From handcrafted to deep local features

Gabriela Csurka, Christopher R. Dance and Martin Humenberger
NAVER LABS Europe, 6 chemin de Maupertuis, 38240 Meylan, France
firstname.lastname@naverlabs.com
www.europe.naverlabs.com

Abstract This paper presents an overview of the evolution of local features from handcrafted to deep-learning-based methods, followed by a discussion of several benchmarks and papers evaluating such local features. Our investigations are motivated by 3D reconstruction problems, where the precise location of the features is important. As we describe these methods, we highlight and explain the challenges of feature extraction and potential ways to overcome them. We first present handcrafted methods, followed by methods based on classical machine learning and finally we discuss methods based on deep-learning. This largely chronologically-ordered presentation will help the reader to fully understand the topic of image and region description in order to make best use of it in modern computer vision applications. In particular, understanding handcrafted methods and their motivation can help to understand modern approaches and how machine learning is used to improve the results. We also provide references to most of the relevant literature and code.

1 Introduction

In the computer vision literature we encounter local features in two main contexts. In the first context, a local feature is the description of a region of an image in terms of characteristics extracted from that region. Such local features are usually called *local descriptors* and are usually represented by vectors of real numbers or binary digits. For example, local descriptors could be obtained by concatenating the pixel intensities or computing a colour histogram of the region. In the second context, a local feature is a distinctive point or area of an image, and such local features are often called *keypoints*, *interest points* or *anchor points*. It is often important both to find the precise location of such a local feature in the image, which is called *detection*, and to extract a corresponding local descriptor. Keypoint detection also usually involves the determination of a set of transformations that may be applied to the region in order to make its descriptor invariant to some geometric or photometric transformations. This paper discusses both detection and description. For clarification, in the remainder of this paper a *local feature* consists of a *keypoint/interest point* and its *descriptor*.

Local features are important for object recognition by the human visual system [Biederman, 1987]. They have also been successful in many computer vision applications including face and object detection and

Updated version of our preprint

recognition, image retrieval, motion detection and tracking, depth-map generation, image stitching, camera calibration, 3D reconstruction, structure from motion (SfM), visual odometry, and visual simultaneous localisation and mapping (VSLAM).

We distinguish three broad categories of local features depending on the applications for which those features were designed [Tuytelaars and Mikolajczyk, 2007]. First, local features may be designed to have a specific semantic interpretation. For instance edges in aerial images often correspond to roads, while blobs often represent impurities in inspection tasks. Second, local features may be designed to be localised accurately and consistently over time. Such local features are important in matching (*e.g.* camera calibration and 3D reconstruction) and tracking (*e.g.* visual odometry) applications. Finally, the set of local features extracted from an image can be used as a robust representation for the recognition and retrieval of objects and scenes. Such local features do not need to have specific semantics or to be accurately localised, as the corresponding applications mainly exploit the statistics of the local-feature set.

In this paper we are mainly interested in the second category of local features, where the aim is to accurately localise and match keypoints that correspond to the same 3D point viewed in different images. Ideally, such local features should be robust to variations in *viewpoint* (geometric deformations) and *lighting* (photometric changes). They should also be robust to image noise, discretisation effects, compression artifacts and blur. Furthermore, the descriptors should be *distinctive*. That is, they should have a rich enough information content that local features corresponding to different 3D points are readily distinguished, while local features corresponding to the same 3D point are readily matched, even in the presence of a large number of such features. On the other hand, many applications require descriptors that are *compact*. That is, they should be of low dimension so that they require little memory to store and may be efficiently matched.

Since the geometric relationship between two images of the same scene is usually a projective transformation, one might argue that local features should be invariant to projective transformations, or that they should be invariant to affine transformations, which approximate small perspective transformations well. One might achieve such invariance in the following way, as illustrated in Figure 1. First one defines a canonical view of the patch and estimates the geometric transformation between the image and the canonical view. Then one transforms the local region into the canonical view and computes the descriptor on this normalised patch. Nevertheless, most local features are only rotation and scale invariant, and they compute normalised patches by first estimating each keypoint's dominant orientation and then its characteristic scale.

To address these requirements, a large set of different keypoint detectors and descriptors have been proposed in the last three decades. In addition to our description, the reader should refer to the comprehensive surveys of Tuytelaars and Mikolajczyk [2007], Krig [2014] and Fan et al. [2014b], as well as the benchmark papers of Mikolajczyk et al. [2005], Mikolajczyk and Schmid [2003], Heinly et al. [2012], Balntas et al. [2017] and Schönberger et al. [2017]. Many of these feature detectors and descriptors are integrated in OpenCV[1] and VLFeat[2].

Aggregates of local features such as the bag of visual words [Sivic et al., 2003, Csurka et al., 2004], and its extensions such as Fisher Vectors [Perronnin and Dance, 2007] and VLAD [Jégou et al., 2010] were

[1] See the section entitled "Feature Detection and Description" in the OpenCV Tutorial at `https://docs.opencv.org/3.1.0/d6/d00/tutorial_py_root.html`.

[2] The Matlab VLFeat library is available at `http://www.vlfeat.org/`.

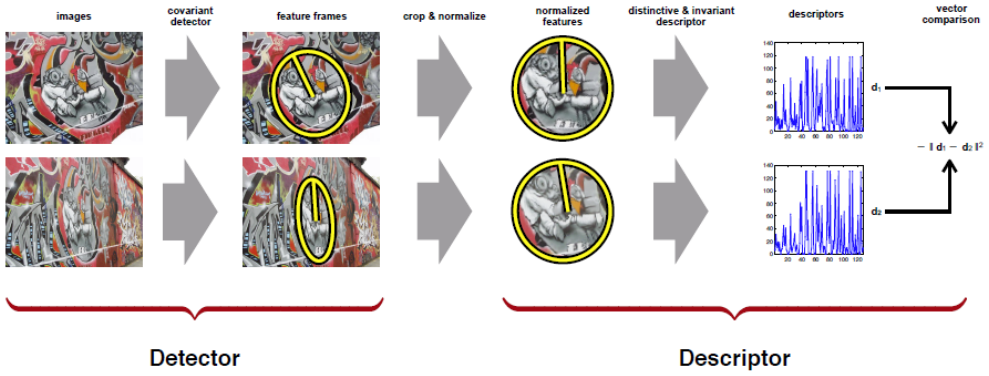

Fig. 1 Local feature detection and description pipeline. A detector is *covariant* to a particular class of geometric transforms if it extracts transformed versions of the same regions when an image is subject to a transform from that class. For instance, regions detected by a scale and rotation covariant detector should scale by $2\times$ and rotate by 30° if the image is scaled by $2\times$ and rotated by 30°. (By courtesy of Andrea Vedaldi.)

the most widely used representations for image classification and retrieval for a timespan of more than a decade. Therefore a tremendous amount of work on learning, combining, and improving local features for object recognition, image classification, and retrieval was published. However, detailed presentation of that literature is out of the scope of this paper which instead focuses on local features that may be accurately localised and consistently matched.

This paper is structured as follows. We present work on handcrafted local features (Section 2) before discussing methods based on classical machine learning (Section 3). Then we present advances in detection and description using deep learning (Section 4). Finally, we summarise the main findings of several benchmark papers (Section 5) and conclude (Section 6).

2 Handcrafted local features

Before the deep-learning revolution, handcrafted local features were a key element of almost all computer vision applications. The requirements for these applications were diverse. Therefore, many different handcrafted detectors and descriptors have been proposed over the last three decades. While several of the papers discussed in this section introduce both a detector and a descriptor, it is often possible to pair the detector from one local feature with the descriptor from another. For this reason, this section first discusses keypoint detectors (Section 2.1), then local feature descriptors given by real vectors that we refer to as *real descriptors* (Section 2.2) and finally descriptors given by binary vectors that we refer to as *binary descriptors* (Section 2.3).

2.1 Keypoint detectors

Handcrafted keypoint detectors may be based on finding corners, analysing intensity derivatives, segmentation, mathematical morphology, saliency and normalised intensity edges, as we now describe.

The earliest detectors were based on finding corners and on analysing intensity derivatives. Corner detectors find maxima of curvature or abrupt changes in the direction of the tangent to an edge [Rosenfeld and Kak, 1982, Wang and Brady, 1995]. Meanwhile, intensity-derivative-based detectors try to find regions that satisfy certain uniqueness and stability criteria, and include the Hessian detector [Zuniga and Haralick, 1983] and Harris detector [Harris and Stephens, 1988] as notable examples. The Hessian detector, also referred to as the determinant of Hessian (DoH) method, is used in the popular speeded up robust features (SURF) algorithm [Bay et al., 2006], where for efficiency, the Hessian is roughly approximated with a set of box filters and no smoothing is applied when going from one scale to the next. Both Harris and Hessian methods were extended by Mikolajczyk and Schmid [2004] to handle affine invariance. Furthermore, one of the most popular keypoint detection methods, the scale-invariant feature transform (SIFT) detector [Lowe, 2004], can be seen as an intensity-derivative-based method. In particular, the SIFT detector uses the difference of Gaussians (DoG) to detect local extrema and the DoG can be viewed as an approximation of the Laplacian of a Gaussian.

Segmentation techniques have also been employed for detection. Such methods either work with junctions on the boundaries of homogeneous regions [Liu and Tsai, 1990] or they work with the homogeneous regions themselves [Corso and Hager, 2005]. Similarly, the maximally stable extremal regions (MSER) detector [Matas et al., 2002], which was developed for estimating disparities in wide-baseline stereo, uses watershed-like segmentation to extract homogeneous intensity regions which are stable over a wide range of thresholds.

Several detectors are based on ideas from mathematical morphology. For instance, SUSAN (univalue segment assimilating nucleus) [Smith and Brady, 1997] computes the fraction of pixels within a neighbourhood which have similar intensity to the center pixel. Corners can then be localised by applying a threshold to this measure and selecting local minima. FAST (features from accelerated segment test) [Rosten and Drummond, 2006] is an extension of SUSAN, whose keypoint detector is significantly faster. The method relies on a set of pixels in a circular pattern to determine a keypoint, and makes comparisons between the intensities of pixels on the circle and the intensity of the pixel at the center of the circle. If a number of consecutive pixels around the circle are consistently brighter than the centre or consistently darker than the centre, then the central pixel is considered to be a good candidate. The process concludes with non-maximum suppression. As originally proposed, FAST is not a scale-space detector, so Lepetit and Fua [2006] extended it to perform scale selection with the Laplacian function.

Salient-region-based detectors exploit the notion that keypoints should exhibit local attributes that are unpredictable compared to the surrounding region. For instance, Kadir et al. [2004] proposed to measure the change in entropy of a grey-value histogram computed in a set of neighbourhoods of variety of positions, scales and affine shapes. Wavelet transformations have also been considered for multi-resolution keypoint detection [Sebe et al., 2003].

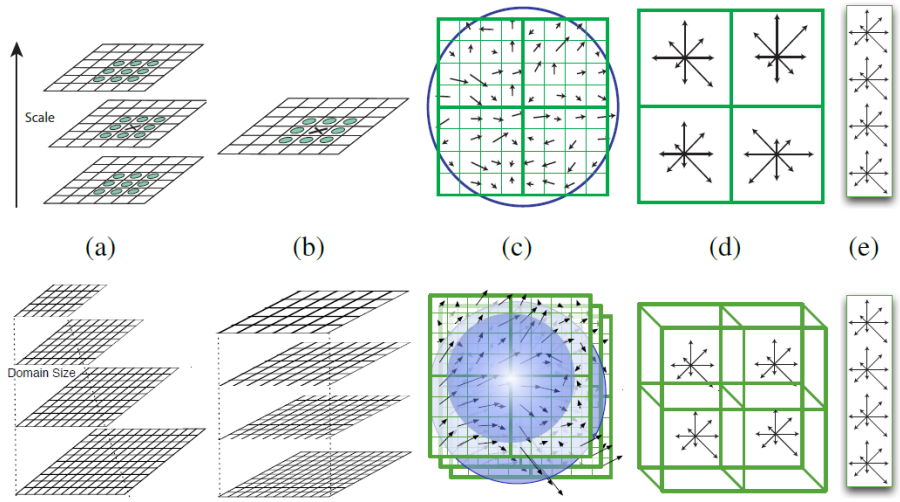

Fig. 2 Top Row: The SIFT [Lowe, 2004] detection and description process. (a) The difference of Gaussians is computed at multiple scales; (b) A scale is selected for each keypoint; (c) Gradient orientations are computed at that scale; (d) The orientations are spatially pooled; (e) This yields histograms that are concatenated and normalised to form the descriptor. Bottom Row: The DSP-SIFT [Dong and Soatto, 2015] detection and description process. (a,b) Patches of different sizes are re-scaled; (c,d) Gradient orientations are computed and pooled across locations and scales; (e) These are concatenated yielding a descriptor of the same dimension as the ordinary SIFT descriptor. (By courtesy of Jingming Dong.)

Rather than working directly with image intensities, EdgeFoci[3] [Zitnick and Ramnath, 2011] works with normalised intensity edges, as its authors hypothesize that such edges are more robust to nonlinear lighting variations and background clutter. In particular, they define an *edge focus* as a point that is equidistant from two edges whose edge orientations are perpendicular, and use such edge foci as keypoints.

2.2 Real descriptors

One of the most widely used local feature descriptors is SIFT [Lowe, 2004, Otero and Delbracio, 2014]. In the literature, SIFT often refers to the whole pipeline shown in Figure 2 (top row), including the detector as well as the descriptor which is based on local gradient histograms computed on a 4×4 grid. SIFT has been extended in several ways. These include the gradient location and orientation histogram (GLOH) [Mikolajczyk and Schmid, 2003], which considers a log-polar grid on which the gradients are averaged to compute the histograms, coloured SIFT (CSIFT) [Abdel-Hakim and Farag, 2006] which exploits colour-invariant characteristics, domain-size pooling SIFT (DSP-SIFT) [Dong and Soatto, 2015], which pools SIFT descrip-

[3] The EdgeFoci page can be found at `http://research.microsoft.com/en-us/um/people/larryz/edgefoci/edge_foci.htm`.

tors across scales, and scale-less SIFT (SLS) [Hassner et al., 2017], which is a subspace representation of SIFT across multiple scales.

Whereas SIFT uses a scale space obtained by smoothing and downsampling the input image, the idea underlying DSP-SIFT [Dong and Soatto, 2015] is to instead use a *size-space* which is obtained by maintaining the same scale as the input image, but considering subsets of it of variable size. Furthermore, while SIFT descriptors are constructed at a selected scale and gradient orientations are pooled in its spatial neighbourhood, DSP-SIFT considers patches of different sizes that are re-scaled and gradient orientations are pooled across locations *and* scales (see Figure 2, bottom row). Note that domain-size pooling (DSP) can also be applied to features other than SIFT.

SLS [Hassner et al., 2017] represents each pixel with a set of SIFT descriptors extracted at multiple scales. The authors show that SLS gives far better matches than descriptors computed at a single selected scale. As this improvement comes at a significant computational cost, the authors propose to represent each set of SIFT descriptors by a low-dimensional, linear subspace and a subspace-to-point mapping is used to get the final descriptors.

Several papers propose descriptors that are faster than SIFT yet still allow for reliable matching. Notably, SURF [Bay et al., 2006] computes descriptors using Haar filter responses obtained from integral images. Also, KAZE (which is the Japanese word for *wind*) [Alcantarilla et al., 2012] uses a similar pipeline to SURF, except that it works in a nonlinear scale space. This nonlinear scale space is built using efficient additive operator splitting techniques and variable conductance diffusion. Accelerated KAZE[4] [Alcantarilla and Nuevo, 2013] uses fast explicit diffusion embedded in a pyramidal framework to dramatically speed up detection.

2.3 Binary descriptors

With the spread of mobile and embedded vision systems, the demand for efficient detection and matching of image features grew. Also, in mobile applications, it is desirable to limit the amount of data sent over the network in order to keep latency and costs down. Moreover, in applications such as tracking and visual simultaneous localisation and mapping (VSLAM), keypoint detection and description often have to be done in real time. These facts have motivated many authors to propose binary descriptors, which require less storage than real descriptors and can be efficiently matched using Hamming distance.

One way to build binary descriptors is to binarise existing real descriptors through quantisation [Gong and Lazebnik, 2011] or hashing [Brown et al., 2005, Gionis et al., 1999, Shakhnarovich et al., 2003, Weiss et al., 2008, Kulis and Grauman, 2009, Strecha et al., 2012]. Alternatively, binary descriptors may be extracted directly from image patches. Such binary descriptors include LBP, the census transform, BRIEF, ORB, BRISK, FREAK and BIO, as we now discuss (see also Figure 3).

[4] (A)KAZE is available at `http://www.robesafe.com/personal/pablo.alcantarilla/kaze.html`.

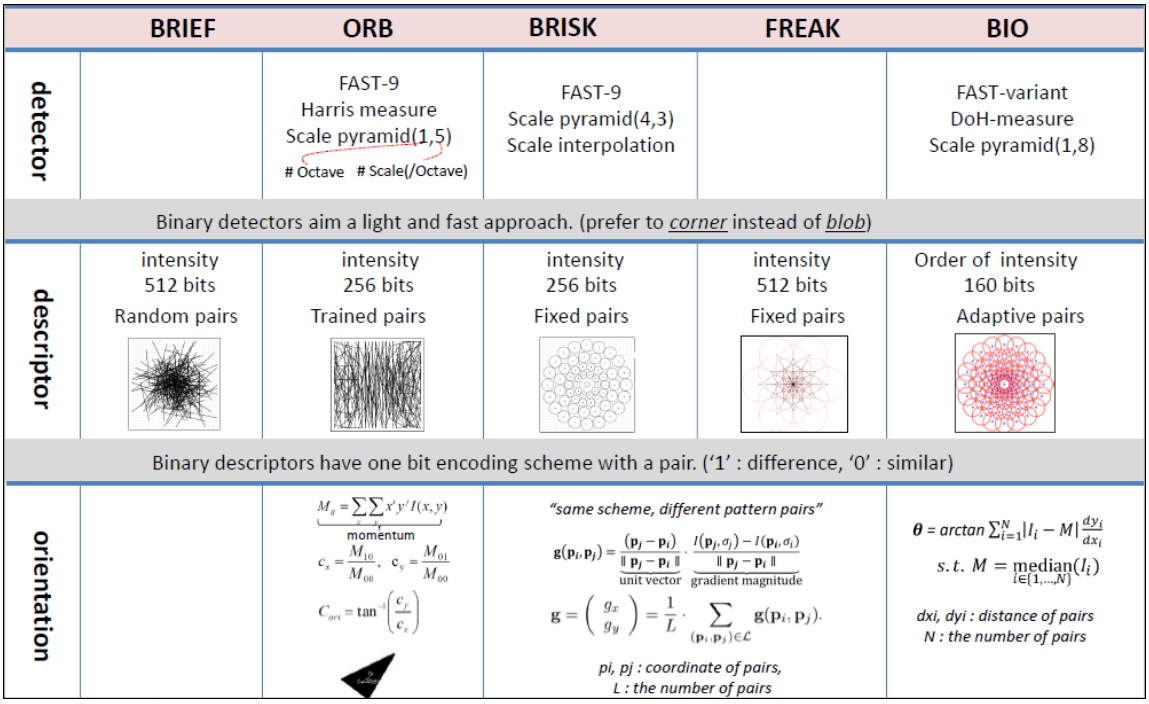

Fig. 3 An illustrative summary of some of the popular binary descriptors from Choi and Kweon [2015]. (By courtesy of Yukyung Choi.)

Local binary patterns (LBP) [Ojala et al., 1996, Pietikäinen et al., 2011] and the census transform [Zabih and Woodfill, 1994] both create a descriptor by comparing the intensity of a given pixel with each of its neighbours' intensities, and encoding 1 if the value is greater and 0 otherwise. Despite their simplicity, LBP and the census transform have proved quite powerful, so they have inspired a large set of variants.

The binary robust independent elementary feature (BRIEF) descriptor [Calonder et al., 2010] uses a randomly-selected distribution of point-pairs relative to a central point to create the descriptor. Oriented BRIEF (ORB) features [Rublee et al., 2011], add rotational invariance to BRIEF by determining corner orientation using FAST [Rosten and Drummond, 2006].

Binary robust invariant scalable keypoints (BRISK) [Leutenegger et al., 2011] is another popular descriptor that uses a sampling pattern consisting of disks of variable sizes arranged in four concentric rings. The pattern is pre-rotated to a characteristic direction to achieve rotation invariance. Finally, the oriented sampling pattern is used to obtain pairwise brightness comparisons which are assembled into the binary descriptor.

Table 1 Popular datasets for local feature detector and descriptor training and evaluation.

Dataset	Nature of ground truth	Reference
Oxford-Affine	homographies	[Mikolajczyk and Schmid, 2003]
Photo-Tourism	corresponding patches	[Winder, 2007]
Fountain and Herzjesu	visibility and optical flow	[Strecha et al., 2008]
EdgeFoci	image sequences	[Zitnick and Ramnath, 2011]
Cornell BigSfM	3D points, tracks, camera info	[Crandall et al., 2013]
Hannover	image sequences, homographies	[Cordes et al., 2013]
RomePatches	corresponding patches, image labels	[Li et al., 2014, Paulin et al., 2015]
1DSfM	3D points and camera info	[Wilson and Snavely, 2014]
DaLI	object deformations	[Simo-Serra et al., 2015a]
WebCam	image sequences	[Verdie et al., 2015]
HPatches	corresponding patches, homographies	[Balntas et al., 2017]
HSequences	image pairs, homographies	[Lenc and Vedaldi, 2018]

The fast retina keypoints (FREAK) descriptor[5] [Alahi et al., 2012] uses a multi-resolution pixel-pair sampling pattern with trained pixel-pairs. This design mimics the human eye in the sense that it has high resolution in the fovea and lower resolution in the periphery.

The robust binary feature using intensity order (BIO) descriptor [Choi et al., 2014] was inspired by the local intensity order pattern (LIOP) descriptor[6] [Wang et al., 2011, 2016] which encodes local intensity ordering information. The authors use a FAST-like binary comparison test and detect keypoints using a fast approximation of the determinant of Hessian (DoH). As ordinal descriptors are insensitive to moderate rank-order errors, they can be quantised into binary descriptors without noticeably degrading performance.

3 Local features based on classical machine learning

We now discuss local feature detectors and descriptors based on classical machine learning, as opposed to *deep* learning which is discussed in Section 4. As with machine learning methods in general [Mohri et al., 2012], such methods involve training on a given dataset in the expectation that this will lead to good performance on new data drawn from a similar distribution. The training procedure can lie anywhere on a spectrum from unsupervised to supervised. On the one hand, purely unsupervised methods need no labelled data. When such unsupervised methods are applied to learning local descriptors, the main idea is often to adapt handcrafted features to the given dataset, for instance by projecting them into a well-chosen low-dimensional space. On the other hand, supervised learning methods require labelled data. In this context, positive labels usually corresponds to "matched keypoints", which are pairs of patches representing different views of the same 3D point. Such matched keypoints can easily be generated synthetically, or extracted from sequences of images of a given scene by leveraging geometric consistency (*i.e.* a known homography or an essential matrix relating different views). Table 1 lists the most popular datasets used to train or evaluate

[5] Code available at `https://github.com/kikohs/freak`.

[6] The code for LIOP is available at `http://zhwang.me/publication/liop/index.html`.

local features, which include Oxford-Affine[7], Photo-Tourism[8], Fountain and Herzjesu[9], Cornell BigSfM[10], 1DSfM[11], RomePatches[12] and HPatches[13].

As in the previous section, we first discuss detectors (Section 3.1), before moving on to real and binary descriptors (Sections 3.2 and 3.3).

3.1 Learning detectors

Several papers have considered using classical machine learning to *speed up* detection while finding the same keypoints as handcrafted methods [Rosten and Drummond, 2006, Leutenegger et al., 2011, Rublee et al., 2011]. Also, Hartmann et al. [2014] learned a classifier that predicts which keypoints are likely to be discarded when matching descriptors among those extracted by a standard keypoint detection algorithm, namely DoG. By using a random forest to learn such "matchability", they showed that the approach can considerably improve and speed up the feature matching stage of a SfM pipeline. However this approach remains limited by the quality of the initial keypoint detector.

Other papers have focused on using learning to improve detector *repeatability*. Given a collection of keypoints detected by a standard detector, Strecha et al. [2009] demonstrate higher repeatability by using a WaldBoost classifier to keep only keypoints that are known to be useful for the task at hand. For instance, in the task of image matching in an urban environment, their classifier learns to focus on more-stable man-made structures and to ignore objects that undergo regular changes such as vegetation and clouds.

Meanwhile Verdie et al. [2015] proposed the temporally invariant learned detector (TILDE), which was designed for repeatable keypoint detection in the presence of drastic illumination changes caused by variations in weather, season and time-of-day, to which keypoint detectors tend to be highly sensitive. To achieve this goal, the authors worked with a collection of sequences of webcam images, where each sequence consists of images acquired at the same location but different times. Using this collection as training data, they learned a variety of regressors to predict how likely a SIFT keypoint detected in one image from a given sequence will also be detected at a nearby point in another image from that sequence. They compared regressors based on piecewise-linear functions, the LeNet-5 CNN [LeCun et al., 1998] and boosted regression trees. While SIFT keypoints were used during training, at runtime the regressor was passed over the entire image, giving a "score map". Then points with a score are selected and subjected to non-maximum suppression. The results showed that using piecewise-linear functions as a regressor gave consistently more reliable keypoints than alternative regressors and than known keypoint detectors such as SURF and MSER. As discussed in Section 5, TILDE remains a state-of-the-art approach to detection in the presence of illu-

[7] Oxford-Affine is available at `http://www.robots.ox.ac.uk/~vgg/data/data-aff.html`.

[8] Photo-Tourism dataset page `http://matthewalunbrown.com/patchdata/patchdata.html`.

[9] Fountain and Herzjesu can be downloaded from `https://cvlab.epfl.ch/data/keypoint`.

[10] Cornell BigSfM is available at `http://www.cs.cornell.edu/projects/bigsfm/`.

[11] 1DSfM dataset page `http://www.cs.cornell.edu/projects/1dsfm/`.

[12] RomePatches is available at `http://lear.inrialpes.fr/people/paulin/projects/RomePatches/`.

[13] HPatches data and benchmark on `https://github.com/hpatches`.

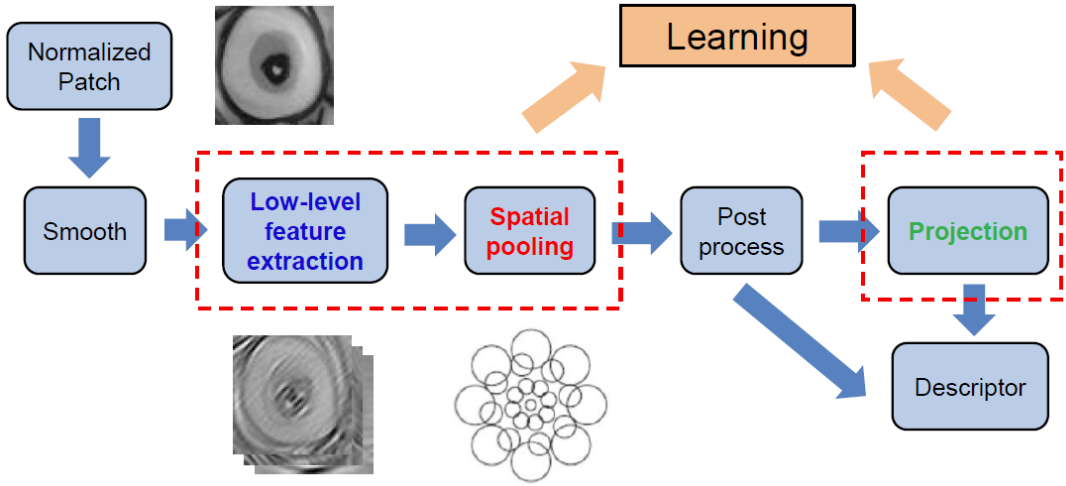

Fig. 4 Discriminative descriptor learning by optimising spatial pooling and feature embedding in Brown et al. [2010]. (By courtesy of Bin Fan.)

mination changes, however it is limited to situations where only keypoints with a common scale are matched.

3.2 Learning real descriptors

Early descriptor-learning methods included unsupervised PCA-SIFT [Ke and Sukthankar, 2004], which uses principal component analysis (PCA) to embed the gradient image of a patch into a new space, and supervised methods using randomised trees [Lepetit and Fua, 2006] and boosting [Babenko et al., 2007] to learn feature representations from matching and non-matching local patch pairs. More recently, Brown et al. [2010] proposed a model which builds on top of handcrafted low-level features, such as steerable filter banks or gradient orientation maps with different spatial pooling, as in the SIFT and DAISY descriptors. Considering both linear and nonlinear transforms for dimensionality reduction, they used linear discriminant analysis (LDA) to select the pooling parameters and to obtain low-dimensional representations (see Figure 4). Meanwhile, Philbin et al. [2010] learned both linear and nonlinear discriminative projections into lower dimensional spaces using a margin-based cost function, which aims to separate matching descriptors from non-matching descriptors. Training data was generated automatically by leveraging geometric consistency.

More recently, Simonyan et al. [2014] proposed convex large-margin formulations of two tasks in descriptor design: the selection of spatial pooling regions, for which the authors use L_1-regularisation; and descriptor dimensionality reduction, for which they use nuclear-norm regularisation. Their method, ConvOpt[14],

[14] Code available for ConvOpt on `http://www.robots.ox.ac.uk/~vgg/research/learn_desc/`.

applies a stochastic optimisation technique called regularized dual averaging [Xiao, 2010] that is well suited to non-smooth sparsity-inducing cost functions. The authors further proposed a weakly-supervised extension, where unlabelled data is exploited using an optimisation objective inspired by the large margin nearest neighbour (LMNN) approach [Weinberger and Saul, 2009].

Wang et al. [2014] aimed to make descriptors robust to affine distortions introduced by viewpoint changes. Their method performs PCA on a collection of affine-warped versions of an input patch. A projection matrix is computed from the first few principal components, in order to describe this collection of affine-warped patches as a linear subspace. To convert this matrix into a descriptor vector, they employ a simple *subspace-to-point mapping* which consists in stacking the upper-triangular elements of the projection matrix into a vector, after scaling the diagonal entries by $1/\sqrt{2}$.

Only a few papers have attempted to directly design descriptors to be robust to deformations that are more general than affine transformations. In the case of images of objects that have undergone nonrigid deformations, there have been two main approaches to finding correspondences given descriptors that are not robust by design. One approach is to enforce global consistency [Cheng et al., 2008, Sanchez-Riera et al., 2010, Torresani et al., 2008] and the other is to include segmentation information in the descriptor and then solve complex optimisation problems to establish matches [Trulls et al., 2013]. Arguably, a better option is to build deformation-invariant descriptors. As such, DaLI[15] (deformation and light invariant) [Simo-Serra et al., 2015a] uses methods from *diffusion geometry* [Gebal et al., 2009, Sun et al., 2009] to build a descriptor for 2D image patches which is invariant to nonrigid deformations and photometric changes. A patch is described in terms of the heat it dissipates onto its neighbourhood over time. To ensure compact descriptors, DaLI uses PCA for dimensionality reduction. Experimental results demonstrate that the DaLI descriptor can simultaneously handle quite complex photometric changes and spatial warps.

3.3 Learning binary descriptors

There is a broad literature on learning binary descriptors for specific applications, such as face recognition [Lei et al., 2014, Lu et al., 2015, 2018, Duan et al., 2017b]. However, this review focusses only on methods that may be relevant for SfM and 3D reconstruction, which include LDAHash, D-BRIEF, RI-LBD, BinBoost, BOLD and RMGD, as we now discuss.

LDAHash [Strecha et al., 2012] computes a projection matrix for a given source of handcrafted descriptors, such as SIFT or SURF, using linear discriminant analysis (LDA). LDA chooses this projection to minimize the ratio of intra-class variance to inter-class variance. Finally, the projections are optimally thresholded to give binary vectors. In D-BRIEF[16] [Trzcinski and Lepetit, 2012] the training data is used to learn linear projections that map image patches to a more discriminative subspace. In order to obtain binary descriptors, the projected patches are simply thresholded. More recently, Duan et al. [2017c] proposed the rotation-invariant local binary descriptors (RI-LBDs), in which each local patch is first categorized into a

[15] Code for DaLI is available at `http://www.iri.upc.edu/people/esimo/research/dali/`.

[16] The code for D-BRIEF is available at `https://cvlab.epfl.ch/research/detect/dbrief`.

"rotational binary pattern". The orientation for each such pattern and a projection matrix that maps each image patch into a binary code are learned jointly.

BBoost[17] features [Trzcinski et al., 2015] are low-dimensional but highly discriminative descriptors computed with a boosted binary hash function. They use weak learners which pool image gradients over particular regions, inspired by handcrafted descriptors like BRIEF [Calonder et al., 2010]. Similarly, the ring-based multi-grouped descriptor (RMGD) [Gao et al., 2015] uses pooling over a polar grid, and it performs boosting to select from the set of all binary comparisons between pairs of grid cells. A circular integral image is used for fast calculation of the binary descriptor. To increase discriminativeness and robustness, the RMGD is built with multiple image properties including intensity, x- and y-gradients, gradient magnitudes, orientations and soft-assigned gradient orientations. The receptive fields descriptor (RFD) [Fan et al., 2014a] uses gradient-orientation maps summed over two kinds of receptive fields, namely rectangular pooling regions or Gaussian pooling regions. Instead of selecting these regions by boosting, RFD selects them by a greedy approach, according to their distinctiveness and correlations.

Whereas most binary descriptors are constructing using the *same* set of measurements for every input patch, binary online learned descriptors (BOLD)[18] [Balntas et al., 2015, 2018] *adapt* the set of measurements depending on the input patch. This adaptation is motivated by the observation that some of the pairwise intensity comparisons made by conventional binary descriptors are unstable to small affine perturbations, for some patches. The adaptation is accomplished by synthesizing multiple small random perturbations of the given input patch. Inspired by LDA, the authors treat these perturbed versions as coming from a single "class", and select comparisons leading to small intra-class distances but large inter-class distances. The selection of comparisons leading to large intra-class distances is made offline from a large set of binary tests using random patches. This online descriptor adaptation process can also be applied to other binary descriptors, and the authors demonstrate that it leads to a consistent improvement in precision-recall curves when applied to BRIEF, ORB and BBoost descriptors.

4 Deep-learning-based local features

Deep learning is an approach to machine learning that involves mapping an input through a cascade of nonlinear processing layers to produce an output [Deng et al., 2014, Schmidhuber, 2015, LeCun et al., 2015, Goodfellow et al., 2016]. The cascade is said to be "deep" if it has many ($\geq$ 3) layers. By automatically learning features across multiple layers, such a system can learn complex functions mapping raw data to outputs directly, without having to rely on handcrafted features. Often, the layers are those of an artificial neural network, trained by some variant of gradient descent, in which gradients are computed by backpropagation. In the field of computer vision, this network is often a convolutional neural network (CNN) with an image as input. Such a CNN consists of a succession of convolutional layers, whose units (neurons) have learnable weights, and other intermediate layers which play a variety of functions, such as introducing non-

[17] Source code for BBoost is available at `https://cvlab.epfl.ch/research/detect/binboost`.

[18] Open source implementation of BOLD is available at `http://vbalnt.io/projects/bold/`.

linearities, pooling to downsample and normalising activations across a batch of inputs.

In the past six years, deep learning has revolutionized computer vision, enabling huge improvements in the state-of-the-art for tasks such as image recognition [Krizhevsky et al., 2012, He et al., 2016, Huang et al., 2017] and pushing the community to propose deep-learning methods for most tasks associated with local features. As in the previous two sections, our discussion of such methods begins by considering keypoint detection (Section 4.1), real descriptors (Section 4.2) and binary descriptors (Section 4.3). However, we conclude the section with a discussion of methods for end-to-end detection and description (Section 4.4).

4.1 Learning detectors with CNNs

Lenc and Vedaldi [2016] discuss covariant point detectors[19]. The authors treat detection as a regression problem in which one learns a function $\phi : \mathcal{X} \to \mathcal{G}$ that maps image patches from a set $\mathcal{X}$ to transformations from a group $\mathcal{G}$. They use a loss that encourages the function ϕ to approximately satisfy the *covariance constraint*, which requires that

$$\phi(gx) = g\phi(x) \quad \text{for all image patches } x \in \mathcal{X} \text{ and transformations } g \in \mathcal{G}.$$

Approximating such functions ϕ with CNNs resembling the compact LeNet model of LeCun et al. [1998], the authors learn three detectors. The first two detectors are covariant with the group of translations, i.e. they are corner detectors. The third detector is covariant with the group of translations and 2D rotations. The authors call the latter detector ROTNET, but we shall use the terminology of Zang et al. [2017] and call it CovDet. To use the CovDet detector, the authors apply the CNN ϕ to the full image, and each pixel votes for a single rotation and translation. These votes are accumulated and used as confidence scores. Only rotations and translations whose confidence score exceeds a threshold are subjected to non-maximum suppression and retained as detected keypoints. Their results show that CovDet performs noticeably better than SIFT when recovering the relative orientation of randomly rotated patch pairs.

The transformation covariant local feature detector (TCovDet)[20] [Zang et al., 2017] is an extension of CovDet. Like CovDet, TCovDet treats detection as a regression problem, learning a CNN ϕ, called the "transformation prediction network" or "transformation regressor", that maps input patches to transformation matrices and selecting keypoints based on voting for transformations. However, the network in TCovDet is trained using patches selected by TILDE, which the authors call "standard patches". Also, the training objective is to recover 24 randomly-selected affine transforms per standard patch, rather than just translations and rotations as in CovDet. Furthermore, the loss is augmented by requiring that the original "untransformed" patch detected by TILDE maps to the identity transformation. The authors argue that this augmented loss resolves a major drawback with CovDet, which is that the regressor ϕ that CovDet learns may not be unique. The results show that TCovDet improves repeatability scores relative to 13 other detectors on three large datasets. Furthermore, TCovDet improves matching scores against four of those detectors in two

[19] Matlab code from the work of Lenc and Vedaldi [2016] is available at `https://github.com/lenck/ddet`.

[20] TensorFlow code for TCovDet is available at `https://github.com/ColumbiaDVMM/Transform_Covariant_Detector`.

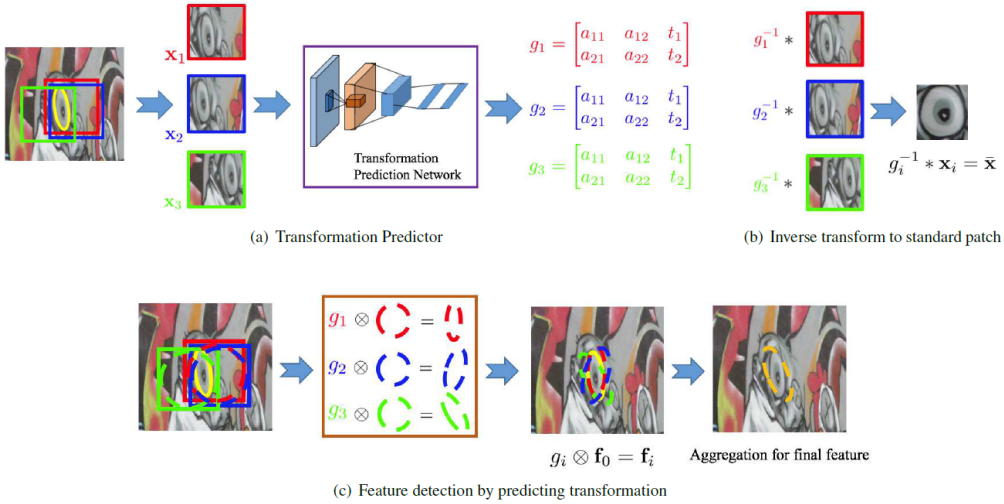

Fig. 5 Illustrating the TCovDet [Zang et al., 2017] keypoint detection framework. (a) The transformation prediction network predicts the geometric transformation of the patch. (b) The inverse of the predicted transformation is such that it warps the patch to a "*standard patch*". (c) The predicted transformations are aggregated to vote for the locations and shapes of keypoints. (By courtesy of Xu Zhang.)

out of the three datasets.

Savinov et al. [2017] treated detector design as the problem of learning a response function, which maps patches to real numbers that determine a *ranking* of image points. The keypoints detected are those points at which the value of the response function is in the top or bottom quantiles of the set of all responses after non-maximum suppression. Such a detector is considered to be good if these quantiles are preserved under the transformations to which the detector is intended to be invariant. The authors propose a variety of neural networks implementing such response functions that they collectively call *quad-networks* since these networks are trained using a quadruplet loss. A *quadruplet loss* is a loss which takes the network outputs for four patches as input. Specifically, say we are given a set of four patches $\mathcal{P} := \{p, q, p', q'\}$ consisting of a pair of patches p, q corresponding to distinct world points and the same pair of patches p', q' after a transformation that we want the detector to be invariant to. The aim is to train weights w of the response function $H_w(\cdot)$ to minimise the loss

$$\text{loss}_w(\mathcal{P}) := \max\{0, 1 - R_w(\mathcal{P})\} \quad \text{where } R_w(\mathcal{P}) := (H_w(p) - H_w(q))(H_w(p') - H_w(q')).$$

This loss vanishes if $R_w(\mathcal{P}) \geq 1$ and is positive otherwise. In order for $R_w(\mathcal{P})$ to be large, it is necessary that either patches p, p' both have much larger responses than patches q, q' respectively, or patches q, q' both have much higher responses than patches p, p' respectively.

The authors compared detectors based on a variety of network architectures with the DoG detector, but with no other handcrafted or learned detector, in two settings. The first setting involved learning a detec-

tor to match RGB images with RGB images. In this setting the authors consider a linear network and a convolutional network with a single hidden layer, finding that both networks outperform the DoG detector with except in one test. The exceptional test measured the robustness of matching to JPEG compression artefacts, which was a transformation not included in training data. The second setting involved matching RGB images with depth images. In this setting they considered three networks: a deep network with 10 convolutional layers interleaved with exponential linear units (ELU) and batch normalisation layers; a shallow fully-connected network; and a deep fully-connected network. The deep fully-connected network outperformed the other detectors on this task when the number of keypoints to be detected per image was moderate (between 500 and 1500).

4.2 Learning real descriptors with CNNs

We now discuss the use of CNNs to learn real descriptors targeting applications involving matching pairs of patches corresponding to the same 3D point. This topic has attracted at least 12 research papers to date. In fact, the earliest work on this topic [Jahrer et al., 2008] predates AlexNet [Krizhevsky et al., 2012], the network whose performance on ImageNet was a key driver of the boom in popularity of CNNs in computer vision. We begin with papers exploring the use of intermediate activations of AlexNet as descriptors, before discussing approaches based on metric learning, in which the network learns not only descriptors but also a function giving a distance between descriptors. The remainder of the section considers CNNs whose outputs are descriptors to be compared with Euclidean distance (with one exception). We structure the discussion around the loss used to train these CNNs, considering pairwise losses, triplet losses, global losses and finally histogram losses.

AlexNet Activations. The success of AlexNet [Krizhevsky et al., 2012] inspired many authors to use the activations of its intermediate layers as descriptors for other datasets and tasks other than the ImageNet classification task for which it was designed. Donahue et al. [2014] showed that such descriptors gave results that surpassed the state of the art at the time on tasks including domain adaptation, fine-grained recognition and scene-type recognition (with *abbey*, *dinner*, *mosque* and *stadium* as categories). Meanwhile, Long et al. [2014] successfully applied the intermediate activations of a network almost identical to AlexNet to the tasks of intra-class alignment, keypoint prediction and keypoint classification.

Fischer et al. [2014] were the first to use the intermediate activations of AlexNet as descriptors for the task of matching patches corresponding to the same 3D point. Soon after, Paulin et al. [2015] also presented matching results using AlexNet. Both papers use the Euclidean distance between descriptors and compare matching mean average precisions (mAP) with those for the SIFT descriptor. However, Fischer et al. [2014] compute descriptors for regions detected with MSER, whereas Paulin et al. [2015] use the Hessian-affine detector. Noting that AlexNet has five convolutional layers followed by three fully-connected layers, the results of Fischer et al. [2014] suggest that the performance of AlexNet-based descriptors is nearly independent of the choice of layer if the best input patch sizes are chosen. Further, the results show that AlexNet-based descriptors clearly outperform SIFT descriptors for a wide range of choice of layer and patch size. In contrast, Paulin et al. [2015] show a clear preference for using AlexNet's fourth convolutional layer, in line with the results of Long et al. [2014], arguing that earlier layers of such networks tend to encode more task-independent information. Moreover, they find higher mAP using SIFT descriptors than AlexNet-based

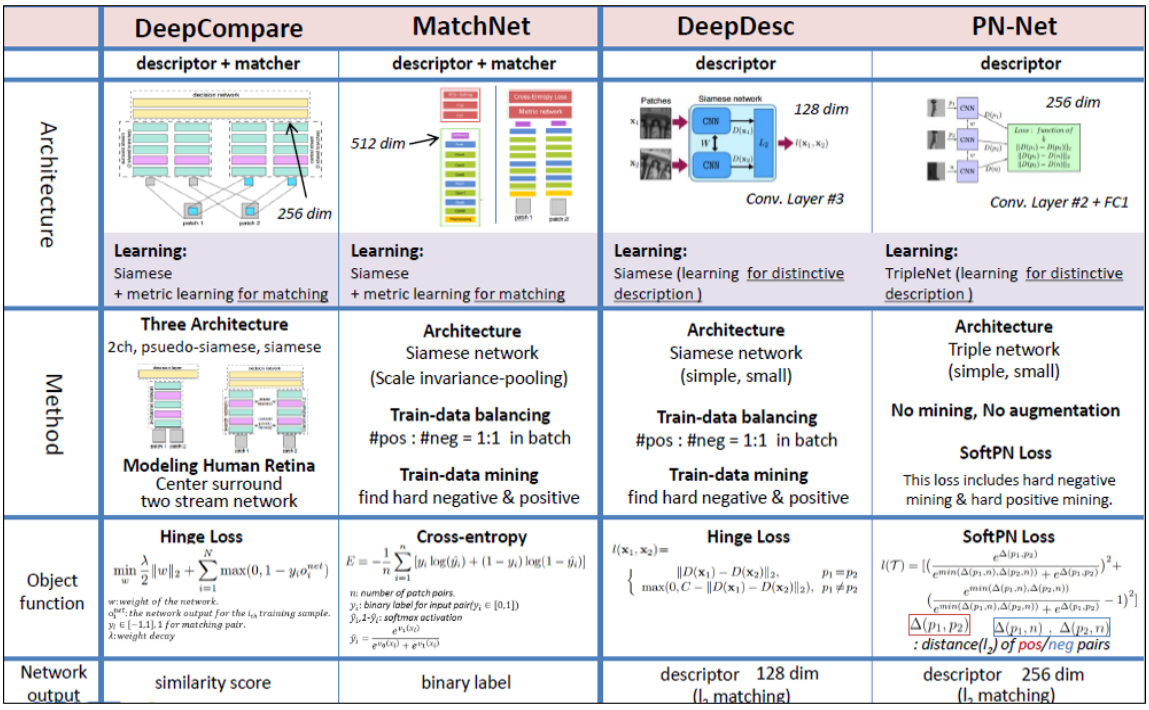

Fig. 6 An overview of some of the first Siamese CNN-based local features learning methods. From left to right, DeepCompare [Zagoruyko and Komodakis, 2015], MatchNet [Han et al., 2015], DeepDesc [Simo-Serra et al., 2015b] and PN-Net [Balntas et al., 2016a]. (By courtesy of Yukyung Choi.)

descriptors.

Both Fischer et al. [2014] and Paulin et al. [2015] propose other descriptors, and demonstrate that they clearly outperform both AlexNet activations and SIFT. The descriptors proposed by Fischer et al. [2014], which Paulin et al. [2015] call *PhilippNet*, use the same network architecture as AlexNet, but train it as follows. A collection of 16000 random 'seed' patches were extracted from Flickr images. Next, 150 random geometric and photometric transformations were applied to each such seed patch. The CNN was trained to associate a single class label to all transformed versions of a given seed patch. Meanwhile, Paulin et al. [2015] proposed a patch-convolutional kernel network (patch-CKN) which exploits a fast and simple stochastic procedure to compute a finite-dimensional feature embedding that approximates a kernel feature map.

Metric Learning. While nearly all other work discussed in this section learns descriptors and compares them with the Euclidean metric, MatchNet[21] [Han et al., 2015] jointly learns a descriptor *and* a metric for comparing descriptors. It computes descriptors by combining a CNN with multiple convolutional and spatial pooling layers plus an optional bottle neck layer. Meanwhile, the metric network passes such descriptors from two patches through three fully-connected layers (see Figure 6). MatchNet is trained by concatenat-

[21] MatchNet code and pre-trained model available at `http://www.cs.unc.edu/~xufeng/matchnet`.

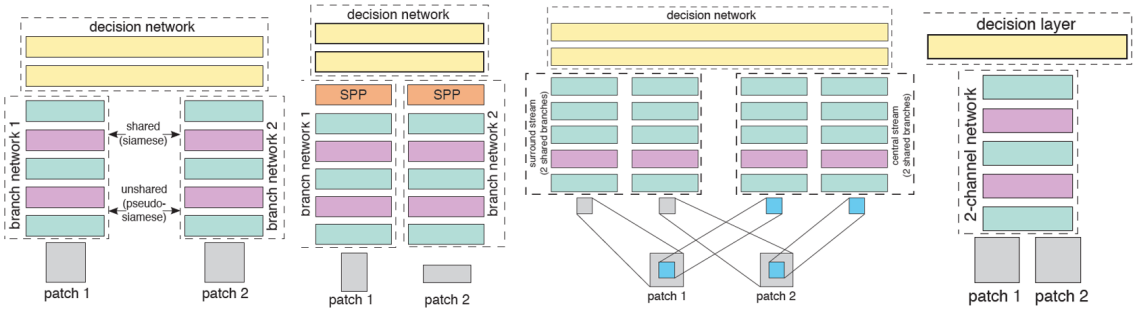

Fig. 7 Various network architectures explored in Zagoruyko and Komodakis [2015]. From left to right: Siamese and pseudo-Siamese networks (the difference between these is that the pseudo-Siamese network does not have shared branches), spatial pyramid pooling (SPP) Siamese network, central-surround two-stream network, and the two-channel network. (By courtesy of Sergey Zagoruyko.)

ing two copies of the descriptor network with a metric network and using a cross-entropy loss, thereby transforming the matching problem into a classification problem. The authors note that this architecture has similarities with to that in Zbontar and LeCun [2016], which was designed to learn a similarity between patches that is used as a matching cost for a stereo algorithm. Although MatchNet improves matching accuracy, it is not obvious how to combine it with fast approximate nearest neighbour algorithms, like hierarchical navigable small worlds [Malkov and Yashunin, 2016], which rely on the use of Euclidean distances.

Similarly, Zagoruyko and Komodakis [2015][22] also focus on learning metrics for comparing patches. An interesting aspect of the paper is that it explores a variety of different neural network models for the task, that are collectively known as *DeepCompare*, namely: Siamese networks, pseudo-Siamese networks and two-channel networks, spatial pyramid pooling (SPP) Siamese networks, as well as two-stream multi-resolution models called central-surround two-stream networks (see Figure 7). In contrast to Siamese networks, whose two streams have identical architectures and identical weights, pseudo-Siamese networks have two branches whose architectures are identical but whose weights are *not* shared, allowing for additional flexibility. Meanwhile, in a two-channel network, pairs of patches are directly fed to the network as two-channel images and hence they are processed jointly. The spatial pyramid pooling (SPP) Siamese network inserts a spatial pyramid pooling layer between the convolutional layers and the fully-connected layers of the network. Such a layer allows the processing of input patches of different sizes. Finally, the central-surround two-stream network consists of two separate streams: surround, which takes full 64×64 patches as input; and central, which takes only the central 32×32 portions of those patches as input. This enables processing to take place at two different spatial resolutions. The paper compares these methods on patch matching and wide-baseline stereo benchmarks, finding that the two-channel network clearly outperforms the others. Nevertheless, of the methods evaluated, the two-channel network would be the most computationally expensive in practice if many pairs of patches have to be tested against each other in a brute-force manner.

Pairwise Losses. To compare two patches, most recent methods apply the same CNN to extract a descriptor from each patch and they take the Euclidean distance between these descriptors. All CNNs discussed in

[22] Source code and trained models are available at `http://imagine.enpc.fr/~zagoruys/deepcompare.html`.

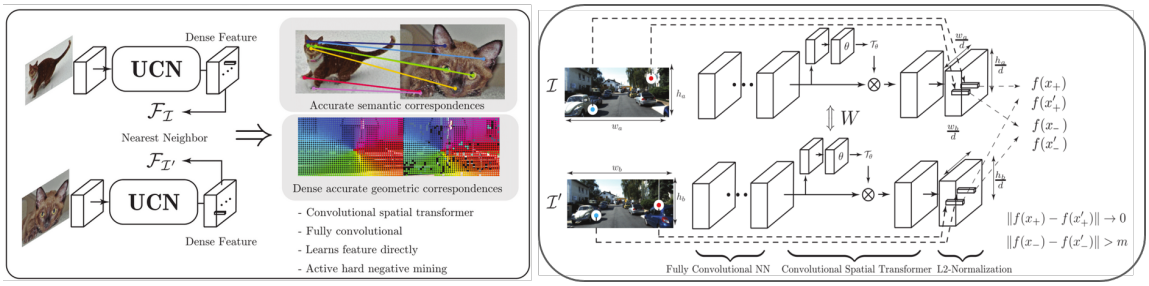

Fig. 8 The UCN architecture [Choy et al., 2016] which can accurately and efficiently learn a metric space for geometric correspondences, dense trajectories or semantic correspondences. It consists of a fully convolutional neural network combined with a convolutional spatial transformer and followed by channel-wise L_2 normalisation. Features that correspond to the positive points (from both images) are trained to be closer to each other, while features that correspond to negative points are trained to be a certain margin apart. (By courtesy of Christopher B. Choy.)

the rest of this section operate like this (except Wei et al. [2018] who apply a Gaussian kernel to a "projection distance"). Furthermore, all these CNNs learn from a large set of pairs of patches that are either known to correspond or known to not correspond (see Figure 6). However, these CNNs differ in the type of loss used during training. In particular, some use *pairwise losses*, where each term involves the distance between two patches, some use *triplet losses*, where each term involves the distance between three patches and others use *global losses*, where each term involves distances between more than three patches.

The use of a pairwise loss for learning real descriptors was first proposed by Jahrer et al. [2008], years before the explosive growth in popularity of CNNs in computer vision. In particular, those authors used a square/exponential loss, which is proportional to the squared Euclidean distance d^2 for patches corresponding to the same 3D point and proportional to a negative exponential $\exp(-\alpha d)$ for some $\alpha \in \mathbb{R}$, for patches corresponding to different 3D points.

DeepDesc[23] [Simo-Serra et al., 2015b] learns 128-dimensional descriptors (the same dimensionality as SIFT) whose Euclidean distances reflect patch similarity (see also Figure 6). Based on the observation that after a certain stage in the learning process, most pairs are correctly classified and no-longer bring an improvement in the performance of the descriptors, they propose a strategy of aggressive mining of "hard" positives and negatives. This is done by selecting, after each forward-propagation, the non-corresponding pairs that are hardest to discriminate and the corresponding pairs that match most poorly. Only such pairs are then backpropagated through the network.

Choy et al. [2016] propose the UCN[24] (Universal Correspondence Network) which also works with Euclidean distance between pairs of L_2-normalised descriptors. Its particularity compared to previous approaches is that it provides a single framework to efficiently handle geometric correspondences, dense trajectories and semantic correspondences. Furthermore, network's input is a whole image, not only an extracted

[23] Torch7 code and pre-trained models for DeepDesc are available at `https://github.com/etrulls/deepdesc-release`.

[24] The UCN code, license agreement, and pre-trained models are available at `http://cvgl.stanford.edu/projects/ucn/`.

patch, making the method suitable for dense correspondence (see Figure 8). The network extracts descriptors from each image using the initial layers of GoogLeNet [Szegedy et al., 2015]. It is trained with a *correspondence contrastive loss*, whose value for a given pair of descriptors f, f' for points x, x' from two images is proportional to

$$\begin{cases} \|f - f'\|_2^2 & \text{if points } x, x' \text{ correspond} \\ (\max\{0, m - \|f - f'\|_2\})^2 & \text{if points } x, x' \text{ do not correspond} \end{cases}$$

where m is a hyperparameter. Inspired by the GPU implementation in Garcia et al. [2010], the authors implement nearest neighbour search as a Caffe [Jia et al., 2014] layer to mine hard negatives on-the-fly. The authors optionally include a *convolutional spatial transformer* in their network, as proposed by Jaderberg et al. [2015]. This transformer attempts to make the extracted features invariant to particular families of transformations. The results of the paper suggest that the transformer is beneficial for semantic correspondence tasks or when there are large geometric transformations between images, but otherwise this component can reduce the quality of matching.

Triplet Losses. Balntas et al. [2016a] proposes the use of a softened version of the triplet loss for training descriptors to be matched with the Euclidean distance[25], calling the resulting network PN-Net, where P and N stand for positive and negative (see Figure 6). This work was continued in Balntas et al. [2016b], where the authors compare a variety of alternative triplet losses and binary-classification-based losses for descriptor training. Also, Yang et al. [2017] propose a set of methods called DeepCD[26] which involve augmenting the PN-Net architecture with an extra stream. This extra stream learns a *complementary descriptor*: that is, a descriptor which is intended to help the descriptor output by the PN-Net stream. The authors experiment with complementary descriptors that are either real vectors compared with Euclidean distance, or binary vectors compared with Hamming distance. The final dissimilarity between two patches is taken as the product of the distance between descriptors from the PN-Net stream and the distance between descriptors from the complementary stream.

Mishchuk et al. [2017] propose HardNet[27], which has the same network architecture as L2-Net, but uses a much simpler loss. This loss maximises the difference in Euclidean distance between the closest positive and closest negative example in each batch and is formulated as follows. Each batch $i = 1, 2, \ldots, n$ has exactly one pair of patches corresponding to the same 3D point, whose descriptors are a_i and p_i. Also, batch i has two sets of patches that do not correspond, whose descriptors form the sets $\mathcal{A}_i$ and $\mathcal{P}_i$. The loss is then

$$L = \frac{1}{n} \sum_{i=1}^{n} \max \left\{ 0, 1 + \|a_i - p_i\|_2 - \min\{ \min_{p \in \mathcal{P}_i} \|a_i - p\|_2 , \min_{a \in \mathcal{A}_i} \|a - p_i\|_2 \} \right\}.$$

The authors explore the effect of batch size on performance, finding that false positive rates decline with increasing batch size until the batch size reaches 512 after which performance saturates. The use of this loss can be seen as a local hard negative mining strategy. Note also that the way the triplets are formed does not require the network to be three-streamed, in contrast to other triplet-based approaches such as TFeat [Balntas

[25] Code for PN-Net is available at `https://github.com/vbalnt/pnnet`.

[26] Code for DeepCD is available at `https://github.com/shamangary/DeepCD`.

[27] HardNet code and pre-trained model are available at `https://github.com/DagnyT/hardnet`.

et al., 2016b].

Wei et al. [2018] propose a new pooling method, called subspace pooling (SP), which enables invariance to a range of geometric deformations such as circular shifts, flipping and in-plane rotation. The basic idea of SP is to model the convolutional feature maps as the linear subspace spanned by their principal components. The authors show that the proposed SP is invariant to all the geometric changes that can be expressed as column permutations of the matrix formed by the feature maps from the last convolution layer stacked as one dimensional features. SP is therefore similar to the bilinear pooling function in Lin et al. [2015], which is equivalent to spatial reordering. Integrating the proposed pooling method with models such as HardNet or TFeat, substantially improves matching results for those models.

Global Losses. TGLoss[28] [Kumar B. G. et al., 2016] is another triplet network, in the sense that the network is applied to three patches at a time during training. However, it replaces the triplet loss with a *global* loss. This global loss involves the mean μ_+ and variance σ_+^2 of the distribution of Euclidean distances between descriptors of matching patches, as well the equivalent mean μ_- and variance σ_-^2 for non-matching patches. The intent is to keep the sum of variances $\sigma_+^2 + \sigma_-^2$ small while ensuring that the different of the means $\mu_- - \mu_+$ is large.

Similarly to DeepDesc, L2-Net[29] [Tian et al., 2017] also uses CNNs to learn high-performance 128-dimensional descriptors whose Euclidean distances reflect patch similarity. However, L2-Net has a CNN with 7 convolutional layers, a final layer that normalises the outputs to unit Euclidean length, and batch normalisation is employed during training, whereas DeepDesc's network has only 3 convolutional layers and no batch normalisation is used. Also, L2-Net abandons DeepDesc's idea of mining hard samples and instead uses batches consists of $p := 128$ correctly-matching pairs of patches plus $p^2 - p$ non-matching pairs from the same patches. This better models the rate of correct and incorrect matches to be expected in real data than the triplet loss usually applied to a Siamese network. Also, L2-Net uses a loss function consisting of three terms: one accounts for the relative distance between descriptors, one controls descriptor compactness, and the other is an extra supervision imposed on the intermediate feature maps. Finally, as the output of L2-Net approximates a zero-mean Gaussian distribution, the authors find that by applying the sign function to this output they obtain a highly-effective set of binary descriptors.

The main idea of global orthogonal regularisation (GOR)[30] [Zhang et al., 2017] is to force features to be "spread-out" in the descriptor space in order to fully utilize the expressive power of that space. This regulariser encourages randomly-sampled non-matching descriptors to resemble uniformly-distributed points on the unit sphere embedded in $\mathbb{R}^d$, where d is the dimension of the descriptor space. It does so by forcing the sample mean and second moment of the inner product of the descriptors of non-matching pairs to be close to zero and $1/d$ respectively. The authors show that adding this loss to models such as DeepDesc or TFeat (discussed above) results in better patch matching.

---

[28] Matlab code and pre-trained models are available at `https://github.com/vijaykbg/deep-patchmatch`.

[29] A Matlab implementation of L2-Net with pre-trained models is available at `https://github.com/yuruntian/L2-Net`.

[30] A TensorFlow implementation is available at `https://github.com/ColumbiaDVMM/Spread-out_Local_Feature_Descriptor`.

Histogram Losses. Most of the systems discussed above, including PN-Net, L2-Net and HardNet, are *trained* to minimise a pairwise or triplet loss, with or without some hard positive or negative mining, but they are *evaluated* in terms of mean average precision (mAP), which is the area under the precision-recall curve. Would one not expect higher mAP if one were to train such systems to directly maximise mAP? He et al. [2018b] propose DOAP (descriptors optimised for average precision) which does exactly that, using the same network architecture as L2-Net and HardNet. The authors propose methods for generating both real-valued and binary descriptors. It is not immediately obvious how to effectively approximate the mAP with a differentiable objective function. The authors do so based on results from their own paper [He et al., 2018a] on hashing and learning binary embeddings, which are based on the idea of building losses using histograms [Ustinova and Lempitsky, 2016]. In particular, they compute a histogram of the Euclidean distances between a query descriptor and all descriptors in a given minibatch, treating all descriptors in a given bin as "ties" which have an equal distance from the query descriptor. They then plug this histogram into an efficient tie-aware formulation of average precision from McSherry and Najork [2008]. To make this formulation of average precision differentiable, they replace the discrete histogram counting operation by partially assigning each inter-descriptor distance to its two closest histogram bins with linear interpolation.

While the basic DOAP descriptors already demonstrate state-of-the-art performance on standard benchmarks as we discuss in Section 5, the authors also experiment with two improvements. The first improvement (DOAP-ST) is to add a spatial transformer module [Jaderberg et al., 2015] to make matching more robust to challenging levels of geometric noise and illumination change. The second improvement (DOAP-LM) is to use *label mining*, which is a clustering method that avoids forcing the system to discriminate between visually similar patches that correspond to distinct 3D points, such as patches from different windows of the same style on the same building.

4.3 Learning binary descriptors with CNNs

Having seen some methods for learning binary descriptors in Section 3.3, it is not surprising that deep-learning methods have also been designed to tackle this problem. Indeed, some of the methods discussed in Section 4.2, such as L2-Net [Tian et al., 2017], also have binary variants. However, such methods essentially apply the sign function to each component of a real descriptor vector, and better-performing alternative methods have been proposed that are specifically designed to learn binary features.

DeepBit[31] [Lin et al., 2016] is an unsupervised deep-learning approach to learning compact binary descriptors for efficient visual object matching. The main idea is to optimise the parameters of a network using a combination of three losses. The first loss forces the binary descriptors to preserve the local data structure by minimising the quantisation loss when the activations of the last layer are projected into binary descriptors. The second loss encourages each bit of the binary descriptor to be evenly distributed, with the intention of making the descriptor more discriminative. Finally, the third loss encourages the descriptor to be invariant to rotations, simply by penalising changes in the descriptor if the input patch is rotated.

[31] Code available at `https://github.com/kevinlin311tw/cvpr16-deepbit`.

DBD-MQ (deep binary descriptor with multi-quantisation) [Duan et al., 2017a] is another unsupervised descriptor learning method. The authors propose a novel and effective approach to quantising real descriptors that they call a K-autoencoder network. This consists of K autoencoders, each of which uses a c-bit binary representation of a given real descriptor. They experiment with $c = 16, 32$ and 64 and find that $K = 4$ gives the best mAP on a matching task. The binary descriptor is the result of concatenating the K binary representations. The autoencoders are trained by analogy with the k-means clustering algorithm: given a real descriptor, they find the autoencoder giving the least reconstruction error; then they backpropagate the error associated with that descriptor through that autoencoder alone.

4.4 End-to-end detection and description of local features

Relative to the number of deep descriptors, only a few end-to-end deep models have been proposed that aim to learn the entire detection and description pipeline (Figure 1). This is in spite of the fact that most handcrafted detectors rely on convolutional filters just like CNNs, as well as the fact that one motivation for deep learning is to work directly with raw input rather than relying on the output of handcrafted (and therefore suboptimal) preprocessing.

Yi et al. [2016a] appear to have been the first authors to propose a fully end-to-end approach to feature detection and description. Their LIFT[32] (learned invariant feature transform) network is composed of three CNN components that feed into each other (see Figure 9): a detector, an orientation estimator and a descriptor. These components are linked with two spatial transformers [Jaderberg et al., 2015] that geometrically manipulate the image patches while preserving differentiability. The first spatial transformer crops the input patch to give a smaller patch centred at the point output by the detector, while the second rotates this smaller patch to the angle determined by the orientation estimator. The authors use the TILDE detector [Verdie et al., 2015]. At training time, they modify TILDE to ensure differentiability by replacing non-maximum suppression with a softargmax function, which is defined as follows. If $S(y) \in \mathbb{R}$ notes the score for image coordinate y output by TILDE for a patch P with domain $\mathrm{dom}(P)$, then the estimated location of the feature point is

$$\mathrm{softargmax}(S) := \frac{\sum_{y \in \mathrm{dom}(P)} y e^{\beta S(y)}}{\sum_{y \in \mathrm{dom}(P)} e^{\beta S(y)}}$$

where $\beta \in \mathbb{R}$ is a hyperparameter. The orientation estimator relies on the CNN proposed by Yi et al. [2016b] which was designed to estimate orientations for matching purposes[33] and which demonstrates significant improvements over SIFT's orientation estimator. Finally, DeepDesc [Simo-Serra et al., 2015b] is used for the descriptor component, since it is a simple network, which does not require metric learning. The authors found it impossible to learn the entire network from scratch. Instead, they initialise the weights by first learning the descriptor, then using that descriptor to learn the orientation estimator, and then using the orientation estimator and descriptor to learn the detector. Finally, the whole network is trained end-to-end with quadru-

[32] Code and pre-trained models are available at `https://github.com/cvlab-epfl/LIFT`.

[33] Code is available at `https://github.com/cvlab-epfl/learn-orientation`.

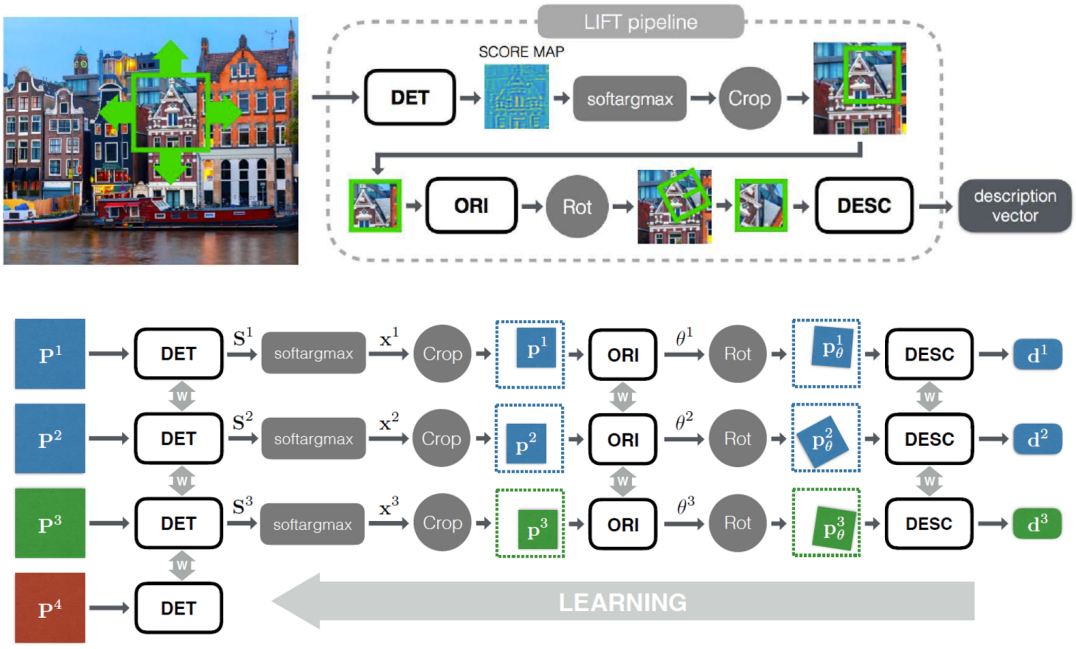

Fig. 9 Top: The LIFT pipeline from Yi et al. [2016a]. Given an input patch $\mathbf{P}$, LIFT applies a detector (DET) followed by a softargmax and a crop operation to provide a smaller patch $\mathbf{p}$ that lies within $\mathbf{P}$. The smaller patch is fed to an orientation estimator (ORI) and rotation operation which provides rotated patches as input for a network that computes descriptors (DESC). Bottom: LIFT's Siamese training architecture that works with quadruplets of patches. Patches $\mathbf{P}^1$ and $\mathbf{P}^2$ (blue) correspond to different views of the same physical point and are used as positive examples to train the descriptor (DESC). Patch $\mathbf{P}^3$ (green) shows a different 3D point which serves as a negative example for DESC. Finally patch $\mathbf{P}^4$ (red) contains no distinctive feature points and is only used as a negative example to train the detector (DET). (By courtesy of Kwang Moo Yi.)

plets of patches, as shown in Figure 9. At test time, scaled versions of the image are fed to the network, and the detector generates score maps at each scale, which are processed by non-maximum suppression to give keypoint locations. The orientation detector and descriptor are then applied only to patches at the detected keypoint locations.

In contrast to the other end-to-end models discussed here, DELF[34] (deep local features) [Noh et al., 2017] was designed to perform more accurate matching and geometric verification for large-scale image retrieval. The authors use visual attention for keypoint selection, arguing that keypoint selection is important for both accuracy and computational efficiency of retrieval systems, since a substantial fraction of local features are irrelevant and may distract such systems. While visual attention based on deep neural networks had previously been employed for many other computer vision tasks [Hong et al., 2015, Xu et al., 2015], it had not previously been used for image retrieval. The DELF pipeline has four main blocks: (i) dense feature extraction, using an intermediate output of a ResNet50 [He et al., 2016]; (ii) attention-based keypoint selection,

[34] DELF code available at `https://github.com/tensorflow/models/tree/master/research/delf`.

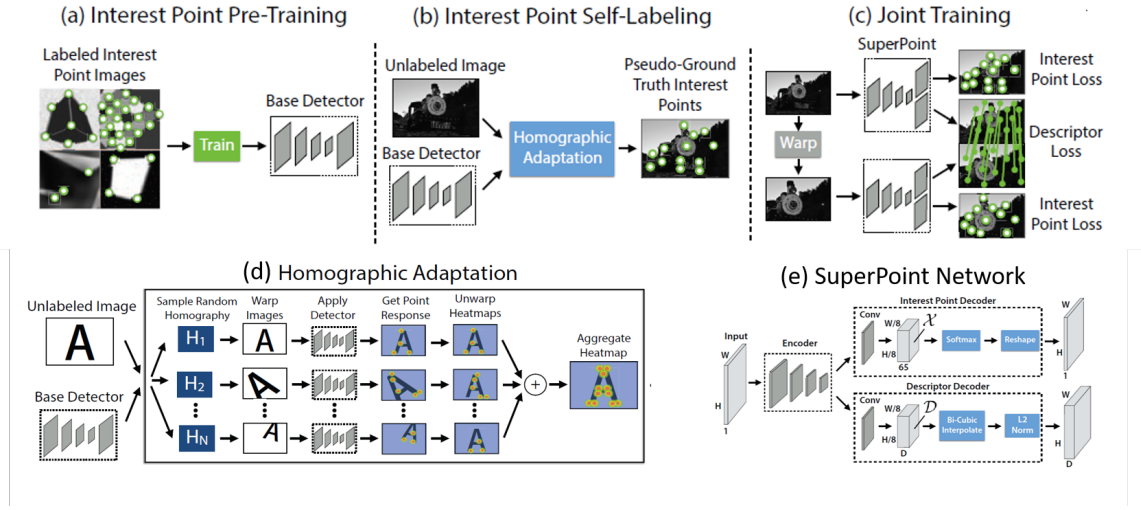

Fig. 10 Overview of the SuperPoint framework for self-supervised training of keypoint detectors from DeTone et al. [2018]. *Keypoints* are here referred to as *interest points*, consistent with the authors of that paper. (a) An initial interest point detector is pre-trained using synthetic data. (b) Then the homographic adaptation procedure automatically generates pseudo-ground truth interest points. (c) The pseudo-ground truth interest points are used to train the SuperPoint network that jointly extracts interest points and descriptors from an image. (d) Homographic adaptation consists in averaging the response of a detector over a set of random homographies. (e) The SuperPoint network consists of a single encoder whose output is fed to two decoders. (By courtesy of Daniel DeTone.)

using a two-layer CNN with a softplus activation at the top (softplus is the function $x \mapsto \log(1+\exp(x))$); (iii) dimensionality reduction, using PCA; and (iv) indexing and retrieval, in which 40 local features from each query image are matched with a database by nearest-neighbour search and the matches are geometrically verified with RANSAC. The results show that DELF significantly outperforms pre-existing methods for image retrieval in terms of its precision-recall curve. In particular, unlike other methods evaluated, it is robust to queries that have no match in the database.

SuperPoint [DeTone et al., 2018] is a framework for self-supervised training of keypoint detectors and descriptors for multiple-view geometry problems. Given an input image, it jointly computes keypoint locations and descriptors in a single forward pass. As shown in Figure 10, first a base keypoint detector is trained on examples from a synthetic dataset consisting of simple geometric shapes for which the keypoint locations are well defined. Then, to improve the repeatability of the detector and enlarge the set of stimuli to which it responds, a process called *homographic adaptation* is applied, which essentially averages the output of a detector over a suitably-chosen distribution of random homographies. Specifically, the authors define the homographic adaptation f^{HA} of detector f applied to image I as the average

$$f^{\mathrm{HA}}(I) := \frac{1}{n}\sum_{i=1}^{n} H_i^{-1}(f(H_i(I)))$$

where $(H_i)_{i=1}^n$ is a sequence of homographies. The keypoints detected by the homographic adaption of the base keypoint detector, which the authors call *pseudo-ground truth interest point locations*, are collected and used as training data for the SuperPoint network. This network has a single encoder whose output is fed into a pair of decoders. One decoder performs interest point detection and the other computes descriptors. The detector is trained with a cross-entropy loss, where labels are derived from the pseudo-ground truth interest point locations. Meanwhile, the descriptor is trained using the sum of a hinge loss for pairs of descriptors that are known to correspondent and another hinge loss for pairs of descriptors that are known to not correspond.

5 Performance comparisons

In this section, we begin by discussing performance comparisons of handcrafted local features on matching and patch retrieval tasks (Section 5.1) and move on to comparisons that also involve deep-learning-based features (Section 5.2). Finally, we discuss an evaluation of the impact of the choice local features on the complex task of image-based reconstruction (Section 5.3).

5.1 Comparisons of handcrafted local features

Heinly et al. [2012] analysed the performance of different handcrafted detector and descriptor pairings on several datasets, namely the Harris, MSER, FAST, BRIEF, ORB, BRISK, SURF and SIFT detectors, and the BRIEF, ORB, BRISK, SURF and SIFT descriptors. The authors examined the impact of both geometric and non-geometric changes (blur, JPEG compression, exposure, day-to-night) on matching. They defined the *matching score* as the ratio of the number of correct matches to the number of features detected in the first image to be matched. In terms of matching scores and in the presence of non-geometric changes, the results showed that BRIEF descriptors beat ORB, BRISK and even SIFT descriptors, and that they did so whichever detector was used. However, as soon as there was any rotation between the images to match, BRIEF descriptors performed awfully compared with ORB and BRISK descriptors. In general, in the presence of geometric changes, the matching scores for the SIFT detector and descriptor were consistently better than those for the other detector-descriptor pairs.

Mishkin et al. [2015] propose the wide multiple baseline stereo (WxBS) dataset[35], which consists of image pairs with a variety of combinations of large changes in geometry, illumination, sensor, appearance and modality. Every image pair is manually labelled with approximately 20 correspondences as ground-truth. The authors explore matching visible-spectrum images with infrared images, matching images acquired with different modes of magnetic resonance imaging, and even matching maps to satellite views (see Figure 11). They show that this is an extremely challenging benchmark and that using only one detector-descriptor model at a time does not perform well. To quantify matching performance, the authors compare the number of images for which at least 15 correct inliers to a homography are found. Of the detector-descriptor combina-

[35] The WxBS dataset can be found at `http://cmp.felk.cvut.cz/wbs/index.html`.

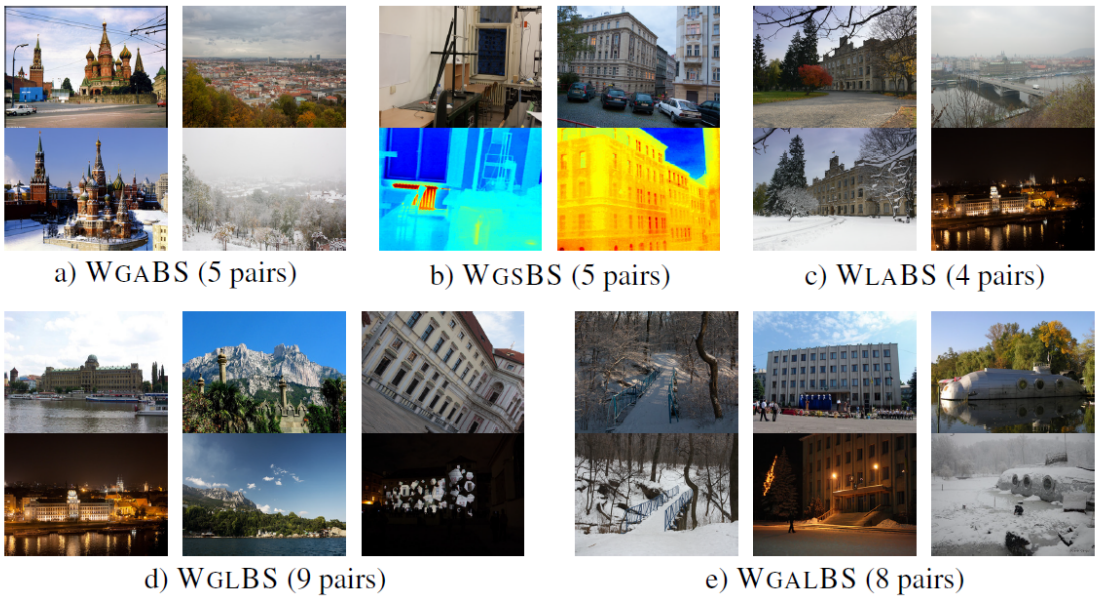

Fig. 11 Example images from the WXBS wide-baselines dataset proposed by Mishkin et al. [2015]. WGABS stands for viewpoint and appearance changes, WGSBS for viewpoint and modality changes, WLABS for lighting and appearance changes, WGLBS for viewpoint and lighting changes, and WGALBS for viewpoint, appearance and lighting changes. (By courtesy of Dmytro Mishkin.)

tions that the authors explore[36], the best results are obtained with models that incorporate multiple detectors and descriptors, such as those proposed by Yang et al. [2007] and Mishkin et al. [2014]. However, in terms of single-detector/descriptor approaches, an adaptive Hessian-affine detector was the best-performing detector, while SIFT and its variants including DAISY [Tola et al., 2010] were the best-performing descriptors. Furthermore, it was observed that most of the descriptors gain significantly from photometric normalisation. One of the main problems in day-to-night matching and matching infrared images is the low number of detected features. A possible approach addressing this problem is iiDoG [Vonikakis et al., 2013], where the SIFT detector's difference of Gaussians is normalised by sum of Gaussians, but this approach cannot be easily be applied to other detectors.

Maier et al. [2017] presented a method to generate ground-truth matches based on the original ground truth of well-known datasets, pointing out that previous notions of a "correct match" were ambiguous, partly because they relied on arbitrarily-set thresholds. For instance, Mikolajczyk et al. [2005] used a threshold of 40% on an "overlap error criterion", Mikolajczyk and Schmid [2003] used a threshold 50% on an "overlap error criterion" whereas Heinly et al. [2012] used a threshold of 2.5 pixels on the location of keypoints. The authors also conducted extensive tests on 133 keypoint-descriptor combinations on the HCI Training 1K

[36] 14 such combinations are explored in the poster associated with the paper, which can be found at `http://cmp.felk.cvut.cz/~mishkdmy/posters/wxbs-2015-poster.pdf`, but only up to 13 combinations were tested, depending on the dataset, in the paper itself.

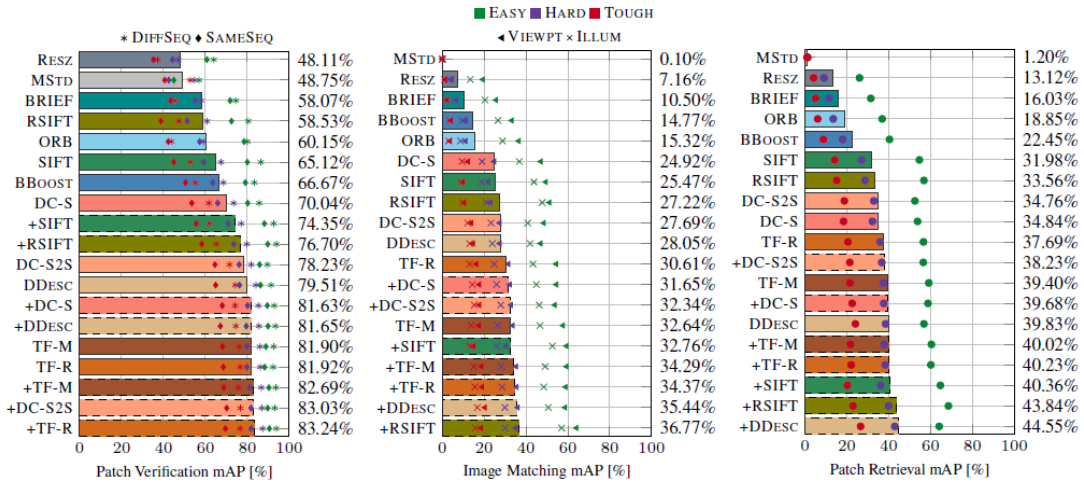

Fig. 12 Mean average precisions (area under the curve) for patch verification, image matching and patch retrieval tasks in the HPatches benchmark [Balntas et al., 2017]. Before extracting descriptors, each patch is randomly perturbed by applying a rotation, translation and anisotropic scaling, each of which is drawn from a uniform distribution. The colours of the markers indicate the "difficulty" of this geometric-transformation noise, which corresponds to the ranges of the uniform distributions. Triangular markers correspond to results from image sequences with viewpoint changes, while crosses correspond to image sequences with photometric changes. Bars show the average of the six variants of each task. Bars with dashed borders and with a + before the descriptor name indicate ZCA projected and normalised features. (By courtesy of Vassileios Balntas.)

flow [Kondermann et al., 2016], KITTI [Moritz and Andreas, 2015] and Oxford [Mikolajczyk and Schmid, 2003] datasets. The best average accuracy was attained by the combination of the BRISK [Leutenegger et al., 2011] detector and the FREAK [Alahi et al., 2012] binary descriptor, with an average accuracy of 81.5%, closely followed by the combination of the MSD [Tombari and Di Stefano, 2014] detector and the optimised ConvOpt [Simonyan et al., 2014] descriptor. Meanwhile, the LATCH [Levi and Hassner, 2015] and BOLD [Balntas et al., 2015] descriptors performed poorly no matter which detector they were paired with, having an average accuracy consistently under 60%.

Sun et al. [2017] propose a dataset that was acquired in a shopping mall for benchmarking image-based localisation algorithms. The authors compare the localisation performance of BRIEF, SURF, SIFT, COV and RSIFT local features. COV is the authors' name for the detector proposed by Perd'och et al. [2009], which is like the Hessian-affine detector [Mikolajczyk and Schmid, 2004] but for two modifications. The first modification is to use the scale-space maxima of the Hessian operator for the initial scale selection. The second modification is that in place of computing a rotation for each patch, they use a *gravity vector* [Philbin et al., 2007], which corresponds to finding the vanishing point of vertical lines in an image. Meanwhile RSIFT [Arandjelović and Zisserman, 2012], also known as RootSIFT, is a simple variant of SIFT in which each component x_i of a SIFT descriptor $x \in \mathbb{R}^{128}$ is replaced by its square root $\sqrt{x_i}$, noting that these components are non-negative by definition. Sun et al. [2017] measure performance in terms of the fraction of all query images whose camera position is estimated within a given distance from the ground truth *and* whose angle is estimated to within 5°. The best performance was obtained with the affine-covariant COV detector coupled with RSIFT descriptors. Of course, these results are for a shopping mall environment where

the gravity vector is usually well defined and one would not expect the COV detector to be the best choice for environments with less man-made constructions.

5.2 Comparisons involving deep-learning-based local features

In contrast to the papers just discussed, recent performance comparisons have tended to consider both handcrafted *and* deep-learning-based local features. Such comparisons are the focus of this section. In particular, we discuss the work of Zang et al. [2017], who compared handcrafted and deep-learning-based detectors, of Balntas et al. [2017], who introduced a new large new dataset called HPatches for evaluating descriptors from the perspective of patch verification, matching and retrieval tasks, as well as discussing three papers [Wei et al., 2018, He et al., 2018b, Lenc and Vedaldi, 2018] that made extensive use of the HPatches data.

Zang et al. [2017] compare several handcrafted feature detectors with FAST [Rosten and Drummond, 2006], TILDE [Verdie et al., 2015], CovDet [Lenc and Vedaldi, 2016] and TCovDet [Zang et al., 2017] on multiple datasets[37]. The authors show that the repeatability TCovDet keypoints is significantly higher than the repeatability of CovDet and TILDE keypoints. CovDet and TILDE keypoints in turn have higher repeatability than handcrafted methods (namely SIFT, SURF, MSER, Harris Laplace, Hessian Laplace, Harris affine and Hessian affine detector) and FAST. In terms of matching scores, TCovDet performs best on two out of three datasets and SIFT on the third dataset, which contains drastic background clutter changes that seem to be handled less-well by TCovDet.

Balntas et al. [2017] show that previous datasets (see Table 1) and protocols for evaluating local features do not unambiguously specify all aspects of evaluation, leading to inconsistencies in results reported in the literature. To overcome these weaknesses, the authors propose a new benchmark called HPatches. This benchmark includes a large new dataset suitable for training and testing modern descriptors. It also unambiguously defines evaluation protocols for several tasks such as matching, retrieval and classification. The authors conducted an exhaustive evaluation comparing the handcrafted features SIFT, RSIFT[38] [Arandjelović and Zisserman, 2012], BRIEF, ORB, BBoost [Trzcinski et al., 2015], as well as the recent deep descriptors DeepCompare (DC) [Zagoruyko and Komodakis, 2015], DeepDesc [Simo-Serra et al., 2015b] and TFeat (TF) [Balntas et al., 2016b]. They also proposed two baseline descriptors, MSTD which is the mean and the standard deviation of the patch, and RESZ which is a vector obtained by resizing the patch to 6×6 and normalising it to have zero mean and unit variance. The results from Balntas et al. [2017] are shown in Figure 12. The learning-based descriptors were trained on the PhotoTourism [Winder, 2007] dataset, which is different from HPatches. Evaluation was done on three benchmark tasks: patch verification, image matching and patch retrieval. The authors also consider applying zero-phase component analysis (ZCA) [Bell and Sejnowski, 1997], which corresponds to multiplying a descriptor vector by $C^{-1/2}$ where C is the covariance matrix of all descriptor vectors. In most cases, post-processing the descriptors by applying ZCA, followed by power law normalisation [Arandjelović and Zisserman, 2012] and L_2-normalisation significantly improved the results, as is apparent in Figure 12 where $+$ indicates use of ZCA. This improvement

[37] The datasets can be downloaded from `https://www.dropbox.com/s/l7a8zvni6ia5f9g/datasets.tar.gz`.

[38] Using a square root (Hellinger) kernel instead of the standard Euclidean distance to measure the similarity between SIFT descriptors.

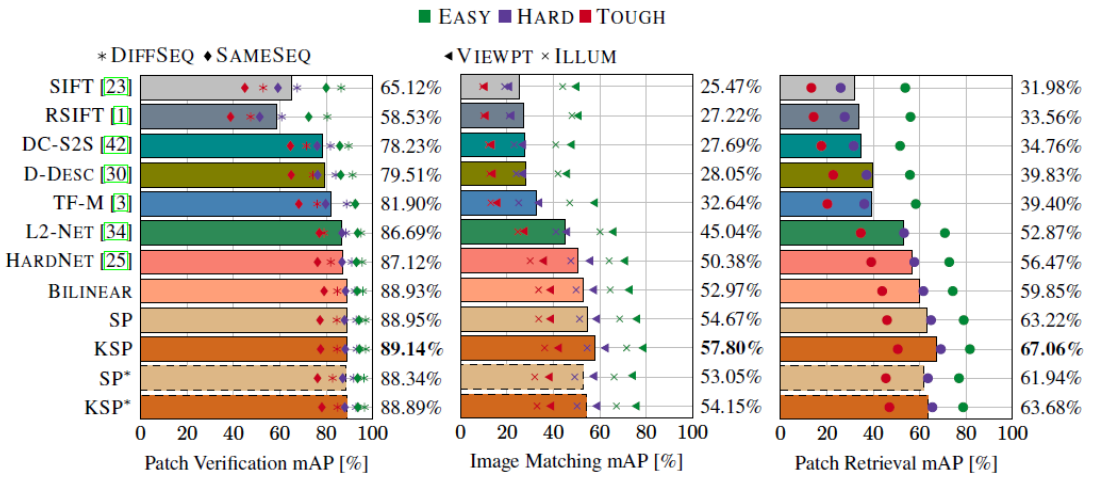

Fig. 13 Patch verification, matching, and retrieval results from on the HPatches dataset from Wei et al. [2018]. The same notation is used as in Figure 12. (By courtesy of Xing Wei.)

had already been observed in previous papers [Ke and Sukthankar, 2004, Arandjelović and Zisserman, 2012].

On the patch verification task in HPatches, deep-learning-based descriptors gave the best mean average precision (mAP). The authors argue that this is because they were jointly optimised with their distance metric to perform well in the verification task. On the image matching task, where descriptors are used to match patches from a reference image to a target image, ZCA whitened and normalised RSIFT surprisingly outperformed the deep-learning-based descriptors. ZCA whitened and normalised RSIFT had almost the highest mAP on the patch retrieval task. The authors observed that binary descriptors have a competitive mAP only for patch verification. In particular, the learning-based binary feature BBoost in general outperformed the handcrafted binary features especially on the patch verification task. Among the deep features, the best matching and retrieval performance was obtained with DeepDesc, followed by TF. However, DeepDesc performed worse on patch verification and of the deep-learning-based methods it has the highest computational cost.

Several recent papers have used the HPatches benchmark dataset and protocol to evaluate other descriptors. These include Wei et al. [2018], whose results are shown in Figure 13, as well as He et al. [2018b] whose results are shown in Figure 14. In Figure 13, the highest mAP is attained by the subspace pooling (SP) descriptors (as described in Section 4.2) and kernelised subspace pooling (KSP) descriptors (in which the marginal triplet loss was combined with a Gaussian kernel). However, these are closely followed by BILINEAR, which is the model of Lin et al. [2015] applied for learning patch matching. Clearly, more recent deep-learning-based approaches such as L2-Net [Tian et al., 2017] and HardNet [Mishchuk et al., 2017] significantly outperform the previous results on HPatches as shown in Figure 12. Furthermore, adding bilinear or subspace pooling to L2-Net further improves performance on all tasks.

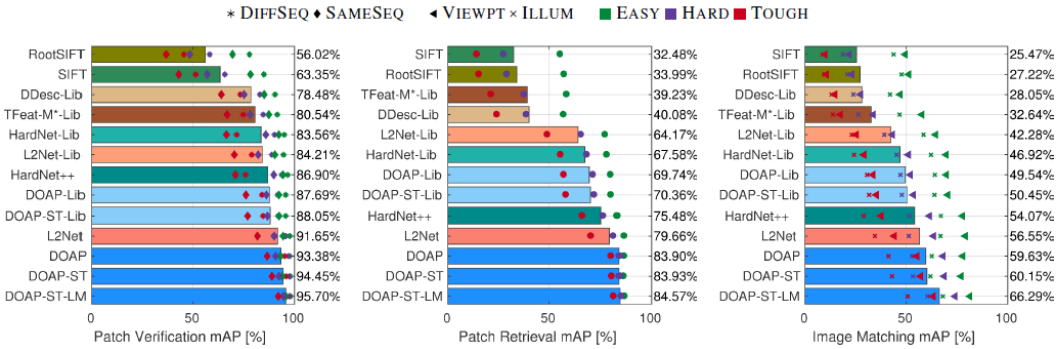

Fig. 14 Patch verification, matching, and retrieval results on the HPatches dataset from He et al. [2018b]. The suffix -Lib means that the model was trained on the Liberty set from the Photo-Tourism dataset [Winder, 2007] instead of HPatches, -ST means that the model includes a spatial transformer network to estimate the geometric transformation, and -LM means that clustering-based label mining was used during training. The same notation is used as in Figure 12. (By courtesy of Kun He.)

Figure 14 compares variants of DOAP [He et al., 2018b] with SIFT and other deep models. These results are somewhat puzzling in the light of the Figure 13, since the mAPs for L2-Net differ by large amounts between the two figures. Assuming that both results are correct, it appears that the SP and KSP descriptors of Wei et al. [2018] are outperformed on every task by DOAP: using the average precision as loss indeed appears to improve the mAP on patch retrieval, image matching and patch verification tasks. Furthermore, the two improvements to DOAP (spatial transformer and label mining) consistently improve the mAP. Notably, label mining increases mAP for image matching by over 6%. These results make DOAP-ST-LM the best-performing descriptor on the HPatches benchmark at the time of writing.

Unlike the above performance comparisons that focus on descriptors or detector-descriptor combinations, Lenc and Vedaldi [2018] focus only on detectors. In particular, the authors propose an improved repeatability measure for detector evaluation. The first improvement is to correct an error in the publicly-available implementation of the repeatability score of Mikolajczyk et al. [2005]. This error causes the repeatability score to vary strongly if the detected regions are simply scaled (magnified) by a common factor, even though the score had been explicitly designed to minimise such variation. This error concerned a heuristic used for accelerating ellipse overlap computation, which had been applied at the wrong stage of the processing pipeline. The second improvement reduces the dependency of the repeatability score on the number of regions detected, which is a tuneable parameter of most detectors. This improvement consists in reporting the average of the repeatability scores attained when 100, 200, 500 and 1000 keypoints are detected per image.

Lenc and Vedaldi [2018] also evaluate 11 existing detectors on 5 datasets using the improved repeatability measure. One of these datasets, which the authors call *HSequences*, consists of 580 image pairs drawn from the same set of 696 images as the HPatches dataset, making it the largest detector-evaluation dataset at the time of writing. In the presence of large illumination changes, they find that the learning-based detector TILDE [Verdie et al., 2015] has a consistently higher repeatability than the other detectors considered. However, in the presence of viewpoint changes, TILDE performs poorly since it aims only for translation invariance and not for scale or affine invariance. Rather, given viewpoint changes, the Hessian affine detector

Table 2 Basic properties of the descriptors evaluated in Schönberger et al. [2017]. Per image average timings were measured on the Oxford5k dataset. Extraction time includes detection time. (By courtesy of Johannes L. Schönberger.)

	RSIFT	RSIFT-PCA	DSP-SIFT	ConvOpt	DeepDesc	TFeat	LIFT
Dimensionality	128	80	128	73	128	128	128
Size (bytes)	128	320	512	292	512	512	512
Platform	CPU	CPU	CPU	GPU	GPU	GPU	GPU
Extraction (s)	9.3	10.5	23.7	49.9	24.3	11.8	212.3
Matching (s)	0.14	0.11	0.14	0.10	0.14	0.14	0.14

tends to have the highest repeatability although it is often outperformed by CovDet [Lenc and Vedaldi, 2016] and TCovDet [Zang et al., 2017].

5.3 Comparison on image-based reconstruction task

The comparative evaluation proposed in Schönberger et al. [2017] goes beyond descriptor matching, to also evaluate the performance of various descriptors on image-based reconstruction tasks using challenging small- and large-scale datasets. Such image-based reconstruction pipelines, often match descriptors in order to produce a graph of corresponding features in multiple views. All subsequent stages of such a pipeline strongly depend on the quantity and the quality of these correspondences. In order to give practical insights, the authors evaluate the impact of the choice of descriptor at four stages of such a pipeline: feature matching, geometric verification, image retrieval, and sparse and dense modelling. They consider RSIFT as a baseline descriptor, along with two advanced variants of it, RSIFT-PCA [Bursuc et al., 2015] and DSP-SIFT [Dong and Soatto, 2015], and they compare these with ConvOpt, DeepDesc, TFeat and LIFT (see Section 4.1). Except for LIFT, which is an end-to-end detector and descriptor network, SIFT's difference-of-Gaussians (DoG) keypoint detector was used for all descriptors and the learning-based approaches were trained on DoG keypoints.

Table 2 shows basic properties of the evaluated descriptors from Schönberger et al. [2017]. We can see that the extraction times of the descriptors vary by an order of magnitude with learned descriptors being up to an order of magnitude slower than handcrafted descriptors, even though the latter run on a GPU. Among the learning-based features, LIFT by far the slowest and as such it is clearly not a practical alternative for processing millions of images as required by some image-based reconstruction use cases. Meanwhile matching times depend largely on the size of the descriptor. Since this size varies by less than a factor of two over the set of descriptors evaluated, the matching times are also not highly variable.

Concerning the evaluation of the impact of the choice of descriptor at different stages of the image-based reconstruction pipeline, we summarise the findings of Schönberger et al. [2017] as follows. In agreement with Heinly et al. [2012], the paper shows that for all stages of the pipeline, blur, day-night, and large viewpoint changes seriously challenge all descriptors. The learned descriptors typically outperformed RSIFT in terms of recall, while RSIFT performed better in terms of precision. Both RSIFT-PCA and DSP-SIFT

outperformed the learned features for almost all metrics and matching scenarios tested. Among the learned descriptors, ConvOpt was found to produce the best overall results and had the lowest variance across the different datasets.

To evaluate the completeness and accuracy of the reconstruction results, the metrics used in Schönberger et al. [2017] were the number of registered images, the number of 3D points in the sparse SfM map output by COLMAP [Schönberger and Frahm, 2016], the number of verified image projections of sparse points and their track lengths, the overall reprojection error, the pose accuracy of the camera locations, as well as the number of reconstructed dense points after multi-view stereo (MVS) reconstruction using CMVS [Furukawa and Ponce, 2010].

The experiments on several datasets, yielded the following observations: on small or easy datasets, the learned descriptors generally perform on a par with or better than RSIFT in terms of the number of sparse points, the number of image observations, and the mean track length; but they performed worse than RSIFT-PCA and DSP-SIFT on these metrics. However, in terms of the number of registered images and the final dense modelling performance and accuracy metrics, all methods produce roughly the same reconstruction quality.

On larger and more challenging datasets, more variation was found when ranking the features using different metrics and datasets. In spite of the superiority of the learned descriptors over RSIFT observed in raw matching evaluation, in the reconstruction evaluation RSIFT sometimes performed better and sometimes worse than the learned descriptors. DSP-SIFT performed the best among all the methods, both in terms of sparse and dense reconstruction results. It consistently produced the most complete sparse reconstructions in terms of the number of registered images and reconstructed sparse points, and its dense models had the most points as a result of accurate camera registration. DSP-SIFT has a slightly higher reprojection error than other methods based on the DoG keypoint detector. This is potentially caused by the descriptor pooling across multiple scales, which improves robustness but also results in less accurate keypoint localisation. LIFT has the largest reprojection error and relatively short tracks on all datasets, indicating inferior keypoint localisation performance as compared to the handcrafted DoG method.

Concerning camera-pose estimation, the ground truth was only available for three datasets: two small ones, namely Fountain and Herzjesu [Strecha et al., 2008]; and the Quad6K from the Cornell BigSfM dataset [Crandall et al., 2013]. On the small datasets all methods performed similarly. On Quad6K, RSIFT performed best, followed by TFeat and DCP-SIFT, with RSIFT-PCA performing worst. In summary, even if learning approaches advanced, handcrafted features still perform on par or better than recent learned features in the practical context of image-based reconstruction.

6 Conclusions

This paper gave an overview of keypoint detectors and descriptors, focussing on local features that are designed to be localised accurately and consistently over time, rather than local features designed for extracting semantics. We traced the evolution of such detectors and descriptors, from classic handcrafted methods

through to more recent learning and deep-learning-based methods. Then we discussed existing benchmark papers, which compare most of those methods. While this discussion may be useful for a reader interested in selecting the best off-the-shelf local features type for a given task and data source at the time of writing, it also suggests that better results may be possible by combining ideas from different state-of-the-art methods. We highlight the following findings.

1. Learned methods are a good choice when image content matters, especially for applications like image matching.
2. Post-processing descriptors by whitening, power-law normalisation, and L_2-normalisation often improves matching results.
3. Among the non-deep-learning-based models, ConvOpt shows good performance across different datasets and tasks, but it was outperformed by several recent deep models. The deep models TFeat, L2-Net and HardNet perform well, but they can be improved by (kernel) subspace pooling (SP, KSP) or bilinear pooling, as well as by adding a global loss (TGLoss) or global orthogonal regularization (GOR).
4. The highest mean average precision on patch verification, matching, and retrieval tasks was attained by DOAP, which directly optimises the average precision instead of using a pairwise or triplet loss.
5. TConvDet was shown to provide the best keypoint repeatability compared to other detectors and combined with SIFT descriptor provides good matching performance.
6. LIFT and SuperPoint are among the few networks that learn keypoint detection, geometric transformation and keypoint description in a single network trained end-to-end.
7. SIFT and many of its variants show high robustness to viewpoint changes, and they remain a good option for most applications.
8. In addition to the advantages of low memory footprint and matching time, deep-learned binary features, such as binary DOAP, provide competitive results on recent benchmarks.
9. Extraction of handcrafted descriptors on a CPU is often much faster than extraction of learned descriptors on a GPU, but there are some exceptions.
10. Even though learning approaches have advanced to the extent that they now attain the highest mean average precision on matching, recent benchmarks targeting their application in image-based reconstruction and localisation pipelines suggest that handcrafted features still perform just as well or even better than recent deep-learned features on such tasks. However, such benchmarks were conducted before methods like SP, KSP and DOAP were published.

Since the applications of computer vision are diverse and the associated data is to a large extent unpredictable, we would argue that learning detectors and descriptors is preferable to manually designing them. By taking such a learning approach, future computer vision algorithms will best cope with different tasks, imaging modalities and environments.

References

Alaa E. Abdel-Hakim and Aly A. Farag. CSIFT: A SIFT descriptor with color invariant characteristics. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast retina keypoint. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Pablo F. Alcantarilla and Adrien Nuevo, Jesús Bartoli. Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *BMVA British Machine Vision Conference (BMVC)*, 2013.

Pablo F. Alcantarilla, Adrien Bartoli, and Andrew J. Davison. KAZE features. In *European Conference on Computer Vision (ECCV)*, 2012.

Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Boris Babenko, Piotr Dollár, and Serge Belongie. Task specific local region matching. In *IEEE International Conference on Computer Vision (ICCV)*, 2007.

Vassileios Balntas, Lilian Tang, and Krystian Mikolajczyk. BOLD - binary online learned descriptor for efficient image matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Vassileios Balntas, Edward Johns, Lilian Tang, and Krystian Mikolajczyk. PN-Net: Conjoined triple deep network for learning local image descriptors. *CoRR*, arXiv:1601.05030, 2016a.

Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *BMVA British Machine Vision Conference (BMVC)*, 2016b.

Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Vassileios Balntas, Lilian Tang, and Krystian Mikolajczyk. Binary online learned descriptors. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 40(3):555–567, 2018.

Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, 2006.

Anthony Bell and Terrence Sejnowski. Edges are the independent components of natural scenes. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 1997.

Irving Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 2(29):115–147, 1987.

Matthew Brown, Richard Szeliski, and Simon Winder. Multi-image matching using multi-scale oriented patches. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

Matthew Brown, Gang Hua, and Simon Winder. Discriminative learning of local image descriptors. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(1):43–57, 2010.

Andrei Bursuc, Giorgos Tolias, and Hervé Jégou. Kernel local descriptors with implicit rotation matching. In *ACM International Conference on Multimedia Retrieval (ICMR)*, 2015.

Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European Conference on Computer Vision (ECCV)*, 2010.

Hong Cheng, Zicheng Liu, Nanning Zheng, and Jie Yang. A deformable local image descriptor. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

Yukyung Choi and In So Kweon. An approach for local feature evaluation. In *IEEE International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 382–383, 2015.

Yukyung Choi, Chaehoon Park, Joon-Young Lee, and In So Kweon. Robust binary feature using the intensity order. In *Asian Conference on Computer Vision (ACCV)*, 2014.

Christopher B. Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2016.

Kai Cordes, Bodo Rosenhahn, and Jörn Ostermann. High-resolution feature evaluation benchmark. In *International Conference on Computer Analysis of Images and Patterns*, 2013.

Jason J. Corso and Gregory D. Hager. Coherent regions for concise and stable image description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

David Crandall, Andrew Owens, Noah Snavely, and Dan Huttenlocher. SfM with MRFs: Discrete-continuous optimization for large-scale structure from motion. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(12):2841–2853, 2013.

Gabriela Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *ECCV Workshop on Statistical learning in computer vision (SLCV)*, 2004.

Li Deng, Dong Yu, et al. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014.

Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *CVPR Workshop on Deep Learning for Visual SLAM*, 2018.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning (ICML)*, 2014.

Jingming Dong and Stefano Soatto. Domain-size pooling in local descriptors: DSP-SIFT. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Yueqi Duan, Jiwei Lu, Ziwei Wang, Jianjiang Feng, and Jie Zhou. Learning deep binary descriptor with multi-quantization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017a.

Yueqi Duan, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Context-aware local binary feature learning for face recognition. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 40(5):1139–1153, 2017b.

Yueqi Duan, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Learning rotation-invariant local binary descriptor. *Transactions on Image Processing*, 26(8):3636–3651, 2017c.

Bin Fan, Qingqun Kong, Tomasz Trzcinski, Zhiheng Wang, Chunhong Pan, and Pascal Fua. Receptive fields selection for binary feature description. *Transactions on Image Processing*, 23(6):2583–2595, 2014a.

Bin Fan, Zhenhua Wang, and Fuchao Wu. *Local Image Descriptor: Modern Approaches*, volume SpringerBriefs in Computer Science. Springer, 2014b.

Philipp Fischer, Alexey Dosovitskiy, and Thomas Brox. Descriptor matching with convolutional neural networks: a comparison to SIFT. *CoRR*, arXiv:405.5769, 2014.

Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(8):1362–1376, 2010.

Yongqiang Gao, Weilin Huang, and Yu Qiao. Local multi-grouped binary descriptor with ring-based pooling configuration and optimization. *Transactions on Image Processing*, 24(12):4820–4833, 2015.

Vincent Garcia, Éric Debreuve, Frank Nielsen, and Michel Barlaud. K-nearest neighbor search: Fast GPU-based implementations and application to high-dimensional feature matching. In *International Conference on Image Processing (ICIP)*, 2010.

Katarzyna Gebal, Jakob Andreas Bærentzen, Henrik Aanæs, and Rasmus Larsen. Shape analysis using the auto diffusion function. *Computer Graphics Forum*, 28(5):1405–1413, 2009.

Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *International Conference on Very Large Data Bases (VLDB)*, 1999.

Yunchao Gong and Svetlana Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*. MIT Press, 2016.

Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C. Berg. MatchNet: Unifying feature and metric learning for patch-based matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1988.

Wilfried Hartmann, Michal Havlena, and Konrad Schindler. Predicting matchability. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Tal Hassner, Shay Filosof, Viki Mayzels, and Lihi Zelnik-Manor. SIFTing through scales. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 39(7):1431–1442, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Kun He, Fatih Cakir, Sarah Adel Bargal, and Stan Sclaroff. Hashing as tie-aware learning to rank. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018a.

Kun He, Yan Lu, and Scan Sclaroff. Local descriptors optimized for average precision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018b.

Jared Heinly, Enrique Dunn, and Jan-Michael Frahm. Comparative evaluation of binary features. In *European Conference on Computer Vision (ECCV)*, 2012.

Seunghoon Hong, Hyeonwoo Noh, and Bohyung Han. Decoupled deep neural network for semi-supervised semantic segmentation. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

Michael Jahrer, Michael Grabner, and Horst Bischof. Learned local descriptors for recognition and matching. In *Computer Vision Winter Workshop*, 2008.

Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *CoRR*, arXiv:1408.5093, 2014.

Timor Kadir, Andrew Zisserman, and Michael Brady. An affine invariant salient region detector. In *European Conference on Computer Vision (ECCV)*, 2004.

Yan Ke and Rahul Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

Daniel Kondermann, Rahul Nair, Katrin Honauer, Karsten Krispin, Jonas Andrulis, Alexander Brock, Güssefeld Burkhard, Mohsen Rahimimoghaddam, Sabine Hofmann, Claus Brenner, and B. Jähne. The HCI benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2016.

Scott Krig. *Computer Vision Metrics: Survey, Taxonomy, and Analysis*. Springer, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep Convolutional Neural Networks. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2012.

Brian Kulis and Kristen Grauman. Kernelized locality-sensitive hashing for scalable image search. In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

Vijay Kumar B. G., Gustavo Carneiro, and Ian Reid. Learning local image descriptors with deep siamese and triplet convolutional networks by minimizing global loss functions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

Zhen Lei, Matti Pietikaïnen, and Stan Z. Li. Learning discriminant face descriptor. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 36(2):289–302, 2014.

Karel Lenc and Andrea Vedaldi. Learning covariant feature detectors. *CoRR*, arXiv:1605.01224, 2016.

Karel Lenc and Andrea Vedaldi. Large scale evaluation of local image feature detectors on homography datasets. In *European Conference on Computer Vision (ECCV)*, 2018.

Vincent Lepetit and Pascal Fua. Keypoint recognition using randomized trees. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(9):1465–1479, 2006.

Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

Gil Levi and Tal Hassner. LATCH: learned arrangements of three patch codes. *CoRR*, abs:1501.03719, 2015.

Yunpeng Li, Noah Snavely, and Dan Huttenlocher. Location recognition using prioritized feature matching. In *European Conference on Computer Vision (ECCV)*, 2014.

Kevin Lin, Jiwen Lu, Chu-Song Chen, and Jie Zhou. Learning compact binary descriptors with unsupervised deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear CNN models for fine-grained visual recognition. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

Song-Tyang Liu and Wen-Hsiang Tsai. Moment preserving corner detection. *Pattern Recognition*, 23(5): 441–460, 1990.

Jonathan L. Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2014.

David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

Jiwen Lu, Xiuzhuang Liong, Venice Erin Zhou, and Jie Zhou. Learning compact binary face descriptor for face recognition. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 37(10):2041–2056, 2015.

Jiwen Lu, Venice Erin Liong, and Jie Zhou. Simultaneous local binary feature learning and encoding for homogeneous and heterogeneous face recognition. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 40(8):1979–1993, 2018.

Josef Maier, Martin Humenberger, Oliver Zendel, and Markus Vincze. Ground truth accuracy and performance of the matching pipeline. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017.

Yu A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *CoRR*, arXiv:1603.09320, 2016.

Jiří Matas, Ondřej Chum, Martin Urban, and Tomas Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. In *BMVA British Machine Vision Conference (BMVC)*, 2002.

Frank McSherry and Marc Najork. Computing information retrieval performance measures efficiently in the presence of tied scores. In *European Conference on Information Retrieval Research*, 2008.

Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(10):1615–1630, 2003.

Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiří Matas, Timor Schaffalitzky, Frederik Kadir, and Luc Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1/2):43–72, 2005.

Anastasiya Mishchuk, Dmytro Mishkin, Filip Radenović, and Jiří Matas. Working hard to know your neighbors margins: Local descriptor learning loss. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2017.

Dmytro Mishkin, Michal Perdoch, and Jiří Matas. MODS fast and robust method for two-view matching. *CoRR*, arXiv:1503.02619, 2014.

Dmytro Mishkin, Jiří Matas, Michal Perdoch, and Karel Lenc. WxBS: Wide baseline stereo generalizations. In *BMVA British Machine Vision Conference (BMVC)*, 2015.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT Press, 2012.

Menze Moritz and Geiger Andreas. Object scene flow for autonomous vehicles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Hyeonwoo Noh, André Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

Ives R. Otero and Mauricio Delbracio. Anatomy of the SIFT method. *Image Processing On Line*, 4:370–396, 2014.

Mattis Paulin, Matthijs Douze, Zaid Harchaoui, Julien Mairal, Florent Perronin, and Cordelia Schmid. Local convolutional features with unsupervised training for image retrieval. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

Michal Perd'och, Ondřej Chum, and Jiří Matas. Efficient representation of local geometry for large scale object retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Florent Perronnin and Christopher R. Dance. Fisher kernels on visual vocabularies for image categorization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

James Philbin, Ondřej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

James Philbin, Michael Isard, Josef Sivic, and Andrew Zisserman. Descriptor learning for efficient retrieval. In *European Conference on Computer Vision (ECCV)*, 2010.

Matti Pietikäinen, Abdenour Hadid, Guoying Zhao, and Timo Ahonen. *Computer Vision Using Binary Patterns*, volume 40. Springer CVIS Series, 2011.

Azriel Rosenfeld and Avinash Kak. *Digital Picture Processing*. Academic Press, 1982.

Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision (ECCV)*, 2006.

Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

Jordi Sanchez-Riera, Jonas Östlund, Pascal Fua, and Francesc Moreno-Noguer. Simultaneous pose, correspondence and non-rigid shape. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-networks: unsupervised learning to rank for interest point detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Johannes L. Schönberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Nicu Sebe, Qi Tian, Etienne Loupias, Michael S. Lew, and Thomas S. Huang. Evaluation of salient point techniques. *Image and Vision Computing*, 21(13-14):1087–1095, 2003.

Gregory Shakhnarovich, Paul Viola, and Trevor Darrell. Fast pose estimation with parameter sensitive hashing. In *IEEE International Conference on Computer Vision (ICCV)*, 2003.

Edgar Simo-Serra, Carme Torras, and Francesc Moreno-Noguer. DaLI: Deformation and light invariant descriptor. *International Journal of Computer Vision*, 115(2):136–154, 2015a.

Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *IEEE International Conference on Computer Vision (ICCV)*, 2015b.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Learning local feature descriptors using convex optimisation. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 36(8):1573–1585, 2014.

Josef Sivic, and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *IEEE International Conference on Computer Vision (ICCV)*, 2003.

Stephen M. Smith and J. Michael Brady. SUSAN a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.

Christoph Strecha, Wolfgang von Hansen, Luc Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

Christoph Strecha, Albrecht Lindner, Karim Ali, and Pascal Fua. Training for task specific keypoint detection. In *Annual Symposium of the German Association for Pattern Recognition (DAGM)*, 2009.

Christoph Strecha, Bronstein Alex, Michael Bronstein, and Pascal Fua. LDAHash: Improved matching with smaller descriptors. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(1):66–78, 2012.

Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. 28(5):1383–1392, 2009.

Xun Sun, Yuanfan Xie, Pei Luo, and Liang Wang. A dataset for benchmarking image-based localization. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Yurun Tian, Bin Fan, and Fuchao Wu. L2-Net: Deep learning of discriminative patch descriptor in euclidean space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(5):815–830, 2010.

Frederico Tombari and Luigi Di Stefano. Interest points via maximal self-dissimilarities. In *Asian Conference on Computer Vision (ACCV)*, 2014.

Lorenzo Torresani, Vladimir Kolmogorov, and Carsten Rother. Feature correspondence via graph matching: Models and global optimization. In *European Conference on Computer Vision (ECCV)*, 2008.

Eduard Trulls, Iasonas Kokkinos, Alberto Sanfeliu, and Francesc Moreno-Noguer. Dense segmentation-aware descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

Tomasz Trzcinski and Vincent Lepetit. Efficient discriminative projections for compact binary descriptors. In *European Conference on Computer Vision (ECCV)*, 2012.

Tomasz Trzcinski, Mario Christoudias, and Vincent Lepetit. Learning image descriptors with boosting. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 37(3):597–610, 2015.

Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends ® Computer Graphics and Vision*, 3(3):177–280, 2007.

Evgenia Ustinova and Victor Lempitsky. Learning deep embeddings with histogram loss. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2016.

Yannick Verdie, Kwang Moo Yi, Pascal Fua, and Vincent LePetit. TILDE: A temporally invariant learned detector. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Vasillios Vonikakis, Dimitrios Chrysostomou, Rigas Kouskouridas, and Antonios Gasteratos. A biologically inspired scale-space for illumination invariant feature detection. *Measurement Science and Technology*, 24(7), 2013.

Han Wang and Michael Brady. Real-time corner detection algorithm for motion estimation. *Image and Vision Computing*, 13(9):695–703, 1995.

Zhenhua Wang, Bin Fan, and Fuchao Wu. Local intensity order pattern for feature description. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

Zhenhua Wang, Bin Fan, and Fuchao Wu. Affine subspace representation for feature description. In *European Conference on Computer Vision (ECCV)*, 2014.

Zhenhua Wang, Bin Fan, Gang Wang, and Fuchao Wu. Exploring local and overall ordinal information for robust feature description. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 38(11): 2198–2211, 2016.

Xing Wei, Yue Zhang, Yihong Gong, and Nanning Zheng. Kernelized subspace pooling for deep local descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

Yair Weiss, Antonio Torralba, and Robert Fergus. Spectral hashing. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2008.

Kyle Wilson and Noah Snavely. Robust global translations with 1DSfM. In *European Conference on Computer Vision (ECCV)*, 2014.

Matthew Winder, Simon or Brown. Learning local image descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

Liu Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, 2010.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning (ICML)*, pages 2048–2057, 2015.

Gehua Yang, Charles V. Stewart, Michal Sofka, and Chia-Ling Tsai. Registration of challenging image pairs: Initialization, estimation, and decision. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(11):1973–1989, 2007.

Tsun-Yi Yang, Jo-Han Hsu, Yen-Yu Lin, and Yung-Yu Chuang. DeepCD: Learning deep complementary descriptors for patch representations. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

Kwang Moo Yi, Eduard Trulls, Vincent V. Lepetit, and Pascal Fua. LIFT: Learned invariant feature transform. In *European Conference on Computer Vision (ECCV)*, 2016a.

Kwang Moo Yi, Yannick Verdie, Pascal Fua, and Vincent LePetit. Learning to assign orientations to feature points. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016b.

Ramin Zabih and John Woodfill. Nonparametric local transforms for computing visual correspondence. In *European Conference on Computer Vision (ECCV)*, 1994.

Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Xu Zang, Felix X. Yu, Svebor Karaman, and Shih-Fu Chang. Learning discriminative and transformation covariant local feature detectors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 2016.

Xu Zhang, Felix X. Yu, Sanjiv Kumar, and Shih-Fu Chang. Learning spread-out local feature descriptors. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

C. Lawrence Zitnick and Krishnan Ramnath. Edge foci interest points. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

Oscar A. Zuniga and Robert M. Haralick. Corner detection using the facet model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1983.