Chapter 1
Feature Representation and Extraction for Image Search and Video Retrieval

Qingfeng Liu, Yukhe Lavinia, Abhishek Verma, Joyoung Lee, Lazar Spasovic and Chengjun Liu

Abstract The ever-increasing popularity of intelligent image search and video retrieval warrants a comprehensive study of the major feature representation and extraction methods often applied in image search and video retrieval. Towards that end, this chapter reviews some representative feature representation and extraction approaches, such as the Spatial Pyramid Matching (SPM), the soft assignment coding, the Fisher vector coding, the sparse coding and its variants, the Local Binary Pattern (LBP), the Feature Local Binary Patterns (FLBP), the Local Quaternary Patterns (LQP), the Feature Local Quaternary Patterns (FLQP), the Scale-invariant feature transform (SIFT), and the SIFT variants, which are broadly applied in intelligent image search and video retrieval.

1.1 Introduction

The effective methods in intelligent image search and video retrieval are often interdisciplinary in nature, as they cut across the areas of probability, statistics, real analysis, digital signal processing, digital image processing, digital video processing, computer vision, pattern recognition, machine learning, and artificial intelligence,

Q. Liu (✉) · J. Lee · L. Spasovic · C. Liu (✉)
New Jersey Institute of Technology, Newark, NJ 07102, USA
e-mail: ql69@njit.edu

C. Liu
e-mail: chengjun.liu@njit.edu

J. Lee
e-mail: jo.y.lee@njit.edu

L. Spasovic
e-mail: spasovic@njit.edu

Y. Lavinia (✉) · A. Verma (✉)
California State University, Fullerton, CA 92834, USA
e-mail: ylavinia@csu.fullerton.edu

A. Verma
e-mail: averma@fullerton.edu

C. Liu (ed.), *Recent Advances in Intelligent Image Search and Video Retrieval*,
Intelligent Systems Reference Library 121, DOI 10.1007/978-3-319-52081-0_1

Fig. 1.1 Example images from the Caltech-256 dataset

just to name a few. The applications of intelligent image search and video retrieval cover a broad range from web-based image search (e.g., photo search in Facebook) to Internet video retrieval (e.g., looking for a specific video in YouTube). Figure 1.1 shows some example images from the Caltech-256 dataset, which contains a set of 256 object categories with a total of 30,607 images [21]. Both the Caltech-101 and the Caltech-256 image datasets are commonly applied for evaluating the performance on image search and object recognition [21]. Figure 1.2 displays some video frames from the cameras installed along the highways. Actually, the New Jersey Department of Transportation (NJDOT) operates more than 400 traffic video cameras, but current traffic monitoring is mainly carried out by human operators. Automated traffic incident detection and monitoring is much needed as operator-based monitoring is often stressful and costly.

The ever-increasing popularity of intelligent image search and video retrieval thus warrants a comprehensive study of the major feature representation and extraction methods often applied in image search and video retrieval. Towards that end, this chapter reviews some representative feature representation and extraction approaches, such as the Spatial Pyramid Matching (SPM) [27], the soft assignment coding or kernel codebook [17, 18], the Fisher vector coding [24, 42], the sparse coding [53], the Local Binary Pattern (LBP) [40], the Feature Local Binary Patterns

Fig. 1.2 Example video frames from the cameras installed along the highways

(FLBP) [23, 31], the Local Quaternary Patterns (LQP) [22], the Feature Local Quaternary Patterns (FLQP) [22, 31], the Scale-invariant feature transform (SIFT) [35], and the SIFT variants, which are broadly applied in intelligent image search and video retrieval.

1.2 Spatial Pyramid Matching, Soft Assignment Coding, Fisher Vector Coding, and Sparse Coding

1.2.1 Spatial Pyramid Matching

The bag of visual words [13, 27] method starts with the k-means algorithm for deriving the dictionary and the hard assignment coding method for feature coding. One representative method is the spatial pyramid matching (SPM) [27] method, which enhances the discriminative capability of the conventional bag of visual words method by incorporating the spatial information.

Specifically, given the local feature descriptors $\mathbf{x}_i \in \mathbb{R}^n$, $(i = 1, 2, \ldots, m)$ and the dictionary of visual words $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_k] \in \mathbb{R}^{n \times k}$ derived from the k-means algorithm, the SPM method counts the frequency of the local features over the visual words and represents the image as a histogram using the following hard assignment coding method:

$$c_{ij} = \begin{cases} 1 & \text{if } j = \arg\min ||\mathbf{x}_i - \mathbf{d}_j||^2 \\ 0 & \text{otherwise} \end{cases} \tag{1.1}$$

In other words, the SPM method activates only one non-zero coding coefficient, which corresponds to the nearest visual word in the dictionary $\mathbf{D}$ for each local feature descriptor $\mathbf{x}_i$. And given one image I with T local feature descriptors, the corresponding image representation is the probability density estimation of all the

local features $\mathbf{x}_i$ in this image I over all the visual words $\mathbf{d}_j$ based on the histogram of visual word frequencies as follows:

$$h = \left[\frac{1}{T} \sum_{i=1}^{m} c_{i1}, \frac{1}{T} \sum_{i=1}^{m} c_{i2}, \ldots, \frac{1}{T} \sum_{i=1}^{m} c_{ik} \right] \tag{1.2}$$

1.2.2 Soft Assignment Coding

The histogram estimation of the density function for the local features $\mathbf{x}_i$ over the visual words $\mathbf{d}_j$, which violates the ambiguous nature of local features, is a very coarse estimation. Therefore, the soft assignment coding [17, 18], or kernel codebook, is proposed as a more robust alternative to histogram.

Specifically, the soft-assignment coding of $\mathbf{c}_{ij}$ is defined as follows:

$$c_{ij} = \frac{exp(-||\mathbf{x}_i - \mathbf{d}_j||^2/2\sigma^2)}{\sum_{j=1}^{k} exp(-||\mathbf{x}_i - \mathbf{d}_j||^2/2\sigma^2)} \tag{1.3}$$

where σ is the smoothing parameter that controls the degree of smoothness of the assignment and $exp(\cdot)$ is the exponential function.

Consequently, given one image I with T local feature descriptors, the corresponding image representation is the probability density estimation of the all the local features $\mathbf{x}_i$ in this image I over all the visual words $\mathbf{d}_j$ based on the kernel density estimation using the Gaussian kernel $K(\mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{\mathbf{x}^2}{2\sigma^2})$ as follows:

$$\begin{aligned} h &= \left[\frac{1}{T} \sum_{i=1}^{m} c_{i1}, \ldots, \frac{1}{T} \sum_{i=1}^{m} c_{ik} \right] \\ &= \left[\frac{1}{T} \sum_{i=1}^{T} w_i K(\mathbf{d}_1 - \mathbf{x}_i), \ldots, \frac{1}{T} \sum_{i=1}^{T} w_i K(\mathbf{d}_k - \mathbf{x}_i) \right] \end{aligned} \tag{1.4}$$

where σ in the Gaussian kernel plays the role of bandwidth in kernel density estimation, and $w_i = \frac{1}{\sum_{j=1}^{k} K(\mathbf{x}_i - \mathbf{d}_j)}$.

1.2.3 Fisher Vector Coding

The kernel density estimation in soft assignment coding is still error-prone due to its limitation in probability in density estimation. Recently, the Fisher vector method [24, 42] is proposed that the generative probability function of the local feature descriptors is estimated using a more refined model, namely the Gaussian mixture

model (GMM). Then the GMM is applied to derive the Fisher kernel, which is incorporated into the kernel based support vector machine for classification. Fisher vector coding method [42] is essentially an explicit decomposition of the Fisher kernel.

As for dictionary learning, unlike the spatial pyramid matching and the soft assignment coding method, the Fisher vector coding method replaces the k-means algorithm with a Gaussian mixture model as follows:

$$\mu_{\boldsymbol{\lambda}}(x) = \sum_{j=1}^{k} w_j g_j(x : \mu_j, \sigma_j) \tag{1.5}$$

where the parameter set $\boldsymbol{\lambda} = \{w_j, \mu_j, \sigma_j, j = 1, 2, \ldots, k\}$ represents the mixture weight, the mean vector, and the covariance matrix of the Gaussian components, respectively. As a result, the visual words are no longer the centroids of the clusters, but rather the GMM components.

As for feature coding, the Fisher vector coding method applies the gradient score of the $j-$th component of the GMM over its parameters (μ_j is used here), instead of the hard/soft assignment coding methods as follows:

$$c_{ij} = \frac{1}{\sqrt{w_j}} \gamma_i(j) \sigma_j^{-1}(x_i - \mu_j) \tag{1.6}$$

where $\gamma_i(j) = \frac{w_j g_j(x_i)}{\sum_{t=1}^{k} w_t g_t(x_i)}$. As a result, given one image I with T local feature descriptors, the corresponding image representation, namely the Fisher vector, is the histogram of the gradient score of all the local features $\mathbf{x}_i$ in this image I:

$$\begin{aligned} h &= \left[\frac{1}{T} \sum_{i=1}^{m} c_{i1}, \ldots, \frac{1}{T} \sum_{i=1}^{m} c_{ik} \right] \\ &= \left[\frac{1}{T} \sum_{i=1}^{T} \frac{1}{\sqrt{w_1}} \gamma_i(1) \sigma_1^{-1}(x_i - \mu_1), \ldots \frac{1}{T} \sum_{i=1}^{T} \frac{1}{\sqrt{w_k}} \gamma_i(k) \sigma_k^{-1}(x_i - \mu_k) \right] \end{aligned} \tag{1.7}$$

1.2.4 Sparse Coding

The sparse coding method deals with the dictionary learning and the feature coding from the reconstruction point of view. Yang et al. [53] applied the sparse coding to learn a dictionary and a vector of coefficients for the feature coding.

Specifically, the sparse coding method is the optimization of the following objective function:

$$\begin{aligned} &\min_{\mathbf{D},\mathbf{W}} \sum_{i=1}^{m} ||\mathbf{x}_i - \mathbf{D}\mathbf{w}_i||^2 + \lambda ||\mathbf{w}_i||_1 \\ &s.t. \quad ||\mathbf{d}_j|| \leq 1, (j = 1, 2, \ldots, k) \end{aligned} \tag{1.8}$$

The sparse coding method applies a reconstruction criterion so that the original local feature descriptor can be reconstructed as a linear combination of the visual words in the dictionary and most of the coefficients are zero. Many methods are proposed for optimizing the objective function, such as the fast iterative shrinkage-thresholding algorithms (FISTA) [8], the efficient learning method [28], as well as the online learning method [36]. After the optimization, both the dictionary and the sparse coding are obtained. Then following the notation in the above sections, the sparse coding method derives the following coding:

$$c_{ij} = w_{ij} \tag{1.9}$$

where w_{ij} is an element in the sparse coding vector $\mathbf{w}_i$.

Consequently, given one image I with T local feature descriptors, the corresponding image representation is computed either using the average pooling method (h_{avg}) or the max pooling method (h_{max}) [46, 53] as follows:

$$\begin{aligned} h_{avg} &= \left[\frac{1}{T} \sum_{i=1}^{m} c_{i1}, \ldots, \frac{1}{T} \sum_{i=1}^{m} c_{ik} \right] \\ h_{max} &= [\max\{c_{i1}\}, \ldots, \max\{c_{ik}\}] \end{aligned} \tag{1.10}$$

We have reviewed in the formal or mathematical means some representative feature representation and extraction methods for intelligent image search and video retrieval. Figure 1.3 shows in a more intuitive and graphical manner the comparison among the Spatial Pyramid Matching (SPM), the Soft Assignment Coding (SAC), the Fisher vector coding, and the sparse coding methods.

1.2.5 Some Sparse Coding Variants

There are a lot of variants of sparse coding methods that are proposed for addressing various issues in sparse coding. Wang et al. [46] proposed the locality-constrained linear coding (LLC) method that incorporates the local information in the feature coding process. Specifically, the LLC method incorporates a locality criterion into the sparse coding criterion to derive a new coding for the local feature descriptor that takes the local information into account. Gao et al. [16] further proposed the Laplacian sparse coding (LSC) that preserves both the similarity and the locality information among the local features. Specifically, the proposed LSC introduces a graph sparse coding criterion into the sparse coding criterion to derive a new coding

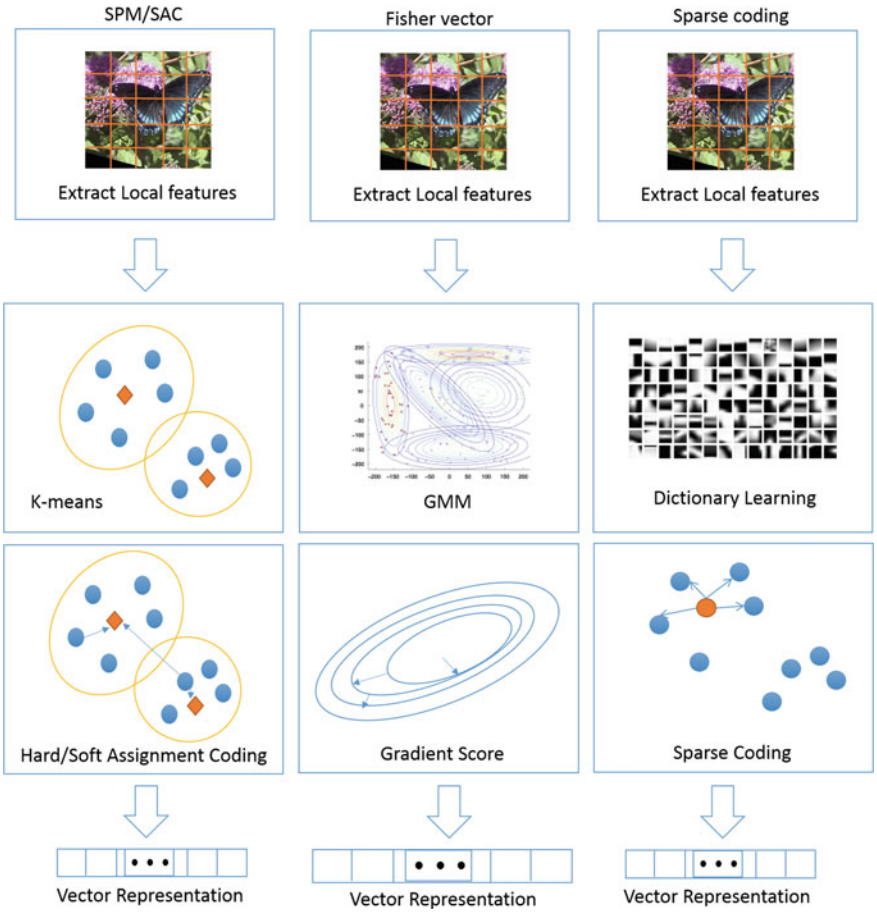

Fig. 1.3 Intuitive and graphical comparison among the Spatial Pyramid Matching (SPM), the Soft Assignment Coding (SAC), the Fisher vector coding, and the sparse coding methods

method to utilize the underlying structure of sparse coding. Zhou et al. [56] proposed a super vector coding method which takes advantage of the probability kernel method for sparse coding. Bo et al. [9] proposed a hierarchical sparse coding methods for image classification, which harnesses the hierarchical structure of the sparse coding.

In addition, many papers on sparse coding focus on developing efficient learning algorithms to derive the sparse coding and the dictionary [8, 20, 28, 36, 47–50], or exploring the data manifold structures [16, 46, 55]. For efficiency optimization, recent research applies screening rules for improving the computational efficiency of the sparse coding (lasso) problem. Ghaoui et al. [20] presented the SAFE screening rule for the lasso problem and sparse support vector machine. Xiang et al. [49] derived two new screening tests for large scale dictionaries and later proposed the DOME test [50] for the lasso problem. Wang et al. [48] proposed the Dual Polytope Projection

(DPP) for the lasso screening problem and later [47] proposed the "Slores" rule for sparse logistic regression screening.

1.3 Local Binary Patterns (LBP), Feature LBP (FLBP), Local Quaternary Patterns (LQP), and Feature LQP (FLQP)

The Local Binary Patterns (LBP) method, which uses the center pixel as a threshold and compares the pixels in its local neighborhood with the threshold to derive a gray-scale invariant texture description, has broad applications in feature representation and extraction for intelligent image search and video retrieval [38–40]. Specifically, some researchers apply the LBP method for facial image representation and then utilize the LBP texture features as a descriptor for face recognition [1, 2]. Other researchers propose a method that fuses the local LBP features, the global frequency features, and the color features for improving face recognition performance [34]. Yet others present new color LBP descriptors for scene and image texture classification [6].

The LBP method is popular for feature representation and extraction because of its computational simplicity and robustness to illumination changes. The limitation of the LBP method comes from the fact that it considers only its local pixels but not the features that are more broadly defined, among which are the edge pixels, the intensity peaks or valleys of an image, the color features [5, 30, 33, 43], and the wavelet features [11, 29, 32], just to name a few.

The Feature Local Binary Patterns (FLBP) method improves upon the LBP method by introducing features that complement the conventional neighborhood used to derive the LBP representation [23]. The FLBP method first defines a True Center (TC) and a Virtual Center (VC): The TC is the center pixel of a neighborhood and the VC is specified on the distance vector by a VC parameter and is used to replace the center pixel of the neighborhood [23]. The FLBP representation is then defined by comparing the pixels in the neighborhood of the true center with the virtual center [23]. It is shown that the LBP method is a special case of the FLBP method [23]: when the TC and VC parameters are zero, the FLBP method degenerates to the LBP method. There are two special cases of the FLBP method: when the VC parameter is zero, the method is called the FLBP1 method; and when the TC parameter is zero, the method is called the FLBP2 method [23]. As these FLBP methods encode both the local information and the features that are broadly defined, they are expected to perform better than the LBP method for feature representation and extraction for intelligent image search and video retrieval [23, 31].

The Local Quaternary Patterns (LQP) method augments the LBP method by encoding four relationships of the local texture [22]:

$$S_{lqp}(g_i, g_c, r) = \begin{cases} 11, \text{ if } g_i \geq g_c + r \\ 10, \text{ if } g_c \leq g_i < g_c + r \\ 01, \text{ if } g_c - r \leq g_i < g_c \\ 00, \text{ if } g_i < g_c - r \end{cases} \tag{1.11}$$

where g_i and g_c represent the grey level of a neighbor pixel and the central pixel, respectively. $r = c + \tau g_c$ defines the radius of the interval around the central pixel and c is a constant and τ is a parameter to control the contribution of g_c to r. For efficiency, the LQP representation can be split into two binary codes, namely the upper half of LQP (ULQP) and the lower half of LQP (LLQP) [22]. As a result, the LQP method encodes more information of the local texture than the LBP method and the Local Ternary Patterns (LTP) method [22]. The Feature Local Quaternary Patterns (FLQP) method further encodes features that are broadly defined. Thus the FLQP method should further improve image search and video retrieval performance when used for feature representation and extraction [22].

1.4 Scale Invariant Feature Transform (SIFT) and SIFT Variants

SIFT [35] is one of the most commonly used local descriptors in intelligent image search and video retrieval. Its power lies in its robustness to affine distortion, viewpoints, clutters, and illumination changes. This makes SIFT an invaluable method in various computer vision applications such as face, object recognition, robotics, human activity recognition, panorama stitching, augmented reality, and medical image analysis.

The SIFT algorithm comprises the following steps: (1) scale space extrema detection, (2) keypoint localization, (3) orientation assignment, (4) keypoint descriptor construction, and (5) keypoint matching. To detect the peak keypoints, SIFT uses Laplacian of Gaussian (LoG), which acts as a space filter by detecting blobs in various scales and sizes. Due to its expensive computation, SIFT approximates LoG with Difference of Gaussian (DoG). The DoG is produced by computing the difference of Gaussian blurring on an image with two different scales that are represented in different octaves in the Gaussian pyramid. Following this, the keypoint extrema candidates are located by selecting each pixel in an image and comparing it with its 8 neighbors and the 9 pixels of its previous and next scales, amounting to 26 pixels to compare. Next is keypoint localization, which analyzes the keypoint candidates produced in the previous step. SIFT uses the Taylor series expansion to exclude the keypoints with low contrast, thus leaving the ones with strong interest points to continue on the next step. Orientation assignment is geared to achieve rotation invariance. At each keypoint, the central derivatives, the gradient magnitude, and the direction are computed, producing a weighted orientation histogram with 36 bins around the keypoint neighborhood. The most dominant, that is, the highest peak of the histograms, of the orientations is selected as the direction of the keypoint. To construct the keypoint

descriptor, a 16×16 neighborhood region around a keypoint is selected to compute the keypoint relative orientation and magnitude. It is further divided into 16 subblocks, each with 4×4 size. An 8-bin orientation histogram is created for each subblock. The 16 histograms are concatenated to form a 128-dimension descriptor. Finally, keypoint matching is done by computing the Euclidean distance between two keypoints. First, a database of keypoints is constructed from the training images. Next, when a keypoint is to be matched, it is compared with the ones stored in a database. The Euclidean distance of these keypoints is computed and the database keypoint with minimum Euclidean distance is selected as the match.

Since its creation, many works have been dedicated to improve SIFT. The modifications are done in various steps of the SIFT algorithm and are proven to increase not only recognition rate but also speed. The following provides brief descriptions of the SIFT and SIFT like descriptors: Color SIFT, SURF, MSIFT, DSP-SIFT, LPSIFT, FAIR-SURF, Laplacian SIFT, Edge-SIFT, CSIFT, RootSIFT, and PCA-SIFT.

1.4.1 Color SIFT

The various color spaces such as RGB, HSV, rgb, oRGB, and YCbCr can be used to enhance SIFT performance [45]. The color SIFT descriptors are constructed by computing the 128 dimensional vector of the SIFT descriptor on the three channels, yielding 384 dimensional descriptors of RGB-SIFT, HSV-SIFT, rgb-SIFT, and YCbCr-SIFT.

Concatenating the three image components of the oRGB color space produces the oRGB-SIFT. Fusing RGB-SIFT, HSV-SIFT, rgb-SIFT, oRGB-SIFT, and YCbCr-SIFT generates the Color SIFT Fusion (CSF). Further fusion of the CSF and the grayscale SIFT produces the Color Grayscale SIFT Fusion (CGSF) [44, 45].

Results of the experiments on several grand challenge image datasets show that oRGB-SIFT descriptor improves recognition performance upon other color SIFT descriptors, the CSF, the CGSF, and the CGSF + PHOG descriptors perform better than the other color SIFT descriptors. The fusion of both Color SIFT descriptors (CSF) and Color Grayscale SIFT descriptor (CGSF) show significant improvement in the classification performance, which indicates that various color-SIFT descriptors and grayscale-SIFT descriptor are not redundant for image classification [44, 45].

1.4.2 SURF

Speeded Up Robust Features (SURF) [7] is a SIFT-based local feature detector and descriptor. SURF differs from SIFT in the following aspects. First, in scale space analysis, instead of using DoG, SURF uses a *fast Hessian detector* that is based on the Hessian matrix and implemented using box filters and integral images. Second, in orientation assignment, SURF computes Haar wavelet responses in the vertical and

horizontal directions within the scale s and radius $6s$ from the interest points. Estimation of the dominant orientation is computed by summing the responses obtained through a sliding 60° angled window. Third, in extracting the descriptor, SURF forms a square region centered at an interest point and oriented according to the dominant orientation. The region is divided into sub-regions. For each sub-region, SURF then computes the sum of the Haar wavelet responses in the vertical and horizontal directions of selected interest points, producing a 64-dimensional SURF feature descriptor.

To improve its discriminative power, the SURF descriptor can be extended to 128 dimensions. This is done by separating the summation computation for d_x and $|d_x|$ according to the sign of d_y ($d_y < 0$ and $d_y \geq 0$) and the computation for d_y and $|d_y|$ according to the sign of d_x. The result is a doubled number of features, creating a descriptor with increased discriminative power. SURF-128, however, performs slightly slower than SURF-64, although still faster than SIFT. This compromise turns out to be advantageous with SURF-128 achieving higher recognition rate than SURF-64 and SIFT. Results in [7] show SURF's improved performance on standard image datasets as well as on imagery obtained in the context of real life object detection application.

1.4.3 MSIFT

Multi-spectral SIFT (MSIFT) [10] takes advantage of near infrared (NIR) to enhance recognition. The NIR occupies the 750–1100 nm region on the wavelength spectrum, and silicon, the primary semiconductor in digital camera chip, is known to have high sensitivity to this region.

The MSIFT descriptor is developed by first decorrelating the RGB-NIR 4-dimensional color vector, followed by linear transformation of the resulting decorrelated components. This produces four components with the first being achromatic (luminance) with roughly the same amount of R, G, B, and high NIR, and the other three consisting of various spectral difference of R, G, B, and NIR. Next, forming the multi-spectral keypoint is done through Gaussian extrema detection in the achromatic first component. It is then followed by creation of 4×4 histogram of gradient orientations of each channel. The color bands are normalized and concatenated to form the final descriptor. Since the resulting RGB-NIR descriptors dimensionality amounts to 512, a PCA dimensionality reduction is applied.

The immediate application of MSIFT is to solve scene recognition problems. As noted above, with silicon's high sensitivity to the NIR region, an MSIFT equipped digital camera would be able to offer enhanced intelligent scene recognition features to users.

1.4.4 DSP-SIFT

Domain Size Pooling (DSP) SIFT [14] defies the traditional scale space step in SIFT descriptor construction and replaces this step with size space. Instead of forming the descriptor from a single selected scaled lattice, in DSP-SIFT, multiple lattices with various domain sizes are sampled, as shown in Fig. 1.4a. To make them all in the same size, each lattice sample is rescaled, making these multiple lattice samples differ in scales, although uniform in size (Fig. 1.4b). Pooling of the gradient orientations is done across these various locations and scales (Fig. 1.4c). These gradient orientations are integrated and normalized by applying a uniform density function (Fig. 1.4d), which then yields the final DSP-SIFT descriptor (Fig. 1.4e).

Authors in [14] report that DSP-SIFT outperforms SIFT by a wide margin and furthermore it outperforms CNN by 21% on the Oxford image matching dataset and more than 5% on the Fischer dataset.

1.4.5 LPSIFT

The layer parallel SIFT (LPSIFT) [12] seeks to implement SIFT on real time devices by reducing the computational cost, time latency, and memory storage of the original SIFT. The modification is done primarily on the most expensive steps of the original SIFT algorithm: scale space Gaussian extrema detection and keypoint localization. The main cause of this bottleneck, as it is observed, is the data dependency of scale image computation. The Gaussian blurring computation on each new image is done sequentially and this proves to cause a large memory expense for the next step in the pipeline: computation of the Difference of Gaussian (DoG). As the DoG pyramid requires at least three scale images to complete, a candidate image needs to wait for the Gaussian blurring operation to compute the next two images, and thus the image must be stored in memory.

LPSIFT solves the problem by introducing layer parallel to Gaussian pyramid construction. To handle simultaneous computation on multiple images, the layer parallel simply merges the kernels on the same level and forwards them to the DoG

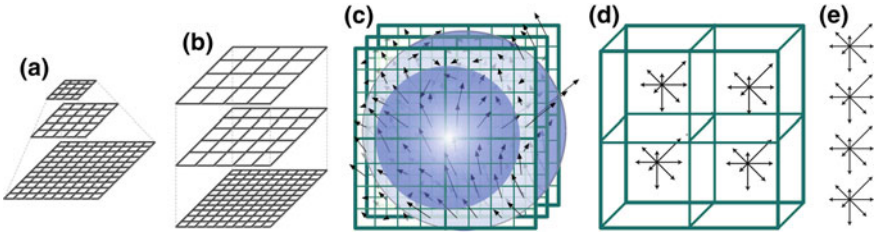

Fig. 1.4 DSP-SIFT methodology

operation. This trick significantly reduces the time latency caused by the sequential flow and also the memory cost needed to store the images. The merged kernel, however, can potentially expand to a size that would cause an increased computational cost. To avoid it, LPSIFT uses integral images that are implemented on modified box kernels of sizes 3×3, 5×5, and 7×7.

The next modification on the original SIFT algorithm takes place in the contrast test. The original SIFT algorithm contrast test aims to remove low contrast keypoints using the Taylor expansion, which is high in complexity. LPSIFT circumvents the use of the expensive Taylor series by modifying the algorithm to exclude low brightness instead of low contrast. The rationale is that low brightness is an accompanying characteristic of low contrast and thus excluding low brightness candidate keypoints would also exclude low contrast candidates.

With this method, LPSIFT manages to reduce 90% of computational cost and 95% of memory usage.

1.4.6 FAIR-SURF

Fully Affine InvaRiant SURF (FAIR-SURF) [41] modifies the SURF algorithm to be fully affine invariant by using techniques used in ASIFT [37]. Like ASIFT, FAIR-SURF uses two camera axis parameters simulation to generate images. It then applies the SURF algorithm for feature extraction. While ASIFT uses finite rotation and tilts to simulate camera angles in generating images and therefore runs much slower, FAIR-SURF selects only certain rotation and tilt angles, thus improving the speed.

The selection process is described as follows. It is observed that SURF maintains fully affine invariant until a certain low angle and that 2 is a balanced value between accuracy and sparsity [37]. The angles under which SURF is fully affine invariant are chosen so as to extend this fully affine invariant trait to the resulting FAIR-SURF. Thus, a list of rotation and tilt angles are formed.

To reduce the number of angles, Pang et al. [41] applied the modified SURF to extract features and perform image matching. They then compared the matching results with the list and selected several to be tested on other images. The matching results became the final list of rotation and tilt angles that are used to simulate images. These images are then used in the next steps of the SURF algorithm.

By using selected images that are simulated using angles under which SURF is fully affine invariant, FAIR-SURF achieves full affine invariance. Its keypoints produce higher matches compared to SURF and ASIFT and its runtime, although 1.3 times slower than the original SURF, is still faster than ASIFT.

1.4.7 Laplacian SIFT

In visual search on resource constrained devices, a descriptor's matching ability and compactness are critical. Laplacian SIFT [51] aims to improve these qualities by preserving the nearest neighbor relationship of the SIFT features. This is because it is observed that the nearest neighbors contain important information that can be used to improve matching ability. The technique utilizes graph embedding [52] and specifically the Laplacian embedding, to preserve more nearest neighbor information and reduce the dimensionality.

The image retrieval process using Laplacian SIFT is implemented in two separate segments: data preprocessing and query processing. Data preprocessing takes a set of feature points and employs Laplacian embedding to reduce the original SIFT 128 feature dimension to a desired feature dimension. The experiment uses a 32-bit representation with 4 dimensions and 8 bits quantization per dimension. Two *kd*-trees are then created on the resulting feature points. These trees are formed to discard the feature points located at the leaf node boundaries.

Query processing takes a set of query feature points and selects the features according to the feature selection algorithms. Two leaf nodes are located and merged. On each leaf node, the algorithm performs nearest neighbor matching and compares the results with a predetermined threshold value.

1.4.8 Edge-SIFT

Edge-SIFT [54] is developed to improve efficiency and compactness of large scale partial duplicate image search on mobile platforms. Its main idea is to use binary edge maps to suppress memory footprint. Edge-SIFT focuses on extracting edges since they preserve spatial cues necessary for identification and matching yet sparse enough to maintain compactness.

The first step to construct Edge-SIFT is creating image patches that are centered at interest points. These patches are normalized to make them scale and rotation invariant. Scale invariance is achieved through resizing each patch to a fixed size, while rotation invariance is through aligning the image patches to make their dominant orientations uniform. An edge extractor is then used to create the binary edge maps with edge pixel values of 1 or 0. The edge map is further decomposed into four sub-edge maps with four different orientations. To overcome sensitivity to registration errors in the vertical direction, an edge pixel expansion is applied in the vertical direction of the maps orientation.

The resulting initial Edge-SIFT is further compressed. Out of the most compact bins, the ones with the highest discriminative power are selected using RankBoost [15]. To reduce the speed of similarity computation, the results are stored in a lookup table. The final Edge-SIFT was proven to be more compact, efficient, and accurate

than the original SIFT. The direct application of the algorithm includes landmark 3D construction and image panoramic view generator.

1.4.9 CSIFT

CSIFT, or Colored SIFT [3], uses color invariance [19] to build the descriptor. As the original SIFT is designed to be applied to gray images, modifying the algorithm to apply to color images is expected to improve the performance. The color invariance model used in CSIFT is derived from the Kubelka Munk theory of photometric reflectance [26] and describes color invariants under various imaging conditions and assumptions concerning illumination intensity, color, and direction, surface orientation, highlights, and viewpoint. The color invariants can be calculated from the RGB color space using the Gaussian color model to represent spatial spectral information [19].

CSIFT uses these color invariants to expand the input image space and invariance for keypoint detection. The gradient orientation is also computed from these color invariants. The resulting descriptor is robust to image translation, rotation, occlusion, scale, and photometric variations. Compared to the grayscale based original SIFT, CSIFT generates higher detection and matching rate.

1.4.10 RootSIFT

RootSIFT [4] is based on the observation that in comparing histograms, the Hellinger kernel generates superior results to the Euclidean in image categorization, object and texture classification. As the original SIFT uses the Euclidean distance to compare histograms, it is expected that using the Hellinger kernel would improve the results. The original SIFT descriptor can be converted to RootSIFT using the Hellinger kernel.

The Hellinger kernel implementation in RootSIFT requires only two additional steps after the original SIFT: (1) L1 normalization of the original SIFT vector, and (2) taking the square root of each element. By executing these steps, the resulting SIFT vectors are L2 normalized. Classification results by applying the RootSIFT on Oxford buildings dataset and the PASCAL VOC image dataset show improvement upon SIFT [4].

1.4.11 PCA-SIFT

PCA-SIFT [25] undergoes the same few steps of the SIFT algorithm (scale space extrema detection, keypoint localization, and gradient orientation assignment) but

modifies the keypoint descriptor construction. PCA-SIFT uses a projection matrix to create the PCA-SIFT descriptor. To form this projection matrix, keypoints are selected and rotated towards their dominant orientation, and a 41×41 patch that is centered at each keypoint is created. The vertical and horizontal gradients of these patches are computed, forming an input vector of size $2 \times 39 \times 39$ with 3,042 elements for each patch. The covariance matrix of these vectors is computed, followed by the eigenvectors and eigenvalues of the matrices. The top n eigenvectors are then selected to construct $n \times$ 3,042 projection matrix, with n being an empirically determined value. This projection matrix is computed once and stored.

The descriptor is formed by extracting a 41×41 patch around a keypoint, rotating it to its dominant orientation, creating a normalized 3,042 element gradient image vector from the horizontal and vertical gradients, and constructing a feature vector by multiplying the gradient image vector with the stored $n \times$ 3,042 projection matrix. The resulting PCA-SIFT descriptor is of size n. With $n = 20$ as the feature space size, PCA-SIFT outperformed the original SIFT that uses 128-element vectors. Results show that using this descriptor in an image retrieval application results in increased accuracy and faster matching [25].

1.5 Conclusion

Some representative feature representation and extraction methods with broad applications in intelligent image search and video retrieval are reviewed. Specifically, the density estimation based methods encompass the Spatial Pyramid Matching (SPM) [27], the soft assignment coding or kernel codebook [17, 18], and the Fisher vector coding [24, 42]. The reconstruction based methods consist of the sparse coding [53] and the sparse coding variants. The local feature based methods includle the Local Binary Pattern (LBP) [40], the Feature Local Binary Patterns (FLBP) [23, 31], the Local Quaternary Patterns (LQP) [22], and the Feature Local Quaternary Patterns (FLQP) [22, 31]. And finally the invariant methods contain the Scale-invariant feature transform (SIFT) [35], and the SIFT like descriptors, such as the Color SIFT, the SURF, the MSIFT, the DSP-SIFT, the LPSIFT, the FAIR-SURF, the Laplacian SIFT, the Edge-SIFT, the CSIFT, the RootSIFT, and the PCA-SIFT.

References

1. Ahonen, T., Hadid, A., Pietikainen, M.: Face recognition with local binary patterns. In: 8th European Conference on Computer Vision, Prague, Czech Republic, pp. 469–481 (2004)
2. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: application to face recognition. IEEE Trans. Pattern Anal. Mach. Intell. **28**(12), 2037–2041 (2006)
3. Aly, A., Farag, A.: Csift: a sift descriptor with color invariant characteristics. In: IEEE International Conference on Computer Vision and Pattern Recognition, New York, NY, pp. 1978–1983 (2006)

4. Arandjelovic, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: IEEE International Conference on Computer Vision and Pattern Recognition, Providence, RI, pp. 2911–2918 (2012)
5. Banerji, S., Sinha, A., Liu, C.: New image descriptors based on color, texture, shape, and wavelets for object and scene image classification. Neurocomputing **117**, 173–185 (2013)
6. Banerji, S., Verma, A., Liu, C.: Novel color LBP descriptors for scene and image texture classification. In: 15th International Conference on Image Processing, Computer Vision, and Pattern Recognition, Las Vegas, Nevada, USA (2011)
7. Bay, H., Tuytelaars, T., Van Gool, L.V.: SURF: speeded up robust features. Comput. Vision Image Underst. **110**(3), 346–359 (2008)
8. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**(1), 183–202 (2009)
9. Bo, L., Ren, X., Fox, D.: Hierarchical matching pursuit for image classification: architecture and fast algorithms. In: Advances in Neural Information Processing Systems, pp. 2115–2123 (2011)
10. Brown, M., Ssstrunk, S.: Multi-spectral sift for scene category recognition. In: IEEE International Conference on Computer Vision and Pattern Recognition, Providence, RI, pp. 177–184 (2011)
11. Chen, S., Liu, C.: Eye detection using discriminatory haar features and a new efficient svm. Image Vision Comput. **33**, 68–77 (2015)
12. Chiu, L., Chang, T.S., Chen, J.Y., Chang, N.Y.C.: Fast sift design for real-time visual feature extraction. IEEE Trans. Image Process. **22**(8), 3158–3167 (2013)
13. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, Prague (2004)
14. Dong, J., Soatto, S.: Domain-size pooling in local descriptors: Dsp-sift. In: IEEE International Conference on Computer Vision and Pattern Recognition, Boston, MA (2015)
15. Freund, Y., Iyer, R., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences. J. Mach. Learn. Res. **4**, 933–969 (2003)
16. Gao, S., Tsang, I.W.H., Chia, L.T.: Laplacian sparse coding, hypergraph laplacian sparse coding, and applications. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 92–104 (2013)
17. Gemert, J.C., Geusebroek, J.M., Veenman, C.J., Smeulders, A.W.: Kernel codebooks for scene categorization. In: ECCV, pp. 696–709 (2008)
18. van Gemert, J.C., Veenman, C.J., Smeulders, A.W.M., Geusebroek, J.M.: Visual word ambiguity. IEEE Trans. Pattern Anal. Mach. Intell. **32**(7), 1271–1283 (2010)
19. Geusebroek, J., Boomgaard, R.v.d., Smeulders, A., Geerts, H.: Color invariance. IEEE Trans. Pattern Anal. Mach. Intell. **23**(12), 1338–1350 (2001)
20. Ghaoui, L.E., Viallon, V., Rabbani, T.: Safe feature elimination in sparse supervised learning. Technical report UC/EECS-2010-126, EECS Department, University of California at Berkeley (2010)
21. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical report, California Institute of Technology (2007)
22. Gu, J., Liu, C.: Local quaternary patterns and feature local quaternary patterns. In: 16th International Conference on Image Processing, Computer Vision, and Pattern Recognition, Las Vegas, Nevada, USA (2012)
23. Gu, J., Liu, C.: Feature local binary patterns with application to eye detection. Neurocomputing **113**, 138–152 (2013)
24. Jegou, H., Perronnin, F., Douze, M., Sanchez, J., Perez, P., Schmid, C.: Aggregating local image descriptors into compact codes. IEEE Trans. Pattern Anal. Mach. Intell. **34**(9), 1704–1716 (2012)
25. Ke, Y., Sukthankar, R.: Pca-sift: a more distinctive representation for local image descriptors. In: IEEE International Conference on Computer Vision and Pattern Recognition, Washington, DC, vol. 2, pp. 506–513 (2004)
26. Kubelka, P.: New contribution to the optics of intensely light-scattering materials, part i. J. Opt. Soc. Am. **38**(5), 448–457 (1948)

27. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: CVPR, pp. 2169–2178 (2006)
28. Lee, H., Battle, A., Raina, R., Ng, A.Y.: Efficient sparse coding algorithms. In: NIPS, pp. 801–808 (2007)
29. Liu, C.: Gabor-based kernel PCA with fractional power polynomial models for face recognition. IEEE Trans. Pattern Anal. Mach. Intell. **26**(5), 572–581 (2004)
30. Liu, C.: Effective use of color information for large scale face verification. Neurocomputing **101**, 43–51 (2013)
31. Liu, C., Mago, V. (eds.): Cross Disciplinary Biometric Systems. Springer, Berlin (2012)
32. Liu, C., Wechsler, H.: Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition. IEEE Trans. Image Process. **11**(4), 467–476 (2002)
33. Liu, Q., Puthenputhussery, A., Liu, C.: A novel inheritable color space with application to kinship verification. In: the IEEE Winter Conference on Applications of Computer Vision, Lake Placid, New York (2016)
34. Liu, Z., Liu, C.: Fusion of color, local spatial and global frequency information for face recognition. Pattern Recognit. **43**(8), 2882–2890 (2010)
35. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
36. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: ICML, p. 87 (2009)
37. Morel, J., Yu, G.: Asift: a new framework for fully affine invariant image comparison. SIAM J. Imaging Sci. **2**(2), 438–469 (2009)
38. Ojala, T., Pietikainen, M., Harwood, D.: Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In: Proceedings of the 12th IAPR International Conference on Pattern Recognition, Jerusalem, Israel, pp. 582–585 (1994)
39. Ojala, T., Pietikainen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. Pattern Recognit. **29**(1), 51–59 (1996)
40. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7), 971–987 (2002)
41. Pang, Y., Lia, W., Yuanb, Y., Panc, J.: Fully affine invariant surf for image matching. Neurocomputing **85**(8), 6–10 (2012)
42. Sanchez, J., Perronnin, F., Mensink, T., Verbeek, J.J.: Image classification with the fisher vector: theory and practice. Int. J. Comput. Vis. **105**(3), 222–245 (2013)
43. Sinha, A., Banerji, S., Liu, C.: New color GPHOG descriptors for object and scene image classification. Mach. Vis. Appl. **25**(2), 361–375 (2014)
44. Verma, A., Liu, C.: Novel EFM- KNN classifier and a new color descriptor for image classification. In: 20th IEEE Wireless and Optical Communications Conference (Multimedia Services and Applications), Newark, New Jersey, USA (2011)
45. Verma, A., Liu, C., Jia, J.: New color SIFT descriptors for image classification with applications to biometrics. Int. J. Biom. **1**(3), 56–75 (2011)
46. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T.S., Gong, Y.: Locality-constrained linear coding for image classification. In: CVPR, pp. 3360–3367 (2010)
47. Wang, J., Zhou, J., Liu, J., Wonka, P., Ye, J.: A safe screening rule for sparse logistic regression. In: Advances in Neural Information Processing Systems, pp. 1053–1061 (2014)
48. Wang, J., Zhou, J., Wonka, P., Ye, J.: Lasso screening rules via dual polytope projection. In: Advances in Neural Information Processing Systems, pp. 1070–1078 (2013)
49. Xiang, Z., Xu, H., Ramadge, P.: Learning sparse representations of high dimensional data on large scale dictionaries. Adv. Neural Inf. Process. Syst. **24**, 900–908 (2011)
50. Xiang, Z.J., Ramadge, P.J.: Fast lasso screening tests based on correlations. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012, pp. 2137–2140 (2012)
51. Xin, X., Li, Z., Katsaggelos, A.: Laplacian sift in visual search. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, Kyoto, Japan (2012)

52. Yan, S., Xu, D., Zhang, B., Zhang, H.J., Yang, Q., Lin, S.: Graph embedding and extensions: a general framework for dimensionality reduction. IEEE Trans. Pattern Anal. Mach. Intell. (2007)
53. Yang, J., Yu, K., Gong, Y., Huang, T.S.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR, pp. 1794–1801 (2009)
54. Zhang, S., Tian, Q., Lu, K., Huang, Q., Gao, W.: Edge-sift: discriminative binary descriptor for scalable partial-duplicate mobile search. IEEE Trans. Image Process. **29**(1), 40–51 (2013)
55. Zheng, M., Bu, J., Chen, C., Wang, C., Zhang, L., Qiu, G., Cai, D.: Graph regularized sparse coding for image representation. IEEE Trans. Image Process. **20**(5), 1327–1336 (2011)
56. Zhou, X., Yu, K., Zhang, T., Huang, T.S.: Image classification using super-vector coding of local image descriptors. In: Proceedings of the 11th European Conference on Computer Vision: Part V, pp. 141–154 (2010)