The Visual Word Booster

DOI:
10.1109/TIP.2018.2830127

Citation for published version (APA):
Dong, X., & Dong, J. (2018). The Visual Word Booster: A Spatial Layout of Words Descriptor Exploiting Contour Cues. *IEEE Transactions on Image Processing*. https://doi.org/10.1109/TIP.2018.2830127

Published in:
IEEE Transactions on Image Processing

The Visual Word Booster: A Spatial Layout of Words Descriptor Exploiting Contour Cues

Xinghui Dong, and Junyu Dong

Abstract—**Although researchers have made efforts to use the spatial information of visual words to obtain better image representations, none of the studies take contour cues into account. Meanwhile, it has been shown that contour cues are important to the perception of imagery in the literature. Inspired by these studies, we propose to use the Spatial Layout of Words (SLoW) to boost visual word based image descriptors by exploiting contour cues. Essentially, the SLoW descriptor utilises contours and incorporates different types of commonly used visual words, including hand-crafted basic contour elements (referred to as "contons"), textons and Scale-Invariant Feature Transform (SIFT) words, deep convolutional words and a special type of words: LBP (Local Binary Pattern) codes. Moreover, SLoW features are combined with Spatial Pyramid Matching (SPM) or Vector of Locally Aggregated Descriptors (VLAD) features. The SLoW descriptor and its combined versions are tested in different tasks. Our results show that they are superior to, or at least comparable to, their counterparts examined in this study. In particular, the joint use of the SLoW descriptor boosts the performance of the SPM and VLAD descriptors. We attribute these results to the fact that contour cues are important to human visual perception and, the SLoW descriptor captures not only local image characteristics but also the global spatial layout of these characteristics in a more perceptually consistent way than its counterparts.**

Index Terms—**Visual words, contours, image descriptors, image features, spatial layout**

I. Introduction

BAG-of-Words (BoW) [39], [64], [67] has been one of the most known image descriptors since firstly introduced in 1999 [39]. However, BoW descriptors are "orderless" because they discard the spatial layout of words [38]. In fact to humans, an image is a meaningful arrangement of local regions and objects rather than only a random mixture of pixels or regions [39]. In a user study, Dong and Chantler [17] showed that human observers tend not to recognise textures when their spatial layouts are scrambled. In addition, Sharan *et al.* [62] found that human observers cannot effectively recognise material categories from locally-ordered but globally-orderless images. These studies reveal the importance of modelling global image layout to the perception of images.

It is known that aperiodic image structure data is retained in global higher order statistics (HOS), e.g. the phase spectrum, rather than the power spectrum [48-49]. Dong *et al.* [17], [19] surveyed 51 texture descriptors and concluded that they only compute HOS on small spatial extent (≤19×19 pixels), if at all (some only utilise the power spectrum). Unfortunately, global phase information is difficult to use due to the phase unwrapping issue [75]. It should be noted that Fourier phase congruency [35] within local image regions does not provide this kind of global phase information [55]. The encoding of the phase data cannot be a local process if it is to be used to capture globally coherent structure. Instead, the relationship between the local characteristics in one image region and those in nearby or even faraway regions needs to be exploited [55]. Accordingly, encoding local spatial data into words [36], [44], [59] and modelling global spatial layout of words [7], [34], [38], [58], [73] have been exploited in order to alleviate the "orderlessness" issue that BoW [64] descriptors encounter. Nevertheless, none of these methods take contour cues into account. In essence, this type of data mainly considers the global layout of local characteristics across different spatial locations and is useful for image discrimination based on the global spatial structure [55].

In this situation, the exploitation of the contour cues in images provides a possible solution to encoding global spatial layouts [26-27], [55]. It has been highlighted in the literature [15], [18], [23], [51], [68] that contour cues are important to human visual perception of imagery. Dong and Chanter [18] examined the importance of contour cues to texture perception. On the basis of contours, they proposed a descriptor, namely, Perceptually Motivated Image Features (PMIF). However, three problems are still remained: (1) PMIF cannot represent small contours well; (2) PMIF cannot encode an image which does not contain the obvious structure; and (3) the longer-range spatial relationship across contours should be exploited.

Motivated by the studies mentioned above, we introduce a new global image descriptor (see Fig. 1 for pipeline), which exploits the spatial layout of words (referred to as "SLoW"). Compared with BoW descriptors [39], [64], [67], this descriptor captures the global spatial layout by encoding the spatial relationship between words both within the same contour and without regard to contours. To our knowledge, this mechanism has not been addressed in the studies [7], [34], [36], [38], [44], [58-59], [73] of incorporating the spatial data into BoW descriptors. The SLoW descriptor exploits HOS over longer ranges compared to the methods [36], [44], [59] that

J. Dong is supported by the National Natural Science Foundation of China (NSFC) (No. 61271405). (Corresponding author: Junyu Dong).

X. Dong is with the Centre for Imaging Sciences, University of Manchester, Manchester, M13 9PT, UK (e-mail:dongxinghui@gmail.com). The majority of this work was complete between taking this appointment and his PhD viva.

J. Dong is with the Department of Computer Science, Ocean University of China, Qingdao, China (e-mail: dongjunyu@ouc.edu.cn).

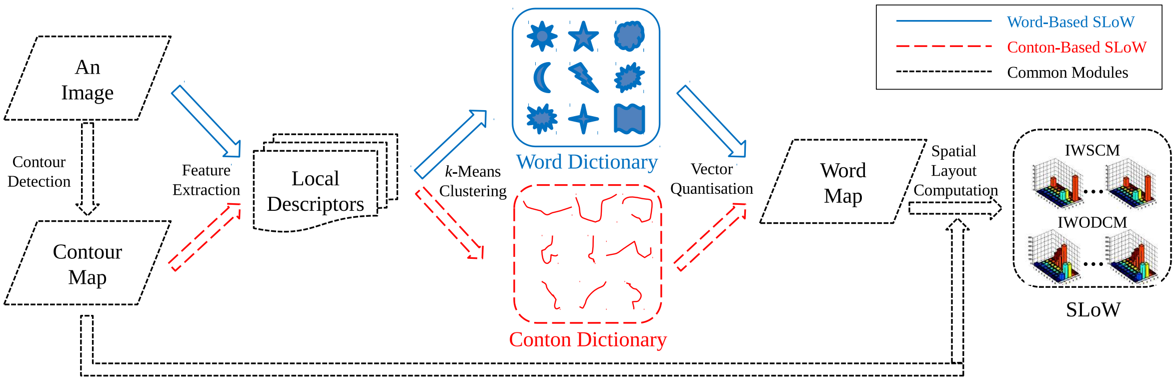

Fig. 1. The two pipelines (shown in red dash and blue solid entities) of the computations of the conton-based and word-based SLoW descriptors respectively. Contons are learnt from the image patches sampled from the contour map while other types of visual words are learnt from the local features extracted from the original image, e.g. raw image patches [67], SIFT [42] and deep convolutional features [10]. Note that *k*-means clustering and vector quantisation are replaced by the LBP [47] computation and the dictionary is replaced by an LBP codebook when LBP codes [47] are used. Please refer to Figs. 5 and 6 for details of the computation of IWSCM (Intra-word Shape Context Matrices) and IWODCM (Intra-word Orientation and Distance Co-occurrence Matrices).

encode the local spatial data into words, and utilises more perceptually consistent information than that the approaches [34], [58], [73] modelling other types of spatial layouts of words use via encoding the shape of contours.

We first test the SLoW descriptor together with the basic contour elements: contons learnt from contour maps. Then, we examine this descriptor along with different types of commonly used visual words, including textons [67], SIFT words [42], deep convolutional words [10] and a special type of words: LBP codes [47]. In addition, the SLoW descriptor is combined with the SPM (Spatial Pyramid Matching) [38] or VLAD (Vector of Locally Aggregated Descriptors) [32] descriptors in order to exploit the merits of both SLoW and SPM or VLAD.

The contributions of this paper are: (1) the definition and utilisation of “contons”; (2) the proposal of the SLoW descriptor by exploiting contour cues; and (3) the indication of the importance of the spatial layout of words exploiting contour cues to visual word based image descriptors. The remainder of this paper is organised as follows. The next section investigates the related work. We then describe the SLoW descriptor and its combined versions in Section III. In Sections IV and V, we test SLoW in different tasks using contons and the other types of words, respectively. Conclusions are drawn in Section VI.

II. Related Work

A. *Bag-of-Words (BoW) Image Descriptors*

Leung and Malik [39] originally applied the bag-of-words (“textons”) descriptor to texture recognition. Normally, BoW descriptors are computed in three stages. First, local features [6], [10], [14], [42], [50], [67], [70], [72] are extracted from images. Second, a word dictionary is learnt from a subset of these features. Third, either vector quantisation [39], [64], [67] or soft assignment [56], [71], [74] is used to map the local features to a “bag” of word labels. A histogram is finally accumulated from the occurrence frequencies of these labels. In addition, the local binary pattern (LBP) methods [47] can be considered as a type of BoW descriptors. However, the descriptors mentioned above discard the spatial layout of local features (encoded by words) and are “orderless” [38].

B. *Merging the Spatial Information into BoW*

Merging the spatial information into BoW approaches has been used to boost their performance. These studies can be divided into two categories (or the hybrid of these [28]): encoding local spatial information into words [36], [44], [59] and modelling global spatial layout of words [7], [38], [58], [73]. The methods of the former category normally use the descriptors which encode local spatial information, such as spatial mean and variance of local image regions [36] and pairs of spatially close SIFT (Scale-Invariant Feature Transform) features [44]. However, these methods do not exploit the global spatial relationship between words and their histogram based feature vectors are still “orderless”.

In terms of the approaches in the second category, they encode the global spatial layout of words using the spatial partitioning [7], [38] of images or the spatial relationship between words [58], [73]. However, the latter has been given less attention. This may be due to the high computational cost. Hence, some methods put their emphasis on reducing the size of word dictionaries [60] or accelerating computational speed [41]. Nevertheless, these methods only capture the relatively local spatial relationship between words. In this context, the global spatial layout cannot be modelled. Khan *et al.* [34] used the orientation of word pairs to encode intra-word spatial layout in longer range but they ignored the distances between words.

To summarise, none of the image descriptors reviewed above take contour cues into account. It is also the case for spatial verification techniques [53], [64] which were used to check the geometrical consistency between two images. Nevertheless, contours are normally comprised of the global structure of images. Thus, the spatial layout of words exploiting contours is important to global image structure discrimination [55].

C. *Contour-Based Image Descriptors*

Contour-based shape descriptors [76] normally encode only a single contour without considering the spatial relationship between contours. Although the bag-of-contour fragments

(BCF) [5], [69] and bag-of-boundaries (BoB) descriptors [3] were successfully used for shape recognition or object retrieval, the histogram features are "orderless". The above descriptors are thus not suitable for capturing the complicated structure information encoded in the contour map of natural images. Ferrari *et al.* [22] and Lim *et al.* [40] learnt different basic contour element sets. However, these elements often contain more than one segment and thereby cannot be used to represent the shape of a single contour. Compared with the PMIF descriptor [18] based on contours, this paper addresses its open issues by representing local contour segments in a more powerful way via conton learning, and modelling the global spatial layout of words in longer ranges.

Contour features were also applied to image registration. Ni *et al*. [46] introduced a two-stage registration scheme: a coarse registration was first conducted based on the contour matching method, and in the second stage, a fine registration was performed using SIFT [42] and a local matching scheme. Differing from the direct use of contours in [46], we introduce an image descriptor by incorporating contour cues into various visual word based descriptors, encoding the words learnt from image patches [67], SIFT features [42], deep convolutional features [10], LBP codes [47], etc.

III. The Spatial Layout of Words (SLoW) Descriptor

Inspired by the importance of contour cues to human visual perception of imagery [15], [18], [23], [51], [68], we introduce a global image descriptor (see Fig. 1 for pipeline) which exploits the spatial layout of words (SLoW) based on contours. First, a contour map is derived from the image in order to model the spatial layout of words. Second, we learn a set of visual words, including contons, textons [67], SIFT words [42] and deep convolutional words [10], from their corresponding local features. In addition, LBP codes [47] are regarded as a special set of words. Third, the spatial layout of those words is modelled via encoding their spatial relationship both within the same contour and without regard to contours. We also combine the SLoW descriptor with other descriptors in order to boost their performance.

A. *Obtaining Contour Maps*

Since we intend to show the importance of contour cues to representation of the spatial layout of words rather than comparing different contour detectors, we only use the *Canny* edge detector [8] to obtain edge maps. For purposes of deleting redundant pixels but keeping continuous edges, erosion operations [24] are applied to the edge map. The derived contour skeleton maps (see Fig. 2) are used instead of edge maps. For simplicity, the term "contour map" is used to refer to the skeleton map. Considering the fact that contour branches make contour representation more complicated, we locate all branch points and break each involved contour into a set of contours through removing these points. By performing connected component labelling and applying the Moore-Neighbour tracing algorithm [24], each contour is traversed from end to end, to derive the exterior boundary sequence of the contour. The traversing sequence of a contour is finally derived from the boundary sequence [18].

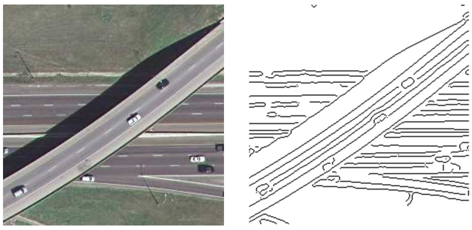

Fig. 2. A land use image [73] and its contour (skeleton) map.

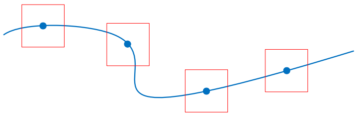

Fig. 3. Sampling of local contour image patches collinearly.

B. *Deriving a Bag of Words*

Visual words are normally learnt from hand-crafted local features [6], [14], [42], [67]. However, the advantage of visual words learnt from deep convolutional features over traditional ones has been addressed [10], [72]. We hence use contour image patches, original image patches [67], SIFT features [42] and deep convolutional features [10] to learn corresponding visual words, i.e. contons, textons, SIFT words and deep words.

1) *Extracting Local Features*

Using Contour Image Patches We learn a set of basic contour elements from contour image patches to represent their local shape characteristics. These basic contour elements are referred to as "contons", in a similar fashion to "textons" [39], [67]. Contons can be regarded as the words learnt from contour maps. In human vision science, it is known that human observers can perceive a contour which comprises a set of disjointed collinear line segments with a blank background [15] or a set of collinear Gabor elements from a randomly placed Gabor elements field [23]. The strong representation power of image patches has also been shown in computer vision [67] since they encode richer local characteristics than the features extracted from these patches.

Inspired by these findings, we represent a contour segment using its surrounding image patch whose centre locates at the middle point of the segment (see Fig. 3), i.e. a contour image patch. The contour image patch retains the original local shape information. It is noteworthy that only a single contour is considered at a time. In other words, we separate each contour from all other contours when it is processed. As a result, we eliminate the influence of other contours on the representation of the current contour. We collinearly sample image patches (see Fig. 3) as this reduces the number of samples used for learning contons but introduces fewer variations of image patches, compared with sampling in the raster-scan order. Also, the central point of a patch sampled using our strategy can be used to approximate the position of the segment contained in this patch. This approximation is useful for describing the shape of the contour from which image patches are sampled.

The width *N* of contour image patches was set as 5, 7, or 9 pixels in this study. The reasons for using small patches are that

(1) we intend to encode basic local shape characteristics rather than global shape information using contons; (2) the computational speed of k-means clustering is faster when it is used along with small image patches to learn contons; and (3) the size of the reasonable conton dictionary learnt using small patches is smaller than that learnt using large patches because the variation of the shapes in the former is fewer than that in the latter. This reduces the dimensionality of final feature vectors.

Using Original Image Patches and SIFT Features It has been shown that textons learnt from original image patches outperformed those learnt from filter responses in the scenario of texture classification [67]. In addition, SIFT [42] features have been widely used to learn visual words for various computer vision applications. Therefore, we examined both types of words together with the SLoW descriptor in this study. The local image patches [67] and SIFT features [42] are densely extracted from original images in order to learn textons and SIFT words respectively. However, only a random subset of the features extracted from a dataset is used.

Using Deep Convolutional Features Deep convolutional features are extracted using a pre-trained CNN model [9], [31], [63], [66], [77]. We use the approach that Cimpoi *et al.* [10] proposed. An image is first resized into six different images at the scales: 2^s ($s \in$ {-2, -1.5, -1, -0.5, 0, 0.5}). Then, each resized image is subtracted by the average colour of the training dataset and sent to CNN. In total, f feature maps are obtained at the last convolutional layer. Given a location in the feature maps, a f-D feature vector is derived. Next, the location of the features computed at different scales is mapped to the original image. Finally, 64-D feature vectors are generated by applying the L_2 normalisation, PCA, whitening and a second L_2 normalisation to the f-D feature vectors in turn.

2) Learning a Word Dictionary

Given that s feature vectors are extracted and represented as x_i ($i = 1,2 \ldots s$), k-means is used to convert x_i from the feature space X to the dictionary space C (see Equation (1)), in which each feature vector is labelled by a word c_j ($j = 1,2 \ldots w$).

$$X = \{x_1, x_2, \ldots x_s\} \xrightarrow{k-means} C = \{c_1, c_2, \ldots c_w\}. \qquad (1)$$

The words learnt using local contour image patches, original image patches, SIFT features and deep convolutional features are termed as contons (see Fig. 4), textons, SIFT words and deep convolutional words respectively. Compared to textons, SIFT words and deep convolutional words, contons are more suitable for sketch-based image retrieval [20] where query images only contain sketches.

3) Quantising the Local Features

Different strategies [64], [71], [74] can be used to map local features into the word space. Since we are more interested in examining the importance of contour cues to the encoding of the spatial layout of words than investigating different mapping strategies, only vector quantisation (VQ) [64] is applied. Given a dictionary C which comprises w words, each feature vector x_i is compared with every word in C and is assigned the label j ($j \in \{1,2, \ldots w\}$) of the word which lies closest to the feature vector in the space X. This process is described as:

$$q(x_i) = argmin_{c_j \in C}\left\|x_i - c_j\right\|^2, \qquad (2)$$

where $q(x_i)$ is the quantisation function of the feature vector x_i and c_j is a word in the dictionary C. After VQ is complete, each feature vector is assigned a word label and its position is described by the location of the central point of the image region from which this feature vector is extracted.

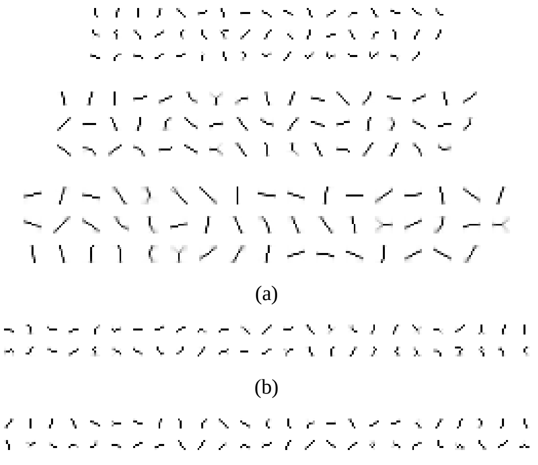

Fig. 4. The contons learnt from different datasets: (a) the 5×5 (top), 7×7 (middle) and 9×9 (bottom) contons learnt from the SBIR dataset [20]; (b) the 5×5 contons learnt from the *Pertex* [12] dataset; and (c) the 5×5 contons learnt from the *JAFFE* dataset [61]. In each group, 50 contons are shown.

C. Deriving a Bag of LBP Codes

The LBP descriptor [47] uses a predefined codebook to quantise local binary patterns (LBP) into a set of codes. The LBP with a circular neighbourhood is defined as:

$$LBP_{P,R} = \sum_{e=0}^{P-1} t\left(r_e - r_f\right) 2^e, \qquad (3)$$

where r_f is the grey level value of the central pixel in a neighbourhood, $r_e (e = 0, 1, \ldots P-1)$ stands for the grey level values of P evenly spaced pixels on a circle of the radius R ($R >$ 0), and $t(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$. Furthermore, the idea of "uniform" was suggested and the grey-scale and rotation invariant description $LBP_{P,R}^{riu2}$ was proposed as

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{e=0}^{P-1} t\left(r_e - r_f\right), & if\ U\left(LBP_{P,R}\right) \leq 2 \\ P+1, & otherwise \end{cases}, \qquad (4)$$

where $U\left(LBP_{P,R}\right) = \left|t\left(r_{e-1} - r_f\right) - t\left(r_0 - r_f\right)\right| + \sum_{e=1}^{P-1} \left|t\left(r_e - r_f\right) - t\left(r_{e-1} - r_f\right)\right|$. We use $LBP_{P,R}^{riu2}$ to compute LBP codes. These codes are considered as a special form of words.

D. Encoding the Spatial Layout of Words

In the case that LBP is used for word generation, each pixel location is assigned an LBP code. In other cases, the pixel location from which a local feature vector is extracted is labelled using a word based on VQ. Due to the importance of contours to perception of imagery [15], [18], [23], [51], [68], we encode the spatial layout of words (including LBP codes) by considering their spatial relationship within the same contour. We also compute the spatial relationship between words

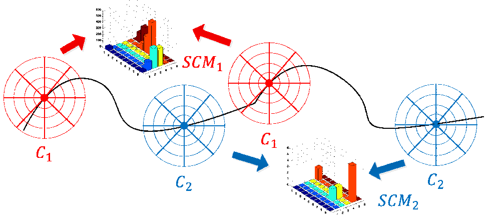

Fig. 5. Computation of shape context matrices (SCMs) for the words: C_1 and C_2 on the same contour. Noted that circles are drawn to show the points used for computing shape contexts rather than indicating the regions involved.

without regard to contours to explore the spatial layout of words across contours.

1) Modelling the Spatial Layout of Words within the Same Contour

The orientation and position of point pairs on a contour are important for describing its shape [6]. We represent the shape characteristics of a contour by encoding the co-occurrence of the orientation and position of word pairs on this contour. The shape context descriptor [6] was designed to capture this type of information. This descriptor computes an $A \times D$ shape context matrix (A angle bins and D *log* distance bins) at each reference point of a contour, encoding the spatial relationship with other reference points. The matching of two contours thus becomes the matching of two series of shape context matrices computed from two reference point sets respectively. When long contours are processed, however, the number of reference points required increases. As a result, the computational cost of matching becomes heavy.

Given a word dictionary learnt from an image dataset, all contour points in a contour map are labelled using words while the non-contour points are ignored. For a contour and a word, we only compute shape contexts from the points labelled by this word. To be specific, we compute a shape context matrix (SCM) of the points in terms of each word and obtain w (number of words) individual $A \times D$ shape context matrices for a contour in total. It should be noted that the distance bins are quantised across all the points (which may be labelled by different words) rather than only the points labelled by the current word on the contour. Fig. 5 shows the computation of the shape context matrices. The w shape context matrices are concatenated into an $A \times D \times w$ intra-word shape context matrix. The merits of this method include that (1) it reduces the computational cost compared with the computation between any pairs of reference points; and (2) it avoids the matching of different reference points as we use a unified word dictionary for all contour points throughout the dataset. In order to further accelerate the computational speed, contour points can be sampled in an interval of several points.

Furthermore, we use w bins of the corresponding BoW [64] histogram to weight the intra-word shape context matrix in order to retain the occurrence frequency information of words. Given an $A \times D \times w$ intra-word shape context matrix computed from a contour, each $A \times D$ sub-matrix is denoted as SCM_j ($j \in \{1,2,\dots w\}$). Meantime, each bin of the BoW histogram computed from the same contour is labelled as $BOWH_j$. Each weight β_j is computed using the formula below:

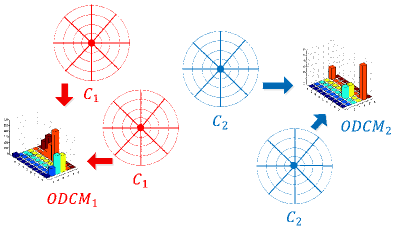

Fig. 6. Computation of the orientation and distance co-occurrence matrices (ODCMs) for the words: C_1 and C_2. Note that circles are drawn to show the points used for computing ODCMs rather than indicating the regions required.

$$\beta_j = BOWH_j / \sum_A \sum_D SCM_j\,,\ j \in \{1,2,\dots w\}. \qquad (5)$$

Then β_j is multiplied by SCM_j to fulfill the weighting.

A weighted intra-word shape context matrix is calculated for each individual contour. All matrices are L_1 normalised to [0,1] separately as different contours may have different lengths. Intuitively, this descriptor encodes the intra-word orientation and position co-occurrences on a contour. It can also be regarded as w co-occurrence matrices of the same word at $A \times D$ different displacements. However, the neighbourhoods used here are contours rather than the square image regions that Haralick *et al.* [29] used. Finally, the mean of all normalised weighted intra-word shape context matrices is computed, which is referred to as "IWSCM", to obtain an average shape representation of all the contours extracted from an image.

It should be noted that Bai *et al.* [5] and Wang *et al.* [69] used shape context [6] to model local contour segments. However, they ignored the spatial relationship between segments. In contrast, we use shape context [6] to describe the global spatial layout of visual words. Therefore, the descriptors that Bai *et al.* [5] and Wang *et al.* [69] proposed are "orderless" local descriptors while the IWSCM descriptor is a global one.

2) Capturing the Spatial Layout of Words without regard to Contours

The spatial relationship between the words across different contours is also important. In addition, the spatial relationship between contours and background (non-contour) regions is useful for many tasks, for example, scene recognition and classification, when original images are used. However, it is challenging to model these types of spatial relationship due to the irregular shape of contours. Hence, we consider the spatial relationship between words without considering contours. Contour points are only used when contons are applied while both contour and non-contour points are used when the other types of words are utilised. We calculated the co-occurrence of the orientation and position of word pairs instead of describing the shape of contours. We only encode the spatial relationship between word pairs labelled by the same word to reduce the computational cost. The other computation, quantisation and weighting operations involved are the same as those used in the previous subsection. An $A \times D \times w$ weighted intra-word orientation and distance co-occurrence matrix (referred to as "IWODCM") is finally obtained (see Fig. 6 for more details).

It is noteworthy that distance bins are computed across the positions of all words rather than those of the current word to obtain a global representation. To reduce the computational cost, only a subset of the positions is used. Different sampling strategies may be considered, such as using a fixed number or ratio of positions for all words. When the words except contons and deep visual words are utilised, we use a fixed ratio (20%) of points for all words. In contrast, when contons are used, the rough shape of each contour needs to be retained as only contour points are considered while the background is blank. In this case, the sampling strategies above may be inaccurate as the optimal number of contons is unknown; while the use of a fixed ratio of contours is not suitable either because contours cannot be detected accurately. We sample contour image patches in an interval of five points collinearly. This reduces the number of sampled contour image patches to approximately 20% but the rough shape of contours is retained. In contrast, smaller intervals will enhance the number of sampled image patches which increases the computational cost of dictionary learning while larger intervals cannot retain the shape of contours.

3) *Generating SLoW Feature Vectors*

Considering the advantages of Spatial Pyramid Matching (SPM) [38], different levels (L) of spatial pyramids can be applied to the IWSCM and IWODCM descriptors. On each level of the pyramids, the IWSCM and IWODCM features are separately extracted and weighted in the same way as that Lazebnik *et al.* [38] proposed. The IWSCM or IWODCM features extracted at all pyramid levels are concatenated into a feature vector. The IWSCM and IWODCM feature vectors are first L_1 normalised to [0,1] and then concatenated into an individual feature vector. This feature vector is referred to as the "SLoW" (Spatial Layout of Words) descriptor, which encodes both the average spatial layout of words computed within the same contour and the spatial layout of words calculated without regard to contours. Therefore, the SLoW descriptor exploits short-range, medium-range and long-range image characteristics in terms of the exploitations of local image regions, contours and images, respectively.

E. *Combining SLoW with other Descriptors*

The SLoW descriptor can be combined with other image descriptors in order to exploit the merits of these descriptors. In this subsection, we introduce three different combinations.

When the SLoW descriptor is computed at multiple levels of spatial pyramids, the dimensionality of the feature vector is high (e.g. this value is $A \times D \times w \times 2 \times 21$ when $L = 3$). This limits the practical application of SLoW to a large dataset. In this situation, we combine the SLoW feature vector extracted at the original resolution ($L = 1$) with the SPM-based BoW feature vector ($L = 3$). This descriptor is named "SLoW+SPM". Using this strategy, the dimensionality of the feature vector is reduced while the merit of SPM is retained.

The VLAD [32] descriptor computed on the basis of the difference between words and local features has shown advantages over the traditional BoW [64] descriptor. Therefore, we also combine the SLoW ($L = 1$) with VLAD feature vectors to exploit this advantage. Correspondingly, this descriptor is referred to as "SLoW+VLAD".

When contons are used, the SLoW descriptor only utilises the contour data. Dong and Chanter [18] have shown that their contour-based descriptor can be boosted by incorporating the contrast data (if this is applicable). We are therefore inspired to combine the conton-based SLoW descriptor with a local variance measure $VAR_{P,R}$ [47]. This measure is expressed as:

$$VAR_{P,R} = \frac{1}{P}\sum_{e=0}^{P-1}(g_e - \mu)^2, \mu = \frac{1}{P}\sum_{e=0}^{P-1} g_e, \qquad (6)$$

where g_e denotes the grey values of P evenly spaced pixels on a circle of the radius R ($R > 0$). We use the multi-scale $VAR_{P,R}$ ("MSVAR") with $(P, R) = (8,1), (16,2)\ \&(24,3)$. A 128-bin histogram is obtained at each scale. In total a 384-bin MSVAR histogram is derived. The MSVAR histogram is L_1 normalised and concatenated with conton-based SLoW features ($L = 1$). This descriptor is referred to as "SLoW+VAR".

TABLE I
KENDALL'S CORRELATION COEFFICIENTS (τ) CALCULATED BETWEEN COMPUTATIONAL AND HUMAN PERCEPTUAL RANKINGS

DESC	SHoG	T&HoG	CCH	SC	PMIF
τ	0.277 [20]	0.223 [20]	-0.014	-0.048	0.209

DESC	BoW	SPM	VLAD	SLoW	SLoW+SPM	SLoW+VLAD
τ	0.178	0.263	0.203	**0.295**	0.269	0.228

Values in bold are the highest τ in the table and our methods are always highlighted in grey (this continues in the following tables).

IV. EXPERIMENTS USING THE SLOW DESCRIPTOR BASED ON CONTONS

In this section, we test the conton-based SLoW descriptor and the two combined descriptors: SLoW+SPM and SLoW+VLAD in four tasks. The first three tasks are a sketch-based image retrieval (SBIR) [20] and two perceptual texture similarity estimation applications [17], [19]. The reasons for using these tasks include: (1) the conton-based SLoW is particularly suitable for SBIR as only contour data is available with query images; (2) we intend to compare the conton-based SLoW with PMIF [18] in the perceptual texture similarity estimation tasks; and (3) the datasets used for the three tasks contain humans' data, which can be used to validate the perceptual consistency for the SLoW descriptor and its combined versions. Due to the importance of the shape data to human face representation, we further apply these descriptors to a fourth task: human facial expression recognition [61].

We use the BoW [64], CCH (Chain Code Histogram) [31], SC (Shape Context) [6], PMIF [18], SPM [38] and VLAD [32] descriptors as baselines for each task. There may also be other baselines obtained from related publications. The SPM [38] and SLoW features are extracted from conton label maps with three spatial pyramid levels while the CCH [31], SC [6] and PMIF [18] features are computed from contour maps. Since the CCH and SC descriptors encode an individual contour, we first compute an L_1 normalised CCH or SC feature vector from each contour and then use the average of all the feature vectors computed from a contour map as the final feature vector. In addition, we learn a conton dictionary from human-drawn sketches [4] and examine its generality to different datasets.

A. *Sketch-Based Image Retrieval (SBIR)*

The method that Eitz *et al.* [20] proposed was used in this experiment. In total 31 human-drawn sketch maps were used as

TABLE II

KENDALL'S CORRELATION COEFFICIENTS (τ) OBTAINED USING SIX CONTON-BASED DESCRIPTORS AT DIFFERENT NEIGHBOURHOOD SIZES

N	5	7	9	5	7	9	5	7	9
DESC	BoW			SPM			VLAD		
τ	0.177	0.177	0.168	0.259	0.259	0.263	0.170	0.194	0.203
DESC	SLoW			SLoW+SPM			SLoW+VLAD		
τ	**0.294**	0.289	0.283	0.260	0.262	0.259	0.198	0.212	0.228

TABLE III

KENDALL'S CORRELATION COEFFICIENTS (τ) DERIVED USING CONTON-BASED METHODS WITH DIFFERENT NUMBERS OF CONTONS ($A = 9, D = 6$ & $N = 5$)

DESC	BoW		SPM		VLAD	
w	50	100	50	100	50	100
τ	0.177	0.178	0.259	0.260	0.170	0.161
DESC	SLoW		SLoW+SPM		SLoW+VLAD	
w	50	100	50	100	50	100
τ	0.294	**0.295**	0.260	0.269	0.198	0.190

TABLE IV

KENDALL'S CORRELATION COEFFICIENTS (τ) OBTAINED USING SLOW WITH DIFFERENT ANGLE AND DISTANCE BINS ($N = 5$ & $w = 50$)

		D				
		1	3	6	12	18
A	9	0.252	0.251	**0.294**	0.289	0.288
	18	0.246	0.246	0.293	0.285	0.283
	36	0.240	0.239	0.282	0.271	0.271

query images and 40 images were retrieved in terms of each query image. Humans' rankings were used as the ground-truth data and Kendall's correlation coefficient ($\partial = 0.05$) was used as the performance metric. Computational rankings were derived using each descriptor and histogram intersection [65]. We also used the top two best descriptors: SHoG and T&HOG that Eitz *et al.* [20] tested as baselines. The results obtained using eight baselines and three SLoW-based descriptors are shown in Table I. It can be observed that (1) both the SPM [38] and SLoW descriptors outperformed BoW; (2) SLoW+SPM and SLoW+VLAD performed better than SPM [38] and VLAD [32] respectively; and (3) the SLoW descriptor produced more consistent results with humans' rankings than its counterparts (including two shape descriptors: CCH [31] and SC [6]).

In the rest of this subsection, we further examine the SLoW-based descriptors in three different aspects.

Neighbourhood Size Table II lists the τ values obtained using BoW [64], SPM [38], VLAD [32] and SLoW-based descriptors when three different neighbourhood sizes were used ($A = 9, D = 6$, and $w = 50$). It can be seen that the SLoW descriptor always outperformed all its counterparts in the same conditions. In addition, SLoW+SPM and SLoW+VLAD performed better than SPM [38] and VLAD [32] respectively.

Number of Contons Table III shows the τ values derived using BoW [64], SPM [38], VLAD [32] and SLoW-based descriptors when different numbers of contons were used ($A = 9, D = 6$ & $N = 5$). As can be seen, the SLoW descriptor outperformed all its counterparts in the same conditions while the SPM [38] and VLAD [32] descriptors performed worse than SLoW+SPM and SLoW+VLAD respectively. Although the performance of those descriptors using 100 contons may be slightly better than that obtained using 50 contons, we will examine SLoW using 50 contons due to the lower feature dimensionality.

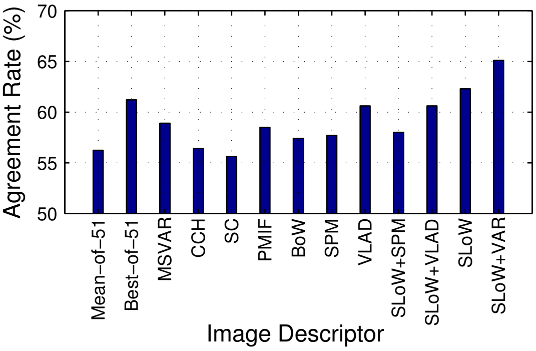

Fig. 7. The agreement rates (%) obtained using nine baselines and four SLoW descriptors against a set of humans' perceptual pair-of-pairs judgements [18].

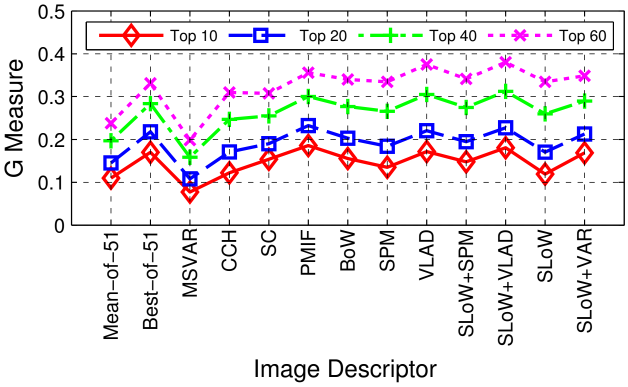

Fig. 8. The G measure values obtained using nine baselines and four SLoW descriptors compared with human perceptual texture rankings [19].

Size of Angle and Distance Bins Table IV displays the τ values obtained using the SLoW descriptor when different numbers of angle and distance bins were used ($N = 5$ and $w = 50$). It can be seen that SLoW obtained its highest performance when nine angle and six distance bins were used.

We therefore mainly examine the SLoW descriptor and its combined versions with the parameters: $N = 5$, $A = 9$, $D = 6$ and $w = 50$ in the following experiments.

B. *Perceptual Texture Similarity Estimation*

Perceptual texture similarity estimation is key to different tasks, including measuring the perceived difference between textures and ranking a set of textures [18]. Dong *et al.* [17], [19] proposed two perceptual texture similarity estimation tasks: pair-of-pairs comparison and texture retrieval. In the former task, two pairs of textures are sent to the algorithm. The output is the decision on which pair is more similar. In the latter task, the algorithm needs to sort the other textures in the database according to their similarity to the query texture. Humans' pair-of-pairs judgements (the left or right pair is more similar) [11] and texture rankings [19] derived from 334 *Pertex* textures [12] were used as ground-truth for the two tasks respectively.

We used the agreement rate (%) to measure the consistency between the human perceptual and computational pair-of-pairs judgements in the pair-of-pairs comparison task [17]. When texture retrieval [19] was conducted, we compared the rankings of the human derived and computational retrievals using the G

TABLE V
THE AVERAGE RECOGNITION RATES (%) AND CORRESPONDING STANDARD DEVIATIONS OBTAINED USING TEN BASELINES AND FOUR SLOW-BASED DESCRIPTORS FOR HUMAN FACIAL EXPRESSION RECOGNITION [61]

DESC	SVM (linear)	SVM (polynomial)	SVM (RBF)	
RR (%)	79.8 [61]	79.8 [61]	81.0 [61]	
DESC	CCH	SC	PMIF	
RR (%)	21.50±1.68	37.98±2.27	63.00±2.12	
DESC	SPM	VLAD	MSVAR	BoW
RR (%)	79.72±2.84	21.97±0.85	37.56±1.52	34.55±2.40
DESC	SLoW+SPM	SLoW+VLAD	SLoW+VAR	SLoW
RR (%)	79.86±1.84	29.01±2.11	86.01±1.38	**86.76±1.75**

measure ($G \in [0, 1]$) [21]. We run this task for the top $N \in \{10,20,40,60\}$ retrieved textures. For the purpose of efficiency, all 334 texture images were down-sampled using Gaussian pyramid to the resolution of 512×512 pixels and only this resolution was used. (Five single pyramid levels and multi-pyramid were used in [17], [19]). However, all the other conditions that Dong *et al.* [17], [19] used were kept constant.

Except the six baselines introduced at the beginning of Section IV, the best and the average of the 51 results obtained by Dong *et al.* [17], [19] and the multi-scale $VAR_{P,R}$ (MSVAR) [47] were also used as baselines. In addition, the SLoW+VAR descriptor was tested. (This descriptor was not tested in the SBIR experiment as only the sketch data is available with query images). Fig. 7 shows the performance derived using the nine baselines and four SLoW-based descriptors in the pair-of-pairs comparison task. It can be seen that (1) the performances of SPM [38], VLAD [32] and MSVAR [47] were boosted by incorporating the SLoW features into these; (2) both SLoW+VAR and SLoW outperformed the other descriptors (including the best one tested in [17]); and (3) SLoW+VAR produced the best performance. Furthermore, Fig. 8 shows the G measure values derived using the same methods as shown in Fig. 7 for the texture retrieval task. As can be observed, the performances of SPM [38], VLAD [32] and MSVAR [47] were improved when combined with SLoW. Nevertheless, the SLoW+VLAD descriptor outperformed all its counterparts.

In [16], Dong discussed the relationship between pair-of-pairs comparison and texture retrieval. It was pointed out that texture retrieval mainly examines the ability of feature descriptors to estimate intra-cluster texture similarity when small numbers of textures are retrieved. This should account for the difference in the performances of SLoW descriptors obtained in the two experiments. By comparing the results shown in Figs. 7 and 8, it can be found that global image descriptors, e.g. SLoW, are more competent for estimating inter-cluster similarity than intra-cluster similarity. In contrast, the opposite trend can be observed for local descriptors.

C. Human Facial Expression Recognition

In this task, we used the *Japanese Female Facial Expression* (*JAFFE*) database [61] which comprises 213 images. Each of the ten subjects showed three or four examples for one of seven expressions. As Shan *et al.* [61] did, we used *JAFFE* images for a 7-class expression recognition task. However, we used original images rather than their cropped normalised versions [61]. This makes the task more challenging but more applied. A 10-fold cross-validation setup was conducted using Support Vector Machines (SVM) [1], [13]. The recognition rate (RR, %) was used as performance measure. The experiment was performed using each descriptor for ten runs (which was not conducted by Shan *et al.* [61]) to obtain different splitting for cross-validation.

We report the results obtained using ten baselines and four SLoW-based descriptors in Table V. It can be observed that: (1) the performances of SPM [38], VLAD [32] and MSVAR [47] were enhanced when combined with SLoW; (2) the VLAD [32] descriptor did not perform well, which impairs the performance of SLoW+VLAD; (3) the SLoW+VAR and SLoW descriptors outperformed all the other descriptors; and (4) the SLoW descriptor performed the best among the 14 descriptors. It should be noted that both SLoW+VAR and SLoW performed better than the best method tested in [61] even though we did not use the cropped face images. We attribute this result to the importance of shape to representation of human faces [37].

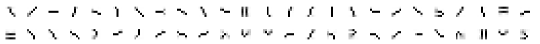

Fig. 9. Fifty 5×5 contons learnt from the human-drawn contours contained in the *BSDS500* dataset [4]. These contons are referred to as “General Contons”.

TABLE VI
KENDALL’S CORRELATION COEFFICIENTS (τ) OBTAINED USING SPM [38] AND SLOW FOR THE SBIR TASK [20] WHEN ORIGINAL AND GENERAL CONTON DICTIONARIES WERE USED

DESC	SPM	SPM-G	SLoW	SLoW-G
τ	0.259	0.257	0.294	0.290

TABLE VII
THE AGREEMENT RATES (%) OBTAINED USING SPM [38] AND SLOW FOR THE PAIR-OF-PAIRS COMPARISON TASK [17] WHEN ORIGINAL AND GENERAL CONTON DICTIONARIES WERE USED

DESC	SPM	SPM-G	SLoW	SLoW-G
AR (%)	57.7	57.8	61.3	60.9

TABLE VIII
THE G MEASURE VALUES ($G \in [0, 1]$) OBTAINED USING SPM [38] AND SLOW FOR PERCEPTUAL TEXTURE RETRIEVAL [19] WHEN ORIGINAL AND GENERAL CONTON DICTIONARIES WERE USED

N	10		20	
DESC	SPM	SPM-G	SPM	SPM-G
G	0.135	0.138	0.184	0.187
DESC	SLoW	SLoW-G	SLoW	SLoW-G
G	0.119	0.118	0.170	0.168
N	40		60	
DESC	SPM	SPM-G	SPM	SPM-G
G	0.266	0.268	0.266	0.268
DESC	SLoW	SLoW-G	SLoW	SLoW-G
G	0.260	0.259	0.260	0.259

D. Does a General Conton Dictionary Exist?

In this subsection, we intend to investigate whether or not a general conton dictionary can be learnt for different image datasets. Arbelaez *et al.* [4] derived a human-drawn image contour dataset: *BSDS500*. This dataset contains 500 contour maps. We learnt 50 5×5 contons from this dataset using the approach introduced in Section III-B-1. For discrimination purposes, we refer to these contons as “General Contons” while

TABLE IX

THE AVERAGE RECOGNITION RATES (%) AND CORRESPONDING STANDARD DEVIATIONS OBTAINED USING SPM [38] AND SLoW FOR HUMAN FACIAL EXPRESSION RECOGNITION [61] WHEN ORIGINAL AND GENERAL CONTON DICTIONARIES WERE USED

DESC	SPM	SPM-G	SLoW	SLoW-G
RR (%)	79.72±2.84	45.87±1.80	**86.76±1.75**	70.14±2.16

TABLE X

THE AVERAGE CLASSIFICATION RATES (%) AND CORRESPONDING STANDARD DEVIATIONS OBTAINED USING THE BoW [64], SPM [38], VLAD [32] AND SLoW-BASED DESCRIPTORS FOR *Pertex* [12] TEXTURE CLASSIFICATION

	DESC	BoW	SPM	VLAD
1	**CR**	98.71±0.40	97.84±0.77	99.31±0.53
6	**(%)**	98.39±0.27	97.21±0.21	99.08±0.13
	DESC	SLoW	SLoW+SPM	SLoW+VLAD
1	**CR**	97.84±0.58	97.07±0.89	**99.70±0.24**
6	**(%)**	97.71±0.26	96.19±0.36	**99.43±0.07**

The number of test images was set as 1 and 6.

the contons learnt from each dataset are named “Original Contons”. Fig. 9 displays the 50 general contons. We repeated the four experiments reported in the previous subsections by replacing original contons with the 50 general contons. To be exact, all conditions were kept constant except that the 50 general contons were used. The parameters of the SLoW descriptor were set as $N=5$, $A=9$, $D=6$ and $w=50$. For simplicity, we only tested the SPM [36] and SLoW descriptors.

We compare the performances derived using original and general contons in this subsection. Tables VI, VII, VIII and IX report the comparison results for the four tasks, respectively, when original and general contons are used. As can be seen, the SPM [38] and SLoW descriptors performed comparably when the two conton dictionaries were used in the SBIR [20] and two perceptual texture similarity estimation tasks [17], [19]. However, this is not the case when the human facial expression recognition task [61] was conducted. In this case, both SPM [38] and SLoW performed better when original contons were used than that they performed when the general contons were used. It is noteworthy that the contours extracted from human face images [61] are more different from human-drawn contours [4] than those extracted from the SBIR [20] or *Pertex* [12] datasets. This finding should account for the difference between the two sets of results obtained using SPM [38] or SLoW when original and general human face contons were used respectively.

E. Summary

In this section, we first tested the SLoW descriptor and its combined versions in four tasks. This descriptor performed better than, or at least comparably to, the baselines. It is also complementary to the SPM [38], VLAD [32] and MSVAR [47] descriptors. We further investigated the universalisability of contons. It was found that the contons learnt from human-drawn contours [4] can be generalised to the SBIR [20] and *Pertex* [12] datasets. However, it was not the case for the *JAFFE* dataset [61]. This result is attributed to the difference between the SBIR [20], *Pertex* [12] and human-drawn [4] contours and the human face contours extracted from the *JAFFE* dataset [61]. It is noteworthy that the conton-based SLoW only uses the contour data. However, the joint use of local contrast data boosted its performance in most cases where these data are applicable. This finding indicates the importance of local non-contour image characteristics. In the next section, we therefore apply the SLoW descriptor along with other types of words which encode richer local information than the contour image patches used by contons.

TABLE XI

THE AVERAGE CLASSIFICATION RATES (%) AND STANDARD DEVIATIONS DERIVED USING THREE BASELINES AND THREE SLoW BASED DESCRIPTORS FOR LAND USE IMAGE CLASSIFICATION [73]

DESC	BoW	SPM	VLAD
***w* = 50**	66.22±0.49	73.15±0.46	82.25±0.35
***w* = 100**	72.55±0.59	77.61±0.39	83.40±0.50
***w* = 200**	76.91±0.48	79.56±0.39	84.78±0.32
DESC	SLoW	SLoW+SPM	SLoW+VLAD
***w* = 50**	76.19±0.33	77.38±0.38	**85.71±0.40**
***w* = 100**	78.60±0.42	79.46±0.39	**86.90±0.28**
***w* = 200**	80.37±0.37	81.38±0.49	**87.32±0.54**

V. EXPERIMENTS USING SLoW DESCRIPTORS BASED ON OTHER TYPES OF WORDS

In this section, we examine the SLoW descriptor derived using other types of words, including textons [67], SIFT words [42], deep convolutional words [10] and LBP codes [47]. The SLoW features were extracted using three spatial pyramid levels for textons, SIFT words and LBP codes. Since local convolutional features had to be sparsely extracted [10], the features sampled on contour points were few, especially, when high spatial pyramid levels were used. In this case, the SLoW features are sparse which impairs their discriminatory power. Therefore, only the original image resolution was used for deep convolutional words. These descriptors are applied to different tasks. For image classificaton tasks, we used the histogram intersection kernel SVM [38] for histogram-based descriptors, including BoW, SPM, SLoW and SLoW+SPM, along with the L_1 normalisation. Without specific statements, the linear kernel SVM (C=10) was used for the other descriptors with the L_2 normalisation, following existing studies [9], [10], [63].

A. Texton-Based SLoW

The *Pertex* dataset [12] was used. Textons [67] were learnt using the approach introduced in Section III-B-1. Each *Pertex* image was devided into 16 equal-sized patches. The number of test patches was set as 1 and 6. The BoW [64], SPM [38] and VLAD [32] descriptors were used as baselines. The results are displayed in Table X. As can be seen, SPM [38] performed worse than the BoW descriptor. Also, the SLoW and SLoW+SPM descriptors performed worse than BoW and SPM respectively. However, SLoW+VLAD performed better than all its counterparts. It is noteworthy that the VLAD descriptor does not encode the spatial data. This may account for the complementary performances of SLoW and VLAD.

B. SIFT Word Based SLoW

The densely sampled SIFT features [42] were used to learn words (see Section III-B-1). The BoW [64], SPM [38] and VLAD [32] descriptors were used as baselines. The *UC Merced Land Use* dataset [73] was used for image classification. Each of 21 classes comprises 100 images. The 2100 images contain

TABLE XII
The average classification rates (%) and standard deviations derived using four baselines and four SLoW based descriptors for UIUC [2] texture classification

DESC	BoW	SPM	VLAD	FC
***w* = 50**	97.56±0.69	97.47±0.59	98.24±1.18	97.32±0.96
***w* = 100**	98.79±0.38	98.74±0.48	97.41±1.01	
***w* = 200**	99.35±0.36	99.35±0.27	96.35±1.09	
DESC	SLoW	SLoW+SPM	SLoW+VLAD	SLoW+FC
***w* = 50**	97.53±0.48	97.56±0.52	**99.21±0.59**	98.35±0.59
***w* = 100**	98.79±0.29	98.76±0.36	**99.56±0.37**	98.82±0.31
***w* = 200**	99.29±0.44	99.29±0.35	**99.59±0.28**	99.32±0.31

TABLE XIII
The average classification rates (%) and standard deviations derived using four baselines and four SLoW based descriptors for land use image classification [73]

DESC	BoW	SPM	VLAD	FC
***w* = 50**	84.58±0.54	86.27±0.46	93.17±0.28	91.35±0.38
***w* = 100**	88.38±0.22	89.20±0.37	92.75±0.32	
***w* = 200**	90.93±0.46	90.61±0.36	93.46±0.33	
DESC	SLoW	SLoW+SPM	SLoW+VLAD	SLoW+FC
***w* = 50**	87.91±0.29	88.05±0.24	**95.14±0.21**	92.99±0.31
***w* = 100**	90.04±0.31	90.51±0.27	**94.73±0.14**	93.40±0.27
***w* = 200**	91.94±0.13	92.04±0.16	**94.94±0.26**	93.95±0.34

various objects and spatial patterns. Five-fold cross-validation was conducted using the SVM classifier [13] and the average classification rate (CR, %) was used as performance measure following the pipeline that Yang and Newsam [73] used. The experiment was conducted using each descriptor for ten runs in order to obtain different splitting for cross-validation.

The classification rates are reported in Table XI. It can be observed that: (1) the SLoW descriptor outperformed BoW [64] and SPM [38] when different numbers of words were used; (2) the performances of SPM [38] and VLAD [32] were improved when combined with SLoW; and (3) the best performance was produced by SLoW+VLAD.

C. Deep Convolutional Word Based SLoW

The method described in Section III-B-1 was used to learn deep convolutional words. In addition to BoW [64], SPM [38] and VLAD [32], the features extracted from the penultimate fully-connected layer of a CNN model, i.e. FC, was used as the fourth baseline. The SLoW descriptor was also combined with FC to test whether or not they are complementary. The SLoW descriptor and the three combined versions were compared with the baselines in image classification and retrieval.

1) Image Classification

Image classification was performed using the *UIUC* [2], *UC Merced Land Use* [73] and *Caltech256* [25] datasets. The pre-trained CNN models: VGG-M [9], Place-CNN [77] and VGG-VD16 [63] were used for the three datasets respectively.

UIUC Seventeen unique classes contained in the *UIUC* dataset [2] were used. This scheme ensures that the classifier is trained with less bias as each class contains the same number of images. In total, 680 *UIUC* images were used. These images were randomly divided into two equal-sized groups. One group was used as training images while the other group was used as test images. This operation was repeated for ten runs in order to produce different splitting. The average classification rate (%) computed across the ten runs was used as performance measure.

TABLE XIV
The average classification rates (%) and standard deviations derived using four baselines and four SLoW based descriptors for Caltech256 image classification [25]

DESC	BoW	SPM	VLAD	FC
***w* = 50**	50.91±0.41	56.76±0.26	77.45±0.25	79.22±0.16
***w* = 100**	59.99±0.46	63.30±0.04	78.51±0.04	
***w* = 200**	65.15±0.45	66.93±0.18	79.44±0.14	
DESC	SLoW	SLoW+SPM	SLoW+VLAD	SLoW+FC
***w* = 50**	55.58±0.33	57.53±0.18	75.24±0.30	**78.74±0.15**
***w* = 100**	62.09±0.30	63.74±0.19	76.34±0.26	**79.33±0.07**
***w* = 200**	65.97±0.21	67.29±0.13	76.67±0.24	**79.81±0.04**

The results derived for texture classification are reported in Table XII. As can be seen, the performances derived using BoW [64], SPM [38], SLoW and SLoW+SPM are close. This finding is similar to that obtained in Section V-A. Nevertheless, the SLoW+VLAD descriptor outperformed all its counterparts even if VLAD [32] did not perform well. Also, the SLoW and FC descriptors are complementary. The joint use of these descriptors performed better than that they did individually.

UC Merced Land Use The experimental setup of the land use image classification was introduced in Section V-B. Table XIII shows the results obtained for land use image classification [73]. Compared with the results shown in Table XI, the deep convolutional word based descriptors performed much better. As can be seen, the SLoW descriptor performed better than BoW [64] and SPM [38] when different numbers of words were used and performed better than FC using 200 words. However, the most obvious finding is that the performances of SPM, VLAD and FC were boosted when combined with SLoW. Besides, SLoW+VLAD outperformed all its counterparts.

Caltech256 The *Caltech256* dataset [25] contains 256 object classes and one background class. In total, 30,607 images are contained in this dataset. Following the setup that Simonyan and Zisserman [63] used, we randomly split each class into the training and test sets. In terms of each class, 60 images were included in the training set while the remaining images were put into the test set. Three different splitting were tested. The average classification rate (%) calculated across these splitting was used as the performance metric. Instead of extracting the FC features using multiple scales [63], we only used the original image resolution. In Table XIV, we report the results generated by the SLoW descriptors and four baselines. It can be observed that the SLoW descriptor outpferformed BoW in the same condition. With the joint use of SLoW, the performances of SPM and FC can be boosted. The best performance was derived using SLoW+FC when 200 deep words were used.

To examine the impact of the kernel functions of SVM, we performed two experiments. First, we conducted grid-search [1] on the parameters of the linear and RBF kernels using 3-fold cross-validation. We randomly selected 60 images from each of the first 100 classes of the *Caltech256* dataset [25]. The results showed that r did not affect the accuracy while C did matter when the RBF kernel was used. The parameters yielding the best result were selected. Specifically, $C = 4096$ and $r = 4096$ were selected for the RBF kernel while $C = 2$ was selected for the linear kernel. The classification experiment was then conducted using these parameters and the full dataset. The

TABLE XV
THE MEAN AVERAGE PRECISION (MAP) DERIVED USING FOUR BASELINES AND FOUR SLOW BASED DESCRIPTORS FOR OXFORD 5K IMAGE RETRIEVAL [53]

DESC	BoW	SPM	VLAD	FC	[45]	[70]	[72]
***w* = 50**	0.356	0.390	0.605				
***w* = 100**	0.417	0.441	0.620	0.419	0.649	0.466	**0.657**
***w* = 200**	0.454	0.475	0.655				

DESC	SLoW	SLoW+SPM	SLoW+VLAD	SLoW+FC
***w* = 50**	0.413	0.404	0.602	0.451
***w* = 100**	0.470	0.461	0.621	0.476
***w* = 200**	0.507	0.495	**0.657**	0.493

TABLE XVI
THE MEAN AVERAGE PRECISION (MAP) DERIVED USING FOUR BASELINES AND FOUR SLOW BASED DESCRIPTORS FOR PARIS 6K IMAGE RETRIEVAL [54]

DESC	BoW	SPM	VLAD	FC	[45]
***w* = 50**	0.446	0.459	0.638		
***w* = 100**	0.517	0.513	0.665	0.641	**0.694**
***w* = 200**	0.560	0.555	0.668		

DESC	SLoW	SLoW+SPM	SLoW+VLAD	SLoW+FC
***w* = 50**	0.493	0.482	0.641	0.657
***w* = 100**	0.554	0.538	0.669	0.668
***w* = 200**	0.603	0.584	0.681	0.680

average classification rates obtained using the two kernels were 79.29±0.18 and 79.29±0.17 respectively. These results suggest that the choice of the linear kernel is reasonable. Second, we performed image classification using the linear kernel and full dataset with the four C ($C \in \{0.1,1,10,100\}$) values that Cimpoi *et al*. [10] used. The classification rates derived using the four C values were 75.43±0.57, 79.21±0.13, 79.22±0.16 and 79.23±0.16 respectively. It is shown that there is not a significant difference in the performance when the C value exceeds 1. In this context, the default C value: 10 is proper.

2) Image Retrieval

We also tested the SLoW descriptor and its combined variants in the scenario of image retrieval. Specifically, the *Oxford 5K* [53] and *Paris 6K* [54] datasets were used. For each dataset, a total of 55 queries were performed. For simplicity, none of the bounding boxes provided with the datasets were used. The mean average precision (mAP) was used to measure the performance of image retrieval. Given a query image, the ranking between this image and other images was computed using the histogram intersection [65] and *Euclidian* distances for histogram and non-histogram based descriptors respectively. Instead of using the combination scheme introduced in Section III-E, we combined the distances computed using two different descriptors when SLoW+SPM, SLoW+VLAD and SLoW+FC were considered. To be exact, let the two distances be denoted as d_1 and d_2, the combined distance d was calculated as:

$$d = \sqrt[n]{{d_1}^n + {d_2}^n}. \tag{7}$$

In the two retrieval experiments, $n = 4$ was used.

Tables XV and XVI report the results derived using the *Oxford 5K* [53] and *Paris 6K* [54] datasets respectively. As can be seen, the performance of the SLoW descriptor was superior to those of BoW and SPM. The joint use of the SLoW descriptor improved the performance of the SPM, VLAD and FC descriptors in most cases. However, the FC descriptor did not generate stable results over the two datasets.

TABLE XVII
THE AGREEMENT RATES (%) DERIVED USING TWO BASELINES AND SLOW-BASED DESCRIPTORS FOR PAIR-OF-PAIRS COMPARISON [17]

DESC	BoW (LBP)	SPM	SLoW	SLoW+SPM
AR (%)	55.3	55.1	**57.9**	**57.9**

TABLE XVIII
THE AVERAGE CLASSIFICATION RATES (%) AND CORRESPONDING STANDARD DEVIATIONS OBTAINED USING TWO BASELINES AND SLOW-BASED DESCRIPTORS FOR PERTEX [12] TEXTURE CLASSIFICATION

DESC	BoW (LBP)	SPM	SLoW	SLoW+SPM
1	88.68±1.94	84.82±1.76	91.59±1.08	**93.65±1.47**
6	85.88±0.47	81.47±0.72	90.30±0.46	**93.09±0.36**

The number of test images was set as 1 and 6.

TABLE XIX
THE AVERAGE CLASSIFICATION RATES (%) AND STANDARD DEVIATIONS DERIVED USING TWO BASELINES AND SLOW-BASED DESCRIPTORS FOR LAND USE IMAGE CLASSIFICATION [73]

DESC	BoW (LBP)	SPM	SLoW	SLoW+SPM
CR (%)	65.12±0.37	70.25±0.66	70.80±0.52	**73.50±0.50**

Furthermore, we compared our results with state-of-the-art results (see Tables XV and XVI). When the *Oxford 5K* [53] dataset was used, our best result was better than, or equal to, those reported in [45], [70], [72]. While the best performance that our methods produced on the *Paris 6K* [54] dataset was slightly worse than that Ng *et al*. [45] reported, they used different strategies to learn words. First, they resized images to 224×224 pixels before feeding these to CNN while we used the original image size. Second, they used all convolutional layers to extract local features. In contrast, we only used the last convolutional layer. However, this study aims to examine the importance of contour cues to word-based descriptors rather than yielding state-of-the-art results on various datasets. Therefore, we ignored the strategies that Ng *et al.* [45] used.

D. LBP-Based SLoW

We further test SLoW descriptors using LBP codes [47]. Since the codebook that LBP uses is obtained in a different manner from the learnt word dictionary, the VLAD descriptor [32] is not applicable. The BoW (i.e. the LBP histogram [47]) and SPM [38] descriptors were used as baselines. The LBP code map was used to compute BoW, SPM, SLoW and SLoW+SPM descriptors. These descriptors were tested in pair-of-pairs comparison [17], texture classification and land use image classification [73]. The experimental setups were introduced in Sections IV-B, V-A and V-B respectively.

The results obtained in the three tasks are reported in Tables XVII, XVIII and XIX respectively. It can be seen that (1) the use of spatial pyramid matching did not boost the performance of LBP [47] in pair-of-pairs comparison [17] and texture classsfication while it did in land use image classification [73]; (2) the SLoW descriptor outperformed both BoW and SPM [38]; and (3) the performance of SPM was boosted when combined with SLoW.

E. Summary

In this section, we examined SLoW-based descriptors using different forms of words learnt from image patches [67], SIFT features [42] and deep convolutional features [10]. In addition, LBP codes [47] were considered as a special type of words and were also used with SLoW. It was shown that the SLoW descriptor outperformed both the BoW [64] and SPM [38] descriptors in all the experiments except texture classification conducted in Sections V-A and V-C. In addition, the SLoW descriptor complemented the SPM [38], VLAD [32] and deep fully-connected (FC) descriptors [10] and improved their performance. We attribute the promising results to the fact that the SLoW descriptor explores the spatial layout of words encoded in the important visual cues: contours. To our knowledge, this global characteristic has not been utilised by other word-based descriptors. (Arandjelovic and Zisserman [3], Bai *et al.* [5] and Wang *et al.* [69] only used local contour characteristics). It is noteworthy that SPM [38] performed worse than BoW [64] in both texture classification experiments either. Since texture is normally regarded as homogenous, the use of spatial pyramids may yield an "average effect" which impairs the discriminatory power of the descriptor. This is also the case when SLoW is applied to texture classification because IWSCM features are computed as the average shape data.

VI. Conclusions and Future Work

Motivated by the importance of contour cues to human visual perception of imagery, we proposed a global image descriptor by exploiting the spatial layout of words encoded in the form of contours. We refer to this descriptor as "Spatial Layout of Words" or "SLoW". We tested the SLoW descriptor together with different types of words, including contons, textons, SIFT words, deep convolutional words and LBP codes. Compared with the bag-of-words (BoW) descriptor, the SLoW descriptor encodes both short-range image characteristics using words and the medium-range and long-range image structure information by computing the spatial layout of words within the same contour and without regard to contours. More importantly, it exploits the important visual cues: contours. This type of global structure information has not been exploited in the existing studies that aim to boost the BoW descriptor by incorporating the spatial information. Besides, the aforementioned SLoW features were combined with Spatial Pyramid Matching (SPM) and Vector of Locally Aggregated Descriptors (VLAD) features in order to incorporate the spatial layout of words into these features. Correspondingly, the combined descriptors were termed as "SLoW+SPM" and "SLoW+VLAD".

The SLoW-based descriptors were tested in different applications along with baselines. Experimental results showed that the SLoW descriptor outperformed both BoW and SPM in most cases. This descriptor also improved the performance of SPM and VLAD when combined with each of these descriptors. Moreover, the performance of the deep fully-connected (FC) descriptor was boosted when used together with SLoW. We attribute these promising results to the fact that the SLoW descriptor performs in a more perceptually consistent manner than its counterparts examined in this study. It is noteworthy that the dimensionality of the SLoW feature vectors is high when multiple spatial pyramids are used. This decreases the computational speed and increases memory requirements. In contrast, the use of SLoW+VLAD descriptor is more practical because it only uses the original image resolution. Also, the best performance was normally derived using SLoW+VLAD where it is applicable.

Since the SLoW descriptor is computed based on contours, it may not perform well in the case that contours cannot be extracted. In this situation, the graph-based spatial relationship [43] may be considered instead of the contour-based one. In future work, therefore, we intend to model the spatial layout of words using the graph-based representation. However, the most important point is that the current work has shown the spatial layout of words encoded in the form of contours can boost the performance of traditional visual word based image descriptors. This may lead to a potential direction for improving other word-based descriptors. In addition, the CNN trained in an end-to-end manner has shown superiority to the pre-trained CNN in the literature. Hence, we will explore the possibility of end-to-end training a CNN with the SLoW descriptor in future.

Acknowledgment

The authors would like to thank the anonymous reviewers for their comments and suggestions to improve this manuscript. Xinghui Dong would also like to thank Prof. Mike Chantler for kindly allowing him to continue his research in the Texture Laboratory.

References

[1] LIBSVM: A library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[2] UIUC. http://www-cvr.ai.uiuc.edu/ponce_grp/data/, 2005.

[3] R. Arandjelovic and A. Zisserman, "Smooth object retrieval using a bag of boundaries," in Proc. IEEE International Conference on Computer Vision, 2011.

[4] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour Detection and Hierarchical Image Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898-916, 2011.

[5] X. Bai, C. Rao, and X. Wang, "Shape Vocabulary: A Robust and Efficient Shape Representation for Shape Matching," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 3935-3949, 2014.

[6] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 24, pp. 509-521, 2002.

[7] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proc. ACM international conference on Image and video retrieval*, pp. 401-408, 2007.

[8] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679-698, 1986.

[9] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the Devil in the Details: Delving Deep into Convolutional Networks," in *Proc. British Machine Vision Conference*, 2014.

[10] M. Cimpoi, S. Maji, I. Kokkinos, and A. Vedaldi, "Deep Filter Banks for Texture Recognition, Description, and Segmentation," *Int'l J. Computer Vision*, vol. 118, no. 1, pp. 65-94, 2016.

[11] A.D.F. Clarke, X. Dong, and M. J. Chantler, "Does Free-sorting Provide a Good Estimate of Visual Similarity," in *Proc. the 3rd International Conference on Appearance*, pp. 17-20, 2012.

[12] A.D.F. Clarke, F. Halley, A.J. Newell, L.D. Griffin, and M.J. Chantler, "Perceptual Similarity: A Texture Challenge," in *Proc. British Machine Vision Conference*, 2011.

[13] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995.

[14] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[15] J. De Winter and J. Wagemans, "The awakening of Attneave's sleeping cat: Identification of everyday objects on the basis of straight-line versions of outlines," *Perception*, vol. 37, no. 2, pp. 245-270, 2008.

[16] X. Dong, “Perceptual Texture Similarity Estimation,” PhD thesis, Heriot-Watt University, 2014.

[17] X. Dong and M. J. Chantler. “The Importance of Long-Range Interactions to Texture Similarity,” in *Proc. the 15th International Conference on Computer Analysis of Images and Patterns*, 2013, vol. 8047, pp. 425-432.

[18] X. Dong and M. J. Chantler, “Perceptually Motivated Image Features Using Contours,” *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5050-5062, 2016.

[19] X. Dong, T. Methven, and M. J. Chantler, “How Well Do Computational Features Perceptually Rank Textures? A Comparative Evaluation,” in *Proc. the ACM 2014 International Conference on Multimedia Retrieval*, 2014, pp. 281-288.

[20] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa, “Sketch-based image retrieval: benchmark and bag-of-features descriptors,” *IEEE Trans. Visual. Comput. Graphics*, vol. 17, no. 11, pp. 1624-1636, 2011.

[21] R. Fagin, R. Kumar and D. Sivakumar, “Comparing Top K Lists,” in *Proc. ACM-SIAM symposium on Discrete algorithms*, pp. 28-36, 2003.

[22] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid, “Groups of Adjacent Contour Segments for Object Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 1, pp. 36-51, 2008.

[23] D. J. Field, A. Hayes and R. F. Hess, “Contour integration by the human visual system: evidence for a local ‘association field’,” *Vision Research*, vol. 33, pp. 173-193, 1993.

[24] R. C. Gonzalez and R. E. Woods, *Digital Image processing*, NJ: Prentice Hall Upper Saddle River, 2002.

[25] G. Griffin, A. Holub, and P. Perona, “Caltech-256 Object Category Dataset,” *Technical Report of California Institute of Technology*, 2007.

[26] S. Grossberg and E. Mingolla, “Neural dynamics of form perception: Boundary completion, illusory figures, and neon color spreading,” *Psychological Review*, vol. 92, pp. 173-211, 1985.

[27] S. Grossberg and E. Mingolla, “Neural dynamics of perceptual grouping: Textures, boundaries, and emergent segmentations,” *Percept. Psychophys.*, vol. 38, pp. 141-171, 1985.

[28] T. Harada, H. Nakayama and Y. Kuniyoshi, “Improving Local Descriptors by Embedding Global and Local Spatial Information,” in *Proc. European Conference on Computer Vision*, 2010.

[29] R.M. Haralick, K. Shanmugam, and I. Dinstein, “Textural Features for Image Classification,” *IEEE Trans. Systems, Man, Cybernetics*, vol. 3, pp. 610-621, 1973.

[30] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[31] J. Iivarinen and A. Visa, “Shape recognition of irregular objects,” in *Proc. SPIE 2904*, 1996.

[32] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid, “Aggregating Local Image Descriptors into Compact Codes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704-1716, 2012.

[33] X. Jia, H. Yang, X. Zhu, Z. Kuang, Y. Niu, and K. Chan, “Reflective Regression of 2D-3D Face Shape Across Large Pose,” in *Proc. the British Machine Vision Conference*, pp. 135.1-135.14, 2016.

[34] R. Khan, C. Barat, D. Muselet and C. Ducottet, “Spatial orientations of visual word pairs to improve Bag-of-Visual-Words model,” in *Proc. British Machine Vision Conference*, 2012.

[35] P. Kovesi, “Phase congruency detects corners and edges,” in *Proc. Australian Pattern Recognition Society Conference*, 2003.

[36] J. Krapac, J. Verbeek, and F. Jurie, “Modeling spatial layout with fisher vectors for image categorization,” in *Proc. IEEE International Conference on Computer Vision*, 2011.

[37] M. Lai, I. Oruç and J. J.S. Barton, “The role of skin texture and facial shape in representations of age and identity,” *Cortex*, vol. 49, no. 1, pp. 252-265, 2013.

[38] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[39] T. Leung and J. Malik, “Recognizing surfaces using three-dimensional textons,” in *Proc. IEEE International Conference on Computer Vision*, vol. 2, pp. 1010-1017, 1999.

[40] J.J. Lim, C.L. Zitnick, and P. Dollar, “Sketch Tokens: A Learned Mid-level Representation for Contour and Object Detection,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[41] D. Liu, H. Gang, P. Viola, and T. Chen, “Integrated feature selection and higher-order spatial feature extraction for object categorization,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[42] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[43] T. Maugey, A. Ortega, and P. Frossard, “Graph-Based Representation for Multiview Image Geometry,” *IEEE Transactions on Image Processing*, vol. 24, no. 5, pp. 1573-1586, 2015.

[44] N. Morioka and S. Satoh, “Building Compact Local Pairwise Codebook with Joint Feature Space Clustering,” in *Proc. European Conference on Computer Vision*, 2010.

[45] J. Y. Ng, F. Yang, and L. S. Davis, “Exploiting Local Features from Deep Networks for Image Retrieval,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 53-61, 2015

[46] X. Ni, C. Cao, L. Ding, T. Jiang, H. Zhang, H. Jia, G. Li, J. Zhao, W. Chen, W. Ji, M. Xu, M. Gao, S. Zheng, R. Tian, C. Liu, and S. Li, “A fully automatic registration approach based on contour and SIFT for HJ-1 images,” *Sci. China Earth Sci.*, vol. 55, pp. 1679-1687, 2012.

[47] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution Grey-Scale and Rotation Invariant Texture Classification with Local Binary Patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 971-987, 2002.

[48] B.A. Olshausen and D.J. Field, “Natural image statistics and efficient coding,” *Network: computation in neural systems*, vol. 7, no. 2, pp. 333-339, 1996.

[49] A.V. Oppenheim and J.S. Lim, “The Importance of Phase in Signals,” *Proceedings of the IEEE*, vol. 69, no. 5, pp. 529-541, 1991.

[50] M. Oszust, “BDSB: Binary descriptor with shared pixel blocks,” *Journal of Visual Communication and Image Representation*, vol. 41, pp. 154-165, 2016.

[51] S. Panis, J. De Winter, J. Vandekerckhove, and J. Wagemans, “Identification of everyday objects on the basis of fragmented outline versions,” *Perception*, vol. 37, pp. 271-289, 2008.

[52] F. Perronnin, J. Sanchez and T. Mensink, “Improving the fisher kernel for large-scale image classification,” in *Proc. European conference on Computer vision*, 2010.

[53] J. Philbin, O. Chum, M. Isard, J. Sivic and A. Zisserman, “Object Retrieval with Large Vocabularies and Fast Spatial Matching,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[54] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[55] F. Phillips and J.T. Todd, “Texture Discrimination Based on Global Feature Alignments,” *J. Vision*, vol. 10, no. 6, pp. 1-14, 2010.

[56] D. Picard and P. Gosselin, “Efficient image signatures and similarities using tensor products of local descriptors,” *Computer Vision and Image Understanding*, vol. 117, no. 6, pp. 680-687, 2013.

[57] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C.* (2nd ed.), Cambridge University Press, 1992.

[58] T. Quack, V. Ferrari, B. Leibe, and L. Van Gool, “Efficient Mining of Frequent and Distinctive Feature Configurations,” in *Proc. IEEE International Conference on Computer Vision*, 2007.

[59] J. Sánchez, F. Perronnin, and T. de Campos, “Modeling the spatial layout of images beyond spatial pyramids,” *Pattern Recognition Letters*, vol. 33, no. 16, pp. 2216-2223, 2012.

[60] S. Savarese, J. Winn and A. Criminisi, “Discriminative Object Class Models of Appearance and Shape by Correlatons,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[61] C. Shan, S. Gong and P.W. McOwan, “Facial expression recognition based on Local Binary Patterns: A comprehensive study,” *Image and Vision Computing*, vol. 27, no. 6, pp. 803-816, 2009.

[62] L. Sharan, C. Liu, R. Rosenholtz and E.H. Adelson, “Recognizing Materials Using Perceptually Inspired Features,” *Int. J. Comput. Vis.*, vol. 103, pp. 348-371, 2013.

[63] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Visual Recognition,” in *Proc. ICLR*, 2015.

[64] J. Sivic and A. Zisserman, “Video Google: a text retrieval approach to object matching in videos,” in *Proc. IEEE International Conference on Computer Vision*, 2003.

[65] M.J. Swain and D.H. Ballard, “Colour Indexing,” *Int. J. Comput. Vis.*, vol. 7, no. 1, pp. 11-32, 1991.

[66] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[67] M. Varma and A. Zisserman, “A Statistical Approach to Material Classification Using Image Patch Exemplars,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 2032-2047, 2009.

[68] J. Wagemans, J. De Winter, H. Op de Beeck, A. Ploeger, T. Beckers, and P. Vanroose, “Identification of everyday objects on the basis of silhouette and outline versions,” *Perception*, vol. 37, pp. 207-244, 2008.

[69] X. Wang, B. Feng, X. Bai, W. Liu and L. Jan Lateck, “Bag of contour fragments for robust shape classification,” *Pattern Recognition*, vol. 47, no. 6, pp. 2116-2125, 2014.

[70] Y. Wang, M. Shi, S. You, and C. Xu, “DCT Inspired Feature Transform for Image Retrieval and Reconstruction,” *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp.4406-4420, 2016.

[71] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, “Locality-constrained Linear Coding for Image Classification,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3360-3367, 2010.

[72] A. B. Yandex and V. Lempitsky, “Aggregating Local Deep Features for Image Retrieval,” in *Proc. IEEE International Conference on Computer Vision*, pp. 1269-1277, 2015.

[73] Y. Yang and S. Newsam, “Spatial pyramid co-occurrence for image classification,” in *Proc. IEEE International Conference on Computer Vision*, 2011.

[74] J. Yang, K. Yu, Y. Gong, and T. Huang, “Linear spatial pyramid matching using sparse coding for image classification,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[75] L. Ying, “Phase unwrapping,” *Wiley Encyclopedia of Biomedical Engineering*, vol. 6, pp. 1-11, 2006.

[76] D. Zhang and G. Lu, “Review of shape representation and description techniques,” *Pattern Recognition*, vol. 37, pp. 1-19, 2004.

[77] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, “Learning Deep Features for Scene Recognition using Places Database,” in *Proc. Advances in Neural Information Processing Systems*, 2014.

Xinghui Dong received his PhD degree from Heriot-Watt University, UK, in 2014. He is currently working as a Research Associate in the Centre for Imaging Sciences, The University of Manchester, UK. His research interests include automatic defect detection, image representation, texture analysis and visual perception.

Junyu Dong received his BSc and MSc from the Department of Applied Mathematics at Ocean University of China in 1993 and 1999 respectively. From 2000 and 2003, he studied in the UK and received his PhD in Image Processing in November 2003, from the Department of Computer Science at Heriot-Watt University. Dr. Dong joined Ocean University of China in 2004 and he is currently a professor and the Head of the Department of Computer Science and Technology. His research interests include machine learning, big data, computer vision and underwater image processing.