Neurocomputing 71 (2008) 1771–1787

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A comprehensive review of current local features for computer vision

Jing Li *, Nigel M. Allinson

Vision and Information Engineering Research Group, Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, S1 3JD, UK

ARTICLE INFO

Available online 29 February 2008

Keywords:
Local features
Feature detectors
Corner detectors
Region detectors
Feature descriptors
Filter-based descriptors
Distribution-based descriptors
Textons
Derivative-based descriptors

ABSTRACT

Local features are widely utilized in a large number of applications, e.g., object categorization, image retrieval, robust matching, and robot localization. In this review, we focus on detectors and local descriptors. Both earlier corner detectors, e.g., Harris corner detector, and later region detectors, e.g., Harris affine region detector, are described in brief. Most kinds of descriptors are described and summarized in a comprehensive way. Five types of descriptors are included, which are filter-based descriptors, distribution-based descriptors, textons, derivative-based descriptors and others. Finally, the matching methods and different applications with respect to the local features are also mentioned. The objective of this review is to provide a brief introduction for new researchers to the local feature research field, so that they can follow an appropriate methodology according to their specific requirements.

1. Introduction

Nowadays, local features (local descriptors), which are distinctive and yet invariant to many kinds of geometric and photometric transformations, have been gaining more and more attention because of their promising performance. They are being applied widely in computer vision research, for such tasks as image retrieval [41,63,73,74,78], image registration [5], object recognition [9,10,30–32,42,50,72], object categorization [20,21,39,46,60,71], texture classification [29,37,40,81,82], robot localization [65], wide baseline matching [22,45,56,61,75,78], and video shot retrieval [67,68].

There are two different ways of utilizing local features in applications: (i) traditional utilization, which involves the following three steps: feature detection, feature description, and feature matching; (ii) bag-of-features [55] and hyperfeatures [1], which include the following four steps: feature detection, feature description, feature clustering, and frequency histogram construction for image representation. The focus of this review is on local features. A local feature consists of a feature detector and a feature descriptor.

Feature detectors can be traced back to the Moravec's corner detector [52], which looks for the local maximum of minimum intensity changes. As pointed by Harris and Stephens [31], the response of this detector is anisotropic, noisy, and sensitive to edges. To reduce these shortcomings, the Harris corner detector [28] was developed. However, it fails to deal with scale changes, which always occur in images. Therefore, the construction of detectors that can cope with this scaling problem is important. Lowe [42] pioneered a scale invariant local feature, namely the scale invariant feature transform (SIFT). It consists of a detector and a descriptor. The SIFT's detector finds the local maximums of a series of difference of Gaussian (DoG) images. Mikolajczyk and Schmid [47] developed the Harris–Laplace detector by combining: (i) the Harris corner detector and (ii) the Laplace function for characteristic scale selection, for scale invariant feature detection. To deal with the viewpoint changes, Mikolajczyk and Schmid [47] put forward the Harris (Hessian) affine detector, which incorporates: the Harris corner detector (the Hessian point detector), scale selection, and second moment matrix based elliptical shape estimation. Tuytelaars and Van Gool developed an edge-based region detector [78], which considers both curved and straight edges to construct parallelograms associated with the Harris corner points. They also proposed an intensity-based detector [78], which starts from the local extrema of intensity and constructs ellipse-like regions with a number of rays emitted from these extrema. Both the edge- and intensity-based methods preserve the affine invariance. Matas et al. [45] developed the maximally stable extremal region (MSER) detector, which is a watershed-like method. The last but not the least affine invariant detector is the salient region detector [33], which locates regions based on an entropy function. The detailed discussions of these methods are given in Section 2.

* Corresponding author.
E-mail addresses: elq06jl@sheffield.ac.uk (J. Li), n.allinson@sheffield.ac.uk (N.M. Allinson).

0925-2312/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2007.11.032

To represent points and regions, which are detected by the above methods, a large number of different local descriptors have been developed. The earliest local descriptor could be the local derivatives [36]. Florack et al. [23] incorporated a number of local derivatives and constructed the differential invariants, which are rotational invariant, for local feature representation. Schmid and Mohr [63] extended local derivatives as the local grayvalue invariants for image retrieval. Freeman and Adelson [25] proposed steerable filters, which are linear combinations of a number of basis filters, for orientation and scale selection to handle tasks in image processing and computer vision research. Marcelja [44] and Daugman [17,18] modeled the responses of the mammalian visual cortex through a series of Gabor functions [17,18], because these functions can suitably represent the receptive field profiles in cortical simple cells. Therefore, Gabor filters can be applied for local feature description. Wavelets, which are effective and efficient for multiresolution analysis, can also represent local features. Textons, e.g., 2D textons, 3D textons [40], and the Varma–Zisserman model [81], have been demonstrated to have good performance for texture classification. A texton dictionary is constructed from a number of textures and a clustering algorithm is applied to select a small number of models to represent each texture. Texton representation is also a good choice for local feature modeling. Van Gool et al. [80] computed them for moments up to second order and second degree based on the derivatives of x and y directions of an image patch. Past research has shown the effectiveness of the SIFT descriptor, which is a 3D histogram for gradient magnitudes and orientations representation. This feature is invariant to changes in partial illumination, background clutter, occlusion, and transformations in terms of rotation and scaling. Schaffalitzky and Zisserman [61] introduced complex filters for generating kernel functions for efficient multi-view matching. Shape context, developed by Belongie et al. [9], describes the distribution of the rest points of a shape with respect to a reference point on the shape for matching. Based on the phase and amplitude of steerable bandpass filters, Carneiro and Jepson [12] proposed phase-based local features, which improve invariance to illumination changes. The spin image, originally developed by Johnson and Hebert [32] for 3D object recognition, has been applied to texture modeling by Lazebnik et al. [37]. Ke and Sukthankar [34] simplified the SIFT descriptor by utilizing principal component analysis (PCA) to normalized gradient patches to achieve fast matching and invariance to image deformations. This method is named as PCA-SIFT. Lazebnik et al. [37] put forward the rotation invariant feature transform (RIFT), which divides each circular normalized patch into concentric rings, each of which is associated with a gradient orientation histogram. A recent study reports the significance of the gradient location and orientation histogram (GLOH), proposed by Mikolajczyk and Schmid [48], which is an extension of the SIFT descriptor. Similar to the PCA-SIFT, GLOH also applies PCA to reduce the dimension of the descriptor. All the preceding descriptors do not consider color information, which can be important in visual representations. To this end, Van De Weijer and Schmid [79] developed four color descriptors, which are histograms of RGB, hue, opponent angle, and spherical angle. Preliminary experiments have demonstrated the effectiveness of these descriptors.

Feature matching is an important step to measure the similarity or dissimilarity between two images, which are represented by two sets of local features, where a similarity metric is constructed based on the correspondences of the local features. In most applications, the following three matching methods are applied: (i) threshold-based matching, (ii) nearest neighbor matching, and (iii) nearest neighbor distance ratio matching. Threshold-based matching finds all possible candidate points in other image for each point in the reference image, in case that the distance between the descriptors of the candidate point and the reference point is below a specified threshold. Nearest neighbor matching algorithms find the point with the closest descriptor to a reference point. Nearest neighbor distance ratio matching utilizes the ratio between the distance of the nearest and the second-nearest neighbors for a reference point. Using which form of matching method depends on a specific application. If a simple and fast strategy is required, the threshold-based matching is often the best choice; if an accurate and effective algorithm is a prerequisition, the nearest neighbor distance ratio matching has distinct advantages. For detector [49,53,64] and descriptor [48,53] performance evaluations, almost all existing works are based on one of these matching methods.

Existing detector performance evaluations [49,53] have demonstrated that: (i) under viewpoint changes, MSER outperform others for both structured and textured images; (ii) under scale changes, Hessian affine detector achieves the best results for both structured and textured images; (iii) under image blur, MSER performs poorly for structured flat images but generally reliable for textured images; others work similarly for both; (iv) under JPEG compression for structured images, Hessian and Harris affine perform best; (v) under illumination changes, all detectors perform well and MSER works best; and (vi) Hessian affine and DoG consistently outperform others for 3D objects.

Previous descriptor evaluations [48,53] revealed that: (i) under rotation changes, all descriptors have a similar performance for structured images. GLOH, SIFT and shape context obtain the best results for textured images although all descriptors do not perform well for this kind of images; (ii) under viewpoint changes, the results are better for textured images than for structured image: the GLOH descriptor perform best for structured images and SIFT obtains the best results for textured images; (iii) under scale changes, SIFT and GLOH obtain the best results for both textured and structured images; (iv) under image blur, the performances of all descriptors are degraded significantly. GLOH and PCA-SIFT obtain the best results; (v) for JPEG compression of structured images, when a high false positive rate is allowable, SIFT works best. Otherwise, PCA-SIFT is the best choice; and (vi) under illumination changes, GLOH performs best for illumination normalized regions. In summary, GLOH obtains the best results, closely followed by SIFT, which is always regarded as state of the art and has been demonstrated to achieve the most stable and efficient results.

The organization of this paper is as following. Detectors are reviewed in Section 2; local descriptors are described in Section 3; and Section 4 concludes.

2. Feature detection

Feature detection is the requisite step in obtaining local feature descriptions. It locates points and regions, and is generally capable of reproducing similar levels of performances to human observers in locating elemental features in a wide range of image types. Most of the existing detectors can be categorized into two types: (i) corner detectors and (ii) region detectors.

2.1. Definitions

If an image is defined as a function $I(\vec{p})$, where the domain of I is the set of locations $\vec{p} = [x, y]^{\mathrm{T}}$ for pixels. The derivative of $I(\vec{p})$ with respect to x is given by $I_x(\vec{p})$ or I_x. The derivative of $I(\vec{p})$ with respect to y is given by $I_y(\vec{p})$ or I_y. A Gaussian kernel with a local

scale parameter σ is defined as $g(\vec{p};\sigma) = (1/2\pi\sigma)\exp(-(\vec{p}^{\mathrm{T}}\vec{p}/2\sigma))$. We introduce the following definitions relevant to the detectors in this paper.

Definition 1 (*Scale Space*). Scale space representation is to characterize an image at different scales. It is universally applied for feature detection and image matching. For a given image $I(\vec{p})$, the corresponding linear scale space representation is a series of responses $L(\vec{p};\sigma)$, which is obtained by convoluting $I(\vec{p})$ with $g(\vec{p};\sigma)$, i.e.,

$$L(\vec{p};\sigma) = \int I(\vec{p}-\vec{q})g(\vec{q};\sigma)\,\mathrm{d}\vec{q} = I(\vec{p})g(\vec{p};\sigma). \tag{1}$$

Definition 2 (*Harris Matrix, Second Moment Matrix, or Structure Tensor*). The Harris matrix, a matrix of partial derivatives, is typically utilized to represent the gradient information in the area of image processing and computer vision research. It was defined in Harris corner detector and the eigenvalues of this matrix determine whether or not a point is a corner. For a given image $I(\vec{p})$, the Harris matrix is defined by

$$A = \nabla I \otimes \nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix}\begin{bmatrix} I_x \\ I_y \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} I_x^2 & I_xI_y \\ I_yI_x & I_y^2 \end{bmatrix}, \tag{2}$$

where $\nabla = \vec{i}(\partial/\partial x) + \vec{j}(\partial/\partial y)$ and $\otimes$ is the outer product.

Definition 3 (*Hessian Matrix*). The Hessian matrix of a given image $I(\vec{p})$ is the matrix of second partial derivatives with respect to $\vec{p}$, i.e.,

$$H = \nabla\nabla^{\mathrm{T}} I = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix}. \tag{3}$$

Definition 4 (*Laplacian Function*). The Laplacian operator is invariant to rotation in a given image $I(\vec{p})$. It is related to scale space representation and can be defined with respect to $\vec{p}$ by

$$\Delta L = \nabla^2 L = (\nabla\cdot\nabla)L = L_{xx} + L_{yy}. \tag{4}$$

It is commonly used for blob detection and usually results in strong positive responses for dark blobs and strong negative responses for bright blobs with similar sizes.

Definition 5 (*Multi-scale Second Moment Matrix*). The multi-scale second moment matrix of a given image $I(\vec{p})$ can be defined as

$$\begin{aligned} \Gamma(\vec{p};\sigma_D,\sigma_I) &= \int \begin{bmatrix} L_x^2(\vec{p}-\vec{q};\sigma_D) & L_x(\vec{p}-\vec{q};\sigma_D)L_y(\vec{p}-\vec{q};\sigma_D) \\ L_y(\vec{p}-\vec{q};\sigma_D)L_x(\vec{p}-\vec{q};\sigma_D) & L_y^2(\vec{p}-\vec{q};\sigma_D) \end{bmatrix} g(\vec{q};\sigma_I)\,\mathrm{d}\vec{q} \\ &= \begin{bmatrix} L_x^2(\vec{p};\sigma_D) & L_x(\vec{p};\sigma_D)L_y(\vec{p};\sigma_D) \\ L_y(\vec{p};\sigma_D)L_x(\vec{p};\sigma_D) & L_y^2(\vec{p};\sigma_D) \end{bmatrix} g(\vec{p};\sigma_I), \end{aligned} \tag{5}$$

where L_x and L_y denote the partial derivatives of $L(\vec{p};\sigma_D)$ with respect to x and y in scale σ_D (differentiation scale), respectively; and the function $g(\vec{q};\sigma_I)$ is a Gaussian kernel with scale σ_I (integration scale). Obviously, σ_D is for scale space smoothing before computing the image derivatives and σ_I is for accumulating non-linear operations on derivative operators onto an integrated image descriptor.

Definition 6 (*Geometric Transformations*). Geometric transformations consist of translation, reflection, rotation, skew, scale, glide—all of which maintain the previous shape; while stretch, shear, enlargement and other topological transformations change the original shape. In a geometric transformation, each pixel at location $\vec{p}_1$ in a reference image is mapped to its corresponding location $\vec{p}_2$ in the target image. Different geometric transformations can be expressed by the same function with different transformation matrix A and translation vector $\vec{b}$, i.e.,

$$\vec{p}_2 = A\vec{p}_1 + \vec{b}. \tag{6}$$

A translation can be achieved by setting a specific $\vec{b}$, while scaling, rotation and reflection can be accomplished by changing A. They results in an affine transformation by performing a combination of a series of geometric operations above simultaneously.

Definition 7 (*Photometric Transformations*). Photometric transformations operate on the values of image pixels, e.g., illumination changes. Take the RGB color space as an example, a photometric transformation is given by

$$\begin{pmatrix} r_2 \\ g_2 \\ b_2 \end{pmatrix} = \begin{bmatrix} \alpha_r & 0 & 0 \\ 0 & \alpha_g & 0 \\ 0 & 0 & \alpha_b \end{bmatrix}\begin{pmatrix} r_1 \\ g_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} \beta_r \\ \beta_g \\ \beta_b \end{pmatrix}, \tag{7}$$

where r, g, b are three different color bands in RGB color space, α and β are scalar illumination parameters.

Definition 8 (*Non-maximum Suppression*). Non-maximum suppression is important for many computer vision tasks, e.g., edge detection and corner detection. An image is scanned along its gradient direction, which should be perpendicular to an edge. Any pixel that is not a local maximum is suppressed and set to zero. As shown in Fig. 1, p and r are the two neighbors along the gradient direction of q. If the pixel value of q is not larger than the pixel values of both p and r, it is suppressed.

2.2. Corner detectors

In this subsection, we review the following operators: (i) the Moravec's corner detector [52], (ii) the Harris corner detector [28], (iii) the smallest univalue segment assimilating nucleus (SUSAN) [69], (iv) the Trajkovic Operator [77], and (v) the high-speed corner detector [59], which are all key corner detectors (Fig. 2).

2.2.1. Moravec's corner detector

Moravec's corner detector [52], one of the earliest corner detectors, was developed in 1977. It seeks the local maximum of the minimum intensity changes by shifting a binary rectangle window over an image. In detail, it operates on each pixel in the image to find whether the pixel is on a corner by considering the similarity between two patches centered at nearby pixels. The detection criteria are as following: (i) if the pixel is in a uniform intensity region, the nearby patches are similar in terms of the sum of squared differences (SSD); (ii) if the pixel is on an edge, the patches belonging to the same side of an edge are similar while patches from different sides are dissimilar; and (iii) if the pixel is on a corner, the patches along the corner are dissimilar from each other. In summary, this operator defines a pixel as on a corner where there is a large intensity variation in every direction from the pixel. In Ref. [52], pixels of interest, which were defined as distinct regions in an image, were applied to image matching.

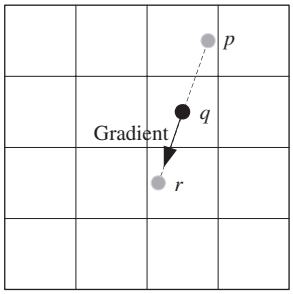


Fig. 1. The non-maximum suppression.

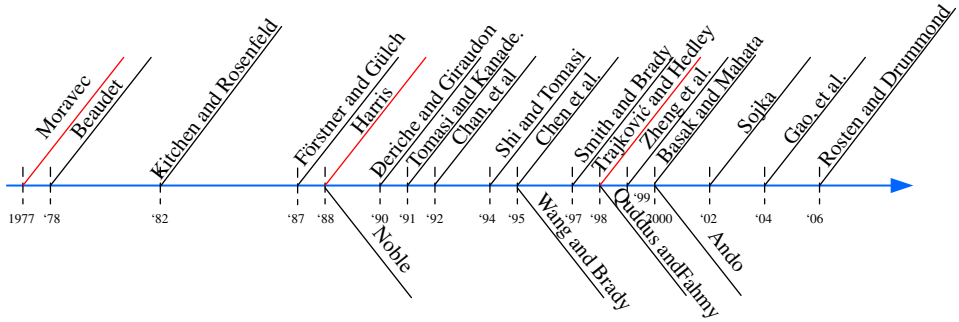


Fig. 2. The timeline of corner detectors (http://www.cim.mcgill.ca/~dparks/CornerDetector/background.htm). Milestones are the Moravec's corner detector [56], the Harris corner detector [31] and the Trajkovic operator [81].

According to Refs. [28,52], the shortcomings of this detector are: (i) the response is anisotropic due to the consideration of shifts only at every 45°; (ii) the response is noisy due to the use of a binary rectangular window; and (iii) the detector responds to edges easily due to the consideration of only the minimum intensity changes.

2.2.2. Harris corner detector

Harris corner detector, also called the Plessey corner detector, was proposed by Harris and Stephens [28] to reduce the weaknesses of the Moravec's corner detector, by: (i) perform an analytic expansion about the shift origin to involve all possible small shifts; (ii) substitute a smooth circular window for the binary rectangular one; and (iii) redevelop the corner response metric to take into account the intensity changes with the direction of shifts, respectively. It can determine whether a pixel is a corner, a point is on an edge, or a point is in a uniform intensity region. This detector first computes the Harris matrix A for each pixel in an image and then computes its eigenvalues λ_1 and λ_2, which indicate the principal curvature of A. The following three rules are applied for justification: (i) if $\lambda_1 \approx 0$ and $\lambda_2 \approx 0$, the pixel is in a uniform intensity region; ii) if $\lambda_1 \approx 0$ and $\lambda_2 \gg 0$, the pixel is a point on an edge; and iii) if λ_1 and λ_2 are both large, a corner is indicated. To reduce the computational complexity, the Harris corner metric [28] $m_h = \det(A) - \kappa \operatorname{tr}^2(A)$ replaces the eigenvalue computation for justification: (i) if m_k is small, the pixel is in a uniform intensity region; (ii) if $m_k < 0$, the pixel is on an edge; and (iii) if $m_k > 0$, the pixel is a corner. The detected Harris points are invariant to rotation. This detector finds locations with large gradient in all directions at a predefined scale, including corners. This detector fails to satisfactorily deal with scaling.

2.2.3. SUSAN

Smith and Brady [69] considered that pixels in a relatively small region are uniform in terms of brightness if these pixels belong to the same object. Based on this point of view, the SUSAN is implemented by comparing brightness within a circular mask C. In detail, the brightness of each pixel within the mask, $\vec{p} \in C$, is compared with that of the nucleus $\vec{p}_0$ (the center of C) by a comparison function:

$$m(\vec{p}) = \exp(-(t^{-1}(I(\vec{p}) - I(\vec{p}_0)))^6), \tag{8}$$

where t is a threshold. A nucleus $\vec{p}_0$ is a corner, if the number of pixels, which are similar to $\vec{p}_0$ in terms of brightness, is less than a given threshold. The SUSAN detector has good reliability. It locates corners accurately and is excellently insensitive to noise. It is relatively fast. However, it performs poorly for blurred images.

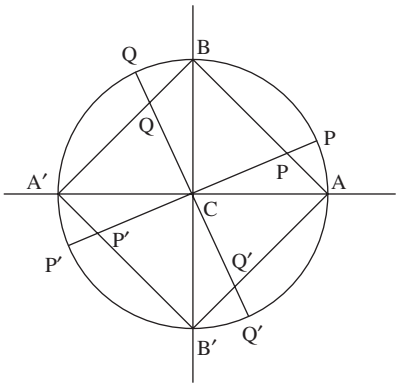


Fig. 3. The neighborhood of nucleus C (the rectangular $ABA'B'$) with linear and circular interpixel positions (P, P', Q and Q'). The figure comes from Ref. [81].

2.2.4. Trajkovic operator

To reduce computational complexity, a fast corner detector is developed by Trajković and Hedley [77]. The chief concept is to investigate the variations of intensity along arbitrary lines passing through a point within a neighborhood of the point, as shown in Fig. 3. Horizontal and vertical intensity variations are computed as

$$\begin{aligned} r_A &= (f_A - f_C)^2 + (f_{A'} - f_C)^2 \\ r_B &= (f_B - f_C)^2 + (f_{B'} - f_C)^2, \end{aligned} \tag{9}$$

where $f_A, f_{A'}, f_B, f_{B'}$ and f_C are pixel values at the locations A, A', B, B' and C, respectively. Then, the corner metric is computed as $R_0 = \min(r_A, r_B)$. If R is below a predefined threshold, the nucleus C is not a corner. Otherwise, the linear- or circular-based interpixel approximation is utilized to check for diagonal edges.

For linear interpixel approximation, define $Y = \min(Y_1, Y_2)$ and $X = r_B - r_A - 2Y$, where $Y_1 = (f_B - f_A)(f_A - f_C) + (f_{B'} - f_{A'})(f_{A'} - f_C)$, $Y_2 = (f_B - f_{A'})(f_{A'} - f_C) + (f_{B'} - f_A)(f_A - f_C)$ and $Z = r_A$, the minimum corner response is $R_l = Z - Y^2/X$ with the constraints $Y < 0$ and $X + Y > 0$; for circular interpixel approximation, define $X = (r_A - r_B)/2$ and $Y = \min(Y_1, Y_2)$, where $Y_1 = (f_A - f_C) \cdot (f_B - f_C) + (f_{A'} - f_C) \cdot (f_{B'} - f_C)$ and $Y_2 = (f_{A'} - f_C) \cdot (f_B - f_C) + (f_A - f_C) \cdot (f_{B'} - f_C)$, the maximum corner response is $R_c = Z - \sqrt{X^2 + Y^2}$ with the constraint $B < 0$.

The fast corner detection is a three-step algorithm:

(1) the corner measure R_0 is computed at every pixel in a low resolution version of the image. Each pixel with its response higher than a predefined threshold T_1 is considered as a "potential corner";

(2) using the full resolution image, for each potential corner:
   (i) compute R_0. If the response is higher than a second threshold T_2, perform (ii); otherwise the pixel is not a corner point and no further computation is needed;
   (ii) use R_l (R_c) to compute a new response. If the response is higher than T_2, then conduct (3); otherwise, the pixel is not a corner point.
(3) pixels with a locally maximal R_l (R_c) are selected as corners, which is called non-maximum suppression.

Because the number of corners is usually a small fraction of the image pixels, it is not meaningful to apply R_l (R_c) to each pixel in the image due to most low responses. Therefore, by firstly using R_0 for each pixel, the fast corner detector largely reduces the computational complexity and thus is very fast. A high response, R_l (R_c) verifies the existence of a corner.

2.2.5. High-speed corner detector

Rosten and Drummond [59] utilized machine learning to speed up the corner detection. The process includes the following three stages:

1. segment test on a Bresenham circle of a center pixel p with radius 3: this step is computationally efficient to remove most of non-corner candidates. It is based on a logistical test, i.e., p is not a corner if a pixel with position x and another pixel with position x+8 are similar to p in terms of intensity. The test will be conducted on 12 consecutive positions;
2. classification-based corner detection: apply ID3 tree classifier [58] to determine whether p is a corner based on 16 features. Each feature is 0, 1, or -1. If a pixel with position x on the Bresenham circle of p is larger (smaller) than p, the corresponding feature is 1 (-1). Otherwise, the feature is 0; and
3. corner verification: the non-maximum suppression is utilized for verification.

2.2.6. Others detectors in the timeline

The Beaudet corner detector [7] is a rotationally invariant operator based on a corner measure, i.e., the determinant of the Hessian matrix. This operator is sensitive to noise because of the computation of second derivatives. Kitchen and Rosenfeld [35] detect corners at the local maximum of a corner measure based on the change of gradient direction along an edge weighted by the gradient magnitude. Similar to the Harris corner detector, the Förstner operator [24] also utilizes a corner measure defined by the second moment matrix. The threshold is determined by local statistics. Nobel [54] modified the Harris corner measure as

$$m_n = (\mathrm{tr}(A) + \varepsilon)^{-1} \det(A). \tag{10}$$

Moreover, the differential geometry of a facet model was applied to represent 2D surface features, including corners. Deriche and Giraudon [19] generalized the Beaudet operator for two scales with the zero crossing of the Laplacian image used to obtain an accurate position of a corner. By analyzing the optical flow equation proposed by Lucas and Kanade [43], Tomasi and Kanade [76] obtained the relationship:

$$\begin{bmatrix} \sum wI_x^2 & \sum wI_xI_y \\ \sum wI_xI_y & \sum wI_x^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} \sum wI_xI_t \\ \sum wI_yI_y \end{bmatrix}. \tag{11}$$

Here, an image is described by $I(x, y, t)$, where x and y are the space variables, t is the time variable and w is a weighting function. Corners are chosen as image locations where the condition is satisfied. Shi and Tomasi [66] proposed a new corner detection algorithm that is strongly based on the Harris corner detector, arguing thatmin(λ_1, λ_2) is a better choice than $m_h = \det(A) - \kappa\,\mathrm{tr}^2(A)$. Chan et al. [13] introduced wavelets for corner detection because wavelets provide multi-resolution (multi-scale) analysis. Corners are detected by a given threshold on a high–high component of an input image, decomposed by a B-spline wavelet in several scales. Chen et al. [14] replaced the B-spline wavelet with the Gaussian function based wavelet for corner detection. Quddus and Fabmy [57] modified Gabor wavelet for corner detection. Recently, Gao et al. [26] applied a different corner measure for Gabor wavelet based corner detection. Wang and Brady [83] designed a corner detector for motion estimation, with an image regarded as a surface and the corner metric $m_w = \nabla^2 I - \alpha|\nabla I|^2$ applied to detect corners. To reduce the computational complexity of the Harris corner detector, Zheng et al. [84] proposed a gradient-direction corner detector based on a simplified corner measure constructed from the Harris corner detector. Basak and Mahata [3] applied neural networks to detect corners. Ando [2] detected corners and edges based on gradient covariance matrix and gradient projection. Sojka [70] applied a measurement function, where the weighting coefficients are computed based on the Bayes' theorem, to justify the variance of the gradient directions.

Both the Harris corner detector and the SUSAN algorithm are reliable, but the latter greatly outperforms the former on localization accuracy and speed. According to Ref. [51], the Kitchen and Rosenfeld detector is the fastest compared with the Harris corner detector and the SUSAN. However, it is worth emphasizing that almost all the corner detectors discussed above can deal with affine transformations.

2.3. Region detectors

In this subsection, we review the following region detectors: (i) the Harris–Laplace region detector [47], (ii) the Harris affine and Hessian affine corner detector [47], (iii) the edge-based region detector [78], (iv) the intensity-based region detector [78], (v) the MSER detector [45] and (vi) the salient region detector [33] as important examples of recently developed region detectors.

2.3.1. Harris–Laplace

Harris–Laplace [47] region detector locates potentially relevant points, *interest points*, with the Harris corner detector and then selects the point with a characteristic scale, which is the extremum of the Laplacian over different scales. To describe the Harris–Laplace detector, we need to define the multi-scale Harris corner measure and the scale-normalized Laplacian operator.

Multi-scale Harris corner measure is given by

$$m_l(\vec{p}; \sigma_D, \sigma_I) = \det(\Gamma(\vec{p}; \sigma_D, \sigma_I)) - \kappa\,\mathrm{tr}^2(\Gamma(\vec{p}; \sigma_D, \sigma_I)), \tag{12}$$

where $\Gamma(\vec{p}; \sigma_D, \sigma_I)$ is the multi-scale second moment matrix; κ is a predefined scalar; and σ_D and σ_I are local scale parameters.

With a given scale parameter σ, the scale-normalized Laplacian operator is

$$\Delta_N L(\vec{p}; \sigma) = \sigma\,\Delta L(\vec{p}; \sigma) = \sigma(L_{xx}(\vec{p}; \sigma) + L_{yy}(\vec{p}; \sigma)), \tag{13}$$

where L_{xx} and L_{yy} are second partial derivatives with respect to x and y, respectively.

Based on the two definitions above, the Harris–Laplace detector contains two steps:

(1) pixels located at the spatial maximum of $m_l(\vec{p}; \sigma_D, \sigma_I)$ are selected as candidate interest points, i.e.,

$$\vec{p} = \arg\max_{\vec{p}} m_l(\vec{p}; \sigma_D, \sigma_I), \quad \sigma_I = \gamma^2 \sigma_D; \tag{14}$$

(2) a characteristic scale is selected for each candidate interest point at the local extremum over scales of $\Delta_N L(\vec{p};\sigma_D)$, i.e.,

$$\sigma_D = \arg\min_{\sigma_D} \Delta_N L(\vec{p};\sigma_D). \tag{15}$$

2.3.2. Harris affine and Hessian affine

Mikolajczyk and Schmid [47] proposed the Harris affine and Hessian affine detectors. The process is as follows:

(1) detect interest points using the multi-scale Harris corner measure, defined in Eq. (12), or the Hessian matrix based detector;
(2) select the characteristic scale for each interest point. This step is the same as the second step for scale selection in the Harris–Laplace detector;
(3) determine the shape associated with an interest point by the eigenvalues and eigenvectors of the second moment matrix A;
(4) normalize the shape into a circle according to $\vec{p}' = A^{1/2}\vec{p}$; and
(5) conduct steps 2–4 for interest points.

2.3.3. Edge-based region detector

Edge-based region detector [78] is chiefly based chiefly on the following two motivations: (i) edges are stable under affine transformations; (ii) edge-based region detection is more effective than corner-based region detection. Edges are further classified as curved edges or straight edges.

For curved edges, the following steps are applied for region detection:

(1) detect interest points $\vec{p}$ by the Harris corner detector;
(2) extract two nearby edges from $\vec{p}$ to $\vec{p}_1$ and $\vec{p}_2$, respectively, through the Canny edge detector [11], as shown in Fig. 4;
(3) the two points $\vec{p}_1$ and $\vec{p}_2$ move away from $\vec{p}$ with the relative speeds, which are related to the equality of relative affine invariant parameters l_1 and l_2:

$$l_i = \int \left| \det\left(\frac{\mathrm{d}\vec{p}_i(s_i)}{\mathrm{d}s_i} \vec{p} - \vec{p}_i(s_i) \right) \right| \mathrm{d}s_i, \quad i = 1, 2, \tag{16}$$

where s_i is an arbitrary curve parameter. For each value l ($l_1 = l_2$), the two points $\vec{p}_1$ and $\vec{p}_2$ associated with the corner $\vec{p}$ define a region $\Omega(l)$, a parallelogram, which is spanned by the vectors $\vec{p}_1 - \vec{p}$ and $\vec{p}_2 - \vec{p}$; and
(4) the points stop when a specific photometric quantity of the texture covered by $\Omega(l)$ achieves an extremum. The following measures are suggested:

$$f_1(\Omega) = \frac{m^1_{00}}{m^0_{00}}, \tag{17}$$

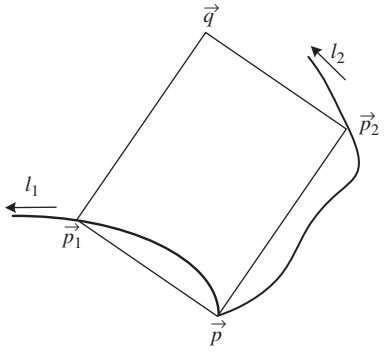


Fig. 4. The edge-based region detector in the case of curved edges. This figure is from Ref. [82].

$$f_2(\Omega) = \left| \frac{\det([\vec{p}_1 - \vec{p}_g \quad \vec{p}_2 - \vec{p}_g])}{\det([\vec{p} - \vec{p}_1 \quad \vec{p} - \vec{p}_2])} \right| \frac{m^1_{00}}{\sqrt{m^2_{00}m^0_{00} - (m^1_{00})^2}}, \tag{18}$$

$$f_3(\Omega) = \left| \frac{\det([\vec{p} - \vec{p}_g \quad \vec{q} - \vec{p}_g])}{\det([\vec{p} - \vec{p}_1 \quad \vec{p} - \vec{p}_2])} \right| \frac{m^1_{00}}{\sqrt{m^2_{00}m^0_{00} - (m^1_{00})^2}}, \tag{19}$$

where $m^n_{pq} = \int_\Omega I^n(x,y)x^p y^q \,\mathrm{d}x\,\mathrm{d}y$, $\vec{p}_g = [\, m^1_{10}/m^1_{00} \quad m^1_{01}/m^1_{00}\,]^{\mathrm{T}}$, and $\vec{q}$ is the corner of the parallelogram opposite to $\vec{p}$, as shown in Fig. 4.

In the case of straight edges, the following steps are applied for region detection:

(1) combine $f_2(\Omega)$ and $f_3(\Omega)$ together; and
(2) take the intersection of the valleys, which are minima of the two functions above, to estimate the parameters s_1 and s_2 along the straight edges.

2.3.4. Intensity-based region detector

The intensity-based region detector [78] is based on intensity extrema detected at multi-scales. The procedure is as following:

(1) the local intensity extrema are detected using non-maximum suppression, as shown in the part (a) of Fig. 5;
(2) given such an extremum, the intensity function along rays emanating from the extremum is

$$f_I(t) = |I(t) - I_0| \left(\max\left(t^{-1} \int_0^t |I(t) - I_0| \,\mathrm{d}t, d \right) \right)^{-1}, \tag{20}$$

where t is the Euclidean arclength along a ray; $I(t)$ is the intensity at position t; I_0 is the intensity extremum; and d is a small number to avoid to be divided by zero. This step is described in the part (b) of Fig. 5;
(3) all points corresponding to the maximum of the intensity function along rays emanating from a same local extremum are linked together to enclose an affine region. Usually, the shape is irregular, as shown in the part (c) of Fig. 5;
(4) an ellipse, which has the same shape moments as the irregular region, is used to describe (replace) it to ensure affine invariance, as shown in the part (d) of Fig. 5; and
(5) the ellipse is enlarged with a ratio 2, as shown in the part (e) of Fig. 5. It is to make the detected region more distinctive and therefore speed up the matching procedure.

2.3.5. Maximally stable extremal region detector

The MSER detector was proposed by Matas et al. [45]. The detection procedure, which is similar to the watershed-based image segmentation, is described by the following steps:

(1) an image is thresholded into a series of binary images, from white to black. The thresholds are in an ascending order;
(2) blobs corresponding to local intensity minimum will appear and merge at some point. All closed components in the image are extremal regions; and
(3) only the extremal regions that remain unchanged over a range of thresholded images are selected as MSERs.

MSERs have the following properties: (i) all intensities in each MSER are either lower (dark extremal region) or higher (bright extremal region) than intensities outside its boundary; and (ii) each MSER is affine invariant for both geometrical and photometrical transformations. It should be borne in mind that the output of the detector is not a binary image.

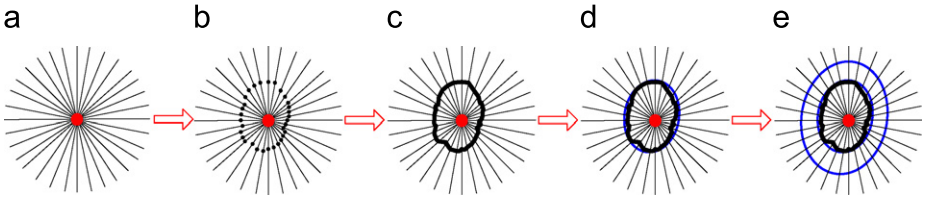


Fig. 5. The intensity-based region detector.

2.3.6. Salient region detector

Kadir and Brady [33] proposed the salient region detector, which is based on the probability density function (pdf) of intensity values $p(I)$ computed over an elliptical region Ω. The detailed procedure is as follows:

(1) at each pixel $\vec{p}$, calculate the entropy of the pdf $p(I)$ over three parameters (s, θ, λ) of an ellipse centered at $\vec{p}$, where s is the scale; θ is the orientation; and λ is the ratio of major axes;
(2) the set of entropy extrema over scales and (s, θ, λ) are recorded as candidate salient regions. The entropy H is

$$H = -\sum_I p(I) \log p(I), \tag{21}$$

(3) the candidate salient regions are ranked according to the magnitude of the derivative of the pdf $p(I; s, \theta, \lambda)$ for each extremum with respect to s, which is described by

$$w = \frac{s^2}{2s-1} \sum_I \left| \frac{\partial p(I; s, \theta, \lambda)}{\partial s} \right|, \tag{22}$$

(4) top P ranked regions are retained by ranking their saliency $y = Hw$ of Ω.

However, in this detector, images have to be manually marked to select the correct correspondences. This step is not practical in most real world applications.

2.3.7. Difference of Gaussian operator and sift detector

DoG method was proposed by Crowley and Parker [15]. It selects the scale-space extrema in a series of DoG images by convoluting an image with DoG functions (with different local scales). The detected points are candidate keypoints.

Lowe [42] extended this operator to deal with scale changes for the SIFT, which is specifically designed for scaling. The extension includes the following three steps: (i) the scale-space extrema detection, (ii) the keypoint localization, and (iii) the orientation assignment. The first step is the same as the DoG operator. In the second step, low contrast candidate points and edge response points along an edge are discarded. This step ensures that the keypoints are more stable for matching and recognition. The final step assigns the dominant orientation to a keypoint. In detail, the following sub-steps are applied to achieve the above objectives.

For a given image $I_0(\vec{p})$, the scale-space extrema detection has the following sub-steps:

(1) to reserve the highest spatial frequencies, i.e., to increase the number of stable keypoints, the linear interpolation is utilized to double the size of a given image: $I_0(\vec{p}) \to I_1(\vec{p})$;
(2) construct the scale space $L(\vec{p}; \sigma_i)$ with a series of $g(\vec{p}; \sigma_i)$, where $1 \leqslant i \leqslant N$ and $N \geqslant 4$. Moreover, $\sigma_i = 2^{(i-1)/(N-3)}\sigma_1$ and σ_1 is a predefined value, e.g., $\sigma_1 = 1.6$;
(3) obtain DoG images $D_j(\vec{p}) = L(\vec{p}; \sigma_{j+1}) - L(\vec{p}; \sigma_j)$, for $1 \leqslant j \leqslant N-1$;
(4) find the local extrema of $D_k(\vec{p})$ for $2 \leqslant k \leqslant N-2$. A point $\vec{q} = [x, y]^{\mathrm{T}}$ in $D_k(\vec{p})$ is a local extremum *if and only if* $D_k(\vec{q})$ is the largest or the smallest among its 26 neighbors, i.e., 8 neighbors in D_k and 9 neighbors in adjacent scales below (D_{k-1}) and above (D_{k+1});
(5) resample $L(\vec{p}; \sigma_i)$, where $\sigma_i = 2\sigma_1$, by taking every second pixel in each row and column. Conduct steps 2–4; and
(6) conduct step 5 many times.

For the keypoints localization, we have the following sub-steps:

(1) reject extrema which have low contrast, thus are sensitive to noise, by calculating

$$\begin{aligned} \vec{z}_e &= \begin{bmatrix} x_e \\ y_e \\ \sigma_e \end{bmatrix} = -\left[\frac{\partial^2 L}{\partial \vec{z}^2}\right]^{-1} \frac{\partial L}{\partial \vec{z}} \\ &= -\begin{bmatrix} \frac{\partial^2 L}{\partial x^2} & \frac{\partial^2 L}{\partial xy} & \frac{\partial^2 L}{\partial x\sigma} \\ \frac{\partial^2 L}{\partial yx} & \frac{\partial^2 L}{\partial y^2} & \frac{\partial^2 L}{\partial y\sigma} \\ \frac{\partial^2 L}{\partial \sigma x} & \frac{\partial^2 L}{\partial \sigma y} & \frac{\partial^2 L}{\partial \sigma^2} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial L}{\partial x} \\ \frac{\partial L}{\partial y} \\ \frac{\partial L}{\partial \sigma} \end{bmatrix}, \end{aligned} \tag{23}$$

and

$$L(\vec{z}_e) = L(\vec{z}) + \frac{1}{2}\frac{\partial^{\mathrm{T}} L}{\partial \vec{z}}\vec{z}_e. \tag{24}$$

The derivatives are approximated by simple differences, e.g., $\partial_\sigma L = (L(\vec{p}; \sigma_{k+1}) - L(\vec{p}; \sigma_{k-1}))/2$. The above calculation is conducted only on local extrema $\vec{z}$ obtained in the scale-space extrema detection, i.e., the first step. This step reduces the extreme point estimation error by replacing a local extreme point $\vec{z}$ with $(\vec{z}_e + \vec{z})/2$, when the distance between $\vec{z}$ and its theoretical estimation $\vec{z}_e$ is larger than 0.5, i.e., $||\vec{z}_e - \vec{z}||_2 > 0.5$. Preserve local extrema if $L(\vec{z}) > 0.03$; and
(2) reject an extreme if the corresponding ratio $r = \lambda_1/\lambda_2$ is larger than 10, where λ_1 and λ_2 are eigenvalues of the Hessian matrix H of this extreme.

The orientation assignment is applied to create keypoints based on each extreme $\vec{z}_m = [x_m, y_m, \sigma_m]^{\mathrm{T}}$ with the following sub-steps:

(1) calculate the magnitude and the orientation of all points in the circular region of an extreme:

$$m(x, y, \sigma) = \sqrt{(L(x+1, y; \sigma) - L(x-1, y; \sigma))^2 + (L(x, y+1; \sigma) - L(x, y-1; \sigma))^2}, \tag{25}$$

$$\theta(x, y, \sigma) = \tan^{-1}\left(\frac{L(x, y+1; \sigma) - L(x, y-1; \sigma)}{L(x+1, y; \sigma) - L(x-1, y; \sigma)}\right); \tag{26}$$

(2) smooth magnitudes with a Gaussian window $(1/2\pi\sigma)$ $\exp(-((x - x_m)^2 + (y - y_m)^2)/2\sigma)$, with the local scale parameter $\sigma = 1.5\sigma_m$;

(3) each magnitude is accumulated to one of 36 predefined bins, $(\pi n)/10$ with $0 \leqslant n \leqslant 35$, if its corresponding orientation belongs to that bin. With this sub-step, a histogram with 36 bins is obtained;
(4) select the following orientations: (i) the orientation to the maximum of the histogram, and (ii) an orientation to a local maximum, whose value is above 80% of the maximum of the histogram;
(5) for each selected orientation, the following procedure is applied for orientation correction: (i) use a parabola to fit the histogram value of the orientation and those of its two neighbors; and (ii) replace the original orientation with the orientation corresponding to the peak position of the parabola; and
(6) create a keypoint $\vec{z}_{kp} = [x_{kp}, y_{kp}, \sigma_{kp}]^{\mathrm{T}}$ in the region, whose orientation is the same as one of the selected orientations. One or many keypoints will be created for a given extreme in this step and each keypoint has an orientation.

The regions detected by region detectors are invariant to more kinds of affine transformations than those detected by corner detectors, which ensures the high performance for image matching. Especially, the MSER, DoG operator, Harris–Laplace and Hessian–Laplace detectors are more suitable for detecting regions that are substantially brighter or darker than their surroundings. Among them, MSER is the optimal one in being able to deal with most varieties of transformations, including scaling, rotation, viewpoint variation, and illumination changes. Furthermore, when dealing with scaling, the SIFT detector, Harris–Laplace, and Hessian-Laplace are better choices than the others.

3. Feature description

Detection is followed by feature description. The simplest descriptor is a vector of pixel values. Ideal descriptors should be not only distinctive, i.e., they should be able to deal with a large number of objects and robust to occlusion and background clutter, but also they need to be invariant to many kinds of variations, both geometric and photometric transformations, as defined in Section 3.1.

A number of descriptors have been proposed in the literature and many of them have been proved to be effective in computer vision research tasks, e.g., robust matching, video tracking, content-based image retrieval, object categorization, image annotation, and robot planning. Existing descriptors can be categorized into the following five types: (i) filter-based descriptors [17,25,61], (ii) distribution-based descriptors [9,34,37,42,48], (iii) textons [40,81,82], (iv) derivative-based descriptors [23,36,63], and (v) others [12,79,80].

3.1. Filter-based descriptors

In this sub-section, we review filter-based descriptors, which are steerable filters [25], Gabor filters [17,18], and complex filters [61].

3.1.1. Steerable filters

The term "steerable filter" [25] describes a set of basis filters, which can synthesize any filter with an arbitrary orientation, i.e., $F^{\theta} = \sum_{i=1}^{N} k_i(\theta) F_i$, where F_i is the ith basis filter; $k_i(\theta)$ is the linear combination coefficient to synthesize the filter F^{θ} with a given orientation θ. A quadrature pair of filters, which means the two filters have identical frequency but one is the Hilbert transform of the other, can be applied to synthesize filters of a given frequency response with arbitrary phase. The derivatives of Gaussian have been demonstrated to be effective in many early vision and image processing tasks. The steerable quadrature filter pairs for the second order (G_2 and H_2), the third order (G_3 and H_3), and the fourth order (G_4 and H_4) of Gaussian are

$$G_2(x,y) = 0.9213(2x^2 - 1)\exp(-(x^2+y^2)), \tag{27}$$

$$H_2(x,y) = (-2.205x + 0.9780x^3)\exp(-(x^2+y^2)), \tag{28}$$

$$G_3(x,y) = (2.472x - 1.648x^3)\exp(-(x^2+y^2)), \tag{29}$$

$$H_3(x,y) = (-0.9454 + 2.959x^2 - 0.6582x^4) \times \exp(-(x^2+y^2)), \tag{30}$$

$$G_4(x,y) = (0.9344 - 3.738x^2 + 1.246x^4) \times \exp(-(x^2+y^2)), \tag{31}$$

$$H_4(x,y) = (2.858x - 2.982x^3 + 3.975x^5)\exp(-(x^2+y^2)), \tag{32}$$

where H_i is the Hilbert transform of G_i, for $i = 2, 3, 4$. Responses of these filters are shown in Fig. 6.

For G_2, it is not difficult to construct the basis filters according to Ref. [25] as

$$G_{2a} = 0.9213(2x^2 - 1)\exp(-(x^2+y^2)), \tag{33}$$

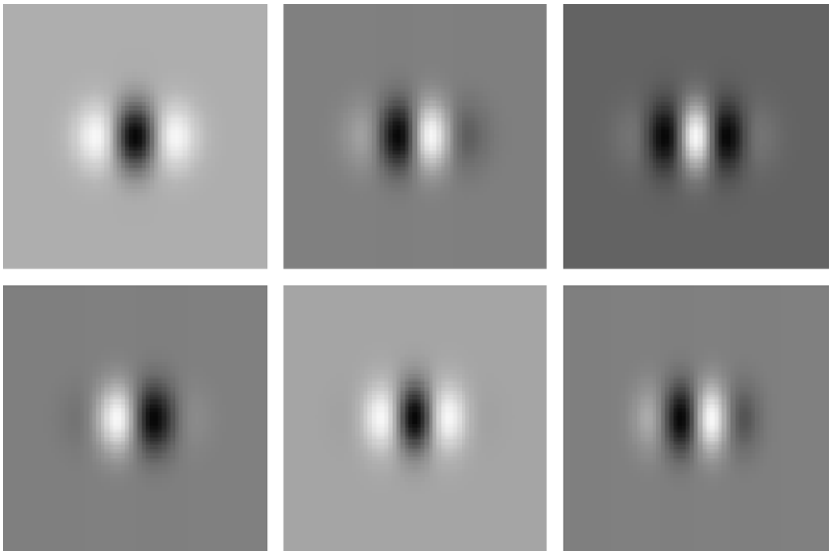


Fig. 6. The upper three sub-figures from left to right are G_2, G_3, and G_4, respectively. The bottom three sub-figures from left to right are H_2, H_3, and H_4, respectively.

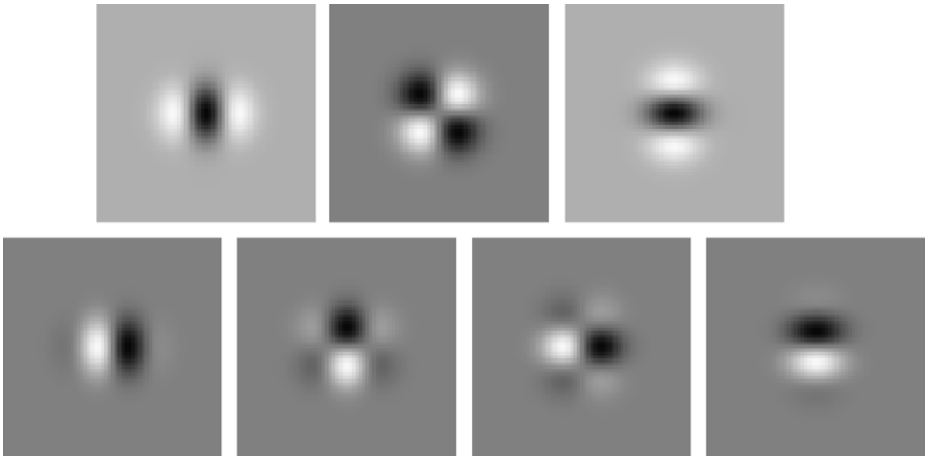


Fig. 7. The upper three sub-figures from left to right are G_{2a}, G_{2b}, and G_{2c}, respectively. The bottom four sub-figures from left to right are H_{2a}, H_{2b}, H_{2c}, and H_{2d}, respectively.

$$G_{2b} = 1.843xy\ \exp(-(x^2+y^2)), \tag{34}$$

$$G_{2c} = 0.9213(2y^2-1)\ \exp(-(x^2+y^2)), \tag{35}$$

and the corresponding linear combination coefficients are

$$k_a(\theta) = \cos^2(\theta), \tag{36}$$

$$k_b(\theta) = -2\ \cos(\theta)\ \sin(\theta), \tag{37}$$

$$k_c(\theta) = \sin^2(\theta). \tag{38}$$

For H_2, the basis filters are

$$H_{2a} = 0.9780(-2.254x + x^3)\ \exp(-(x^2+y^2)), \tag{39}$$

$$H_{2b} = 0.9780(-0.7515 + x^2)y\ \exp(-(x^2+y^2)), \tag{40}$$

$$H_{2c} = 0.9780(-0.7515 + y^2)x\ \exp(-(x^2+y^2)), \tag{41}$$

$$H_{2d} = 0.9780(-2.254y + y^3)\ \exp(-(x^2+y^2)), \tag{42}$$

and the corresponding linear combination coefficients are

$$k_a(\theta) = +\cos^3(\theta), \tag{43}$$

$$k_b(\theta) = -3\cos^2(\theta)\ \sin(\theta), \tag{44}$$

$$k_c(\theta) = +3\ \cos(\theta)\sin^2(\theta), \tag{45}$$

$$k_d(\theta) = -\sin^3(\theta). \tag{46}$$

The responses of G_{2a}, G_{2b}, and G_{2c} are shown in the top three sub-figures in Fig. 7. The responses of H_{2a}, H_{2b}, H_{2c}, and H_{2d} are shown in the bottom four sub-figures in Fig. 7.

3.1.2. Gabor filters

Research findings from cognitive psychology and psychophysics research [44] suggest that Gabor filters based on image decomposition are biologically relevant to human image understanding and recognition. Consequently, Gabor filters have been applied for local feature representation within a pure computer vision context.

Marcelja [44] and Daugman [17,18] modeled the responses of the visual cortex by Gabor filters, as they are similar to the receptive field profiles in the mammalian cortical simple cells. Daugman [17,18] developed the 2D Gabor filters, a set of local spatial bandpass functions, which have good spatial localization, orientation selectivity, and frequency selectivity. Lee [38] gave a good introduction to image representation using Gabor filters. A Gabor filter is the product of an elliptical Gaussian envelope and a complex plane wave, defined as

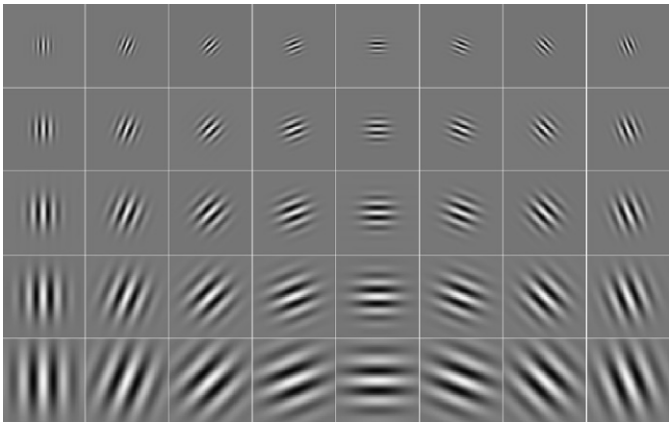


Fig. 8. The real part of Gabor filters in five different scales and eight different directions.

$$\psi_{s,d}(x,y) = \psi_{\vec{k}}(\vec{z}) = \frac{||\vec{k}||}{\delta^2}\cdot\exp\left(-\frac{||\vec{k}||^2\cdot||\vec{z}||^2}{2\delta^2}\right) \times\left[\exp i\vec{k}\cdot\vec{z} - \exp\left(-\frac{\delta^2}{2}\right)\right], \tag{47}$$

where $\vec{z}=[x,y]$ is the variable in a spatial domain and $\vec{k}$ is the frequency vector, which determines the scale and orientation of Gabor filters, $\vec{k} = k_s\, e^{i\phi_d}$, where $k_s = k_{\max}/f^s$, $k_{\max}=\pi/2$, $f=2$, $s=0,1,2,3,4$, and $\phi_d = \pi d/8$, for $d=0,1,2,3,4,5,6,7$. Examples of the real part of Gabor filters are presented in Fig. 8, where Gabor functions (full complex functions) are with five different scales and eight different orientations, making a total of 40 Gabor functions. The number of oscillations under the Gaussian envelope is determined by $\delta = 2\pi$. The term $\exp(-\sigma^2/2)$ is subtracted in order to make the kernel DC–free, and thus insensitive to the average illumination level.

3.1.3. Complex filters

Schaffalitzky and Zisserman [61] applied the following filters for local feature representation,

$$K_{mn} = (x+iy)^m(x-iy)^n G(x,y), \tag{48}$$

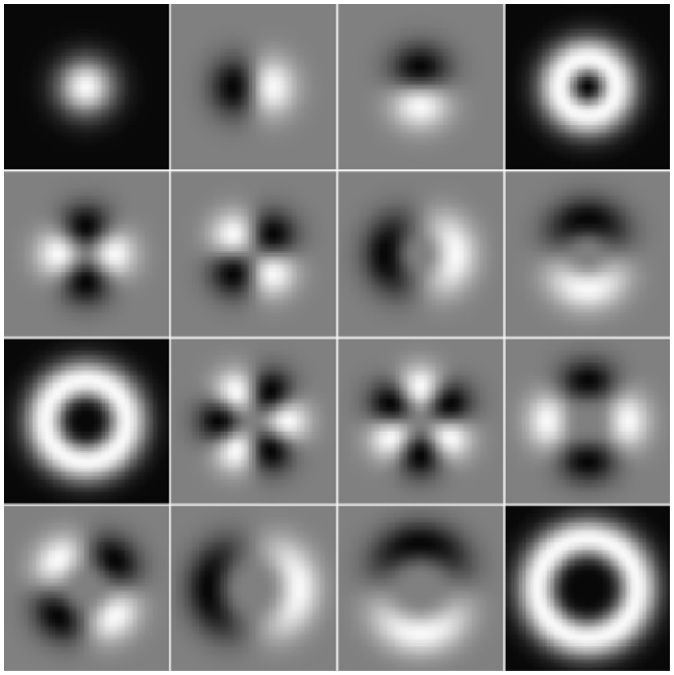


Fig. 9. The 16 responses of complex filters for image decomposition.

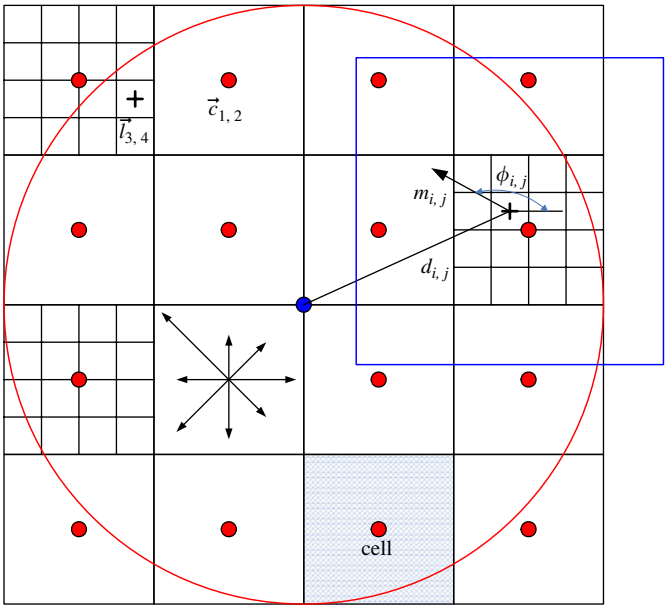


Fig. 10. The 4×4 histogram in SIFT.

where $G(x,y)$ is a Gaussian; $m+n \leqslant 6$ and $m \geqslant n$. There are 16 different filter responses for a given image, as shown in Fig. 9. The responses of these filters: (i) are similar to derivatives of a Gaussian; (ii) are rotationally invariant; and (iii) act only on changing their relative phases but not the magnitudes.

3.2. Distribution-based descriptors

Distribution-based descriptors usually obtain better performance compared with other types of local descriptors. We review the following prevailing ones: the SIFT [42], the PCA-SIFT [34], the shape context [9], the spin image [37], the RIFT [37], and the GLOH [48].

3.2.1. Descriptor of sift

The SIFT descriptor [42] for each keypoint $\vec{z}_{kp}$ (with scale σ_{kp} and orientation θ_{kp}) is a 128 dimensional vector created by first computing the gradient magnitude and orientation in the neighborhood of the keypoint. It contains 16 orientation sub-histograms, and each consists of 8 bins. In detail, we have the following steps (Fig. 10):

(1) generate the center $\vec{c}_{i,j} = [-6+4(i-1), -6+4(j-1)]^{\mathrm{T}}$, where $1 \leqslant i,j \leqslant 4$, for each cell;
(2) generate the location $\vec{l}_{i,j} = [-8.5+i, -8.5+j]^{\mathrm{T}}$, where $1 \leqslant i,j \leqslant 16$, for each point;
(3) generate the orientation bin $\varphi_i = (i-5)\pi/4$, where $1 \leqslant i \leqslant 8$;
(4) calculate the rotation matrix

$$R = \begin{bmatrix} \cos(\theta_{kp}) & -\sin(\theta_{kp}) \\ \sin(\theta_{kp}) & \cos(\theta_{kp}) \end{bmatrix};$$

(5) rotate and translate centers and locations according to $\vec{c}_{i,j} \leftarrow R\vec{c}_{i,j} + \vec{z}_{kp}$ and $\vec{l}_{i,j} \leftarrow R\vec{l}_{i,j} + \vec{z}_{kp}$;
(6) the gradient magnitude $m_{i,j}$ and the orientation $\phi_{i,j}$ of $\vec{l}_{i,j}$ are sampled around the keypoint $\vec{z}_{kp}$ on scale σ_{kp};
(7) compute the x-coordinate weighting vector $\vec{w}_{i,j}^{(x)} \in R^{16}$ of the location $\vec{l}_{i,j}$ according to: $\vec{w}_{i,j}^{(x)} = [\max(1 - |\vec{c}_{m,n}(1) - \vec{l}_{i,j}(1)|/4, 0)]_{1 \leqslant m,n \leqslant 4}$, where $\vec{c}_{m,n}(1)$ is the x-coordinate of $\vec{c}_{m,n}$ and $\vec{l}_{i,j}(1)$ is the x-coordinate of $\vec{l}_{i,j}$. Compute $\vec{w}_{i,j}^{(y)} = [\max(1 - |\vec{c}_{m,n}(2) - \vec{l}_{i,j}(2)|/4, 0)]_{1 \leqslant m,n \leqslant 4}$. Construct the location weighting vector $\vec{w}_{i,j}^{(l)} = [\vec{w}_{i,j}^{(x)}(m,n) \times \vec{w}_{i,j}^{(y)}(m,n)]_{1 \leqslant m,n \leqslant 4}$. Repeat $\vec{w}_{i,j}^{(l)} \in R^{16}$ 8 times to obtain $\vec{w}_{i,j}^{(L)} \in R^{128}$ because we have 8 orientations for each cell;
(8) compute the orientation weighting vector $\vec{w}_{i,j}^{(O)} \in R^8$: (i) $\vec{\vartheta} = mod(\phi_{i,j} - \theta_{kp} - \vec{\varphi} + \pi, 2\pi) - \pi$, where $\vec{\varphi} = [\varphi_i]_{1 \leqslant i \leqslant 8}$ and $\vec{\vartheta} \in R^8$; (ii) $\vec{w}_{i,j}^{(o)} = \max(1 - 4|\vec{\vartheta}|/\pi, 0)$; and (iii) repeat $\vec{w}_{i,j}^{(o)} \in R^8$ 16 times to obtain $\vec{w}_{i,j}^{(O)} \in R^{128}$ because we have 16 cells for each orientation;
(9) the 128 dimensional histogram is $\vec{h}_{kp} = \sum_{i,j=1}^{16} [\vec{w}_{i,j}^{(L)}(k) \cdot \vec{w}_{i,j}^{(O)}(k)]_{1 \leqslant k \leqslant 128} w_{i,j}^{(G)} m_{i,j}$, where $w_{i,j}^{(G)}$ is a Gaussian weighting factor. It is obtained by $w_{i,j}^{(G)} = \exp(-d_{i,j}^2/(2 \times 8^2))/(2\pi \times 8^2)$, where $d_{i,j}$ is the Euclidean distance between $\vec{z}_{kp}$ and $\vec{l}_{i,j}$; and
(10) normalization: (i) $\vec{h}_{kp} = \vec{h}_{kp} / \sum_{k=1}^{128} \vec{h}_{kp}(k)$; (ii) $\vec{h}_{kp} \leftarrow \min(\vec{h}_{kp}, 0.2)$; and (iii) $\vec{h}_{kp} = \vec{h}_{kp} / \sum_{k=1}^{128} \vec{h}_{kp}(k)$;
(11) The calculated $\vec{h}_{kp} \in R^{128}$ is the descriptor of the keypoint $\vec{z}_{kp}$ with scale σ_{kp} and orientation θ_{kp}.

3.2.2. PCA-SIFT

Ke and Sukthankar [34] proposed the PCA based SIFT (PCA-SIFT), which performs more efficiently than the SIFT descriptor for matching. The SIFT detector is applied here to detect keypoints. The PCA-SIFT descriptor is constructed with the following steps:

(1) a 41×41 patch centered at each keypoint is extracted at a given scale and N image patches are collected;
(2) rotate each image patch according to its dominant orientation to a canonical direction;
(3) for each image patch, the gradient maps computed both on orthogonal and vertical directions are combined together to generate a feature vector, which has $39 \times 39 \times 2$ elements;

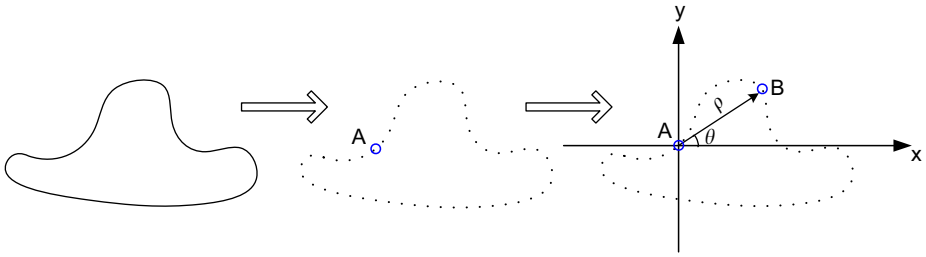


Fig. 11. The representation of a reference point '*A*' with respect to another arbitrary point '*B*' on the edge.

(4) each feature vector is normalized to unit length to minimize the effects of illumination changes; and
(5) PCA is performed to reduce the feature dimension from 3042 to 20 based on the collected image patches.

Although the PCA-SIFT is simpler and faster than the SIFT, it loses some discriminative information during the process of dimension reduction.

3.2.3. Shape context

Shape context, a robust and simple algorithm to find correspondences between shapes, is a 3D histogram of edge point locations and orientations introduced by Belongie et al. [9]. Location is quantized into 9 bins using a log-polar coordinate system as displayed with the radius set to 6, 11 and 15 and orientation is quantized into 4 bins (horizontal, vertical and two diagonals), resulting in a 36 dimensional vector. It is similar to the SIFT descriptor, but is based on edges extracted by the Canny detector. The procedure to obtain shape contexts is as following:

(1) extract the shape contours of an image by an edge detector;
(2) sample N points from each shape contour to represent the shape, each sampling point is considered as a reference point;
(3) for each reference point, construct a coordinate with the reference point as its origin;
(4) compare every reference point, e.g., the point A, with each of the other $N-1$ points, e.g., the point B, in the form of ρ and θ as shown in Fig. 11, which results in a $N-1$ dimensional vector for each reference point;
(5) express each vector in the polar coordinate as the function of $\log \rho$ and θ, where 5 bins are used for $\log \rho$ and 12 bins are used for θ to construct a two-dimensional histogram, as shown in Fig. 12; and
(6) count the number of points with the same $\log \rho$ and θ falling into the same bin.

In summary, the shape context describes the distribution of the rest points of the shape with respect to the reference point on the shape. Therefore, finding correspondences between shapes is equivalent to finding the point that has the most similar shape context on the other shape. In Ref. [9], shape contexts were applied for shape matching and object recognition.

3.2.4. Spin image

Spin image was first used in 3D object recognition by Johnson and Hebert [32], and later applied for planar images classification by Lazebnik et al. [37], where it is also called intensity-domain spin image, as it is a two-dimensional histogram representing the distribution of intensity in the neighborhood of a center point, whose two dimensions are d and i, where d is the distance from the center point and i is the intensity value. The procedure is as following:

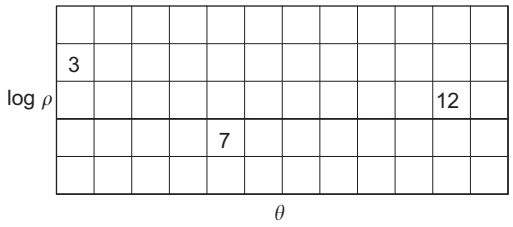


Fig. 12. The shape context histogram.

(1) extract a sparse set of affine-covariant elliptical regions from a texture image using the Harris affine or Laplacian blob detectors, which detect complementary types of structures;
(2) normalize each elliptical region into a unit circle to reduce the affine ambiguity to a rotational one;
(3) before computing the spin images, slightly blur the normalized patches using a Gaussian kernel to reduce the potential sensitivity of the normalization to noise and resampling;
(4) quantize intensities of each normalized patch into 10 bins; and
(5) compute a 10 bin normalized histogram for each of 10 rings centered on the region, resulting in a 10×10 dimensional descriptor.

To achieve invariance to affine transformation of the image intensity function, i.e., $I \rightarrow aI+b$, the intensity function range is normalized within the support region of the spin image (Fig. 13).

3.2.5. Rotation invariant feature transform

The RIFT was also proposed by Lazebnik et al. [37]. The major steps are as follows:

(1) extract a sparse set of affine-covariant elliptical regions from a texture image using the Harris affine or Laplacian blob detectors, which detect complementary types of structures;
(2) normalize each elliptical region into a unit circle to reduce the affine ambiguity to a rotational one;
(3) divide the circular normalized patch into concentric rings with equal width, where 4 rings are used; and
(4) compute a gradient orientation histogram with 8 orientations within each ring, resulting in a 4×8 dimensional descriptor.

To ensure rotation invariance, this orientation is measured at each point from a high intensity value pointed to a low intensity value, i.e. from a white region to a black region, as illustrated in Fig. 14.

However, the RIFT is sensitive to flip transformations of the normalized patch.

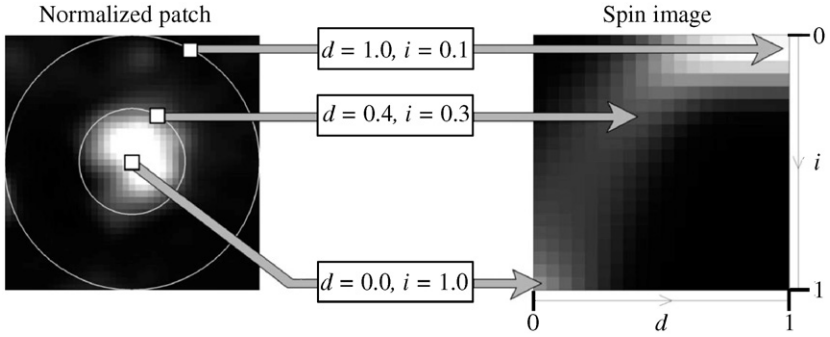


Fig. 13. Spin image, which comes from Ref. [37].

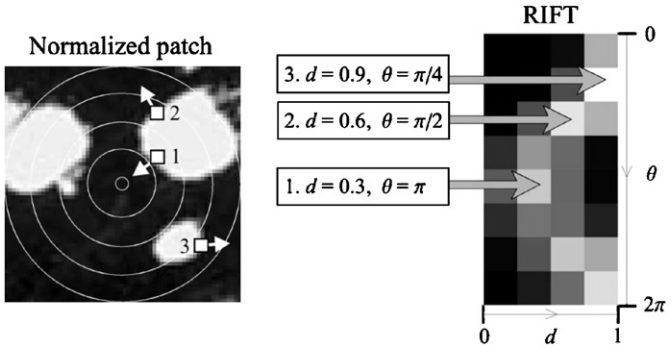


Fig. 14. Rotation invariant feature transform (RIFT), which comes from Ref. [37].

3.2.6. Gradient location and orientation histogram

The GLOH [48] extend the SIFT. Similar to the PCA-SIFT, it applies PCA to reduce the dimension of the feature vector. In the GLOH, the SIFT is computed for 17 locations and 16 orientations bins in a log-polar co-ordinate system, resulting in 17×6 dimensional descriptors, where 17 locations bins are 3 bins for radial direction with 3 different radii and 8 bins for angular direction. PCA is used to alleviate the dimensions to 128, which corresponds to the 128 largest eigenvectors of the covariance matrix obtained from 47,000 image patches collected from a variety of images.

3.3. Textons

Textures can be characterized by their responses to a set of filters [16,40,62,81,82] and this Section reviews texton-based feature representations. Many object categorization algorithms [1,27,37,55] are relevant to texton models. There are usually four steps:

1. local appearance vectors generation: (i) collect n images, each with size $M_i \times N_i$ for $1 \leqslant i \leqslant n$; (ii) select a set of filters, e.g., Gabor filters, as a filter bank; (iii) convolute each image with the filter bank and generate n filtered image sets; and (iv) cluster each image set based on a clustering algorithm, e.g., *K-Means*, into K_i cluster centers $\vec{c}_{i,j}$ in R^L for $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant K_i$. These cluster centers are termed as *appearance vectors*. This procedure is shown in Fig. 15;
2. global appearance vectors clustering or dictionary generation: (i) collect all cluster centers $\vec{c}_{i,j}$; (ii) clustering these $\sum_{i=1}^{n} K_i$ centers to K centers $\vec{a}_i \in R^L$ with $1 \leqslant i \leqslant K$; and iii) set $\vec{a}_i \in R^L$ as initial cluster centers and apply a clustering algorithm to update these centers based on all vectors in the filtered image sets. The final K centers $\vec{a}_i \in R^L$ with $1 \leqslant i \leqslant K$ are global appearance vectors or the dictionary. The procedure is illustrated in Fig. 16;
3. pseudo-image generation for representation: (i) generate a filtered image set for a given image based on the filter bank; and (ii) assign each vector a class label by comparing it with the global appearance vectors or dictionary according to the nearest neighbor rule. This procedure is shown in Fig. 17; and
4. texton generation: (i) stack each filter to a vector and combine all vectors as a matrix $F \in R^{(L_1 \times L_2) \times L}$; (ii) calculate the pseudo-inverse of F and multiply it with appearance vectors as texton vectors $\vec{t}_i \in R^{(L_1 \times L_2)}$ with $1 \leqslant i \leqslant K$; and (iii) unstack each texton vector $\vec{t}_i$ to texton $T_i \in R^{L_1 \times L_2}$, a matrix. The procedure is shown in Fig. 18.

Different types of textons have been developed for feature characterization. The key issue for the texton construction is the filter bank selection. For example, Leung and Malik [40] selected 48 filters: first and second derivatives of Gaussians with 6 orientations and 3 scales, 8 Laplacian of Gaussian filters, and 4 Gaussians. They are shown in Fig. 19(a); Schmid [62] selected 13 rotationally invariant filters $F(r,\sigma,\tau) = F_0(\sigma,\tau) + \cos(\pi\tau r/\sigma)\exp(-r^2/(2\sigma^2))$, as shown in Fig. 19(c); and Varma and Zisserman [81,82] developed the maximum response filter bank based on the root filter set, as shown in Fig. 19(b). This filter bank consists of 38 filters: a Gaussian filter, a Laplacian of Gaussian filter, an edge filter with 6 orientations and 3 scales, and a bar filter with 6 orientations and 3 scales. Different from other methods, Varma and Zisserman selected only one response from 6 orientations for both edge and bar filters in each scale, i.e., 8 responses are selected to characterize features.

3.4. Derivative-based descriptors

Local derivatives [36] were explored by Koenderink and Van Doorn. Florack et al. [23] incorporated a number of local derivatives and constructed the differential invariants, which are rotational invariant, for local feature representation.

One of the most popular derivative-based descriptors is the local grayvalue invariants developed by Schmid and Mohr [63], which are a set of differential invariants computed to third-order from the local derivatives. Let I as an image and σ as a given scale.

The local derivative of order N at a point $\vec{x} = (x_1, x_2)$ is $L_{i_1 \cdots i_k \cdots i_n}(\vec{x}, \sigma)$, which is obtained by convoluting I with Gaussian derivatives $G_{i_1 \cdots i_k \cdots i_n}(\vec{x}, \sigma)$, and $i_k \in \{x_1, x_2\}$.

Then the local grayvalue invariants can be described as $F = [f_i]_{1 \leqslant i \leqslant 9}$, where

$$f_1 = L = IG(\vec{x}, \sigma), \tag{49}$$

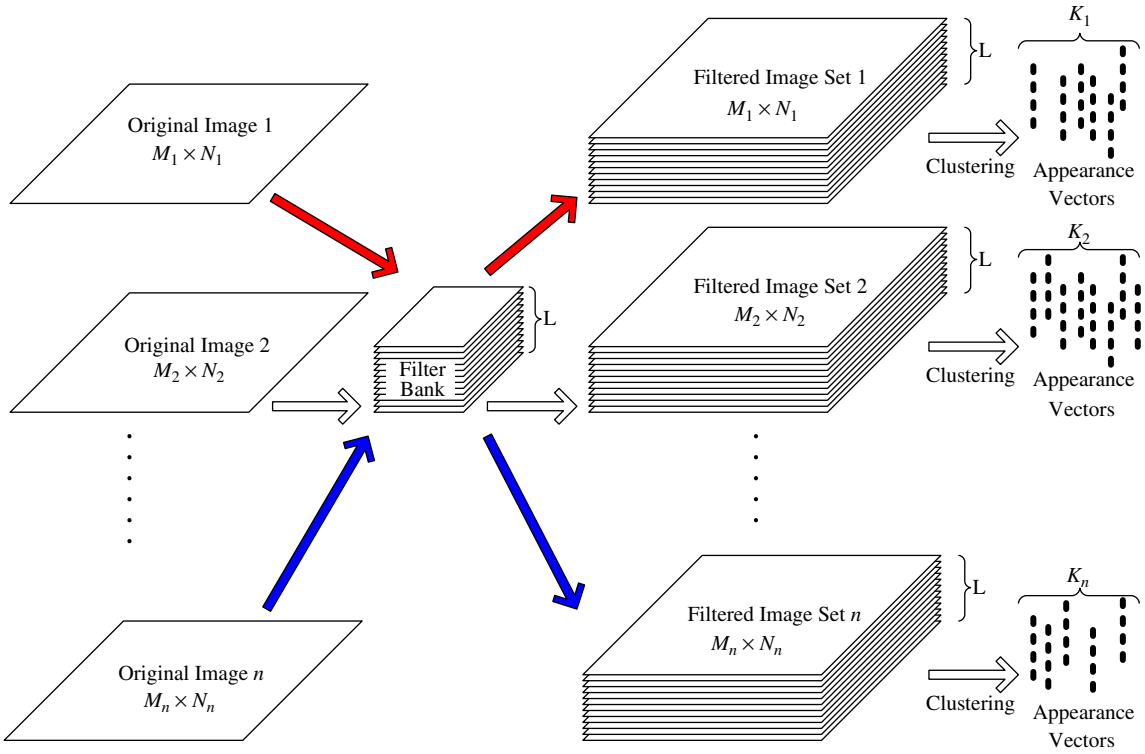


Fig. 15. Local appearance vectors generation.

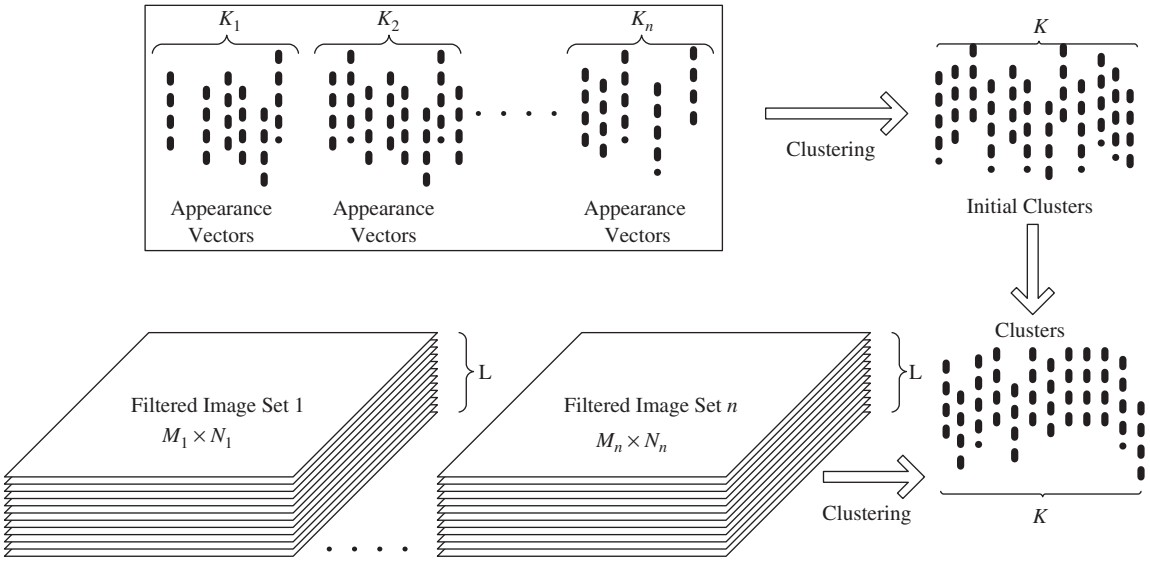


Fig. 16. Dictionary generation.

$$f_2 = L_i L_i = \sum_{i=1}^{2} L_{x_i}^2, \tag{50}$$

$$f_3 = L_i L_{ij} L_j = \sum_{i,j=1}^{2} L_{x_i} L_{x_i x_j} L_{x_j}, \tag{51}$$

$$f_4 = L_{ii} = \sum_{i=1}^{2} L_{x_i x_i}, \tag{52}$$

$$f_5 = L_{ij} L_{ji} = \sum_{i,j=1}^{2} L_{x_i x_j} L_{x_j x_i}, \tag{53}$$

$$f_6 = \varepsilon_{ij}(L_{jkm} L_i L_k L_m - L_{jkk} L_i L_m L_m) = \sum_{i,j,k,m=1}^{2} \varepsilon_{ij}(L_{x_j x_k x_m} L_{x_i} L_{x_k} L_{x_m} - L_{x_j x_k x_k} L_{x_i} L_{x_m} L_{x_m}), \tag{54}$$

$$f_7 = L_{iij} L_j L_k L_k - L_{ijk} L_i L_j L_k = \sum_{i,j,k=1}^{2} (L_{x_i x_i x_j} L_{x_j} L_{x_k} L_{x_k} - L_{x_i x_j x_k} L_{x_i} L_{x_j} L_{x_k}), \tag{55}$$

$$f_8 = -\varepsilon_{ij} L_{jkm} L_i L_k L_m = -\sum_{i,j,k,m=1}^{2} \varepsilon_{ij} L_{x_j x_k x_m} L_{x_i} L_{x_k} L_{x_m}, \tag{56}$$

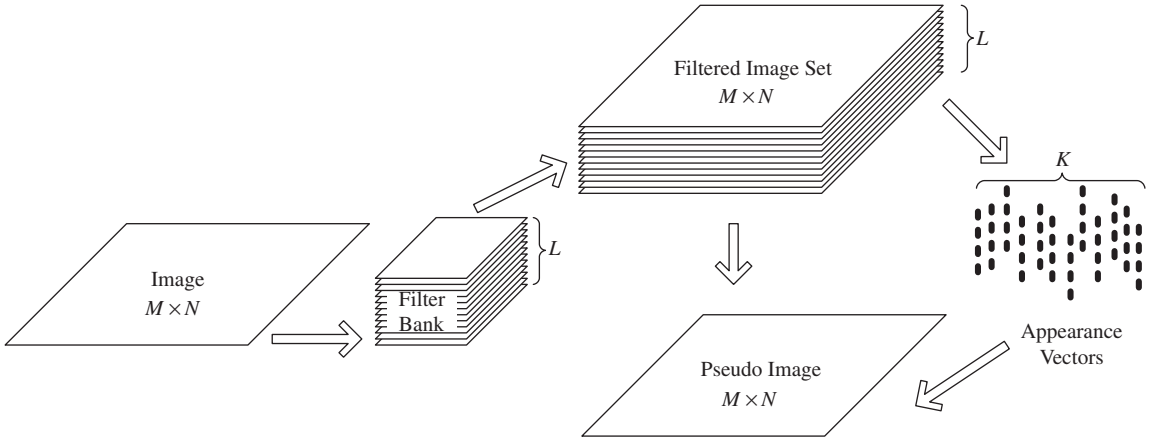


Fig. 17. Pseudo image generation for representation.

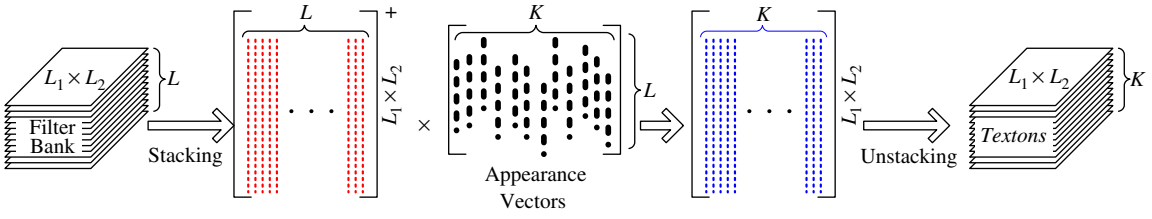


Fig. 18. Texton generation.

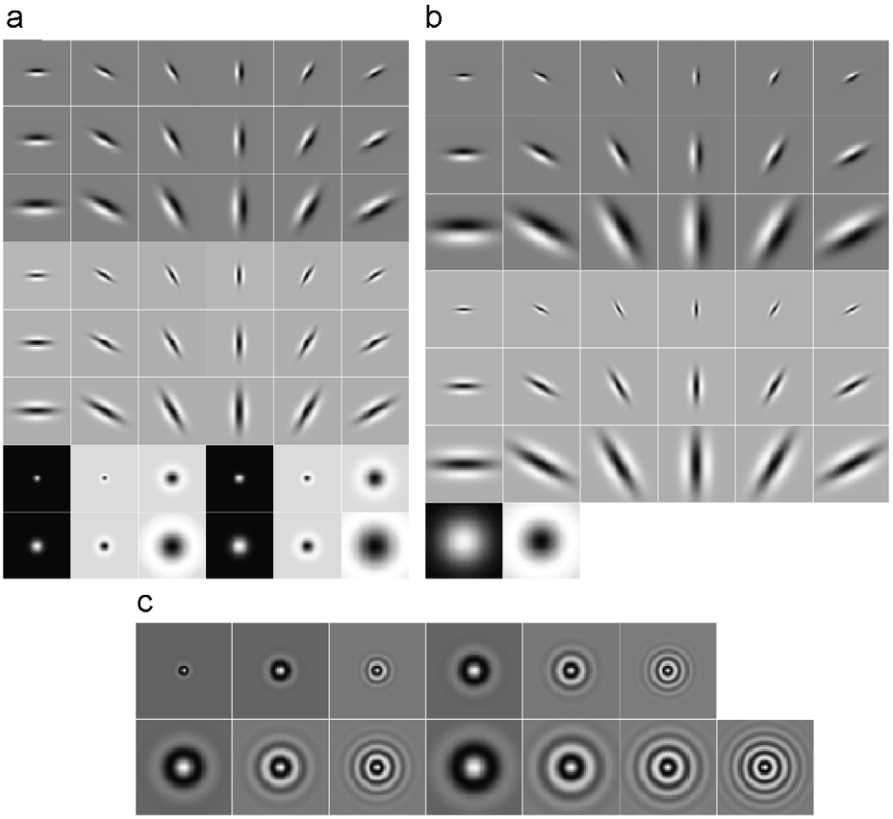


Fig. 19. Filter banks for texton-based feature representation: (a) Leung and Malik filter bank; (b) Varma and Zisserman filter bank; (c) Schmid filter bank.

$$f_9 = L_{ijk}L_iL_jL_k = \sum_{i,j,k=1}^{2} L_{x_jx_jx_k}L_{x_i}L_{x_j}L_{x_k}, \quad (57)$$

where ε_{ij} is the 2D anti-symmetric epsilon tensor defined by $\varepsilon_{12} = -\varepsilon_{21} = 1$ and $\varepsilon_{11} = \varepsilon_{22} = 0$.

The major steps to obtain the descriptor are: (i) interest points are extracted by the Harris corner detector; and (ii) a rotationally invariant descriptor is computed for each of the interest points, according to Eqs. (49)–(57).

3.5. Others

Apart from the four types of descriptors above, there are also other local features, e.g., the moment-based descriptor [80], the phase-based local features [12], and the color-based descriptors [79].

3.5.1. Moment-based descriptors

Generalized moment invariants [80] are computed up to second order and second degree for derivatives of an image patch: $M^a_{pq} = (1/xy)\sum_{x,y} x^p y^q I^a_d(x,y)$, of order a and degree $p+q$. I_d is the image gradient in the direction $d = x, y$.

3.5.2. Phase-based local features

Multi-scale phase-based local features [12] utilize both the phase and the amplitude responses of the complex-valued steerable bandpass filters for feature description. They can deal with scale changes and are likely to provide improved invariance to illumination changes. Moreover, this type of feature descriptor outperforms others when applied to images that are subjected to 2D rotation. By using such quadrature pair filters [25], a phase-based local feature is obtained and turned to a specific orientation θ and scale σ. The pair filters are described as follows:

$$g(\vec{p},\sigma,\theta) = G_2(\sigma,\theta)I(\vec{p}),$$
$$h(\vec{p},\sigma,\theta) = H_2(\sigma,\theta)I(\vec{p}), \quad (58)$$

where $G_2(\sigma,\theta)$ is the second derivative of a Gaussian function, $H_2(\sigma,\theta)$ is the approximation of Hilbert transform of $G_2(\sigma,\theta)$, and σ is the standard deviation of the Gaussian kernel used to derive $G_2(\sigma,\theta)$ and $H_2(\sigma,\theta)$. A complex polar representation can be described as

$$g(\vec{p},\sigma,\theta) + ih(\vec{p},\sigma,\theta) = \rho(\vec{p},\sigma,\theta)\exp(i\phi(\vec{p},\sigma,\theta)), \quad (59)$$

where $\rho(\vec{p},\sigma,\theta)$ is the local amplitude information and $\phi(\vec{p},\sigma,\theta)$ is the local phase information.

The procedure to obtain the phase-based local features is

(1) detect interest points by the Harris corner detector [28]. However, instead of using the Harris corner measure, a more convenient threshold criterion is proposed:

$$R(\vec{p}) = \frac{\lambda_2(\vec{p})}{c + (1/2)(\lambda_1(\vec{p}) + \lambda_2(\vec{p}))'}, \quad (60)$$

where λ_1 and λ_2 ($\lambda_1 \geqslant \lambda_2$) are eigenvalues of the Harris matrix; c is obtained according to the histogram of $R(\vec{p})$ for different images, herein, $c = 1$. Every point with $R(\vec{p}) \geqslant 0.5$ is selected as an interest point;

(2) compute the saturated amplitude $\tilde{\rho}(\vec{p},\sigma,\theta)$, which is allowed to reduce the system's sensitivity to brightness changes:

$$\tilde{\rho}(\vec{p},\sigma,\theta) = 1 - \exp\left(\frac{-\rho^2(\infty,\sigma,\theta)}{(2\sigma_\rho^2)}\right), \quad (61)$$

when $\rho(\vec{p},\sigma,\theta)$ is high enough, $\tilde{\rho}(\vec{p},\sigma,\theta)$ should be roughly constant; when $\rho(\vec{p},\sigma,\theta)$ is small, $\tilde{\rho}(\vec{p},\sigma,\theta) \approx 0$;

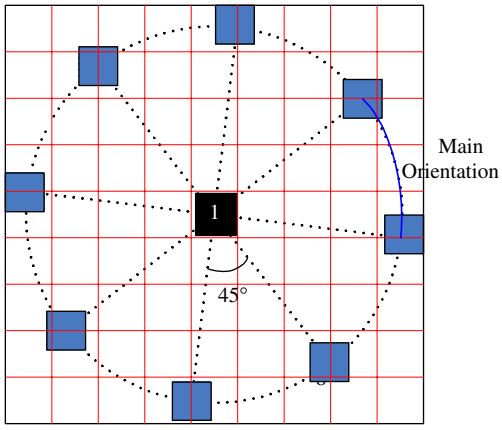


Fig. 20. Configuration of the phase-based local descriptor. The figure comes from Ref. [12].

(3) several sample points $\{\vec{p}_{i,m}\}_{1\leqslant m\leqslant M}$ in a region around each interest point $\vec{p}_i$ are taken into account. As shown in Fig. 20, a specified sampling pattern is used with the center point $\vec{p}_{i,1}$ representing the specific interest point $\vec{p}_i$. At each spatial point $\vec{p}_{i,m}$, the filters are steered to N equally spaced orientations:

$$\theta_n(\vec{p}_i) = \theta_M(\vec{p}_i) + (n-1)\frac{180^\circ}{N}, \quad \text{for } n = 1,2,\ldots,N, \quad (62)$$

where $\theta_M(\vec{p}_i)$ is the main orientation of the pixel determining both the orientations that the filters are steered to and the positions of the sample points $\vec{p}_{i,m}$; and

(4) the filter response evaluated at $\vec{p}_{i,m}$ and steered to orientation $\theta_n(\vec{p}_i)$ is denoted by $\tilde{\rho}(n,m)\exp(\phi_i(n,m))$, where $n = 1,2,...,N$, and $m = 1,2,...,M$. A NM-dimensional feature vector $\vec{F}(\vec{p}_i)$ is obtained by combining these responses.

3.5.3. Color-based descriptors

Color-based local features are based on color information. Four color descriptors were proposed by Weijer and Schmid [79], which are histograms of *rgb*, *hue*, *opponent angle* and *spherical angle*. The first two descriptors called color invariants are based on zero-order invariants; while the others named as color angles are based on first-order invariants. To test their effectiveness against shape-based descriptors, e.g., the SIFT, they were applied for matching, retrieval, and classification. For color objects, a pure color-based approach achieves better performance than a shape-based one. In general, the combination of color and shape descriptors outperforms the pure shape-based approach. The color descriptors are given as following:

(1) for *rgb*, the normalized r can be described as

$$r = \frac{R}{R+G+B}. \quad (63)$$

Normalized g and b have similar equations as Eq. (63). The normalized *rgb* can be considered to be invariant to lighting geometry and viewpoint;

(2) for *hue*

$$hue = \arctan\left(\frac{O_1}{O_2}\right), \quad (64)$$

where $O_1 = (R-G)/\sqrt{2}$ and $O_2 = (R+G-2B)/\sqrt{6}$ are opponent colors;

(3) for *opponent angle*

$$\theta^O_x = \arctan\left(\frac{(O_1)_x}{(O_2)_x}\right), \quad (65)$$

where $(O_1)_x = (R_x - G_x)/\sqrt{2}$ and $(O_2)_x = (R_x + G_x - 2B_x)/\sqrt{6}$ are the derivatives of the opponent colors. They are invariant to specular variations. The opponent angle is robust to geometric changes; and

(4) for *spherical angle*

$$\theta_x^S = \arctan\left(\frac{(\theta_1)_x}{(\theta_2)_x}\right), \tag{66}$$

where $(\theta_1)_x = (G_xR - RG_x)/\sqrt{R^2+G^2}$ and $(\theta_2)_x = (R_xRB + G_xGB - B_xR^2 - B_xG^2)/\sqrt{(R^2+G^2)(R^2+G^2+B^2)}$ are changes in the two directions perpendicular to the object color, which are invariant to geometric variations.

4. Conclusion

Because of the promising properties and capabilities of local features, they are being utilized broadly for various kinds of computer vision applications, e.g., object recognition, object class recognition, texture classification, image retrieval, robust matching and video data mining.

In this review, detectors are divided into corner detectors and region detectors. For example, corner detectors consist of Moravec's corner detector, Harris corner detector, SUSAN, Trajkovic operator, and high-speed corner detector; and region detectors discussed are the Harris–Laplace, Harris affine and Hessian affine, edge-based region detector, intensity-based region detector, maximally stable extremal region (MSER) detector, salient region detector, difference of Gaussian (DoG) operator and scale invariant feature transform (SIFT) detector.

Local descriptors introduced were following five types: (i) filter-based descriptors, (ii) distribution-based descriptors, (iii) textons, (iv) derivative-based descriptors, and (v) others. Further details were given on filter-based descriptors involving steerable filters, Gabor filters, and complex filters; distribution-based descriptors comprise SIFT descriptor, PCA based SIFT (PCA-SIFT), shape context, spin image, rotation invariant feature transform (RIFT), and gradient location and orientation histograms (GLOH); textons contain Leung and Malik 3D textons, Varma and Zisserman model, and Schmid filter bank; derivative-based descriptors consist of local derivatives, Florack descriptor, local grayvalue invariants; others are generalized moment invariants, phase-based local features, and color-based local features.

According to current performance evaluation studies, we can make the following tentative conclusions for detectors and descriptors: (i) MSER combined with SIFT descriptor usually performs the best for flat images but poorly for 3D objects; (ii) Hessian affine and DoG combined with an arbitrary descriptor consistently outperform other approaches for 3D objects; and (iii) SIFT and GLOH are generally accepted as the most effective descriptors. However, new detectors and descriptors are constantly being reported, and some of these may surpass the performance of currently accepted standards and increasingly demonstrate the power of feature-rich portrayal of imagery.

References

[1] A. Agarwal, B. Triggs, Hyperfeatures—multilevel local coding for visual recognition, Technical Report RR-5655, INRIA, 2005.

[2] S. Ando, Image field categorization and edge/corner detection from gradient covariance, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2) (2000) 179–190.

[3] J. Basak, D. Mahata, A connectionist model for corner detection in binary and gray images, IEEE Trans. Neural Netw. 11 (5) (2000) 1124–1132.

[5] J. Bauer, H. Bischof, A. Klaus, K. Karner, Robust and fully automated image registration using invariant features, ISPRS, DVD Proc. (2004) 12–23.

[7] P.R. Beaudet, Rotational invariant image operators, in: Proceedings of the IEEE International Conference on Pattern Recognition, 1978, pp. 579–583.

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

[10] A.C. Berg, T.L. Berg, J. Malik, Shape matching and object recognition using low distortion correspondence, in: Proceedings of the IEEE International Conference Computer Vision Pattern Recognition, vol. 1, 2005, pp. 26–33.

[11] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 8 (6) (1986) 679–698.

[12] G. Carneiro, A.D. Jepson, Multi-scale phase-based local features, in: Proceedings of the IEEE International Conference on Computer Vision Pattern Recognition, vol. 1, 2003, pp. 736–743.

[13] A.K. Chan, C.K. Chui, J. Zha, Q. Liu, Corner detection using spline wavelets, in: Proceedings of the SPIE Curves and Surfaces Computer Vision Graphics II, vol. 1610, 1992, pp. 311–322.

[14] C. Chen, J. Lee, Y. Sun, Wavelet transformation for gray-level corner detection, Pattern Recognit. 28 (6) (1995) 853–861.

[15] J.L. Crowley, A.C. Parker, A representation for shape based on peaks and ridges in the difference of low-pass transform, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1984) 156–168.

[16] O.G. Cula, K.J. Dana, Compact representation of bidirectional texture functions, in: Proceedings of the IEEE International Computer Vision and Pattern Recognition, vol. 1, 2001, pp. 1041–1047.

[17] J.G. Daugman, Two-dimensional spectral analysis of cortical receptive field profile, Vis. Res. 20 (1980) 847–856.

[18] J.G. Daugman, Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters, J. Opt. Soc. Am. 2 (7) (1985) 1160–1169.

[19] R. Deriche, G. Giraudon, Accurate corner detection: an analytical study, in: Proceedings of the IEEE International Conference on Computer Vision, 1990, pp. 66–70.

[20] G. Dorko, C. Schmid, Selection of scale-invariant parts for object class recognition, in: Proceedings of the IEEE International Conference on Computer Vision, vol. 1, 2003, pp. 634–639.

[21] R. Fergus, P. Perona, A. Zisserman, Object class recognition by unsupervised scale-invariant learning, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2003, pp. 264–271.

[22] V. Ferrari, T. Tuytelaars, L. Van Gool, Wide-baseline multiple-view correspondences, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, vol. 1, 2003, pp. 718–725.

[23] L.M.J. Florack, B.M. Ter Haar Romeny, J.J. Koenderink, M.A. Viergever, General intensity transformations and differential invariants, J. Math. Imaging Vis. 4 (2) (1994) 171–187.

[24] W. Förstner, E. Gülch, A fast operator for detection and precise location of distinct points, corners and centres of circular features, ISPRS Intercommission Conf. Fast Processing Photogrammetric Data, 1987, pp. 281–305.

[25] W. Freeman, E. Adelson, The design and use of steerable filters, IEEE Trans. Pattern Anal. Mach. Intell. 13 (9) (1991) 891–906.

[26] X. Gao, F. Sattar, R. Venkateswarlu, Corner detection of gray level images using Gabor wavelets, in: Proceedings of the IEEE International Conference on Image Processing, vol. 4, 2004, pp. 2669–2672.

[27] K. Grauman, T. Darrell, The pyramid match kernel: discriminative classification with sets of image features, in: Proceedings of the IEEE International Conference on Computer Vision, vol. 2, 2005, pp. 1458–1465.

[28] C. Harris, M. Stephens, A combined corner and edge detector, Alvey Vision Conference, 1988, pp. 147–151.

[29] E. Hayman, B. Caputo, M. Fritz, J.O. Eklundh, On the significance of real-world conditions for material classification, Proceedings of European Conference on Computer Vision, Lecture Notes in Computer Science, vol. 3024, 2004, pp. 253–266.

[30] G. Hetzel, B. Leibe, P. Levi, B. Schiele, 3D object recognition from range images using local feature histograms, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, vol. 2, 2001, pp. 394–399.

[31] A. Johnson, M. Hebert, Object recognition by matching oriented points, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 1997, pp. 684–689.

[32] A.E. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3D scenes, IEEE Trans. Pattern Anal. Mach. Intell. 21 (5) (1999) 433–449.

[33] T. Kadir, A. Zisserman, M. Brady, An affine invariant salient region detector, in Proceedings of the European Conference on Computer Vision, Lecture Notes in Computer Science, vol. 3021, 2004, pp. 228–241.

[34] Y. Ke, R. Sukthankar, PCA-SIFT: A more distinctive representation for local image descriptors, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, vol. 2, 2004, pp. 506–513.

[35] L. Kitchen, A. Rosenfeld, Gray level corner detection, Pattern Recognit. Lett. 1 (1982) 95–102.

[36] J. Koenderink, A. van Doorn, Representation of local geometry in the visual system, Biol. Cybernet. 55 (6) (1987) 367–375.

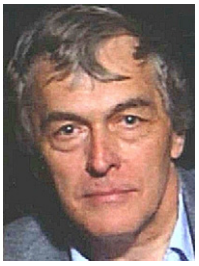
[37] S. Lazebnik, C. Schmid, J. Ponce, A sparse texture representation using local affine regions, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1265–1278.

[38] T.S. Lee, Image representation using 2D Gabor wavelets, IEEE Trans. Pattern Anal. Mach. Intell. 18 (10) (2003) 959–971.

[39] B. Leibe, B. Schiele, Interleaved object categorization and segmentation, in: Proceedings of the British Machine Vision Conference, 2003, pp. 759–768.

[40] T. Leung, J. Malik, Representing and recognizing the visual appearance of materials using three-dimensional textons, Int. J. Comput. Vis. 43 (1) (2001) 29–44.
[41] J. Li, N. Allinsion, D. Tao, X. Li, Multitraining support vector machine for image retrieval, IEEE Trans. Image Process. 15 (11) (2006) 3597–3601.
[42] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[43] B.D. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: International Joint Conference on Artificial Intelligence, 1981, pp. 674–679.
[44] S. Marcelja, Mathematical description of the responses of simple cortical cells, J. Opt. Soc. Am. 70 (11) (1980) 1297–1300.
[45] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions, in: Proceedings of the British Machine Vision Conference, 2002, pp. 384–393.
[46] K. Mikolajczyk, B. Leibe, B. Schiele, Local features for object class recognition, in: Proceedings of the IEEE International Conference on Computer Vision, vol. 2, 2005, pp. 1792–1799.
[47] K. Mikolajczyk, C. Schmid, Scale & affine invariant interest point detectors, Int. J. Comput. Vis. 60 (1) (2004) 63–86.
[48] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Trans. Pattern Anal. Mach. Intell. 27 (10) (2005) 1615–1630.
[49] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. Van Gool, A comparison of affine region detectors, Int. J. Comput. Vis. 65 (1/2) (2005) 43–72.
[50] K. Mikolajczyk, A. Zisserman, C. Schmid, Shape recognition with edge-based features, in: Proceedings of the British Machine Vision Conference, vol. 2, 2003, pp. 779–788.
[51] F. Mokhtarian, F. Mohanna, Performance evaluation of corner detectors using consistency and accuracy measures, Comput. Vis. Image Understand. 102 (1) (2006) 81–94.
[52] H. Moravec, Towards automatic visual obstacle avoidance, in: Proceedings of the International Joint Conference on Artificial Intelligence, 1977, p. 584.
[53] P. Moreels, P. Perona, Evaluation of features detectors and descriptors based on 3D objects, in: IEEE International Conference Computer Vision, vol. 1, 2005, pp. 800–807.
[54] J.A. Noble, Finding corners, Image Vis. Comput. 6 (2) (1988) 121–128.
[55] E. Nowak, F. Jurie, B. Triggs, Sampling strategies for bag-of-features image classification, in: Proceedings of the European Conference on Computer Vision, Lecture Notes in Computer Science, vol. 3954, 2006, pp. 490–503.
[56] P. Pritchett, A. Zisserman. Wide baseline stereo matching, in: Proceedings of the IEEE International Conference on Computer Vision, 1998, pp. 754–760.
[57] A. Quddus, M.M. Fahmy, Corner detcction using Gabor-type itering, in: Proceedings of the IEEE International Symposium Circuits and Systems, vol. 4, 1998, pp. 150–153.
[58] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1986) 81–106.
[59] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, in: European Conference on Computer Vision, vol. 1(1), 2006, pp. 430–443.
[60] F. Rothganger, S. Lazebnik, C. Schmid, J. Ponce, 3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, vol. 2(2), 2003, pp. 272–277.
[61] F. Schaffalitzky, A. Zisserman, Multi-view matching for unordered image sets, or 'how do I organize my holiday snaps?', in: Proceedings of the European Conference on Computer Vision, vol. 1, 2002, pp. 414–431.
[62] C. Schmid, Constructing models for content-based image retrieval, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, vol. 2, 2001, pp. 39–45.
[63] C. Schmid, R. Mohr, Local grayvalue invariants for image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 19 (5) (1997) 530–534.
[64] C. Schmid, R. Mohr, C. Bauckhage, Evaluation of interest point detectors, Int. J. Comput. Vis. 37 (2) (2000) 151–172.
[65] S. Se, D. Lowe, J. Little, Vision-based mobile robot localization and mapping using scale-invariant features, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2001, pp. 2051–2058.
[66] J. Shi, C. Tomasi, Good features to track, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 593–600.
[67] J. Sivic, A. Zisserman, Video Google: a text retrieval approach to object matching in videos, in: Proceedings of the International Conference on Computer Vision, vol. 2, 2003, pp. 1470–1477.
[68] J. Sivic, F. Schaffalitzky, A. Zisserman, Object level grouping for video shots, Int. J. Comput. Vis. 67 (2) (2006) 189–210.
[69] S.M. Smith, J.M. Brady, SUSAN—a new approach to low level image processing, Int. J. Comput. Vis. 23 (1) (1997) 45–78.
[70] E. Sojka, A new algorithm for detecting corners in digital images, in: Proceedings of the Spring Conference on Computer Graphics, 2002, pp. 55–62.
[71] D. Tao, X. Li, W. Hu, S.J. Maybank, X. Wu, Supervised Tensor Learning, Knowledge and Information Systems, Springer, 2007.
[72] D. Tao, X. Li, X. Wu, S.J. Maybank, General tensor discriminant analysis and Gabor features for gait recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2007).
[73] D. Tao, X. Tang, X. Li, Y. Rui, Kernel direct biased discriminant analysis: a new content-based image retrieval relevance feedback algorithm, IEEE Trans. Multimedia 8 (4) (2006) 716–727.
[74] D. Tao, X. Tang, X. Li, X. Wu, Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 28 (7) (2006) 1088–1099.
[75] D. Tell, S. Carlsson, Wide baseline point matching using affine invariants computed from intensity profiles, in: Proceedings of the European Conference on Computer Vision, Lecture Notes in Computer Science, vol. 1842, 2000, pp. 814–828.
[76] C. Tomasi, T. Kanade, Shape and motion from image streams: a factorization method—Part 3. Detection and tracking of point features, Technical Report, CMU-CS-91-132, Carnegie-Mellon University, 1991.
[77] M. Trajkovic, M. Hedley, Fast corner detection, Image Vis. Comput. 16 (2) (1998) 75–87.
[78] T. Tuytelaars, L. Van Gool, Matching widely separated views based on affine invariant regions, Int. J. Comput. Vis. 59 (1) (2004) 61–85.
[79] J. Van De Weijer, C. Schmid, Coloring local feature extraction, in: Proceedings of the European Conference on Computer Vision, Lecture Notes in Computer Science, vol. 3952, 2006, pp. 334–348.
[80] L. Van Gool, T. Moons, D. Ungureanu, Affine/photometric invariants for planar intensity patterns, in: Proceedings of the European Conference on Computer Vision, Lecture Notes in Computer Science, vol. 1064, 1996, pp. 642–651.
[81] M. Varma, A. Zisserman, Classifying images of materials: achieving viewpoint and illumination independence, in: Proceedings of the European Conference on Computer Vision, Lecture Notes in Computer Science, vol. 2352(3), 2002, pp. 255–271.
[82] M. Varma, A. Zisserman, A statistical approach to texture classification from single images, Int. J. Comput. Vis. 62 (1/2) (2005) 61–81.
[83] H. Wang, M. Brady, Real-time corner detection algorithm for motion estimation, Image Vis. Comput. 13 (9) (1995) 695–703.
[84] Z. Zheng, H. Wang, E.K. Teoh, Analysis of gray level corner detection, Pattern Recognit. Lett. 20 (2) (1999) 149–162.

Jing Li is currently a Ph.D. student with the Department of Electronic and Electrical Engineering in the University of Sheffield. Her research interests include content-based image retrieval, database organization, and image processing. She published at IEEE Transactions on Image Processing (TIP), IEEE International Conference on Web Intelligence, and IEEE International Conference on Intelligent Engineering Systems. She is a reviewer for the International Journal of Computer Mathematics (IJCM), International Journal of Image and Graphics (IJIG) and International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI).



Nigel M. Allinson holds degrees in Electrical and Electronic Engineering from the Universities of Bradford and Cambridge. He is currently the Chair of Image Engineering at the University of Sheffield and previously held the same title at UMIST. His research interests are the development of novel imaging devices and systems (and currently leads the UK Basic Technology consortium on active pixel sensors), image transmission and processing, and pattern recognition. He has published over 250 scientific papers and patents, and has co-founded three spinout companies based on research within his group.