. As shown in Fig. 1, ATC and CTA represent concurrent paradigms that can be implemented to tackle distributed visual content analysis. In particular, according to ATC, sensing nodes acquire the content, extract semantic information from it in the form of visual features that are subsequently exploited directly on the node or compressed and transmitted to a central unit in order to carry out a given high-level task.

Most reference hardware and networking platforms that could possibly enable the ATC paradigm, such as smartphones or Visual Sensor Networks (VSN), have strict constraints on available computational capabilities, transmission bandwidth, and energy resources [2]. Hence, efficient algorithms for visual feature extraction, compression, and transmission are key to the success of ATC. Since a decade ago, algorithms for feature extraction are being constantly improved, with the aim of generating compact, discriminative, and low-complexity descriptors.

To cope with bandwidth scarcity, *ad hoc* coding algorithms tailored to visual features have been recently proposed. Such algorithms can be split into two main categories: local feature compression and global feature encoding methods. The former exploits the inherent redundancy within a feature or within sets of feature to efficiently reduce the number of bits needed to represent a descriptor. Such approaches are usually inspired by traditional image and video coding techniques, comprising a transform aimed at exploiting spatial or temporal redundancy, or a projection of the signal into a lower-dimensional space, along with *ad hoc* entropy coding algorithms.

As to global feature encoding, local features extracted from a still image are aggregated so as to create a single signature [5]. Global feature encoding algorithms aim at digesting the large amount of information pertaining to local features and to their spatial relationship, creating a signature that is able to effectively yet concisely describe the entire image. Such approach is particularly suitable to large-scale applications, in which matching sets of local features is computationally expensive or even unfeasible. Nonetheless, global features are not able to completely describe the spatial relationship between local features, thus being unsuitable to applications that require geometric verification, such as calibration, structure-from-motion, and object tracking.

Besides such two main coding approaches, entirely devoted to compression of visual features, several hybrid coding techniques are being proposed. They address the problem of jointly encoding images (or video sequences) and visual features. Within this broad category, several approaches are being pursued. On the one hand, some methods modify the traditional image or video coding pipelines so as to preserve the quality of the features that are extracted from lossy content [6]. On the other hand, features and visual content can be jointly encoded in an efficient fashion, achieving a tradeoff between the quality of visual content and the effectiveness of features [7].

A recent line of research addresses the extraction and compression of visual feature starting from video sequences. To this end, the content is processed either on a frame-by-frame basis or considering Groups-Of-Pictures (GOP). Regarding the extraction of features, temporal redundancy can be exploited to speed up the feature extraction process [8]. Besides, the problem of extracting temporally coherent features has been thoroughly addressed in the previous literature. Temporally stable detectors and consistent descriptors lead to significant improvements in both accuracy and coding efficiency, especially considering tracking scenarios [9]. As to the compression of features extracted from video, several different lossy and lossless architectures have been proposed, targeting either local [9, 10] or global [11] features. Such architectures usually take inspiration from the traditional video coding techniques, adapting the coding process to the signal at hand. As in the case of video coding, temporal redundancy can be

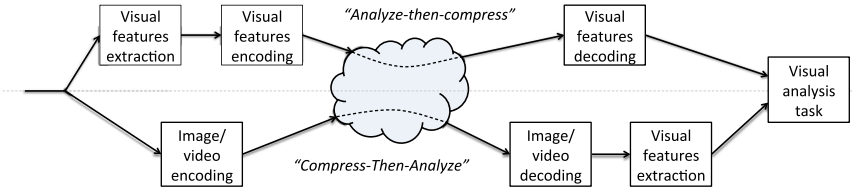

Fig. 1. Pipelines for the *"Analyze-Then-Compress"* and *"Compress-Then-Analyze"* paradigms.

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

Table 1. Summary of the methods presented in this survey.

Feature extraction Section II	Detector	Moravec [16], Kitchen [17], Harris [18], Shi-Tomasi [19], Lee *et al.* [20], SUSAN [21], LoG [22], MSER [23], Harris-affine [24], Hessian affine [24], DoG [25], FAST [26], SURF [27], Trujillo *et al.* [28], THRIFT [29], CenSurE [30], AGAST [31], BRISK [32], KAZE [33], TILDE [34], BIK-BUS [35]
	Descriptor	Schmid and Mohr [36], Shape context [37], SIFT [25], GLOH [38], HoG [39], SURF [27], DAISY [40], BRIEF [41], BRISK [32], ORB [42], FREAK [43], DBRIEF [41], BAMBOO [44], BINBOOST [45]
Feature coding Sections III and IV	Local compression	PCA-SIFT [46], Similarity Sensitive Coding (SSC) [47], Locality Sensitive Hashing (LSH) [48], Spectral Hashing (SH) [49], Semi-Supervised Hashing (SSH) [50], CHoG [51], Transform Coding (KLT) [52], Low-bitrate [53], Multi-stage quantization [54], Product Quantization [55], LDA-Hash [56], Rate-accuracy [57], Predictive coding [58], Cluster coding [59]
	Global encoding	Bag-of-Words (BoW) [5], Pyramid Kernel [60], Tree codebook [61], Kernel codebook (KC) [62], Sparse coding [63], Locality-constrained Linear Coding (LLC) [64], Hamming Embedding (HE) [65], VLAD [66], Fisher Kernel (FK) [67], Super Vector [68], Bag-of-Binary-Words [69], BVLAD [70]
	Other	Location coding [71, 72], SIFT-Preserving JPEG [73] and H.264/AVC [74], Chen and Moulin [75], Hybrid ATC (HATC) [7], Interframe patch [9] and descriptor [76] coding, VideoSIFT [10], VideoBRISK [77]
Feature networking Section V	–	Yang *et al.* [78, 79], feature extraction offloading [80–82], lossy feature transmission [3], Mobile Visual Search [83]

exploited to encode visual features, providing a significant coding gain with respect to the case of still images.

Similar works in the previous literatures focus on either feature extraction [12, 13] or encoding [14, 15]. In this work, we propose a comprehensive survey on algorithms and methods for constructing and exploiting compact visual features, meticulously addressing each step of the pipeline, i.e. feature extraction, compression, and transmission. To the best of the authors' knowledge, this is the first attempt at offering a complete overview of the problem.

The rest of the paper is organized as follows: Section II presents visual feature extraction algorithms, highlighting their main characteristics. Feature encoding techniques are illustrated and compared in Section III. Section IV addresses the problem of extracting and encoding visual features from video sequences and Section V illustrates networking techniques tailored to the context of visual features. Finally, conclusions are drawn in Section VI. For the convenience of the reader, Table 1 offers a summary of methods and algorithms presented in this survey[1].

II. LOCAL FEATURE EXTRACTION

We distinguish between two main classes of visual features: local features, capturing the local information of a given interest point or region of interest, and global features, yielding a compact signature for the input image, based on its content. Global representations are often built starting from a set of local features, by applying proper pooling or aggregation functions. Section III-C thoroughly explores the problem of building global representations starting from local features, whereas in the following we address the extraction of local features from visual content.

[1]An interactive collection of references can be found at http://home.deib.polimi.it/baroffio/surveyFeat/

The definition of local feature is not univocal, heavily depending on the problem at hand and on the type of application. Nonetheless, the feature extraction process usually comprises two main steps: (i) a detector, that identifies keypoints (e.g. blobs, corners, and edges) within an image, and (ii) a keypoint descriptor that assigns to each detected keypoint a descriptive signature consisting of a set of (either real-valued or binary) values, based on the visual characteristics of the image patch surrounding such keypoint.

A) Keypoint detectors

Detecting interest points within an image is the first step toward visual feature extraction. A keypoint detector should be able to identify salient points under very different imaging conditions, such as illumination, contrast, point of view, etc. Hence, a key requirement for a feature detector is repeatability, that is, the ability of the algorithm to detect the same physical interest point in two or more images representing the same scene under different imaging conditions. Depending on the application, several different definitions of keypoints have been proposed. In particular, edges, corners, blobs, and ridges represent instances of interest points, each targeting and capturing peculiar image properties. Table 2 offers an overview and a taxonomy of the most common feature detection algorithms. In the following, we will describe the two most common classes of keypoint detectors, that is, corner and blob detectors.

1) Corner detectors

The first attempts at extracting image features date back to the late 1970s. At that time, early computer vision systems were proposed, aimed at understanding scenes and enabling robot navigation. Such early attempts were able to detect corners by first applying segmentation to the input image to separate physical objects, and by subsequently analyzing

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

Table 2. Overview of the most common local feature detectors.

	Year	Edge	Corner	Blob	Scale	Affine
Moravec [16]	1979		✗			
Kitchen and Rosenfeld [17]	1980		✗			
Harris and Stephens [18]	1988	✗	✗			
Shi–Tomasi [19]	1994		✗			
Lee *et al.* [20]	1995		✗		✗	
SUSAN [21]	1995	✗	✗			
LoG [22]	1998			✗	✗	
MSER [23]	2002			✗	✗	✗
Harris affine [24]	2002	✗	✗		✗	✗
Hessian affine [24]	2002			✗	✗	✗
DoG [25]	2004			✗	✗	
FAST [26]	2005		✗			
SURF [27]	2006			✗	✗	
Trujillo and Olague [28]	2006		✗	✗	✗	
CenSurE [30]	2008			✗	✗	
AGAST [31]	2010		✗			
BRISK [32]	2011		✗		✗	
KAZE [33]	2012			✗	✗	
TILDE [34]	2014			✗	✗	

their shapes. Such methods suffer from segmentation errors and their performance is consistently impaired by noise and cluttered textures.

Kitchen and Rosenfeld observe that corners correspond to changes of edge direction, and introduce an algorithm that is able to detect corners exploiting edge intensity and direction information [17]. Despite being effective on artificial and simple shapes, such an approach is sensitive to noise and not accurate when considering natural scenes.

Moravec was the first to define image features back in 1979, proposing an automated robot navigation system [16]. According to his proposal, a point is considered a good visual feature if: (i) it can be detected in multiple views of the same scene, and (ii) it is sufficiently significant and distinguishable from other regions. In particular, Moravec [16] identifies corners as good visual features and proposes a method to effectively detect them. The key observation behind his approach is that corners have a high variance along the two orthogonal directions. The algorithm tests each pixel within the input image to check whether a corner is present. To this end, the patch centered in a candidate corner is extracted, and the similarity between such a patch and nearby overlapping ones is evaluated. In particular, 25 neighboring patches are considered, sampled horizontally, vertically, and along the two diagonals. Given a candidate corner patch centered in (x, y) and a neighboring patch shifted by $(\Delta x, \Delta y)$ pixels, both sampled from the image I, their similarity is evaluated by means of the sum of squared differences (SSD) as

$$d_{\Delta x,\Delta y}(x,y) = \sum_{(x_i,y_i)\in\mathcal{N}(x,y)} [I(x_i,y_i) - I(x_i-\Delta x, y_i-\Delta y)]^2, \quad (1)$$

where $\mathcal{N}(x_i, y_i)$ represents the neighborhood of the candidate corner point (x, y).

Since a smaller SSD indicates a higher patch similarity and thus lower cornerness, the candidate corner strength, or cornerness measure, is defined as the minimum of the SSDs between the candidate patch and its neighboring ones. Finally, points corresponding to local cornerness maxima are detected as stable features. To this end, non-maxima suppression is performed: a keypoint is detected if its cornerness measure is higher than a given threshold and it is a local maximum within an arbitrarily sized neighborhood.

Despite its effectiveness, one of the main drawbacks of Moravec corner detector is that it is anisotropic, that is, it is not invariant to rotation. In fact, only edges along the four main directions – horizontal, vertical and, along the diagonal – are correctly discerned from corners.

Harris and Stephens propose a joint corner and edge detection algorithm [18], overcoming the issues of Moravec's approach. They build upon the same idea, that is, corners are points with high-intensity variance along all directions. Exploiting Taylor expansion, the difference between a candidate corner patch and a neighboring one is:

$$d_{\Delta x,\Delta y}(x,y) \simeq [\Delta x \;\; \Delta y] D(x,y) \begin{bmatrix}\Delta x \\ \Delta y\end{bmatrix}, \quad (2)$$

where $D(x, y)$ is the structure tensor matrix, capturing the intensity structure of the candidate patch, based on local gradients. In particular, it is possible to infer information about the candidate patch intensity structure analyzing the two eigenvalues λ_1 and λ_2 of the structure tensor matrix. The values of such eigenvalues are proportional to the amount of intensity variation along the directions specified by the corresponding eigenvectors. In particular, the higher an eigenvalue, the faster the intensity variation along the corresponding direction. Hence, if both λ_1 and λ_2 are sufficiently large, a corner is identified, whereas if $\lambda_1 \gg \lambda_2$, an edge is identified. Harris measure is capable of efficiently capturing the cornerness of a candidate point (x, y) as

$$R(x,y) = det(D(x,y)) - \alpha \cdot tr(D(x,y))^2. \quad (3)$$

Shi and Tomasi propose a keypoint detector inspired by the Harris corner detector, targeting object tracking scenarios [19]. Shi and Tomasi observe that a stable corner corresponds to two large eigenvalues of the structure tensor matrix and hence use the value of the smallest eigenvalue as a cornerness measure, i.e. $R(x, y) = \min(\lambda_1, \lambda_2)$, where λ_1 and λ_2 are the two eigenvalues of the structure tensor matrix $D(x, y)$ computed at a given point (x, y). Features are then ranked according to an *ad hoc* statistical measure that indicates the temporal consistency of each keypoint with respect to an affine image motion model.

Lee *et al.* propose to use a wavelet transform to identify corners at different scales [20]. By representing the signal in the wavelet domain, they are able to detect both arcs and corners at multiple scales.

In 1995, Smith and Brady introduce *Smallest Univalue Segment Assimilating Nucleus* (SUSAN) [21], a combined edge and corner detector. The algorithm analyzes a circular region around a candidate edge or corner point. Within

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

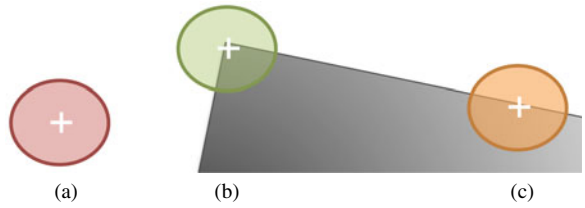

Fig. 2. The key idea behind SUSAN. In flat regions (a), almost all the pixels have an intensity similar to that of the nucleous (white cross). In edge regions (c), approximately half of the pixels have an intensity similar to that of the nucleous. In corner regions (b), less than half of the pixels have an intensity similar to that of the nucleous.

such region, the USAN value is computed as the number of pixels belonging to the region and having an intensity value similar to that of the nucleous, i.e. the center of the region. Corners and edges correspond to some characteristic USAN values, as shown in Fig. 2, and can thus be easily detected.

Mikolajczyk and Schmid propose an affine-invariant version of the Harris detector [24]. Their approach is able to detect features at multiple scales and is robust to affine transformations, exploiting an affine Gaussian scale-space.

Features from Accelerated Segment Test (FAST) [26], introduced by Rosten in 2005, is the first instance of corner detectors based on the *Accelerated Segment Test (AST)*. The main idea behind such approach is that every corner is surrounded by a circular arc of pixels whose intensities are all higher or lower than the circle center. Considering a Bresenham circle of radius r consisting of k pixels $p_1, \ldots, p_k$, FAST compares the intensity of such pixels with the one of the pixel corresponding to the center of the circle. A candidate point is detected as a corner if at least n contiguous pixels, out of the k ones, are all brighter or darker than the center by at least a threshold t.

The *AST* can be efficiently implemented resorting to machine learning and decision trees, allowing negative corner responses to be discarded with just few operations, thus yielding a high computational efficiency. AGAST [31] improves the performance of FAST by proposing an optimization framework tailored to the AST decision tree building process. Furthermore, AGAST allows for the definition of more generic, application-dependent *AST* and for the computation of the corresponding decision trees.

BRISK [32] further refines the process, introducing a scale-invariance version of the *AST*-based detector.

2) Blob detectors

Early computer vision researchers identify corners as points of interest within an image and thus a good fit for feature extraction. Blobs represent regions of an image that differ in terms of one or more visual characteristics, such as color or brightness, compared with the surrounding area. Being peculiar regions of images that can be detected under different imaging conditions, blobs emerged as an effective alternative to corners.

Lindeberg observed that filtering an image with a *Laplacian of Gaussian* (LoG) leads to large positive and negative responses corresponding to dark and bright blobs, respectively [22]. He proposes a scale-invariant blob detector that is capable of extracting arbitrarily sized blobs. To this end, given an input image $I(x, y)$ and considering a given scale σ, a scale-space representation $L(x, y, \sigma)$ is obtained by convolving the image with a Gaussian kernel:

$$L(x, y, \sigma) = I(x, y) * g(x, y, \sigma),$$
$$g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2/2\sigma^2)}. \quad (4)$$

Then, the Laplacian operator is applied to such scale-space representation according to

$$\nabla^2 L(x, y, \sigma) = L_{xx}(x, y, \sigma) + L_{yy}(x, y, \sigma), \quad (5)$$

where L_{xx} (L_{yy}) denotes the second-order partial derivative along the x (y)-axis.

Finally, blobs correspond to local extrema of the scale normalized LoG response $R(x, y, \sigma) = \sigma^2 \nabla^2 L(x, y, \sigma)$, that consists in a three-dimensional (3D) space composed of both spatial coordinates (x, y) and scale σ. Thresholding and non-maxima suppression are usually exploited to identify such extrema.

Matas *et al.* propose *Maximally Stable Extremal Regions* (MSER), an alternative blob detector that allows for the extraction of affine regions sufficiently uniform in terms of pixel intensity [23]. Without loss of generality, consider a gray-scale image $I \in \{0, 255\}^{M \times N}$. Furthermore, consider a thresholding $I(\tau)$ of the image I, obtained by fixing a proper threshold value τ within the set $\{0, 255\}$. In particular, assume that all the pixels whose intensity is lower than τ are set to zero (black), whereas the remaining one are set to 255 (white). If $\tau = 0$, all the pixels are white. Increasing the value of τ, some black regions appear, corresponding to local intensity minima. The *Extremal Regions* are all the spatially connected regions obtained by thresholding the image with all possible values of τ. The *MSER* are a subset of *Extremal Regions* that satisfy a stability criterion based on region areas. According to such criterion, regions are enforced to have similar shapes and dimensions across a large set of possible thresholds.

Simlarly, Kim and Grauman propose a Boundary-Preserving Local Region detector (BPLR) [84], which robustly identifies different regions in an image in a shape and boundaries preserving manner.

Alongside the Harris affine corner detector, Mikolajczyk and Schmid propose an affine feature detector based on the Hessian matrix [24]. In particular, consider an affine scale space pyramid $L(x, y, \sigma_1, \sigma_2)$ obtained by smoothing the input image with bivariate Gaussian kernels at different scales. For each point of the scale-space, consider the Hessian matrix

$$H(x, y, \sigma_1, \sigma_2) = \begin{bmatrix} L_{xx}(x, y, \sigma_1, \sigma_2) & L_{xy}(x, y, \sigma_1, \sigma_2) \\ L_{xy}(x, y, \sigma_1, \sigma_2) & L_{yy}(x, y, \sigma_1, \sigma_2) \end{bmatrix}, \quad (6)$$

where L_{xx} (L_{yy}) is the second-order partial derivative along the x (y)-axis and L_{xy} the second-order mixed derivative.

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

Features corresponds to scale-space entries $L(x, y, \sigma_1, \sigma_2)$ that correspond to extrema of both the determinant and the trace of the Hessian matrix. Differently from the case of Harris detector, picking the extrema of the determinant of the Hessian matrix penalizes elongated regions corresponding to edges.

Lowe proposes to approximate the *LoG* operator by means of a *Difference of Gaussians* (DoG) [25]. To this end, a scale-space is obtained by subsequently filtering the input image with Gaussian kernels with constantly increasing standard deviation. Then, adjacent Gaussian-smoothed images are subtracted to build a DoG scale-space. As in the case of LoG, local extrema of such scale-space, corresponding to stable features, can be detected by means of non-maxima suppression. DoG significantly reduces the computational complexity of the scale-space building process with respect to LoG. Nonetheless, DoG scale-space construction represents the computational bottleneck of the keypoint detection process, leaving space for further optimization.

Bay *et al.* propose *Speeded Up Robust Features* (SURF), a fast feature extraction algorithm [27]. It comprises both a keypoint detector and a keypoint descriptor. As to the former, it aims to efficiently compute an approximation of the Hessian matrix resorting to a combination of 2d box-like filters, and making use of integral images. Despite being computationally efficient, SURF filters are anisotropic and hence not completely robust against image rotations.

Agrawal *et al.* propose CenSurE [30], a blob detection algorithm that, approximates the LoG operator by means of center-surround kernels, obtained as combinations of 2d box-like filters.

Alcantarilla and Bartoli propose KAZE [33], a scale-invariant feature detector based on the Hessian matrix. Differently from DoG and SURF, KAZE exploits non-linear diffusion filtering to build a scale-space representation of the input image, preserving edges and boundaries and thus yielding improved localization accuracy and distinctiveness.

3) Machine learned detectors

Visual feature detectors presented so far have been developed so that the extracted interest points satisfy given properties or possess particular characteristics. To this end, feature detection operators such as LoG, SUSAN, and FAST have been mostly handcrafted and optimized resorting to trial and error procedures. Nonetheless, with the advent of effective statistical and computational models and powerful computing hardware, machine learning techniques are being exploited to automatically learn effective feature detection operators. Differently from traditional approaches, driven by human-defined intuitions, such approaches aim to automatically learn detection operators resorting to a set of training examples.

In this context, Trujillo and Olague propose a genetic programming framework that is able to automatically synthesize keypoint detection operators [28]. According to such approach, the quality of a feature detector can be evaluated by means of three key properties: (i) separability between detected points, (ii) amount of local information content, and (iii) stability to imaging conditions. To learn detection operators that maximize such properties, a set of low-level operations (e.g. image derivatives and pixel-wise summation) are defined. Then, an instance of detection operator is defined as a combination of an arbitrary number of low-level operations. Finally, evolutionary models are exploited to explore the search space of operators instances, so that the key properties are satisfied.

In 2014, Verdie and Yi introduce TILDE [34], a machine-learned detector that is invariant to drastic changes in imaging conditions (e.g. night/day, partial occlusion or clutter). First, they build a novel training dataset of image patches, corresponding to keypoints that are stably detected under a large set of different imaging conditions. Then, they exploit a linear regression procedure to define an operator that is able to accurately detect such stable features.

The performance of keypoint detection algorithms, in terms of detection stability under different imaging conditions and computational efficiency, have been thoroughly evaluated [12, 13, 85]. Efficient detectors based on *AST*, such as FAST[2], AGAST, and BRISK, approach the performance of traditional algorithms such as SURF and DoG in terms of detection repeatability, at a much lower computational complexity.

Finally, a number of 3D keypoint detectors have been proposed [29, 35]. Such algorithms are capable of identifying salient 3D structures in depth maps or 3D point clouds.

B) Keypoint descriptors

Early attempts at matching stereo images for tracking and image understanding are based solely on keypoint detectors. Moravec [16], and Harris and Stephens [18] algorithms represent the cornerstones of such early computer vision applications. Besides image keypoint positions, local image content (e.g. in terms of intensity, texture, and color) can be effectively exploited to match pairs of images. In this sense, keypoint descriptors aim to assign to each detected keypoint a concise signature, consisting of a set of values that capture local visual characteristics of the surrounding image patch. Zhang *et al.* [86] propose to use a simple descriptor, consisting of the intensity values of the pixels surrounding a given keypoint. Such pixel-level windows can be matched resorting to either the SSD or the *Normalized Cross Correlation* (NCC), in a process similar to video coding's motion estimation. On the one hand, such simple representation is sufficient to match contiguous frames extracted from the same video sequence and whose visual content is highly correlated. On the other hand, it is not sufficiently robust to changes in imaging conditions, and thus it is not suitable to general-purpose image matching.

In this context, more and more powerful keypoint descriptors have been devised. They can be categorized in two broad groups, according to the data type of the features

[2] Note that FAST detector is not invariant to scale changes.

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

Table 3. Overview of the most common local feature descriptors.

	Year	Real-valued	Binary	Intensity	Gradient	Rotation Inv.	Default size (bytes)
Schmid and Mohr [36]	1997	✗					32
Shape context [37]	2002	✗					144
SIFT [25]	2004	✗			✗	✗	512
GLOH [38]	2005	✗			✗	✗	512
HoG [39]	2005	✗			✗	✗	124
SURF [27]	2006	✗			✗	✗	256
DAISY [40]	2010	✗			✗	✗	400
MROGH [87]	2010	✗			✗	✗	192
BRIEF [88]	2011		✗	✗			64
BRISK [32]	2011		✗	✗		✗	64
ORB [42]	2011		✗	✗		✗	32
FREAK [43]	2012		✗	✗		✗	64
DBRIEF [41]	2012		✗	✗		✗	4
BAMBOO [44]	2013		✗	✗		✗	8
BINBOOST [45]	2013		✗		✗	✗	8
Radial Gradient [89]	2013	✗			✗	✗	16

they generate: (i) real-valued descriptors and (ii) binary descriptors. Table 3 offers an overview of the most common keypoint descriptors.

1) Real-valued descriptors

The first keypoint description algorithms assign to each detected keypoint a compact signature consisting of a set of real-valued elements. In [36], an image retrieval system is proposed, based on Harris corner detector and local grayvalue invariants. They assign to each detected corner a descriptor based on a set of differential invariants, up to the third order. In particular, such approach is invariant with respect to image rotation. Furthermore, scale invariance is achieved by computing descriptors at different scales.

The work in [37] propose *Shape Context*, a feature extraction algorithm that captures the local shape of a patch. Considering a patch surrounding the point (x, y), edge detection is applied to identify edges. Then, a radial grid is applied to the patch, and a histogram centered in (x, y) counts the number of edge points falling in a given spatial bin.

David Lowe introduces *Scale Invariant Feature Transform* (SIFT) [25]. SIFT computes for each keypoint a real-valued descriptor, based on the content of the surrounding patch in terms of local intensity gradients. In particular, considering a keypoint identified in a given spatial position (x, y) and at a given scale σ, an image patch centered in (x, y) and having size proportional to σ is extracted. Within such image patch, local gradients are computed and used to estimate the prominent keypoint orientation. Then, local gradients are pooled over a 4×4 grid, oriented according to the prominent keypoint orientation, as depicted in Fig. 3. Finally, local gradient orientations are quantized using eight angular bins, and for each one of the 16 regions composing the 4×4 grid, a weighted histogram of gradients is computed. In particular, each local gradient contributes to the bin corresponding to the nearest orientation, for an amount proportional to its magnitude. The final SIFT descriptor consists of 128 elements.

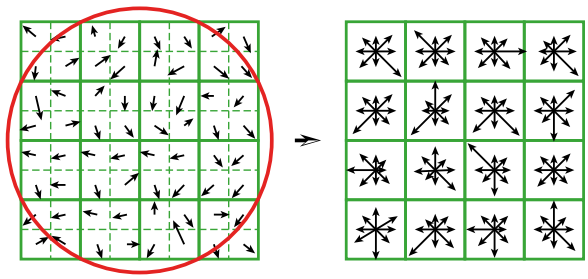

Fig. 3. SIFT descriptor building process. (Left) Local gradients are computed and pooled on a 16×16 grid around the keypoint (shown as 8×8 here for simplicity). (Right) for each cell of the overlying 4×4 grid a 8D weighted histogram of gradients is computed.

Given its remarkable performance, SIFT has been often used as starting point for the creation of other descriptors.

Mikolajczyk and Schmid propose *Gradient Location and Orientation Histogram* (GLOH) [38], a descriptor inspired by SIFT. Instead of using a 4×4 spatial grid, they propose to pool the gradients in 17 radial bins. Furthermore, differently from SIFT, gradient orientations are quantized using 16 angular bins. Finally, Principal Component Analysis (PCA) is applied to the 272-dimensional descriptor in order to reduce its dimensionality, leading to a 128-dimensional real-valued descriptor.

Dong and Soatto propose DSP-SIFT [90], in which pooling of gradient orientations is performed across different scales. Morel and Yu propose ASIFT [91], a fully affine invariant version of SIFT computed starting from a set of simulated images obtained from the original one. Another example of an affine-invariant approach is given by the ASR descriptor [92], which uses PCA to represent affine-warped patches compactly and efficiently.

Similarly to SIFT and GLOH, the DAISY [40] descriptor is obtained by spatially pooling local gradients within ad-hoc circular regions arranged on concentric circles of

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

increasing radius. Differently from SIFT and GLOH, DAISY has been designed with the aim of extracting descriptors in predefined locations that are densely sampled on an uniform grid, bypassing the keypoint detection stage. Being densely sampled, multiple descriptors may exploit the same local gradients. DAISY optimizes the computation resorting to gradient channels, so that a local gradient is computed just one time and shared among multiple descriptors.

In the context of pedestrian detection, Dalal and Triggs propose *Histogram of Oriented Gradients* (HOG) [39], a descriptor based on spatial pooling of local gradients. Although the approach is similar to SIFT, to address the problem of detecting human-like shapes *Histogram of Gradients* are computed on a dense grid of locations, skipping keypoint detection. Dalaal and Triggs observe that gradient strengths have large variations due to local illumination properties, and thus propose a contrast normalization technique to enhance the descriptor accuracy.

Besides an efficient keypoint detector, SURF [27] includes a fast gradient-based descriptor. In particular, given a keypoint, its main orientation is computed by analyzing local gradient orientations, similarly to the case of SIFT. Local gradient responses along x- and y-axis are efficiently extracted exploiting particular wavelet filters, that can be computed fast resorting to integral images. Then, such responses are pooled on a 4×4 grid, and for each bin of the grid a compact representation is built by applying simple summations.

Fan *et al.* propose MROGH [87], a 192-dimensional local descriptor, which differs from the aforementioned ones in three aspects: (i) achieving rotation invariance without computing a dominant orientation for the keypoint, (ii) pooling intensity gradients in an adaptive strategy based on their intensity orders, and (iii) constructing the descriptors by relying on multiple support regions in order to increase their discriminative power.

Along the same line, Girod and co-workers propose rotation invariant features based on the Radial Gradient Transform [89]. According to such methods, the extracted gradients are intrinsically oriented and thus invariant to image rotations, allowing for very efficient computation of local features.

A different approach is taken by Wang *et al.* with their Local Intensity Order Pattern (LIOP) descriptor [93]. LIOP describes an image patch using local ordinal information of the pixels composing the patch, resulting in a 144-dimensional descriptor robust to intensity changes, image rotation, viewpoint change, image blur and compression.

2) Binary descriptors

Despite yielding a good matching accuracy for a large set of tasks, real-valued gradient-based local descriptors such as SIFT or HOG require computationally intensive processes to be extracted, especially when considering low-power devices such as mobiles, smart cameras, or visual network sensing nodes. Binary descriptors, usually based on pairwise intensity comparisons, recently emerged as an efficient yet accurate alternative to real-valued features. Most binary feature extraction algorithms do not require the computation of local image gradients or local derivatives, thus being computationally efficient. Furthermore, binary features can be efficiently matched resorting to fast Hamming distance computation [88], resulting in significant speedup, especially considering large-scale applications.

Calonder *et al.* introduce *Binary Robust Independent Elementary Features* (BRIEF) [88], a local binary keypoint description algorithm partially inspired by *Random Ferns* [94] and *Local Binary Patterns* [95]. Exploiting pairwise comparisons between smoothed pixel intensities, it results in very fast computation. Considering a keypoint identified at location (x, y), the surrounding image patch is extracted. Within such patch, n_d pairs of pixel locations $(x_i^1, y_i^1), (x_i^2, y_i^2), i = 1, \ldots, n_d$ are randomly selected. For each couple of pixel locations, a binary value is obtained performing a pairwise intensity comparison, defined as

$$D_i(p) = \begin{cases} 1 & \text{if } p(x_i^1, y_i^1) > p(x_i^2, y_i^2) \\ 0 & \text{otherwise} \end{cases}, \qquad (7)$$

where p represents a smoothed version of the original input image. Finally, the BRIEF descriptor for the keypoint under consideration is obtained by concatenating the n_d binary values $D_i(p), i = 1, \ldots, n_d$ obtained by performing the n_d pairwise intensity comparisons.

Leutenegger *et al.* propose *Binary Robust Invariant Scalable Keypoints* (BRISK) [32], a binary intensity-based descriptor inspired by BRIEF. Each binary dexel (descriptor element) of BRISK is obtained, as in the case of BRIEF, by performing a pairwise intensity comparison. Differently from BRIEF, the location of the pairs of pixels are sampled on an *ad hoc* concentric pattern, as depicted in Fig. 4. Furthermore, differently from BRIEF, BRISK is able to produce scale- and rotation-invariant descriptors. In particular, considering the BRISK sampling pattern of pixel locations, long-range pairwise comparisons are exploited to estimate the prominent orientation of a feature, whereas short-range ones are used to build the actual binary descriptor. Scale invariance is obtained by rescaling the pattern according to the inherent scale σ of the detected keypoint.

Similarly to the case of BRISK, *Fast REtinA Keypoints* (FREAK) [43] uses a novel sampling pattern of points inspired by the human visual cortex, whereas *Oriented and*

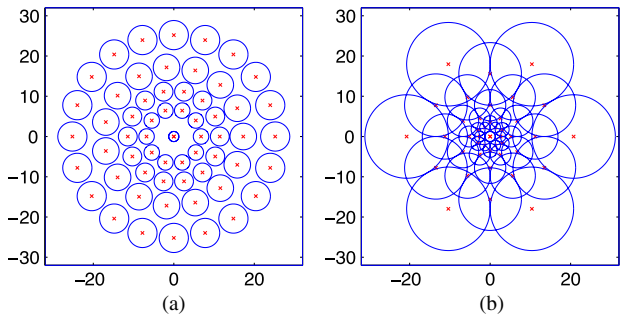

Fig. 4. (a) BRISK and (b) FREAK patterns of pixel locations (in red) used to perform pairwise intensity comparisons. Blue circles corresponds to Gaussian kernel used to smooth local pixel intensities.

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

Rotated BRIEF (ORB) [42] adapts the BRIEF descriptor, so that it achieves rotation invariance.

Byrne and Shi propose Nested Shape Descriptors [96], constructed by pooling oriented gradients over a large geometric structure, which is constructed with a nested correlation structure. Such descriptors are an example of binary descriptors using gradients instead of intensity comparisons and are shown to obtain performance very similar to SIFT on affine image-matching tasks.

3) Machine-learned descriptors

Similarly to the case of keypoint detectors, most of traditional feature descriptors are the result of human intuitions and thus handcrafted. Nonetheless, the availability of large sets of annotated training data is recently being exploited in order to learn effective yet compact feature descriptors.

Winder *et al.* [97] optimize the *DAISY* descriptor [40] exploiting a large dataset of image patches and resorting to machine-learning techniques.

Besides optimizing traditional handcrafted descriptors, machine learning can be used to implement feature extraction algorithms from scratch. *Discriminative BRIEF* (D-BRIEF) [41] learns discriminative feature representations starting from the pixel-level data. In particular, consider the vector $\mathbf{x}$ containing all the pixel intensity values of the image patch p surrounding a given keypoint. Each D-BRIEF descriptor element $D_i(p), i = 1, \ldots, n_d$ is obtained as a thresholded projection of the values of the vector $\mathbf{x}$, that is,

$$D_i(p) = \mathrm{sgn}(\mathbf{w}_i^T \mathbf{x} + \tau_i), i = 1, \ldots, n_d, \tag{8}$$

where $\mathbf{w}_i$ is a vector containing the weights of the ith projection (or linear combination) of the input patch p and τ_i is an arbitrary binarization threshold. The projection vectors $\mathbf{w}_i, i = 1, \ldots, n_d$ and the thresholds $\tau_i, i = 1, \ldots, n_d$ are obtained by minimizing the classification error on the training dataset of patches, exploiting gradient descent. To obtain a fast extraction algorithm, each projection is approximated by means of a combination of few simple kernels (e.g. Gaussians and box-filters).

Binary descriptors from AsymMetric BOOsting (BAMBOO) [44] exploits a greedy boosting procedure inspired by Adaboost to learn a pattern of pairwise smoothed intensity comparisons, used to build a binary descriptor with a procedure similar to that of BRIEF. In particular, each pairwise comparison of smoothed pixel intensities can be expressed as a thresholding of a projection of the image patch, composed by two (or more) box filters, as shown in Fig. 5. Besides learning novel patterns, BAMBOO can be exploited to train traditional binary descriptors, such as BRISK and FREAK, on task-dependent patch datasets, significantly improving their accuracy by selecting their most discriminative descriptor elements.

BINary BOOSTed descriptor (BINBOOST) [45] is a gradient-based binary descriptor obtained exploiting a boosting procedure. Considering the image patch surrounding a given keypoint, local intensity gradients are first computed similarly to the case of traditional real-valued descriptors such as SIFT or SURF. Then, SIFT would pool the resulting gradients on a handcrafted 4×4 grid. Instead, BINBOOST exploits a boosting procedure to learn discriminative gradient pooling functions. On the one hand, BINBOOST yields a high level of matching precision and a high grade of specialization, achieved by performing training on task-specific datasets. On the other hand, BINBOOST requires the computation of local gradients and complex pooling functions, resulting in a computational complexity comparable with that of SIFT.

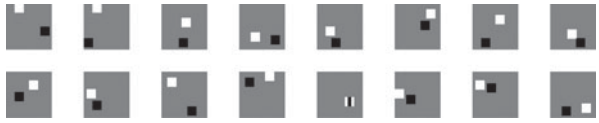

Fig. 5. The best 16 pairwise smoothed intensity comparisons learned by BAMBOO, exploiting a dictionary of box- and Haar-like filters.

Finally, *Binary Online Learned Descriptor* (BOLD) [98] combines the advantages of classical binary descriptors with the improved performance of learning-based descriptors. In particular, BOLD adapts the binary tests to the content of each patch and demonstrates performance that matches that of SIFT with a computational complexity similar to BRIEF.

Besides traditional machine learning approaches, the availability of fast parallel computation architectures such as GPU and cluster computing facilities have favored the deep learning revolution, and a whole new line of research is addressing the problem of extracting and matching local features resorting to deep learning techniques. For what concerns, feature extraction, deep learning have been successfully exploited to improve the quality of the descriptors and their invariance with respect to changes in imaging conditions [99–101]. Deep-learning methods have been also applied to the context of 3D and multi-view features, again demonstrating dramatic improvements int the quality of local representations and allowing for the recognition and matching of complex 3D shapes [102, 103]. Deep learning is also used for matching features. As an example, Fischer *et al.* proposed a feature matching strategy based on deep neural networks, achieving better results than the ones obtained by the traditional feature matching pipelines [104]. Finally, deep learning can be used to perform feature extraction and matching simultaneously: Han *et al.* propose *MatchNet* [105], a unified approach consisting of a deep convolutional network that extracts features from patches and a network of three fully connected layers that computes a similarity between the extracted features. Such a unified approach is shown to improve the accuracy over the previous state-of-the-art results on patch matching datasets. Nonetheless, the use of deep learning in the context of computing and matching local features is still vastly unexplored and it is still a hot topic. Being able to spot complex relationship between visual data, deep neural networks, coupled with very large dataset of image patches could lead to dramatic improvements in terms of feature invariance and robustness.

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

4) Performance evaluation

Many works have tried to compare the performance of different feature extraction algorithms. For instance, the works in [12, 13, 24, 38, 85] all describe comparative studies on the performance of both detectors and descriptors, each work incrementing the set of tested algorithms with the most recent advances in the field. A common denominator of such works is that they do not identify a single winning technology, as this would require extensive testing over all possible visual content analysis tasks. Therefore, such works generally compare different detector/descriptor combinations over a series of standard tests on publicly available datasets. For what concerns detectors, the processing time and the reliability (i.e., the ability to detect the same keypoint under different viewing conditions) are generally used as performance metrics. As for descriptors, the processing time and the percentage of correct true positive matches are generally adopted. *Ad hoc* performance measures may be used if a particular visual task is under test: as an example, in the case of image retrieval or object recognition, the Mean of Average Precision (MAP) is generally used. Conversely, in the case of object tracking, structure-from-motion or camera calibration, the precision in estimating the homography between two images is adopted [10]. Tables 4 and 5 report the MAP value and the homography estimation precision obtained running different couples of feature detector/descriptor over several publicly available image datasets for image retrieval and object tracking, while Table 6 reports the processing time needed for feature extraction in different configurations. The inspection of such results confirms that SIFT features generally obtain very good results, and that is why they are widely accepted as the gold standard solution for feature extraction in several analysis tasks. At the same time, the computational time needed for their extraction is extremely high. This constitutes a limit in those cases where visual content analysis must be performed in real-time or using low-power hardware. Conversely, low-complexity binary features such as BRISK or BAMBOO sometimes perform at par or even outperform their real-valued counterpart such as SIFT or SURF, at just a fraction of the required computational time. This result is very promising as it forms the basis for visual content analysis on low-cost and low-power architectures such as embedded systems, mobile platforms, and VSN.

Table 4. MAP on Oxford, Turin, and Zurich Building dataset.

Detector	Descriptor	Oxford	Turin	Zurich
DoG	SIFT	0.438	0.795	0.792
SURF	SIFT	**0.584**	0.765	0.779
SURF	SURF	0.387	0.772	0.695
ORB	ORB	0.371	0.702	0.743
BRISK	BRISK	0.460	0.798	0.803
ORB	FREAK	0.357	0.675	0.690
SURF	BRISK	0.501	**0.832**	0.763
SURF	FREAK	0.436	0.814	0.670
SURF	BINBOOST	0.213	0.567	0.473
SURF	BAMBOO	0.478	0.789	0.744
BRISK	BINBOOST	0.243	0.521	0.489
BRISK	BAMBOO	0.457	0.787	**0.813**

Table 5. Homography estimation precision on the Visual Tracking Dataset [106].

Detector	Descriptor	HEP
SIFT	SIFT	**0.71**
SURF	SURF	0.68
SURF	BRISK	0.64
BRISK	BRISK	0.69
SURF	BINBOOST	0.66
BRISK	BINBOOST	0.64

Table 6. Average amount of time required to compute 500 local descriptors.

Descriptor	Time (ms)
SIFT	43.5
SURF	13.4
BRISK	2.11
ORB	1.36
FREAK	1.09
BAMBOO	2.79
BINBOOST	97.2

III. VISUAL FEATURE COMPRESSION

In recent years, visual features have been successfully exploited in a number of high-level applications. Distributed analysis tasks such as augmented reality, content-based search, assisted navigation, require visual data, either in the form of pixel-level information (CTA) or visual features (ATC), to be transmitted over a network. Furthermore, most of such applications require visual content to be matched against large-scale databases.

Nonetheless, the whole visual content analysis process should be performed in an efficient fashion, since small delay, typically of the order of tens or hundreds of milliseconds, and high frame rates are required. To this end, feature compactness is key since it allows a very large amount of visual information to be efficiently stored and queried. Moreover, concise feature-based representations can be efficiently transmitted in bandwidth-constrained scenarios such as VSN or congested mobile networks [2].

In this context, *ad hoc* coding methods tailored to visual features are key to the success of distributed visual analysis architectures. Again, such coding methods can be classified into two broad categories: local feature compression and global feature encoding. According to the former, thoroughly covered in Section III-A, local features extracted from an image are compressed resorting to either lossy or lossless coding. Usually, the location information of each feature is as well compressed and transmitted, allowing for the use of geometric verification methods to refine feature matches (see Section III-B).

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

Table 7. Overview of visual feature coding methods.

	Year	Real input	Bin input	Local	Global
PCA-SIFT [46]	2004	✗		✗	
SSC [47]	2006	✗		✗	
LSH [48]	2008	✗		✗	
SH [49]	2011	✗		✗	
SSH [50]	2012	✗		✗	
CHoG [51]	2009	✗		✗	
KLT [52]	2009	✗		✗	
Low-bitrate [53]	2012	✗		✗	
Multi-stage [54]	2012	✗		✗	
Product Q [55]	2013	✗		✗	
LDA-Hash [56]	2012	✗		✗	
Rate-accuracy [57]	2013		✗	✗	
Predictive [58]	2013		✗	✗	
Cluster [59]	2013		✗	✗	
BoW [5]	2004	✗			✗
Tree [61]	2006	✗			✗
Pyramid [60]	2006	✗			✗
KC [62]	2008	✗			✗
Sparse Coding [63]	2009	✗			✗
LLC [64]	2010	✗			✗
HE [65]	2008	✗			✗
VLAD [66]	2010	✗			✗
Fisher [67]	2010	✗			✗
Super Vector [68]	2010	✗			✗
BoBW [69]	2013		✗		✗
BVLAD [70]	2014		✗		✗

A different approach is taken by global features, presented in Section III-C, that create a global representation of an entire frame by pooling and encoding a set of local features. Such global representations are essential when considering very large-scale applications, where matching efficiency is crucial. By discarding keypoint location information, such methods do not enable geometric verification. Nonetheless, some global feature encoding approaches are able to capture spatial information to some extent, by pooling and aggregating features using *ad hoc* spatial patterns. Table 7 offers an overview of the most common feature coding algorithms.

A) Local feature compression

The implementation of distributed visual analysis architectures calls for effective methods to reduce the dimensionality of local features. Gradient-based descriptors such as SIFT and HoG represent the state of the art for a number of applications, and since their introduction a growing body of research has been investigating effective compression techniques tailored to such signals.

1) Real-valued descriptors

Yan Ke and Sukthankar propose *PCA-SIFT* [46]. Similarly to the case of SIFT, gradients are computed within the image patch surrounding each identified keypoint. Differently from SIFT, such gradients are not pooled and aggregated on a spatial grid. Instead, PCA is exploited to project the data into a lower-dimensional space. Such a projection can be learned offline, resorting to a large training set of patches along with the corresponding local gradients, and then efficiently applied to input samples. Efficiently projecting the gradient maps, *PCA-SIFT* generates very compact yet discriminative local features.

Shakhnarovich proposes SSC [47], a machine-learning approach that learns how to embed a real space into a binary space, preserving distances between elements. Shakhnarovich tests such an algorithm on SIFT descriptors, to quantize their element into binary values, significantly reducing the number of bits needed to store local features.

Yeo *et al.* propose a novel local feature compression method, based on LSH [48]. Consider a random projection to be applied to a descriptor vector. In particular, such a projection splits the descriptor space in two regions by means of a hyperplane. The key intuition behind the approach is that, if two descriptors are close, then they lie on the same side of the hyperplane for a large set of projections. Hence, for each projection, a one-bit hash can be computed based on the side of the hyperplane a projected descriptor falls in. Finally, a binary hash is obtained by concatenating the results of a number of random projections, and descriptors can be matched resorting to Hamming distance. The process has been further refined by Kulis and Grauman [107].

Weiss *et al.* propose *Spectral Hashing* [49, 108]. Instead of using random projections as in the case of LSH, such approach applies PCA on the input data to identify the k-principal components, and then creates a hashing function based on such components.

Wang *et al.* propose a set of *SSH* techniques [50] that can be effectively applied to local features in the context of large-scale search. Such approaches exploit a partially annotated training dataset to learn a set of projections that lead to highly discriminative hashes of the input signal. In particular, *Sequential Projection Learning* offers the best performance iteratively optimizing the output hash. According to such a method, the projection learned at each step is able to improve the hash accuracy, making up for errors due to previously learned projections.

Strecha *et al.* introduce *LDAHash* [56], a hashing technique tailored to real-valued local features. Such a technique exploits a large training dataset of descriptors. The descriptors are annotated, so as to recognize the ones corresponding to the same physical point. Then, a set of projections and binarization thresholds are learned, in order to map the real-valued descriptor space into a low-dimensional binary space. Regarding the projections, they are learned resorting to *Linear Discriminant Analysis*, so that the covariance between projected descriptors referring to the same physical entity is minimized, and at the same time the covariance between descriptors of different classes is maximized.

Chandrasekhar *et al.* propose *Compressed Histogram of Gradients* (CHoG) [51], a very compact gradient-based local feature. Similarly to SIFT and GLOH, it computes and pools gradients within the image patch surrounding each detected keypoint, so as to generate a descriptor composed of a number of histograms of gradients. CHoG models such descriptors as tree structures, and exploits tree coding algorithms to reduce the number of bits needed to encode each feature.

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

Furthermore, a method to match descriptors in the compressed domain is proposed, so that descriptors need not to be decoded before being matched, yielding significant improvements in terms of both memory consumption and computational efficiency.

Moreover, Chandrasekhar *et al.* propose a compression architecture tailored to real-valued features, based on the Karhunen-Loève Transform (KLT) [14, 52]. In particular, KLT is applied in order to decorrelate the input descriptor elements, the resulting transformed values are quantized and finally symbols are entropy coded.

Redondi and Cesana propose a coding architecture that exploits the correlation between features extracted from the same frame [53]. In particular, the optimal descriptor coding order is computed so as to minimize the expected bitrate needed to encode the features in a predictive fashion. Similarly to [52], KLT is used to decorrelate descriptor elements.

Jegou *et al.* propose a local feature compression algorithm based on product quantization [55]. According to such approach, a P-dimensional input descriptor $\mathbf{d}_i$ is split into m subvectors, each consisting of P/m elements. Then, the m subvectors are quantized separately, yielding the m-quantized symbols $q_{i,1}, \ldots, q_{i,m}$. The global quantization value Q_i for the input descriptor $\mathbf{d}_i$ is obtained as $Q_i = \prod_{j=1}^{m} q_{i,m}$.

Chen *et al.* resorts to a multi-stage quantization process to improve coding efficiency [54]. First, they apply coarse vector quantization to a P-dimensional input descriptor. Being a lossy process, a P-dimensional residual error is generated. Then, product quantization is applied on such residual, yielding improved distinctiveness.

Even though many studies evaluate the performance of local feature compression algorithms [13, 109], it is difficult to identify the best approach for all tasks and scenarios. A common denominator of all the experiments is that the accuracy of compressed real-valued features tends to saturate at about 140–170 bits/feature. That is, 140–170 bits are capable of capturing the characteristics of the image patch surrounding a keypoint, whereas richer representations do not yield significant accuracy gains.

2) Binary descriptors

Most feature compression and hashing techniques are tailored to the class of real-valued features such as SIFT or HoG. Nonetheless, the advent of fast yet accurate binary feature extraction algorithms such as BRISK calls for effective coding methods tailored to such binary signals. In particular, the peculiar binary nature of such class of features should be taken into account when designing *ad hoc* coding algorithms. [110] propose a lossless binary feature coding technique. The main idea behind such approach is that binary descriptor elements, usually being the result of pairwise intensity comparisons, are correlated. Thus, a greedy technique is developed so as to find the permutation of descriptor elements that minimizes the conditional entropy of the signal, so that coding efficiency is maximized.

Ascenso *et al.* propose a predictive coding architecture tailored to binary features [58, 111]. Similarly to what has been done in [53], the correlation between features extracted from the same frame is exploited in order to improve coding efficiency. In particular, extracted binary descriptors are permuted so as to minimize the expected bitrate, resorting to a greedy procedure. Then, entropy coding is exploited to encode the prediction residual between couples of features that are contiguous within such permutation.

Furthermore, Ascenso *et al.* introduce a clustering-based coding technique tailored to binary local descriptors [59]. In particular, given a set of binary descriptors extracted from a frame, similar features are grouped in the same cluster. Then, within each cluster, correlation between features is exploited to efficiently encode the descriptors in a predictive fashion.

In summary, considering local binary descriptors, lossless compression yields a bitrate reduction of up to 30%.

B) Coding of keypoint locations

Keypoint location information is essential for a number of visual content analysis tasks such as object localization and tracking and structure from motion. Furthermore, content-based retrieval architectures based on local features often exploit a geometric consistency check to refine the matches between query and database images, and thus need to know the position of local features and their relationship. In this context, a body of research addresses the problem of efficiently encoding the location of keypoints detected in a frame. A naive approach is based on scalar quantization of keypoint coordinates, followed by entropy coding of the quantized symbols [10]. Tsai *et al.* [72] observe that feature are usually clustered around highly textured regions, and thus the probability of finding one or more keypoints in a given area depends on the presence of other keypoints nearby such area. To this end, a spatial grid is applied to the input frame, and a histogram counting the number of keypoints lying in each spatial bin is constructed. Then, a context-based arithmetic coder is used to efficiently encode the number of keypoints in each bin, exploiting the spatial context, that is, the number of keypoints lying in neighboring bins. Recently, the rate-accuracy performance of such histogram-based location coder has been enhanced resorting to complex coding contexts [71].

C) Global feature encoding

Besides local feature compression methods, global features are proposed as a way of compactly representing visual content. The key idea behind these approaches is to create a global, compact signature for an entire image, based on the set of local features extracted from it. Global representations are much more concise than local ones, requiring a lower amount of memory to be stored and less bandwidth to be transmitted.

The simplest global feature encoding method is based on a well-known information retrieval model, i.e.

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

"Bag-of-Words" (BoW). In the context of text-based retrieval, the key assumption of such a model is that, given a large dictionary of words, each document can be represented as a "BoW", that is, a histogram counting the number of occurrences of each word within the document. Hence, each document is represented by a single histogram, that is, a vector of real-valued entries. Given a query document, it is possible to find the most relevant retrieval results by simply computing the distances between such histograms. Furthermore, since the size of the dictionary is much greater than the size of a document, histograms are sparse, and the matching efficiency can be improved making use of *ad hoc* tools such as inverted indices. Similarly, images can be represented by means of a BoW model. Although the meaning of "words" in the context of text-retrieval is straightforward, it should be clearly defined in the case of content-based image retrieval. In such a case, each word (or visual word) can be thought of as an image patch, having distinctive visual characteristics. Local features are an effective way of describing the characteristics of an image patch, and are thus a good fit for the problem at hand. In this respect, in the context of computer vision such a model is often referred to as as "Bag-of-Features" or "Bag-of-Visual-Words". Considering P-dimensional real-valued or binary descriptors, a dictionary with K visual words can be represented by means of K P-dimensional descriptors, each representing a different visual word.

1) Global encoding of real-valued features

Sivic and Zisserman propose *Video Google*, an image-matching approach based on "BoW" [5]. To construct a dictionary of visual words, a large number of real-valued P-dimensional descriptors $\mathbf{d}_i \in \mathbb{R}^P$ are computed starting from a training set of images. Then, the descriptors are vector quantized into K clusters, whose centroids $\mathbf{v}_k, k = 1, \ldots, K$, represent the actual visual words composing the dictionary $\mathbf{V}$. Dealing with real-valued features, k-means is exploited to cluster the training set of descriptors into a number of visual words. Once the dictionary has been defined, a "BoW" representation can be computed for each input image. In particular, given an image, local features are extracted. Then, each feature is associated with the most similar visual word composing the dictionary, i.e. the dictionary centroid with minimal Euclidean distance with respect to the input feature, as shown in Fig. 6(a). Finally, the image is represented by means of a histogram that counts the occurrences of all dictionary words. A database is built by assigning a BoW representation to each image, so that it can be efficiently queried. To improve matching accuracy, histogram vectors are normalized according to a *tf–idf* scheme, which is common in the text-based retrieval [112]. Finally, given a query image, a global BoW representation is built and it is matched against database entries, resorting to, e.g. cosine similarity. Relevant results correspond to database entries whose cosine similarity with respect to the query is higher than an arbitrary threshold.

Considering systems based on the "BoW" model, the size of the visual vocabulary influences the matching accuracy. In the case of large-scale retrieval, up to hundreds of thousands or even millions of visual words are needed to obtain performance saturation. In this regard, Nister and Stewenius [61] refine the model proposed by Sivic *et al.* by introducing a vocabulary tree. In particular, the vocabulary is built as a hierarchical structure, where each level refines the partitioning of the descriptor space. Using such an architecture and *ad hoc* matching algorithms, large dictionaries can be seamlessly exploited, achieving high task accuracy, without significantly affecting matching performance.

"BoW" represents a simple model that enables fast yet accurate large-scale image matching. Nonetheless, by building a unique, global representation for a frame, it completely disregards the position of local features and their relationship. Such information, if included in the image

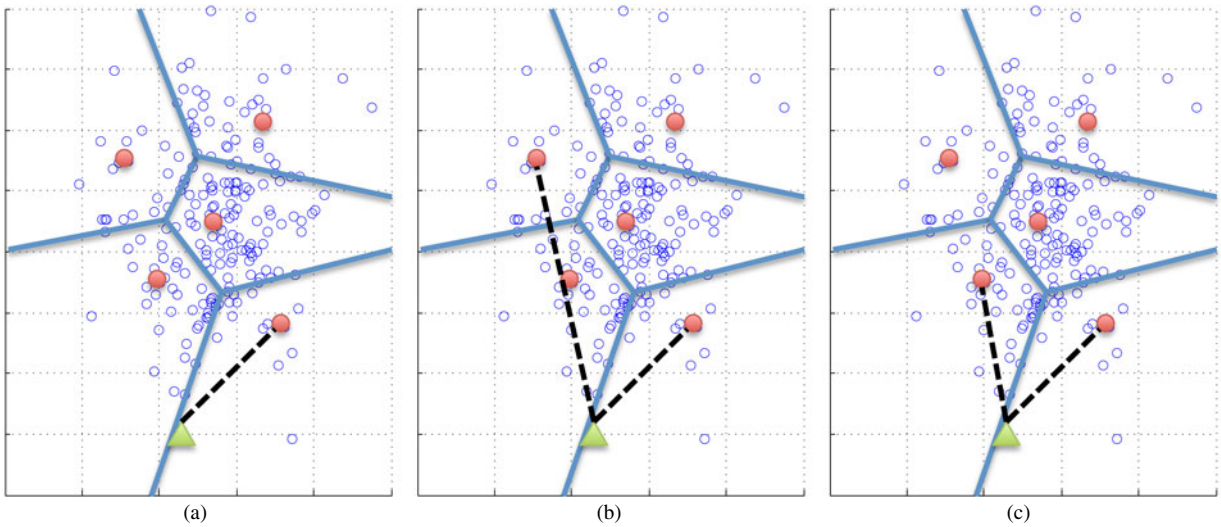

Fig. 6. (a) "Bag-of-Features" assigns the input feature (green triangle) to its nearest visual word (b) *Sparse Coding* approximates the input feature as a combination of few words (c) *Locality-constrained Linear Coding* constraints the visual words composing the sparse combination to be near to the input feature in the descriptor space.

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

representation, can significantly improve the discriminative and matching capabilities of a global feature. In this sense, Lazebnik *et al.* propose *Spatial Pyramid Kernel* [60], a global feature encoding aimed to address such issue. Considering an input image, features are extracted resorting to state-of-the-art algorithms such as SIFT. Then the image is partitioned into increasingly fine non-overlapping spatial sub-regions, and a "BoW" representation is computed separately for each sub-region. Finally, a hierarchical global feature representation is obtained by concatenating the histograms corresponding to all the sub-regions. Besides such pyramidal feature encoding, *ad hoc* efficient distance metrics for the representations at hand are introduced, aimed to enable the matching of *Spatial Pyramid Kernels*.

Gemert *et al.* identify a few issues of the "BoW" model, and propose KC [62]. In particular, according to traditional "BoW" approach, features are hard-assigned to the first nearest centroid in the descriptor space. Gemert *et al.* point out two main weaknesses related to such approach: (i) assignment plausibility and (ii) assignment uncertainty. As regards the former, it is strictly related to the curse of dimensionality [113]. Since visual vocabularies are usually learned by clustering a high-dimensional space (e.g., $P = 128$ for SIFT), such space may not be densely covered by learned centroids. In this context, at encoding time, the nearest centroid of an input local feature may be really distant in the descriptor space, and the hard-assignment of such feature to the corresponding centroid may not be plausible. Furthermore, an input local feature may be almost equally distant to two or more visual words, and thus hard-assignment would pertain a high uncertainty, which significantly hurts the performance of the "BoW" approach. To tackle such an issue, Gemert *et al.* propose to soft-assign local features to multiple nearest visual words by using smoothing kernels, significantly improving the discriminative power of global representations. Philbin *et al.* propose a similar global feature encoding method based on *Soft Coding* [114], achieving comparable accuracy performance.

Jianchao Yang *et al.* [63] propose a soft-assignment approach based on *Sparse coding*. In order to learn a visual vocabulary, "BoW" applies k-means clustering to solve the following problem:

$$\begin{aligned} \min_{\mathbf{q}_i, \mathbf{V}} \quad & \sum_{i=1}^{N} \|\mathbf{d}_i - \mathbf{q}_i \mathbf{V}\|^2 \\ \text{s.t.} \quad & \|\mathbf{q}_i\|_0 = 1, \|\mathbf{q}_i\|_1 = 1, \end{aligned} \tag{9}$$

where $\mathbf{V}$ represents the visual vocabulary, that is, a matrix with K rows $\mathbf{v}_k, k = 1, \ldots, K$, corresponding to the K dictionary words, and $\mathbf{q}_i$ a vector that assigns feature $\mathbf{d}_i$ to the nearest word in the matrix $\mathbf{V}$. Constraints force $\mathbf{q}_i$ to be a vector of zeros, containing a single one, so as to hard-assign training feature $\mathbf{d}_i$ to its first nearest neighbor when applying k-means clustering. Yang *et al.* propose to substitute k-means clustering with a *Sparse coding* approach, that aims to solve the following problem:

$$\begin{aligned} \min_{\mathbf{q}_i, \mathbf{V}} \quad & \sum_{i=1}^{N} \|\mathbf{d}_i - \mathbf{q}_i \mathbf{V}\|^2 + \lambda \|\mathbf{q}_i\|_1 \\ \text{s.t.} \quad & \|\mathbf{v}_k\|_2 \leq 1, \quad \forall k = 1, \ldots, K. \end{aligned} \tag{10}$$

The hard-assignment constraint in equation (9) is substituted with a penalization term that favors sparsity. That is, each training local features contributes to a small number of centroids (see Fig. 6(b)) and, in particular, λ controls the assignment sparsity. *Sparse Coding* is applied during both training and encoding, to learn the set of visual words composing the dictionary and to create global representations via soft histograms of words, respectively.

Wang *et al.* further improve the performance of global representations based on *Sparse Coding*, introducing LLC [64]. They observe that *Sparse Coding* approximates an input feature as a combination of few visual words. The selected visual words may be very distant to the input local feature in the descriptor space, possibly reducing the discriminative power of the global encoding. Instead, besides seeking a sparse combination of visual words that minimizes the distance with respect to the input features, LLC requires that the selected visual words are *local* to the input feature, that is, they are near to the input feature in the descriptor space, as shown in Fig. 6(c). Such constraint can be efficiently included in the optimization problem, yielding discriminative global representations that can be computed fast.

Jegou *et al.* present HE [65], an alternative approach to construct global representations starting from local descriptors, aimed to tackle the curse of dimensionality. Instead of exploiting soft assignment and high-dimensional dictionaries, *HE* defines a small number of coarse centroids and builds discriminative representations based on assignment residual. In particular, each input feature is hard-assigned to the closest centroid, but its location within the Voronoi cell defined by the centroid is refined with a short binary signature. In particular, each bit of the binary signature is obtained by analyzing the position of the feature with respect to a hyperplane that subdivides the Voronoi cell.

Jegou *et al.* also introduce *Vector of Locally Aggregated Descriptors* (VLAD) [66]. VLAD builds more complex yet representative global encodings by pooling the residual error due to the assignment of features to visual centroids. In particular, consider a P-dimensional descriptor $\mathbf{d}_i \in \mathbb{R}^P$ extracted from an image, and a visual vocabulary $\mathbf{V}$, consisting of K P-dimensional visual words, i.e. $\mathbf{V} = \{\mathbf{v}_k\}, k = 1, \ldots, K$. Assigning each descriptor to nearest centroid results in a P dimensional assignment residual, that is, $\mathbf{r}_i = \mathbf{d}_i - \mathbf{v}_{\mathrm{NN}_i}$, where $\mathbf{v}_{\mathrm{NN}_i} \in \mathbf{V}$ represents the nearest dictionary centroid with respect to the input feature $\mathbf{d}_i$. VLAD builds a global image representation by assigning each input descriptor to its nearest visual words, and by pooling the resulting assignment residuals. In particular, for each visual word $\mathbf{v}_k, k = 1, \ldots, K$, a P-dimensional vector is obtained by summing the residuals $\mathbf{r}_i$ relative to

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

Table 8. Image classification accuracy achieved by global feature encoding algorithms as reported in [15].

Descriptor	No. of words	MAP
PASCAL VOC 2007		
BoW	25 k	0.561
KC	25 k	0.563
LLC	25 k	0.577
Fisher	256	**0.617**
Super Vec	1024	0.582
CALTECH 101		
BoW	8 k	0.742
KC	8 k	0.759
LLC	8 k	0.769
Fisher	256	**0.778**
Super Vec	–	–

descriptors associated to such word. That is,

$$\mathbf{g}_k = \sum_{i:\mathrm{NN}(\mathbf{d}_i)=\mathbf{v}_k} \mathbf{r}_i = \sum_{i:\mathrm{NN}(\mathbf{d}_i)=\mathbf{v}_k} \mathbf{d}_i - \mathbf{v}_k, \quad k = 1, \ldots, K, \tag{11}$$

where $\mathbf{g}_k, k = 1, \ldots, K$ is the set of K P-dimensional vectors composing the VLAD representation, and $\mathrm{NN}(\mathbf{d}_i)$ indicates the nearest neighbor of $\mathbf{d}_i$ within the visual vocabulary $\mathbf{V}$. The final VLAD representation is a $K \times P$ vector obtained by concatenating the vectors $\mathbf{g}_k$. Differently from "BoW", that requires up to millions of visual words, VLAD yields discriminative global representations using as few as tens or hundreds of centroids. Finally, dimensionality reduction techniques such as PCA or Hashing can be used to reduce the dimensionality of the VLAD feature vector.

Perronin *et al.* propose to use *Fisher Kernels* [115] to build an effective global representation, starting from a set of local features. *Fisher Kernels* are able to combine the power of generative models, e.g. Gaussian Mixture Models (GMM), and discriminative classifiers, e.g. SVM. In particular, a visual vocabulary is built by fitting a GMM to a large set of training descriptors. The GMM associates each feature vector $\mathbf{d}_i$ to the mode (or centroid) $\mathbf{v}_k$ with a strength q_{ik}, based on the posterior probability of $\mathbf{d}_i$ having been generated from such Gaussian mode. The values of q_{ik} may be viewed as soft assignment weights. VLAD simply computes the deviation vector of feature $\mathbf{d}_i$ with respect to the nearest centroid, and pools all the deviations relative to the same centroid. Similarly, *Fisher Kernel* computes the mean deviation vector of features with respect to each centroid, weighting each contribution with q_{ik}. Besides first order statistics, that is, mean deviation, the covariance deviation vector with respect to each GMM centroid is computed. The final representation is obtained by concatenating both first- and second-order deviation vectors relative to all centroids. Perronin significantly improve the performance of *Fisher Kernel* encoding by introducing effective normalization techniques [67], achieving state-of-the-art performance in terms of global encoding accuracy for image classification [15], as shown in Table 8.

Zhou *et al.* introduce *Super vector* coding (SV) [68]. Similarly to the case of *Fisher Kernel*, SV soft assigns each feature vector $\mathbf{d}_i$ to codebook centroid $\mathbf{v}_k$ by means of weight q_{ik}. Then, a global representation is built based on two terms: (i) pooled first-order deviations and (ii) mass of feature clusters. As regards the former, identically to the case of *Fisher Kernel*, it is obtained by pooling the mean deviation of features with respect to centroids. As to the latter, for each centroid $\mathbf{v}_k, k = 1, \ldots, K$, the associated cluster mass s_k can be computed as

$$s_k = s \cdot \sqrt{\frac{1}{N} \sum_{i=1}^{N} q_{ik}}, \quad k = 1, \ldots, K, \tag{12}$$

where s is a constant and N is the total number of input local features. In practical terms, the mass of cluster s_k indicates how much the input features $\mathbf{d}_i, i = 1, \ldots, N$ contribute, in terms of weights q_{ik}, to centroid $\mathbf{v}_k$.

Finally, a number of methods have been proposed in order to directly construct global image representations [116, 117], without resorting to local features as an intermediate step, but are out of the scope of this survey.

In the context of image classification, the performance of global feature encoding approaches has been thoroughly evaluated and compared [15]. Table 2 shows the classification accuracy, in terms of *Mean Average Precision*, for a subset of encoding methods, as reported in [15][3]. Global feature encoding has been recently outperformed by more complex methods based on deep neural networks [118].

2) Global encoding of binary features

Traditional approaches aim to find a low-dimensional global representation for a set of real-valued features. With the advent of computationally efficient yet discriminative binary descriptors, a growing body of research is addressing the problem of constructing effective global encodings tailored to such category of local features. In particular, considering traditional real-valued features such as SIFT, the process of building a visual codebook is usually based on a clustering of the real-valued descriptor space $\mathbb{R}^P$. Instead, in the case of binary descriptors, alternative techniques should be developed in order to cluster the P-dimensional binary space $\{0, 1\}^P$. To this end, k-means can be adapted to the peculiar nature of the signal at hand, or it can be substituted by means of *ad hoc* clustering algorithms such as k-medians or k-medoids, yielding comparable results [11, 69, 119].

More recently, more effective global encodings tailored to binary local features have been proposed. Steinbach and co-workers propose BVLAD [70], an adaptation of the VLAD feature encoding algorithm to the context of binary local features. Similarly to VLAD, each local feature $\mathbf{d}_i$ is assigned to the nearest centroid $\mathbf{v}_{\mathrm{NN}_i}$, in terms of Hamming distance, and the assignment residual is computed as $\mathbf{r}_i = \mathbf{d}_i \oplus \mathbf{v}_{\mathrm{NN}_i}$, where $\oplus$ represents the exclusive or (XOR)

[3] *Spatial Pyramid Kernels* are used in combination with all the proposed approaches.

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

operator, i.e. a difference operator in the binary space. Identically to the case of VLAD, residual vectors are pooled over the centroids, and their dimensionality is reduced resorting to PCA.

D) Hybrid visual content/feature encoding

As described in Section I, the ATC paradigm is getting more and more attention in both the scientific community and the industry. ATC moves part of the analysis to sensing nodes, that extract, encode and transmit visual features to a sink node that performs higher level analysis. On the one hand, ATC makes a more efficient use of storage and transmission resources compared to the traditional CTA paradigm, yielding compact yet discriminative signatures. Furthermore, avoiding the image or video encoding process inherent to CTA, it generates visual features that are not affected by distortion introduced by pixel-level coding, such as ringing and block boundary artifacts. On the other hand, in ATC the pixel-level visual content is unavailable at the sink nodes and it can not be shown to the users or used for other purposes.

To overcome such issues, several hybrid paradigms for visual content analysis, aimed to combine the benefits of both ATC and CTA, have been proposed recently. In 2011, Chao and Steinbach propose to adapt the JPEG image compression method so as to preserve the quality of visual features that are extracted from lossy images [73]. Similarly, H.264/AVC video coding architecture can be modified, so that the lossy encoding process does not significantly affect the quality of visual features extracted from decoded frames [74].

Instead of modifying image or video coding primitives, a number of novel paradigms aim to efficiently encode and transmit both pixel- and feature-level representations. In particular, Chen and Moulin [75] propose a solution to jointly encode images and global representations based on "BoW". A feature enhancement layer is computed and attached to the traditional pixel-level stream, so as to improve the quality of global features extracted from lossy content, possibly impaired by coding artifacts. Similarly, Baroffio *et al.* propose *"Hybrid-Analyze-Then-Compress"* (HATC) [7], a novel visual analysis paradigm aimed to encode and transmit both pixel-level representations and high-quality local features. In particular, since keypoint detection is strongly affected by coding artifacts, the location of keypoints extracted from original, lossless content is sent to the sink node, so that it is possible to detect stable keypoints. Furthermore, a descriptor enhancement layer is encoded and sent, so as to refine local descriptors possibly impaired by pixel-level coding.

IV. FEATURES EXTRACTED FROM VIDEO SEQUENCES

A number of visual content analysis tasks such as object tracking and event detection require visual features to be extracted and processed on a temporal basis, from a sequence of frames. In this context, a body of research is being carried out to introduce effective architectures for the extraction and compression of visual features starting from video sequences.

A) Feature extraction from video sequences

Typically, when considering applications based on video sequences, visual features are extracted and processed on a frame-by-frame basis [8, 77]. In other cases, a GOP is processed concurrently to construct a feature-based temporal representation [11]. In the context of object tracking, stability and repeatability of keypoints detected in contiguous frames is key to achieving good performance. As comprensively presented in Section II, Shi and Tomasi [19] propose some modifications to the Harris corner detector, so that detected keypoints are stable over time and are thus suitable for tracking applications. Triggs [120] thoroughly analyzes the problem of detecting keypoints that are robust with respect to changes in imaging conditions, such as illumination, contrast, and viewpoint. Kläser *et al.* [121] propose a spatio-temporal descriptor based on histograms of 3D gradients computed with respect to the two spatial dimensions and the temporal one. More recently, the works in [9] introduce a temporally coherent keypoint detector. According to such approach, only keypoints that can be accurately detected in a set of contiguous frames are retained, whereas non-repeatable detections are discarded, improving both task accuracy and feature coding performance.

Several tasks are time critical and require frames to be processed at a high rate. In this context, traditional feature extraction algorithms have been modified so that they can be efficiently run on low-power devices [122]. Furthermore, temporal redundancy inherent to video sequences can be exploited to efficiently detect keypoints on a frame-by-frame basis, significantly reducing the computational time needed to process each frame [123].

B) Coding features extracted from video sequences

Besides extracting features from video content, several research studies have been conducted to address the problem of encoding visual features extracted from video sequences.

[8, 10] propose *VideoSIFT*, a coding architecture tailored to real-valued local features such as SIFT or SURF, extracted from video sequences. The coding architecture is inspired by traditional video coding techniques, and it aims to adapt the main building blocks of well-established video coding architectures to the context of local features. A similar architecture [77] has been proposed for binary local features such as BRISK and FREAK.

Makar *et al.* propose an architecture for encoding local features extracted from video sequences. According to such architecture, temporally coherent keypoints are detected, and the patches surrounding such keypoints are encoded

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

in a predictive fashion [9]. Alternatively, descriptors can be encoded in place of patches in a predictive fashion [76]. The temporal correlation between video frames can be exploited to efficiently encode keypoint locations. In particular, methods presented in Section III-B can be adapted to the context of keypoints extracted from video content [8, 76].

Besides local features, the problem of encoding global representations extracted from contiguous frames has been recently addressed [11]. Finally, a number of works addressed the problem of building spatio-temporal features for action detection, recognition, and classification [124, 125], but they fall outside of the scope of the paper.

V. VISUAL FEATURE TRANSMISSION AND NETWORKING

The last decades have seen huge technological leaps that are enabling a whole new range of applications and services. On the one hand, more and more powerful yet compact devices are being introduced. In particular, smartphones, tablets, and smart cameras pervade our everyday lives, and wearable devices will supposedly have a similar impact in next years. On the other end, Internet and the web are becoming ubiquitous, connecting billions of people in social networks, and offering advanced distributed services such as cloud computing. Besides, Wireless Sensor Networks (WSN) and cellular networks are expected to play a big role in the evolution toward an "Internet-of-Things".

In the context of visual content analysis, a number of applications are performed in a distributed fashion, requiring cooperation between sensing devices and central processing nodes. Applications such as mobile visual search, smart camera networking, smart surveillance, computer-assisted driving are gaining popularity and are based on distributed computation. As presented in Sections I and III-D, CTA and ATC are alternative paradigms for distributed applications. The former is a traditional approach that has been successfully exploited in VSN and content-based search applications [126, 127]. Methods based on such a paradigm have been thoroughly investigated and are out of the scope of the survey. Instead, we will present algorithms and methods for feature transmission, networking, and cooperation tailored to the ATC paradigm.

Low-energy consumption plays a crucial role in VSN. Computational and networking capabilities of sensing nodes are usually severely constrained to limit the energy consumption. In this context, ATC represents a promising solution, since it requires a small amount of data to be transmitted to central processing nodes [2].

Yang *et al.* [78, 79] propose a system for object recognition based on a smart camera network. In particular, a number of cameras are deployed over a region, so as to acquire the same scene from different point of views, and connected with a central processing node that performs the analysis. Since the acquired views refer to the same physical scene, features extracted from such views are correlated, too. Correlation is exploited to efficiently encode features from multiple views and to transmit them to the sink node.

Dan *et al.* [80, 128, 129] propose an architecture for offloading part of the computational burden due to feature extraction on wireless sensor nodes. They consider a network composed of sensing nodes, that acquire visual content, and processing nodes, that can be exploited to offload part of the computation. In particular, each acquired image is split in subregions that are assigned to cooperating processing nodes. Each node performs a subtask by extracting features from the assigned regions. Furthermore, the computational load on network nodes can be balanced by properly assigning subtasks.

Similarly, Redondi *et al.* [81, 82] propose a framework for cooperative feature extraction on low-power visual sensor nodes. Several different network configurations and protocols are proposed and empirically evaluated in terms of speed up of feature extraction task, network lifetime, and energy consumption.

Baroffio *et al.* [3] show that network condition can severely affect the accuracy of visual content analysis tasks. In particular, noisy channels and transmission errors may lead to packet loss and transmission delay, impairing the performance of the system. Reliable transfer protocols achieve good task accuracy, since packet loss is prevented, at the cost of increased network delay.

Besides VSN, the ATC paradigm has been effectively implemented in the context of mobile visual search. Girod *et al.* [83] thoroughly analyze the problem and show that ATC represents the most effective option in terms of bandwidth-accuracy performance. In particular, they propose an object retrieval system based on the transmission of CHoG features [51] and compressed keypoint positions. The performance of the mobile visual search system is evaluated in terms of a number of key metrics such as query accuracy, response delay, transmission bitrate, and energy consumption.

VI. CONCLUSION

Distributed visual content analysis is an interesting problem related to a large number of applications, including advanced surveillance, mobile visual search, and augmented reality that is having a huge impact on our everyday lives. Until few years ago, handcrafted features such as SIFT and SURF represented the state of the art for visual content analysis. In particular, SIFT is widely regarded as the gold standard in the context of local feature extraction, and has been partially adopted by the MPEG *Compact Descriptors for Visual Search* (CDVS) [4, 130] standard, which includes: (i) an optimized implementation of SIFT, along with a local feature compression architecture based on multi-stage quantization [54], (ii) a global feature algorithm based on *Fisher Kernels*, and (iii) a keypoint location coding module based on histograms of keypoint positions [83].

SIFT-based solutions have been successfully exploited in both centralized systems running on powerful servers and

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

on portable devices like smartphones. Nonetheless, their computational complexity is still quite high- for low-power devices, and thus they could not be the better choice in the case of limited computational resources and high frame rates [12].

Binary local features such as BRISK and FREAK have been introduced as fast alternatives to SIFT. Some studies proved that they approach the quality of SIFT in terms of discriminative power, while being up to 20 times faster [44]. They are thus a good choice for scenarios in which computational resources are limited, like in the case of VSN nodes or any other low-power or battery-operated devices.

Traditional machine learning techniques like boosting and bagging have been successfully exploited to build effective descriptors [45] or to improve the accuracy performance of existing methods [44]. In the meanwhile, the use of deep learning techniques to detect and describe keypoints looks a promising area of research.

We proposed an overview of the most successful techniques for extracting, encoding, and transmitting compact representations of visual content, describing their evolution during the last two decades. Furthermore, we highlighted and compared the characteristics of each solution, providing indications for some different visual analysis tasks. Such overview may serve as an entry point and a reference for further research in the area.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] Hu, M.-K.: Visual pattern recognition by moment invariants. IRE Trans. Inf. Theory, **8** (2) (1962), 179–187.

[2] Baroffio, L.; Cesana, M.; Redondi, A.; Tagliasacchi, M.: Compress-then-analyze vs. analyse-then-compress: two paradigms for image analysis in visual sensor networks, in *IEEE Int. Workshop on Multimedia Signal Processing (MMSP) 2013*, Pula, Italy, September 2013, 278–282.

[3] Baroffio, L.; Cesana, M.; Redondi, A.; Tagliasacchi, M.: Performance evaluation of object recognition tasks in visual sensor networks, in *2014 26th Intl. on Teletraffic Congress (ITC)*, September 2014, 1–9.

[4] Duan, L.-Y.; Gao, F.; Chen, J.; Lin, J.; Huang, T.: Compact descriptors for mobile visual search and MPEG CDVS standardization, in *2013 IEEE Int. Symp. on Circuits and Systems (ISCAS)*, May 2013, 885–888.

[5] Sivic, J.; Zisserman, A.: Video google: a text retrieval approach to object matching in videos, in *Int. Conf. on Computer Vision*, 2003, 1470–1477.

[6] Chao, J.; Steinbach, E.: Preserving sift features in JPEG-encoded images, in *2011 18th IEEE Int. Conf. on Image Processing (ICIP)*, September 2011, 301–304.

[7] Baroffio, L.; Cesana, M.; Redondi, A.; Tagliasacchi, M.; Tubaro, S.: Hybrid coding of visual content and local image features, in *2015 IEEE Int. Conf. on Image Processing (ICIP)*, September 2015, 2530–2534.

[8] Baroffio, L.; Cesana, M.; Redondi, A.; Tubaro, S.; Tagliasacchi, M.: Coding video sequences of visual features, in *2013 20th IEEE Int. Conf. on Image Processing (ICIP)*, September 2013, 1895–1899.

[9] Makar, M.; Tsai, S.S.; Chandrasekhar, V.; Chen, D.; Girod, B.: Inter-frame coding of canonical patches for mobile augmented reality, in *2012 IEEE Int. Symp. on Multimedia (ISM)*, 2012, 50–57.

[10] Baroffio, L.; Cesana, M.; Redondi, A.; Tagliasacchi, M.; Tubaro, S.: Coding visual features extracted from video sequences. IEEE Trans. Image Process., **23** (5) (2014), 2262–2276.

[11] Baroffio, L.; Canclini, A.; Cesana, M.; Redondi, A.; Tagliasacchi, M.; Tubaro, S.: Coding local and global binary visual features extracted from video sequences. IEEE Trans. Image Process., **24** (11) (2015), 3546–3560.

[12] Canclini, A.; Cesana, M.; Redondi, A.; Tagliasacchi, M.; Ascenso, J.; Cilla, R.: Evaluation of low-complexity visual feature detectors and descriptors, in *2013 18th Int. Conf. on Digital Signal Processing (DSP)*, 2013, 1–7.

[13] Miksik, O.; Mikolajczyk, K.: Evaluation of local detectors and descriptors for fast feature matching, in *2012 21st Intl. Conf. on Pattern Recognition (ICPR)*, November 2012, 2681–2684.

[14] Chandrasekhar, V. et al.: Survey of SIFT compression schemes, in *2nd Int. Workshop on Mobile Multimedia Processing*, 2010, 35–40.

[15] Chatfield, K.; Lempitsky, V.; Vedaldi, A.; Zisserman, A.: The devil is in the details: an evaluation of recent feature encoding methods. British Machine Vision Conf., **2** (4) (2011), 8–20.

[16] Moravec, H.P.: Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover. Ph.D. thesis, Stanford University, California, Department of Computer Science, Stanford, CA, USA, 1980. AAI8024717.

[17] Kitchen, L.; Rosenfeld, A.: Gray-level corner detection. Pattern Recognit. Lett., **1** (2) (1982), 95–102.

[18] Harris, C.; Stephens, M.: A combined corner and edge detector, in *Proc. Alvey Vision Conf.*, Alvey Vision Club, 1988, 23.1–23.6.

[19] Shi, J.; Tomasi, C.: Good features to track, in *1994 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'94)*, 1994, 593–600.

[20] Lee, J.; Sun, Y.; Chen, C.: Multiscale corner detection by using wavelet transform. IEEE Trans. Image Process., **4** (1) (1995), 100–104.

[21] Smith, S.M.; Brady, J.M.: SUSAN - a new approach to low level image processing. Int. J. Comput. Vis., **23** (1) (1997), 45–78.

[22] Lindeberg, T.: Feature detection with automatic scale selection. Int. J. Comput. Vis., **30** (2) (1998), 79–116.

[23] Matas, J.; Chum, O.; Urban, M.; Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image Vis. Comput., **22** (10) (2004), 761 – 767. British Machine Vision Computing 2002.

[24] Mikolajczyk, K.; Schmid, C.: Scale & affine invariant interest point detectors. Int. J. Comput. Vis., **60** (1) (2004), 63–86.

[25] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis., **60** (2) (2004), 91–110.

[26] Rosten, E.; Drummond, T.: Machine learning for high-speed corner detection, in *Eur. Conf. on Computer Vision*, vol. **1**, May 2006, 430–443.

[27] Bay, H.; Tuytelaars, T.; Van Gool, L.J.: Surf: speeded up robust features, in *Eur. Conf. on Computer Vision*, 2006, 404–417.

[28] Trujillo, L.; Olague, G.: Synthesis of interest point detectors through genetic programming, in *Proc. 8th Annual Conf. on Genetic and*

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

Evolutionary Computation, GECCO '06, New York, NY, USA, 2006, 887–894. ACM.

[29] Flint, A.; Dick, A.; Hengel, A.V.D.: Thrift: local 3d structure recognition, in *9th Biennial Conf. of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications*, December 2007, 182–188.

[30] Agrawal, M.; Konolige, K.; Blas, M.: Censure: center surround extremas for realtime feature detection and matching, in *Computer Vision ECCV 2008* (D. Forsyth, P. Torr, A. Zisserman, eds), vol. **5305** of *Lecture Notes in Computer Science*, 102–115. *Springer*, Berlin, Heidelberg, 2008.

[31] Mair, E.; Hager, G.D.; Burschka, D.; Suppa, M.; Hirzinger, G.: Adaptive and generic corner detection based on the accelerated segment test, in *Eur. Conf. on Computer Vision (ECCV'10)*, September 2010, 183–196.

[32] Leutenegger, S.; Chli, M.; Siegwart, R.: Brisk: binary robust invariant scalable keypoints, in *Int. Conf. on Computer Vision*, 2011, 2548–2555.

[33] Alcantarilla, P.F.; Bartoli, A.; Davison, A.J.: Kaze features, in *Proc. 12th Eur. Conf. on Computer Vision – Volume Part VI*, ECCV'12. *Springer–Verlag*, Berlin, Heidelberg, 2012, 214–227.

[34] Verdie, Y.; Yi, K.M.; Fua, P.; Lepetit, V.: TILDE: a temporally invariant learned detector. *CoRR*, Abs./1411.4568, 2014.

[35] Filipe, S.; Itti, L.; Alexandre, L.A.: Bik-bus: biologically motivated 3d keypoint based on bottom-up saliency. IEEE Trans. Image Process., **24** (1) (2015), 163–175.

[36] Schmid, C.; Mohr, R.: Local grayvalue invariants for image retrieval. IEEE Trans. Pattern Anal. Mach. Intell., **19** (5) (1997), 530–534.

[37] Belongie, S.; Malik, J.; Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell., **24** (4) (2002), 509–522.

[38] Mikolajczyk, K.; Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell., **27** (10) (2005), 1615–1630.

[39] Dalal, N.; Triggs, B.: Histograms of oriented gradients for human detection, in *Int. Conf. on Computer Vision & Pattern Recognition* (C. Schmid, S. Soatto, C. Tomasi, eds), vol. **2**, 886–893. INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot, 2005.

[40] Tola, E.; Lepetit, V.; Fua, P.: DAISY: an efficient dense descriptor applied to wide baseline stereo. IEEE Trans. on Pattern Anal. Mach. Intell., **32** (5) (2010), 815–830.

[41] Trzcinski, T.; Lepetit, V.: Efficient discriminative projections for compact binary descriptors, in *Proc. 12th European Conf. on Computer Vision – Volume Part I*, ECCV'12. Springer-Verlag, Berlin, Heidelberg, 2012, 228–242.

[42] Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G.: Orb: an efficient alternative to sift or surf, in *2011 IEEE Int. Conf. on Computer Vision (ICCV)*, November 2011, 2564–2571.

[43] Alahi, A.; Ortiz, R.; Vandergheynst, P.: Freak: Fast retina keypoint, in *Computer Vision and Pattern Recognition (CVPR)*, 2012, 510–517.

[44] Baroffio, L.; Cesana, M.; Redondi, A.; Tagliasacchi, M.: BAMBOO: a fast descriptor based on asymmetric pairwise boosting, in *2014 IEEE Int. Conf. on Image Processing (ICIP)*, October 2014, 5686–5690.

[45] Trzcinski, T.; Christoudias, M.; Lepetit, V.; Fua, P.: Boosting binary keypoint descriptors, in *Computer Vision and Pattern Recognition*, 2013, 2874–2881.

[46] Ke, Y.; Sukthankar, R.: Pca-sift: a more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proc. 2004 IEEE Computer Society Conf.*, vol. **2**, June 2004, II–506–II–513.

[47] Shakhnarovich, G.: Learning Task-specific Similarity. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2005.

[48] Yeo, C.; Ahammad, P.; Ramchandran, K.: Rate-efficient visual correspondences using random projections, in *15th IEEE Intl Conf. on Image Processing, 2008. ICIP 2008.*, October 2008, 217–220.

[49] Weiss, Y.; Torralba, A.; Fergus, R.: Spectral hashing, in *Proc. 22 Annual Conf. on Advances in Neural Information Processing Systems 21*, Vancouver, British Columbia, Canada, 8–11 December 2008, 1753–1760.

[50] Wang, J.; Kumar, S.; Chang, S.: Semi-supervised hashing for large-scale search. IEEE Trans. Pattern Anal. Mach. Intell., **34** (12) (2012), 2393–2406.

[51] Chandrasekhar, V.; Takacs, G.; Chen, D.; Tsai, S.; Grzeszczuk, R.; Girod, B.: Chog: compressed histogram of gradients a low bit-rate feature descriptor, in *IEEE Conf. on Computer Vision and Pattern Recognition, 2009 (CVPR 2009)*, June 2009, 2504–2511.

[52] Chandrasekhar, V.; Takacs, G.; Chen, D.; Tsai, S.S.; Singh, J.; Girod, B.: Transform coding of image feature descriptors, 2009.

[53] Redondi, A.; Cesana, M.; Tagliasacchi, M.: Low bitrate coding schemes for local image descriptors, in *Int. Workshop on Multimedia Signal Processing*, September 2012, 124–129.

[54] Chen, J.; Duan, L.-Y.; Ji, R.; Wang, Z.: Multi-stage vector quantization towards low bit rate visual search, in *2012 19th IEEE Int. Conf. on Image Processing (ICIP)*, September 2012, 2445–2448.

[55] Jegou, H.; Douze, M.; Schmid, C.: Product quantization for nearest neighbor search. IEEE Trans. Pattern Anal. Mach. Intell., **33** (1) (2011), 117–128.

[56] Strecha, C.; Bronstein, A.M.; Bronstein, M.M.; Fua, P.: Ldahash: Improved matching with smaller descriptors. IEEE Trans. Pattern Anal. Mach. Intell., **34** (1) (2012), 66–78.

[57] Redondi, A.; Cesana, M.; Tagliasacchi, M.: Rate-accuracy optimization in visual wireless sensor networks, in *Int. Conf. on Image Processing*, October 2012, 1105–1108.

[58] Ascenso, J.; Pereira, F.: Lossless compression of binary image descriptors for visual sensor networks, in *2013 18th Int. Conf. on Digital Signal Processing (DSP)*, July 2013, 1–8.

[59] Monteiro, P.; Ascenso, J.: Clustering based binary descriptor coding for efficient transmission in visual sensor networks, in *Picture Coding Symp. (PCS), 2013*, December 2013, 25–28.

[60] Lazebnik, S.; Schmid, C.; Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in *2006 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, vol. **2**, 2006, 2169–2178.

[61] Nister, D.; Stewenius, H.: Scalable recognition with a vocabulary tree, in *2006 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, vol. **2**, 2006, 2161–2168.

[62] Gemert, J.C.; Geusebroek, J.-M.; Veenman, C.J.; Smeulders, A.W.: Kernel codebooks for scene categorization, in *Proc. 10th European Conf. on Computer Vision: Part III, (ECCV '08)*, *Springer–Verlag*, Berlin, Heidelberg, 2008, 696–709.

[63] Yang, J.; Yu, K.; Gong, Y.; Huang, T.: Linear spatial pyramid matching using sparse coding for image classification, in *IEEE Conf. on Computer Vision and Pattern Recognition, 2009 (CVPR 2009)*, June 2009, 1794–1801.

[64] Wang, J.; Yang, J.; Yu, K.; Lv, F.; Huang, T.; Gong, Y.: Locality-constrained linear coding for image classification, in *2010 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2010, 3360–3367.

[65] Jegou, H.; Douze, M.; Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search, in *Proc. 10th*

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

European Conf. on Computer Vision: Part I (ECCV '08), *Springer–Verlag*, Berlin, Heidelberg, 2008, 304–317.

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

[67] Perronnin, F.; Sánchez, J.; Mensink, T.: Improving the fisher kernel for large-scale image classification, in *Proc. 11th Eur. Conf. on Computer Vision: Part IV (ECCV'10)*, *Springer–Verlag*, Berlin, Heidelberg, 2010, 143–156.

[68] Zhou, X.; Yu, K.; Zhang, T.; Huang, T.S.: Image classification using super-vector coding of local image descriptors, in *Proc. 11th European Conf. on Computer Vision: Part V (ECCV'10)*, *Springer–Verlag*, Berlin, Heidelberg, 2010, 141–154.

[69] Paratte, J.: Sparse Binary Features for Image Classification. Master's thesis, Ecole Polytechnique federale de Lausanne (EPFL), Lausanne, Switzerland, 2013.

[70] Van Opdenbosch, D.; Schroth, G.; Huitl, R.; Hilsenbeck, S.; Garcea, A.; Steinbach, E.: Camera-based indoor positioning using scalable streaming of compressed binary image signatures, in *2014 IEEE Int. Conf. on Image Processing (ICIP)*, October 2014, 2804–2808.

[71] Tsai, S.S. et al.: Improved coding for image feature location information, 2012.

[72] Tsai, S.S.; Chen, D.; Takacs, G.; Chandrasekhar, V.; Singh, J.P.; Girod, B.: Location coding for mobile image retrieval, in *Proc. 5th Int. ICST Mobile Multimedia Communications Conf. (Mobimedia '09)*, ICST, Brussels, Belgium, Belgium, 2009, 8:1–8:7. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[73] Chao, J.; Steinbach, E.: Preserving sift features in JPEG-encoded images, in *2011 18th IEEE Intl. Conf. on Image Processing (ICIP)*, September 2011, 301–304.

[74] Chao, J.; Steinbach, E.: Sift feature-preserving bit allocation for h.264/avc video compression, in *2012 19th IEEE Int. Conf. on Image Processing (ICIP)*, September 2012, 709–712.

[75] Chen, S.D.; Moulin, P.: A two-part predictive coder for multitask signal compression, in *2014 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, 2035–2039.

[76] Makar, M.; Chandrasekhar, V.; Tsai, S.S.; Chen, D.; Girod, B.: Interframe coding of feature descriptors for mobile augmented reality. *IEEE Trans. Image Process.*, **23** (8) (2014), 3352–3367.

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

[78] Naikal, N.; Yang, A.Y.; Sastry, S.S.: Towards an efficient distributed object recognition system in wireless smart camera networks, in *2010 13th IEEE Conf. on Information Fusion (FUSION)*, 2010, 1–8.

[79] Yang, A.Y.; Maji, S.; Christoudias, C.M.; Darrell, T.; Malik, J.; Sastry, S.S.: Multiple-view object recognition in band-limited distributed camera networks, in *Third ACM/IEEE Int. Conf. on Distributed Smart Cameras, 2009. ICDSC 2009.*, August 2009, 1–8.

[80] Eriksson, E.; Dan, G.; Fodor, V.: Prediction-based load control and balancing for feature extraction in visual sensor networks, in *2014 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, 674–678.

[81] Redondi, A.; Baroffio, L.; Cesana, M.; Tagliasacchi, M.: A mathematical programming approach to task offloading in visual sensor networks, in *IEEE Vehicular Technology Conf. (VTC2015)*, May 2015. http://home.deib.polimi.it/redondi/greeneyes/publications/2015VTCRedondi.pdf.

[82] Redondi, A.; Cesana, M.; Tagliasacchi, M.; Filippini, I.; Dán, G.; Fodor, V.: Cooperative image analysis in visual sensor networks. *Ad Hoc Networks*, 2015, 1–5.

[83] Girod, B. et al.: Mobile visual search. *IEEE Signal Process. Mag.*, **28** (4) (2011), 61–76.

[84] Kim, J.; Grauman, K.: Boundary preserving dense local regions, in *2011 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2011, 1553–1560.

[85] Moreels, P.; Perona, P.: Evaluation of features detectors and descriptors based on 3d objects. Int. J. Comput. Vis., **73** (3) (2006), 263–284.

[86] Zhang, Z.; Deriche, R.; Faugeras, O.; Luong, Q.: A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Artif. Intell., **78** (1–2) (1995), 87–119.

[87] Fan, B.; Wu, F.; Hu, Z.: Aggregating gradient distributions into intensity orders: a novel local image descriptor, in *2011 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2011, 2377–2384.

[88] Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P.: Brief: binary robust independent elementary features, in *European Conf. on Computer Vision*, 2010, 778–792.

[89] Takacs, G.; Chandrasekhar, V.; Tsai, S.S.; Chen, D.; Grzeszczuk, R.; Girod, B.: Fast computation of rotation-invariant image features by an approximate radial gradient transform. IEEE Trans. Image Process., **22** (8) (2013), 2970–2982.

[90] Dong, J.; Soatto, S.: Domain-size pooling in local descriptors: DSP-SIFT, in *IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 2015*, Boston, MA, USA, 7–12 June 2015, 5097–5106.

[91] Morel, J.-M.; Yu, G.: Asift: A new framework for fully affine invariant image comparison. SIAM J. Image. Sci., **2** (2) (2009), 438–469.

[92] Wang, Z.; Fan, B.; Wu, F.: *Computer Vision – ECCV 2014: Proc. 13th European Conf.*, Zurich, Switzerland, 6–12 September, 2014, Part VII, Chapter – Affine Subspace Representation for Feature Description. *Springer International Publishing*, Cham, 2014, 94–108.

[93] Wang, Z.; Fan, B.; Wu, F.: Local intensity order pattern for feature description, in *2011 Int. Conf. on Computer Vision*, November 2011, 603–610.

[94] Ozuysal, M.; Calonder, M.; Lepetit, V.; Fua, P.: Fast keypoint recognition using random ferns. IEEE Trans. Pattern Anal. Mach. Intell., **32** (3) (2010), 448–461.

[95] Ojala, T.; Pietikainen, M.; Harwood, D.: Performance evaluation of texture measures with classification based on kullback discrimination of distributions, in *Proc. 12th IAPR Int. Conf. on Pattern Recognition, 1994. Vol. 1 – Conf. A: Computer Vision amp; Image Proc.*, vol. **1**, October 1994, 582–585.

[96] Byrne, J.; Shi, J.: Nested shape descriptors, in *2013 IEEE Int. Conf. on Computer Vision*, December 2013, 1201–1208.

[97] Winder, S.; Hua, G.; Brown, M.: Picking the best daisy, in *IEEE Conf. on Computer Vision and Pattern Recognition, 2009 (CVPR 2009)*, June 2009, 178–185.

[98] Balntas, V.; Tang, L.; Mikolajczyk, K.: Bold - binary online learned descriptor for efficient image matching, in *2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2015, 2367–2375.

[99] Osendorfer, C.; Bayer, J.; Urban, S.; van der Smagt, P.: Convolutional neural networks learn compact local image descriptors, in *Neural Information Processing*, 2013, 624–630. *Springer*.

[100] Zagoruyko, S.; Komodakis, N.: Learning to compare image patches via convolutional neural networks, in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2015)*, Boston, MA, USA, 7–12 June 2015, 2015, 4353–4361.

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

[101] Simo-Serra, E.; Trulls, E.; Ferraz, L.; Kokkinos, I.; Fua, P.; Moreno-Noguer, F.: Discriminative learning of deep convolutional feature point descriptors, in *Int. Conf. on Computer Vision*, 2015, 118–126.

[102] Boscaini, D.; Masci, J.; Melzi, S.; Bronstein, M.M.; Castellani, U.; Vandergheynst, P.: Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks, in *Computer Graphics Forum*, vol. **34**, 2015, 13–23. Wiley Online Library.

[103] Karpushin, M.; Valenzise, G.; Dufaux, F.: Improving distinctiveness of brisk features using depth maps, in *2015 IEEE Int. Conf. on Image Processing (ICIP)*, September 2015, 2399–2403.

[104] Fischer, P.; Dosovitskiy, A.; Brox, T.: Descriptor matching with convolutional neural networks: a comparison to sift. *arXiv:1405.5769*, 2014.

[105] Han, X.; Leung, T.; Jia, Y.; Sukthankar, R.; Berg, A.C.: Matchnet: unifying feature and metric learning for patch-based matching, in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2015)*, Boston, MA, USA, 7–12 June 2015, 3279–3286.

[106] Gauglitz, S.; Höllerer, T.; Turk, M.: Evaluation of interest point detectors and feature descriptors for visual tracking. Int. J. Comput. Vis., **94** (3) (2011), 335–360.

[107] Kulis, B.; Grauman, K.: Kernelized locality-sensitive hashing. IEEE Trans. Pattern Anal. Mach. Intell., **34** (6) (2012), 1092–1104.

[108] Weiss, Y.; Fergus, R.; Torralba, A.: Multidimensional spectral hashing, in *Proc. 12th Eur. Conf. on Computer Vision – ECCV 2012*, Florence, Italy, 7–13 October 2012, Part V, 2012, 340–353.

[109] Chandrasekhar, V.; Takacs, G.; Chen, D.M.; Tsai, S.S.; Makar, M.; Girod, B.: Feature matching performance of compact descriptors for visual search. In *Data Compression Conf. (DCC 2014)*, March 2014, 3–12.

[110] Redondi, A.; Baroffio, L.; Ascenso, J.; Cesana, M.; Tagliasacchi, M.: Rate-accuracy optimization of binary descriptors, in *20th IEEE Int. Conf. on Image Processing*, Melbourne, Australia, September 2013, 2910–2914.

[111] Monteiro, P.; Ascenso, J.: Coding mode decision algorithm for binary descriptor coding, in *Proc. 22nd European Signal Processing Conf. (EUSIPCO), 2014*, September 2014, 541–545.

[112] Yang, J.; Jiang, Y.-G.; Hauptmann, A.G.; Ngo, C.: Evaluating bag-of-visual-words representations in scene classification, in *Proc. Int. Workshop on Workshop on Multimedia Information Retrieval (MIR '07)*, New York, NY, USA, 2007, 197–206. ACM.

[113] Bellman, R.; Bellman, R.E.: Dynamic Programming, series P (Rand Corporation), *Princeton University Press*, Princeton, NJ, 1957.

[114] Philbin, J.; Chum, O.; Isard, M.; Sivic, J.; Zisserman, A.: Lost in quantization: improving particular object retrieval in large scale image databases, in *IEEE Conf. on Computer Vision and Pattern Recognition, 2008 (CVPR 2008)*, June 2008, 1–8.

[115] Perronnin, F.; Dance, C.: Fisher kernels on visual vocabularies for image categorization, in *IEEE Conf. on Computer Vision and Pattern Recognition, 2007 (CVPR '07)*, June 2007, 1–8.

[116] Lefebvre, F.; Czyz, J.; Macq, B.: A robust soft hash algorithm for digital image signature, in *Proc. 2003 Int. Conf. on Image Processing, 2003. ICIP 2003*, vol. **2** and **3**, September 2003, II–495–8.

[117] Oliva, A.; Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. Int. J. Comput. Vis., **42** (3) (2001), 145–175.

[118] Chatfield, K.; Simonyan, K.; Vedaldi, A.; Zisserman, A.: Return of the devil in the details: delving deep into convolutional nets. In *British Machine Vision Conf.*, 2014.

[119] Galvez-Lopez, D.; Tardos, J.D.: Real-time loop detection with bags of binary words, in *2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, September 2011, 51–58.

[120] Triggs, B.: Detecting keypoints with stable position, orientation and scale under illumination changes, in *8th Eur. Conf. on Computer Vision (ECCV '04)* (T. Pajdla, J. Matas, eds), vol. **3024** of *Lecture Notes in Computer Science (LNCS)*, Prague, Austria. *Springer-Verlag*, 2004, 100–113. This research was supported by the European Union FET-Open research project VIBES.

[121] Kläser, A.; Marszałek, M.; Schmid, C.: A spatio-temporal descriptor based on 3d-gradients, in *British Machine Vision Conf.*, September 2008, 995–1004.

[122] Baroffio, L.; Canclini, A.; Cesana, M.; Redondi, A.; Tagliasacchi, M.: Briskola: brisk optimized for low-power arm architectures, in *2014 IEEE Int. Conf. on Image Processing (ICIP)*, October 2014, 5691–5695.

[123] Baroffio, L.; Cesana, M.; Redondi, A.; Tagliasacchi, M.; Tubaro, S.: Fast keypoint detection in video sequences, in *2016 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, March 2106, 1342–1346.

[124] Wang, H.; Kläser, A.; Schmid, C.; Liu, C.-L.: Dense trajectories and motion boundary descriptors for action recognition. Int. J. Comput. Vis., **103** (1) (2013), 60–79.

[125] Wang, H.; Schmid, C.: Action recognition with improved trajectories, in *IEEE Int. Conf. on Computer Vision*, Sydney, Australia, 2013, 3551–3558.

[126] Paniga, S.; Borsani, L.; Redondi, A.; Tagliasacchi, M.; Cesana, M.: Experimental evaluation of a video streaming system for wireless multimedia sensor networks, in *2011 10th IFIP Annu. Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, June 2011, 165–170.

[127] Yu, W.; Sahinoglu, Z.; Vetro, A.: Energy efficient JPEG 2000 image transmission over wireless sensor networks, in *IEEE Global Telecommunications Conf., 2004 (GLOBECOM '04.)*, vol. **5**, November 2004, 2738–2743.

[128] Dan, G.; Khan, M.; Fodor, V.: Characterization of surf and brisk interest point distribution for distributed feature extraction in visual sensor networks. IEEE Trans. Multimed., **17** (5) (2015), 591–602.

[129] Eriksson, E.; Dan, G.; Fodor, V.: Real-time distributed visual feature extraction from video in sensor networks, in *2014 IEEE Int. Conf. on Distributed Computing in Sensor Systems (DCOSS)*, May 2014, 152–161.

[130] MPEG. Compact descriptors for visual search. http://mpeg.chiariglione.org/standards/mpeg-7/compact-descriptors-visual-search.

Luca Baroffio received the M.Sc. degree (2012, cum laude) in Computer Engineering and the Ph.D. degree (2016) in Information Technology both from Politecnico di Milano, Milan, Italy. In 2013, he was visiting scholar at "Instituto de Telecomunicações, Lisbon", Portugal. His research interests are in the areas of multimedia signal processing and visual sensor networks.

Alessandro E. C. Redondi received the M.S. in Computer Engineering in July 2009 and the Ph.D. in Information Engineering in 2014, both from Politecnico di Milano. From September 2012 to April 2013 was a visiting student at the EEE Department of the University College of London (UCL). Currently, he is an Assistant Professor at the "Dipartimento di Elettronica, Informazione e Bioingegneria – Politecnico di Milano" and his research activities are focused on algorithms and protocols for Visual Sensor Networks and on the analysis of computer networks data.

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press

Marco Tagliasacchi is currently an Assistant Professor at the "Dipartimento di Elettronica e Informazione Politecnico di Milano", Italy. He received the "Laurea" degree (2002, cum Laude) in Computer Engineering and the Ph.D. in Electrical Engineering and Computer Science (2006), both from Politecnico di Milano. He was visiting academic at the Imperial College London (2012) and visiting scholar at the University of California, Berkeley (2004). His research interests include multimedia forensics, multimedia communications (visual sensor networks, coding, quality assessment), and information retrieval. Dr. Tagliasacchi co-authored more than 120 papers in international journals and conferences, including award winning papers at MMSP 2013, MMSP2012, ICIP 2011, MMSP 2009, and QoMex 2009. He has been actively involved in several EU-funded research projects.

Stefano Tubaro completed his studies in Electronic Engineering at the Politecnico di Milano, Italy, in 1982. Since December 2004 he has been appointed as a Full Professor of Telecommunication at the "Dipartimento di Elettronica, Informazione e Bioingegneria of the Politecnico di Milano (DEIB-PoliMi)". His current research interests are on advanced algorithms for video and sound processing. Stefano Tubaro authored over 150 publications on international journals and congresses. In the past few years, he has focused his interest on the development of innovative techniques for image and video tampering detection and, in general, for the blind recovery of the "processing history" of multimedia objects. Stefano Tubaro is the Head of the Telecommunication Section of DEIB-PoliMi, and the Chair of the IEEE SPS Italian Chapter; moreover he coordinates the research activities of the Image and Sound Processing Group (ISPG). He is a member of the IEEE MMSP and IVMSP TCs.

https://doi.org/10.1017/ATSIP.2016.13 Published online by Cambridge University Press