PPD: A Robust Low-computation Local Descriptor for Mobile Image Retrieval

Congxin Liu, Jie Yang and Deying Feng
Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University
Shanghai 200240, China
[e-mail: phillx, jieyang@sjtu.edu.cn, fdy629@163.com]
*Corresponding author: Congxin Liu

Received February 2, 2010; revised April 1, 2010 and May 6, 2010; accepted June 4, 2010; published June 30, 2010

Abstract

This paper proposes an efficient and yet powerful local descriptor called phase-space partition based descriptor (PPD). This descriptor is designed for the mobile image matching and retrieval. PPD, which is inspired from SIFT, also encodes the salient aspects of the image gradient in the neighborhood around an interest point. However, without employing SIFT's smoothed gradient orientation histogram, we apply the region based gradient statistics in phase space to the construction of a feature representation, which allows to reduce much computation requirements. The feature matching experiments demonstrate that PPD achieves favorable performance close to that of SIFT and faster building and matching. We also present results showing that the use of PPD descriptors in a mobile image retrieval application results in a comparable performance to SIFT.

Keywords: Local descriptor, image matching, mobile image retrieval, SIFT

This research is supported by national science foundation, China, No: 60805001 and aviation science fund, China, No: 2008CZ57. We would like to thank the anonymous reviewers for the valuable comments.

DOI: 10.3837/tiis.2010.06.007

1. Introduction

Recently, local invariant features have shown promise for image retrieval tasks [1][2][3][4] because they can be computed efficiently, robust to partial occlusion, and relatively insensitive to changes in viewpoint. The rough steps of image retrieval using local invariant features contains: first, extracting interest points (keypoints) and corresponding local neighbor regions around them for all of the images in the image lib. Typically, interest points are located at local peaks in scale-space [5][6][7], and filtered to preserve only those points that are likely to remain stable over various transformations. The local neighbor regions are generally elliptic regions whose size depend on the characteristic scale of the interest points; Second, certain features are extracted from these local neighbor regions and then form vectors called descriptors. Ideally, descriptor should be distinctive (reliably discriminating one interest point from others), robust to noise, partial occlusion, detection displacements and invariant to geometric and photometric deformations. Third, all the descriptors are clustered to acquire a great number of cluster centers (also vectors); Fourth, all of the images in the lib are encoded using these cluster centers to obtain respective content metric. When a query image is obtained, the above processing steps other than clustering are re-executed to get its content metric. Comparing the content metric against all of the metrics in the image lib, we could find the most similar image to the query image in the lib.

How to effectively build mobile image retrieval is the major concern of this paper. In practice, the rough process for mobile image retrieval is shown as follows. The interest point detection and local image region description are first accomplished on mobile terminals and then the resulting feature vectors are sent to remote data processing centers via communication network where the image retrieval task is carried out. After comparison and ranking, a few candidate images are sent back to the original mobile phones for further operations. Considering the low computation capability of mobile terminals, the computation efficiency on the mobile phone becomes a bottleneck for enhancing the overall performance of the system. Thus, developing both a detector and descriptor which are fast to compute without much performance loss is of great significance for an efficient mobile image retrieval system. In this paper, we focus on the descriptor and try to find an efficient feature representation. Our primary motivation is to lessen the computation cost and to achieve a low memory usage for mobile phones.

In this paper, we propose a phase-space based local image descriptor (PPD). The original idea of using phase space, in which all possible states of a system are represented, comes from physics. In a phase space, every degree of freedom or parameter of the system is represented as an axis of a multidimensional space. If an image surface is regarded as a system and its pixel location *(x,y)* is considered as vector variable, then the triples (pixel intensity, horizontal gradient, and vertical gradient) can represent the state of an image at each sample position. Due to brightness changes, intensity value cannot serve as a state component in the context of image matching. Therefore we construct a reduced phase space for describing local image patterns. Each region in phase space corresponds to an entry of the histogram in our algorithm. In comparison with the state-of-the-art descriptor SIFT [6], PPD is more efficient since it neither computes gradient orientation nor applies any interpolation to the feature representation while yet preserving considerable distinctiveness. The following experiments demonstrate the promise of PPD in online applications.

The remainder of this paper is organized as follows: In Section 2, we review the previous work in interest point description. In Section 3, the details of PPD are presented. In Section 4, we provide detailed experiment results comparing PPD with SIFT and SURF on feature matching experiments and also in the context of a mobile image retrieval application, Section 5 concludes the paper.

2. Related Work

Many different descriptors have been proposed to describe the appearance of a local image region. Popular descriptors include differential invariants [8], steerable filters [9], complex filters [10], moment invariants [11], spin image [12], SIFT (Scale-Invariant Feature Transform) [6], Shape Context [13]. The detailed performance evaluations of these descriptors were presented in [14][15], where it was shown that the high-dimensional representations based on histograms of localized gradient orientations such as SIFT outperform other descriptors by a certain margin in matching images of both planar surfaces and 3D objects [16]. Various refinements have been proposed in the literature to improve on the gradient orientation based descriptors. For example, Ke and Sukthankar developed the PCA-SIFT [17] which represents the surface of an image patch by the principal components of the normalized gradient patch. The computation burden of this PCA-SIFT is comparable to SIFT since the process of forming the normalized gradient patch also involves many interpolation operations, which is somewhat similar to SIFT's 3D interpolation in terms of computation consumption. In addition, applying PCA also slows down the feature computation. GLOH (Gradient Location Orientation Histogram) [14] modifies the SIFT representation by using alternative spatial sampling strategy and PCA for dimensionality reduction. 128-dimensional GLOH was proved to be more distinctive than SIFT but it requires more computation cost.

To improve the computation efficiency in dense computation, Tola et al. introduced an alternative spatial weighting scheme (Gaussian kernel function) and replaced the smoothed weighted histogram used by SIFT with sum of convolutions [18]. Jie Chen et al. presented a robust and efficient dense descriptor based on differential excitation and gradient orientation [19], which obtains a favorable performance in texture classification and face detection. In order to retrieve leaf image effectively, Yoon-Sik Tak and Eenjun Hwang introduced a new shape representation, indexing, and matching scheme [20].

To compute more quickly for sparse feature points, Bay et al. provided an efficient implementation of SIFT by applying the integral image for the computation of image derivatives [21]. However, its implementation process is computationally a bit complicated and the distinctive character of the descriptor can be further enhanced. Another two low-cost descriptors have also been reported in the literature [22][23]. CS-LBP (simplified center-symmetric local binary patterns) [22] and Contrast Context Histogram [23] both exploited the contrast of pixel value to reduce computation cost. The former needs to set a contrast threshold in the case of flat regions, which increases its sensitivity to the parameter. The latter lacks the description of the correlation between adjacent pixels in its feature representation. Due to the out-performance of SIFT representation in common deformation, this paper also focuses on this algorithm and explores the simplified alternatives to it.

3. The PPD Descriptor

In this section, we briefly describe the SIFT, then introduce PPD and discuss the difference between them. Finally we demonstrate the greater computation efficiency on PPD representation as compared with SIFT.

3.1 Review of the SIFT Algorithm

The SIFT descriptor is a 3D smoothed histogram in which two dimensions are image gradient locations, and the additional dimension is the image gradient orientation. The location, over a local image patch, is quantized into 4×4 location grids. The gradient orientation, in each single grid, is quantized into 8 orientations. For each grid, an orientation based histogram is created. All these orientation histograms over all 4×4 grids are concatenated to form the SIFT descriptor, which is a 128-dimensional description vector. The histogram based on gradient orientation contributes much to SIFT in distinctiveness and robustness due to its insensitivity to initial misregistration error.

The smoothed properties of SIFT are mainly attributed to the following two reasons. First, a Gaussian window is overlaid over the local image region to assign a weight to the gradient magnitude of each sample point. The purpose of this weighting function is to avoid sudden changes in the descriptor due to small misalignment error of keypoint (interest point) and to reduce the influences of gradient magnitudes which are far from the keypoint. Second, a trilinear interpolation is used to distribute the gradient magnitude of each sample point into adjacent 4×2 (4×2 denotes 4 nearest location grids to the sample point and 2 adjacent orientations of the gradient sample respectively) histogram bins, the main purpose of the interpolation is to weaken boundary affect in which the descriptor abruptly changes as a sample shifts smoothly from one grid to another or from one orientation to another [6]. As a result, each bin in the histogram is composed of weighting summation of magnitudes of image gradients around its grid center.

Furthermore, to cancel the effect of affine illumination change and non-linear illumination change, the descriptor is normalized to a unit vector in which each value is limited to no more than 0.2, and then re-normalized to a unit vector. All the above characteristics result in the remarkable out-performance of SIFT.

3.2 The PPD Descriptor

In this section, we discuss PPD representation in detail. It is assumed that the keypoints and corresponding local neighbor regions have been extracted from an image and the local neighbor regions have been transformed into normalized forms. The algorithm for PPD is carried out on the normalized image patches (41×41 pixels [24]), and then the keypoint locations are located at the centers of the patches. The dominant orientation of the image patch is defined as the direction of the smoothed gradient over this patch, in which a Gaussian weighting function with the standard variance equal to half of the width of the image patch is used to assign a weight to the *(dx,dy)* of each sample point. In general, a large integration scale can make orientation estimation robust to the keypoint location errors.

3.2.1 Primary PPD Representation

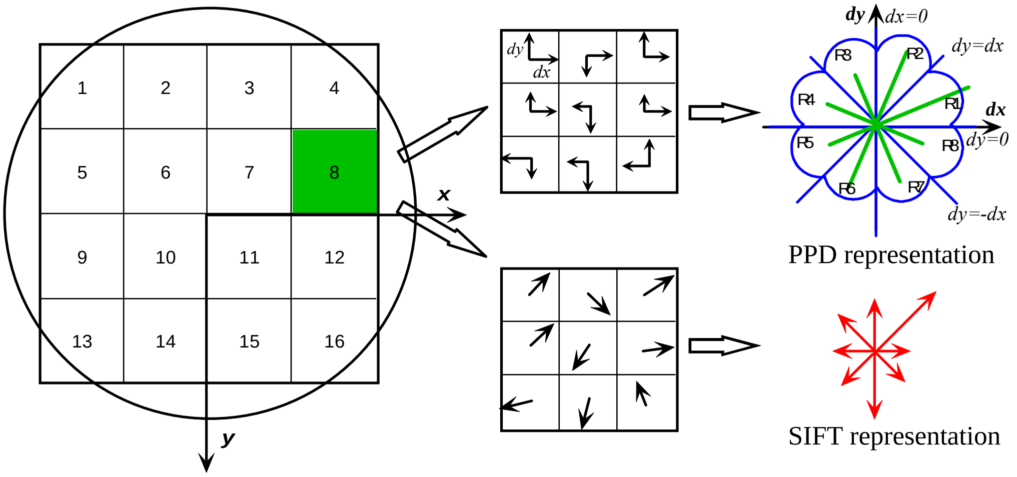
The idea of PPD is based on a phase-space (the coordinates are composed of *dx, dy*) partition over the image patch, see **Fig. 1**, in which a naive phase-space partition scheme is illustrated. From **Fig. 1**, it can be observed that PPD differs from SIFT in that it constructs description vector based on 8 continuous regions rather than 8 discrete orientations used by SIFT.

PPD can be summarized in the following steps:

(1) Given an image patch, compute both the horizontal and vertical gradient maps . The horizontal gradient, vertical gradient and corresponding image gradient magnitude are first computed at each sample point over the image patch. To guarantee invariance to orientation change, the coordinates of the descriptor are rotated relatively to the assigned dominant orientation. Both the new image coordinates and the gradient maps at each sample point can be obtained from the old ones by a linear transformation **A**.

$$\mathbf{x}_{new} = \mathbf{A}\mathbf{x}_{old} \qquad \mathbf{A} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$$

Where θ is the dominant orientation; $\mathbf{x}_{new}$ represents the new image coordinates or gradient maps and $\mathbf{x}_{old}$ refers to the old corresponding ones. The gradient maps are illustrated with small arrows at each sample location in the middle of Fig. 1.



PPD representation

SIFT representation

4×4 sub-regions over an image patch   3×3 set of image gradients

On the left, the image patch is first divided into 4×4 sub-regions. For the PPD representation, we compute *(dx, dy)* at each pixel on the image patch. Then, according to the relation between *dx* and *dy*, the corresponding phase-space region can be determined for each gradient sample. On the upper right, an eight-partition phase-space for the 8th sub-region is illustrated and the length of green line segments represents the sum of gradient norm within the regions. With respect to SIFT, however, we must compute both the gradient norm and orientation at each pixel, as shown in the middle of the figure.

Fig. 1. The difference between PPD and SIFT

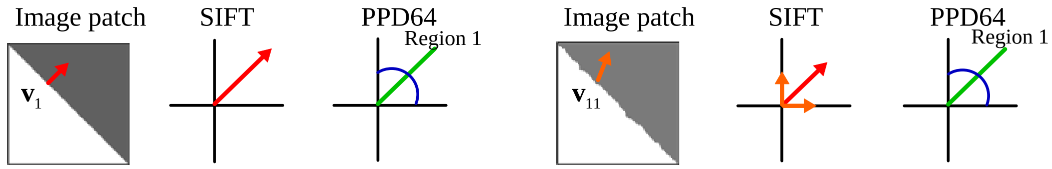
(2) Building the descriptor based on a specific phase-space partition scheme.

In the new coordinate system, the input image patch is first divided into 4×4 sub-regions, and one specific phase-space scheme is applied to them. For example, an eight region-partition scheme, in which the whole phase-space is evenly divided into eight regions, is shown in Fig. 1. According to this scheme, algorithm 1 outputs the PPD representation.

Based on the above algorithm, we can calculate a histogram *{bin1, bin2, ……, bin8}* for each sub-region, concatenating the 4×4 histograms in the order illustrated on the left side of Fig. 1 to form an enhanced 128-dimensional descriptor, which is a 3D histogram. The histogram for the eighth sub-region can be obtained from the upper-right of Fig. 1, where the length of each green line segment serves as the corresponding entry of the histogram. It allows for both significant shift in gradient positions and deviation in gradient orientations by creating histograms over 4×4 sample regions and 8 phase-space regions. A gradient sample, shifting randomly over 3×3 set of sample locations, rotating within a certain range, makes the same contribution to the histogram, as can be seen in Fig. 1 and Fig. 2. PPD outperforms SIFT about 10% performance as compared with SIFT without an interpolation in the dimension of orientation. Hence, PPD is more robust to distortions.

Algorithm 1 – PPD construction over one image patch

```
Input: one image patch and being divided into 4×4 sub-regions
for 4×4 sub-regions
    {bin1, bin2, ……, bin8}:={0, 0, 0, ……, 0}
    for each gradient sample point (dx,dy) on one sub-region
        norm=sqrt(dx*dx+dy*dy)*w(x,y)          % compute the weighted gradient norm
                                               % w(x,y) is a Gaussian function with the
                                               % variance equal to half of the width of
                                               % the image patch.
        if dx>0{
                dy>dx          bin1← bin1+norm
                0<dy<dx        bin2← bin2+norm
                -dx<dy<0       bin3← bin3+norm
                else           bin4← bin4+norm
        }
        else{
                dy>-dx         bin5← bin5+norm
                0<dy<-dx       bin6← bin6+norm
                dx<dy<0        bin7← bin7+norm
                else           bin8← bin8+norm
        }
        store {bin1, bin2, ……, bin8}
    end
end
```



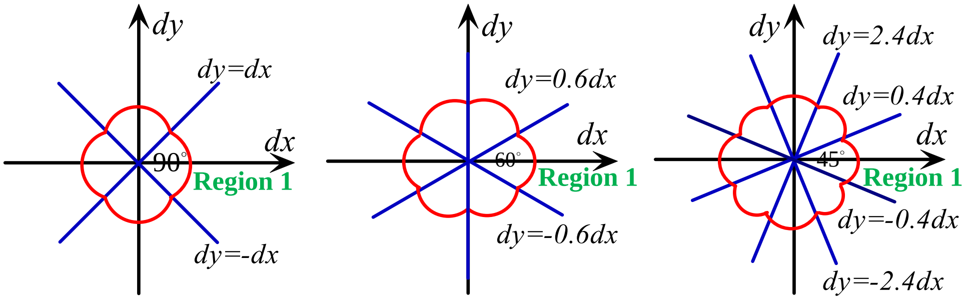
(a) Clean (b) Noisy

Due to projective distortions or noise, a gradient sample $\boldsymbol{V_1}$ is transformed into $\boldsymbol{V_{11}}$. If not beyond the range of the region 1, $\boldsymbol{V_{11}}$ makes the same contribution to PPD as $\boldsymbol{V_1}$. Therefore PPD is more robust to the variations of gradient directions than the locally operating SIFT.

Fig. 2. The robustness of PPD for the variations of gradient directions. The image patchs are from [21]

3.2.2 Refined PPD Representations

Some important issues about PPD in practical applications should be further considered. First, how to reduce the boundary affect? The region layout scheme illustrated in **Fig. 1** usually cause significant boundary affect because some gradients close to the horizontal direction often commute between the region 1 and the region 8 due to the initial misregistration errors. This decreases the robustness of the descriptor when the horizontal direction has been aligned with the dominant orientation. To address the problem, a refined region partition scheme is proposed in **Fig. 3c**, in which we align the central axis of region 1 with the horizontal direction and assume the dominant orientation is in accordance with it. As a result, the gradients around the dominant orientation are almost partitioned into an identical region (region 1), which tends to reduce boundary affect and increase the robustness of the descriptor. Our experiments have shown that a significant performance increase (around 10-15%) occurs when the scheme is put into practice.



(a) 4-region partition (b) 6-region partition (c) 8-region partition

Fig. 3. Refined PPD representations

Second, how to reduce memory consumption and further increase matching speed? For embedded devices, lower memory usage and higer computation efficiency are critical to practical applications, therefore finding low-dimensional description schemes is also necessary. Two alternative low-dimensional counterparts to the 8-region scheme are proposed in **Figs. 3a** and **3b**, namely 4-region scheme and 6-region one.

Third, some tricks may be taken for the convenience of computation. For the 4-region partition scheme, an offset-angle of *π/4* radians is first subtracted from the dominant orientation and then the descriptor coordinates are rotated to align with this new dominant orientation. Next, we can determine the histogram entry of a gradient sample by judging the configurations of the signs of the gradient maps (see **Fig. 1**) at the given sample location. Algorithm 2 gives the simplified algorithm of building a PPD representation according to the improved 4-region partition scheme in **Fig. 3a**.

Algorithm 2 – the simplified version of PPD64

```
Input: one image patch
$\theta=\theta-\pi/4$, $\mathbf{X}_{new}=A\mathbf{X}_{old}$
The image patch is then divided into 4×4 sub-regions
for 4×4 sub-regions
        {bin1, bin2, bin3, bin4}:={0, 0, 0, 0}
        for each gradient sample point (dx,dy) on one sub-region
        norm=sqrt(dx*dx+dy*dy)*w(x,y)
            if dx>0{                                   % only judging the sign
                    dy>0                  bin1← bin1+norm
                    else                  bin2← bin2+norm
            }
            else{
                    dy>0                  bin3← bin3+norm
                    else                  bin4← bin4+norm
            }
        store {bin1, bin2, bin3, bin4}
        end
end
```

From the algorithm 2, we can see that the simplified PPD is computationally very simple. The similar operations can be extended to the other two schemes.

3.2.3 Normalization

To cancel the effect of affine illumination change, the PPD descriptor turns into a unit vector. The unitization process can cancel the effect of scale factor on the descriptor. In addition, an identical offset added to each pixel value does not cause any change of gradient magnitude which is based on pixel value difference. Therefore, the unit form of PPD representation remains invariant under affine change in illumination.

To reduce effect of the non-linear illumination change, we limit each value in the unit feature vector to be no larger than 0.35, and re-normalize it to unit length. The value of 0.35 was determined experimentally using images containing different amount of illuminations. Note that it is different from the threshold 0.2 suggested in [6]. This is because the first histogram entry for each sub-region accumulates more gradient norms than the other entries in the histogram. Therefore, it is reasonable to assign a larger value.

3.2.4 Computation Complexity

In this section, we discuss the computation burden of PPD and SIFT. Both of the algorithms are carried out on the normalized image patches. SIFT needs to compute gradient (magnitude and orientation) at each sample location, in which the computation of gradient orientation involves relatively time-consuming inverse tangent computation. The algorithm for PPD circumvents and simplifies the problem by comparing the size of horizontal gradient and vertical gradient at each sample position or judging configurations of their signs to obtain the corresponding bins in the histogram. In our experiments, the histogram entry for a gradient sample can be determined by twice or thrice comparisons, which is rather convenient to compute.

Moreover, to avoid all boundary affects, SIFT carries out trilinear interpolation to smoothen the histogram, but PPD representation doesn't take any interpolation, because (1) region based statistics is robust to the variations of gradient orientation, as can be seen in **Fig. 2**. (2) a useful measurement is proposed to weaken boundary affect in the direction dimension as described in Section 3.2.2. (3) in most cases, the performance of PPD is close to that of SIFT, as shown in the experiments below.

Concerning the overall performance in terms of both discriminance and computation efficiency, we decide to give up interpolation. As a result, PPD is approximately 4 times faster than SIFT at the cost of sacrificing a little discriminative power in the commonly occurring photometric scenes. The detailed results are shown in the following experiments.

4. Experiment Evaluation

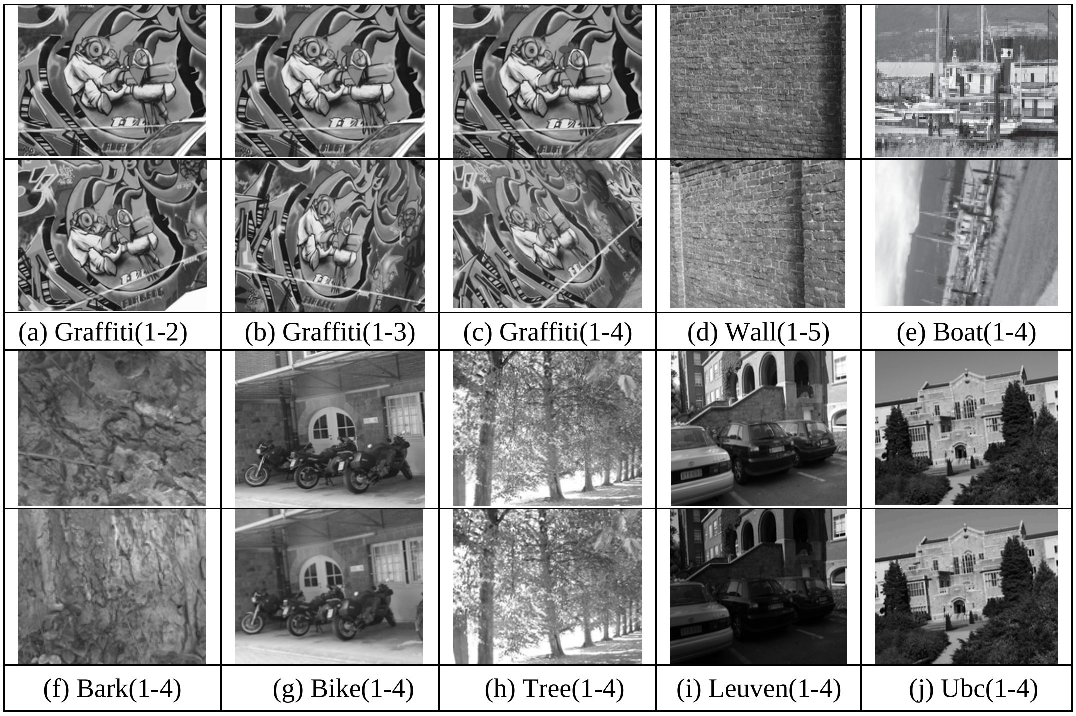
4.1 Data Set

In this section, we evaluate PPD descriptor and compare it with the state-of-the-art descriptor SIFT on a standard dataset and a real-application dataset. The former one comes from [24], a popular dataset for the evaluation of local feature properties. The dataset consists of eight image sequences including textured-images and structured-images. There are different deformations among these images, like viewpoint change, scale and rotation change, light change, blur, and JPEG compression. The test images of the standard dataset used in the experiments are shown in **Fig. 4**. These image pairs are either planar scene or captured from fixed-position camera during acquisition. Thus, the relation between them can be modeled by a 2D homography matrix; by the way, the image set has also provided homographies for some pair of images. The second dataset is captured by a scanner and several mobile phones, which is used in the last experiment of mobile image retrieval, as discussed in detail in Section 4.4.4.

4.2 Interest Region Detection and Normalization

Since solutions to localization and description are independent [14], we could choose different local image region detection algorithms, such as MSER [25], Harris-Affine, Hessian-Affine [5], EBR, IBR [26], Affine saliency [27], and DOG [6]. A detailed performance comparison between them has been presented in [28].

In the case of feature matching experiment, we choose Hessian-Affine and Harris-Affine for that they have been widely used in descriptor evaluation [14]. Hessian-Affine could detect blob-like points, which are less likely at the positions of depth-difference pixel points and favor local planarity and smoothness assumption.



(a) Graffiti(1-2) (b) Graffiti(1-3) (c) Graffiti(1-4) (d) Wall(1-5) (e) Boat(1-4)

(f) Bark(1-4) (g) Bike(1-4) (h) Tree(1-4) (i) Leuven(1-4) (j) Ubc(1-4)

Fig. 4. Data set 1. Viewpoint change for structured scene (a) 20°, (b) 30°, and (c) 40°, for textured scene (d) 50°; scale+rotation for structured scene (e), and for textured scene (f); blur for structured scene (g), and for textured scene (h); (i) light change ; (j) JPEG compression.

Harris-Affine extractes corner points which possess excellent robustness under geometric and photometric transformations but often lie close to a depth discontinuity, thus Hessian-Affine regions have higher detection accuracy than Harris-Affine ones. Both detector output elliptic regions whose sizes depend on the characteristic scale. Before computing descriptors, these elliptic regions should be normalized to 41×41 image patches [24]. Note that the practically effective computation scope in our experiments is a circular region with the radius of 29 for guaranteeing orientation invariance.

In addition, to evaluate the performance of descriptors on scale invariant detectors, we select DOG (a very popular scale invariant detector, a simplified version of multi-scale LOG detector [7], detecting blob-like points as well) as the localization algorithm in the experiment of mobile image retrieval, since severe viewpoint changes are not the main deformations in this image dataset.

4.3 Evaluation Metric

We adopt the metrics in [14][5] to evaluate the performance of PPD, which are based on the number of correct matches and false matches obtained for a pair of images. More specifically, the following three metrics are adopted:

(1) **Correspondence metric:** $$\frac{\left\|\mathbf{D}_{k1}-\mathbf{D}_{k22}\right\|_2}{\left\|\mathbf{D}_{k1}-\mathbf{D}_{k21}\right\|_2} > \beta$$

Where k_1 represents a keypoint in one image; k_{21}, k_{22} refer to two keypoints in another image; D_{ki} represents the descriptor for k_i, $i \in \{1, 21, 22\}$; D_{k21}, D_{k22} are the first and the second nearest neighbor to D_{k1} respectively; β is the match threshold. The distance ratio, which can reflect the underlying distribution of descriptors in feature space, is commonly used to identify the correspondences most likely to be correct.

(2) **Correct match metric :** $$1-\frac{R_{\mu_a} \bigcap \mathbf{A}^T R_{\mu_b} \mathbf{A}}{R_{\mu_a} \bigcup \mathbf{A}^T R_{\mu_b} \mathbf{A}} < \varepsilon$$

Where R_μ represents the elliptic region defined by $\mathbf{x}^T \mu \mathbf{x} = 1$; $\mathbf{A}$ is a locally affine transformation of the homography between the two images; $R_{\mu_a} \bigcap \mathbf{A}^T R_{\mu_b} \mathbf{A}$ and $R_{\mu_a} \bigcup \mathbf{A}^T R_{\mu_b} \mathbf{A}$ represent the area intersection and area union respectively; ε refers to the overlap error.

(3) **Recall versus 1-precision:**

$$recall = \frac{\#of\ \ correct\ match}{\#of\ \ ground\ truch\ correspondence}$$

and

$$1-precision = \frac{\#of\ \ false\ match}{\#of\ \ total\ match}$$

This indicator can fully reflect the discriminative power of the descriptor. The recall versus 1-precision curve is obtained by varying the match threshold β and a perfect descriptor would give a recall close to 1 for any 1-precision. Note that the number of total match is determined by the correspondence metric and the numbers of correct match and false match need to be further determined by the correct match metric.

4.4 Experiment Results

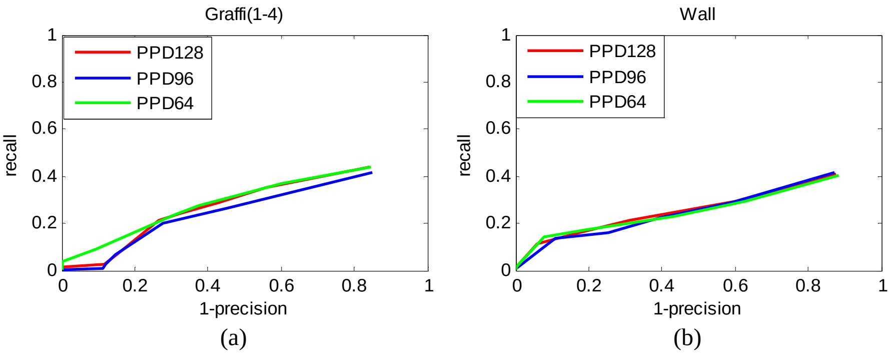
Because the SIFT-like representations have been identified as being most resistant to common image deformations [14], we choose to compare PPD with SIFT and SURF [21] (another simplified version of SIFT with excellent performance) in the experiments. Since the characteristic scale is not discernible over the normalized patches [24], we use the gradient to replace SURF's original Haar wavelet response and thus implement an approximate version of SURF on the normalized patches, which is called A-SURF64 in the following experiments. The bin codes of interest region detection (Harris-Affine, Hessian-Affine) can be downloaded from [24]; DOG and SIFT representation [29] are re-implemented by us for the experiments. The octaves for DOG are limited not to be beyond 6 and the initial scale is set to 1.6 [6]. ε is set to 50% [14]. All the test programs are running on a LAPTOP of AMD 1.9GHz, 2GM.

4.4.1 Scheme Selection

In practical application of image retrieval, choosing an appropriate phase-space partition scheme is quite necessary. Generally, more bins can readily capture more intensity pattern and more spatial information in the image patch. However, due to the sensitivity to noise or the limitations of the proposed scheme, the performance of a descriptor is likely to degrade or maintain at a certain level when more bins are added. Furthermore, more bins requires more storage space and more matching time. We test the performances of PPD with three partition schemes (4-region, 6-region and 8-region) on Hessian-Affine regions using the standard dataset. The experiment results are shown in **Figs. 5-8**.

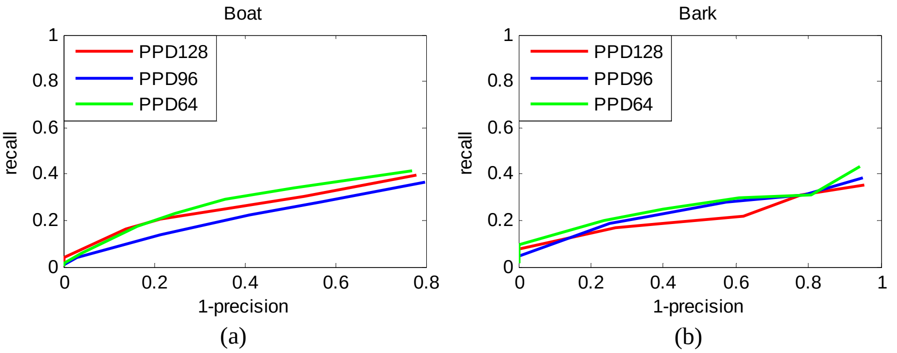
On the whole, there is a minor difference among them in all plots. Looking closely, we can also observe that 4-region scheme obtains a slightly better score in the geometric transformation scenes as compared with the other two schemes and comparable score in the other scenes. This can be attributed to the following two reasons. First, the geometric transformation of images often results in variations of the dominant orientations and more feature location errors, which usually enhance boundary affect between the bins of the histograms. The better performance of 4-region scheme in such scenes is due to the fact that it can offer larger regions compared with the other two schemes and these larger regions tend to tolerate more orientation errors and feature location errors. Second, lower dimensional feature vectors are in general less distinctive than their high-dimensional counterparts. When orientation estimation and feature location are sufficiently accurate, the high-dimensional description vectors are able to capture more structure information of image patterns. Thus, the outperformance of 8-region scheme is not surprising in most cases of the remaining non-geometric scenes. As can also be observed, however, the 8-region scheme is closely followed by the 4-region one and even surpassed by the latter (shown in **Fig. 8b**) in Ubc image set. This also confirms that the finer subdivisions of histogram bins appear to be less robust [21] provided that no other measures are taken to reduce the effect of inaccuracy introduced by image transformations and region detection.

In the next experiments, we only use the 4-region PPD to compare performance with SIFT due to its reasonable dimension and relatively leading performance in all of the scenes.



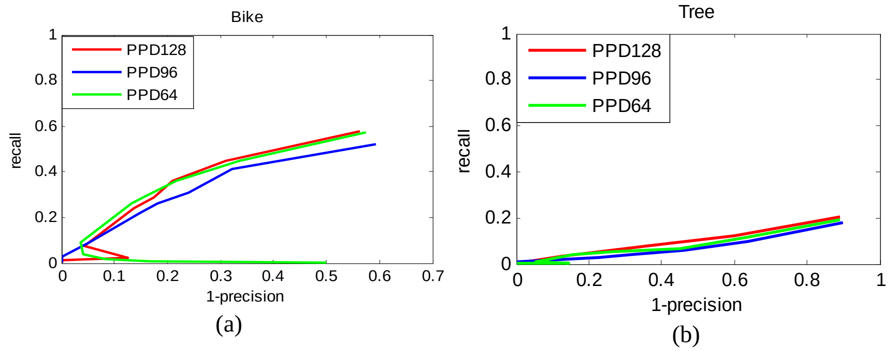
(a) (b)

Fig. 5. Viewpoint for structured scene (a), textured scene(b)



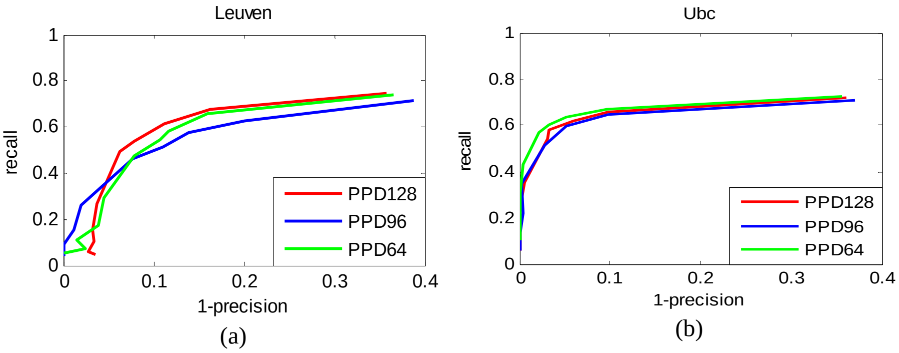
(a) (b)

Fig. 6. Scale+rotation for structured scene (a), textured scene(b)



(a) (b)

Fig. 7. Blur for structured scene (a), textured scene(b)



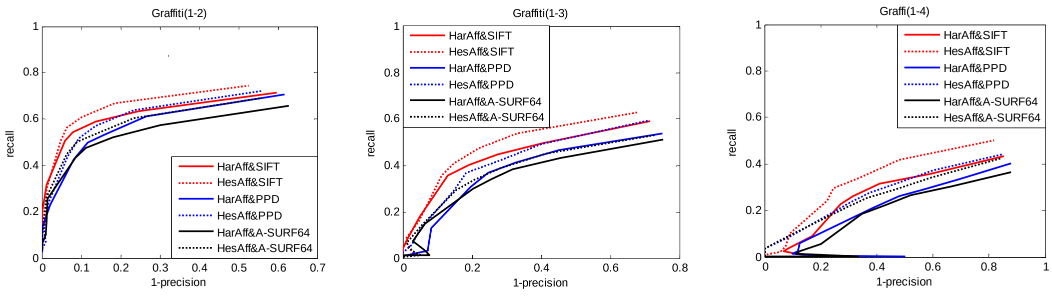
(a) (b)

Fig. 8. Illumination change (a), JPEG compression (b)

4.4.2 Recall versus 1-precision Performance Curve

Viewpoint changes:

We first evaluate the PPD descriptor under viewpoint changes using the "Graffiti" set. The experiment is conducted between the first image and the remaining images in the set in the order of gradually increasing transformations, only the results for three image pairs (shown in **Figs. 4a**, **4b**, and **4c**) are given due to the space limit. The viewpoint changes for them are approximately (a) 20° , (b) 30°, (c) 40°. In **Fig. 9a**, which shows the results for the image pair (1st, 2nd), we can observe that PPD obtains a lower score than SIFT for both Hessian-Affine regions and Harris-Affine regions. As the amount of viewpoint changes increases, the performance difference of the two algorithms also increases slightly, as shown in **Figs. 9a**, **9b**, and **9c**. Overall, the performance of PPD is comparably close to that of SIFT if the viewpoint changes are less than 40°, which commonly occurs in photometric scenes. In addition, A-SURF is getting increasingly closer to PPD with the severity of the transformation. This demonstrates that SURF representation is more robust to viewpoint change compared to PPD. The similar results, which are not shown for space limit either, can be obtained in the textured scenes (the Wall set).



(a) (b) (c)

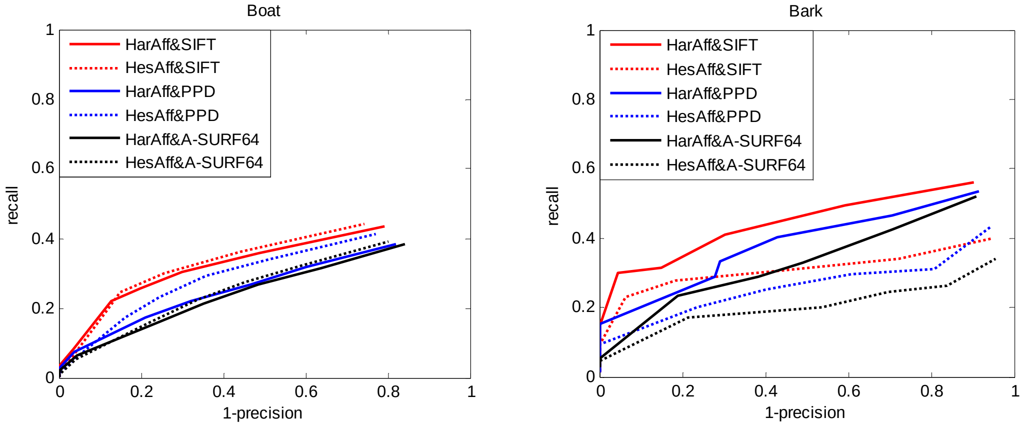
Fig. 9. Viewpoint changes for structured scene: (a) 20°, (b) 30°, and (c) 40°

Scale change and rotation:

Next, we test the performance of PPD for combined scale change and rotation. As can be observed from **Fig. 10**, there is a minor difference in performance between PPD and SIFT in both the Boat pair (shown in **Fig. 4e**) and the Bark pair (shown in **Fig. 4f**). As the 1-precision increases, PPD shows a trend to surpass SIFT; for example, when 1-precision reaches about 0.9, PPD outperforms SIFT in the Bark pair. Also, we can observe that the performance gap between them decreases compared with viewpoint changes. This is because the weakening geometric transformation leads to more accurate local image regions and reduces the impact of boundary affect on the descriptors. Accordingly, the advantage of SIFT's smoothed histogram is less distinct as compared with viewpoint changes.

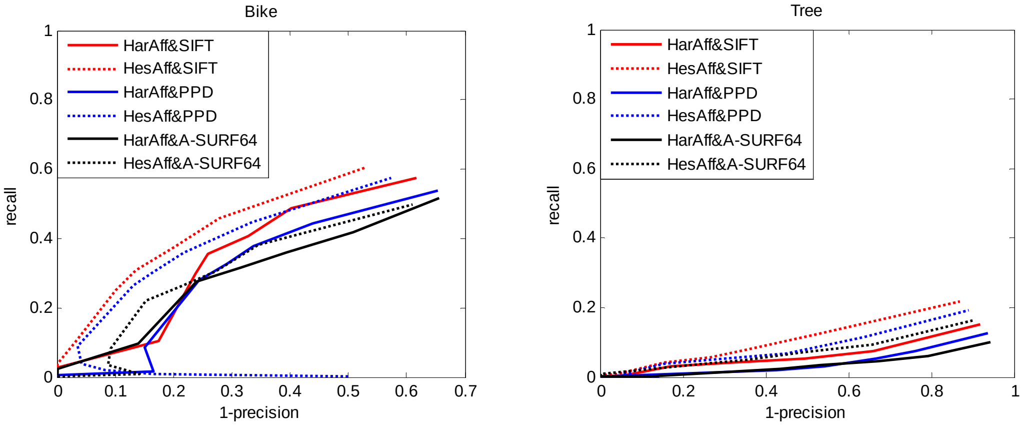
Images blur:

we also evaluate our descriptor under significant image blur. **Fig. 11a** shows the results for the structured scene (shown in **Fig. 4g**), and **Fig. 11b** for the textured scene (shown in **Fig. 4h**). PPD achieves comparable performance to SIFT in both of the scenes. In the textured scene, the significant amount of blur makes local image regions nearly identical, therefore the two algorithms both obtain rather low scores. The relatively low-dimensional PPD64 is not enough to discriminate these local image regions distinctively. In **Fig. 7b**, PPD128 outperforms PPD64 and achieves a better score close to that of SIFT.



(a) (b)

Fig. 10. Scale+rotation for structured scene (a), textured scene(b)



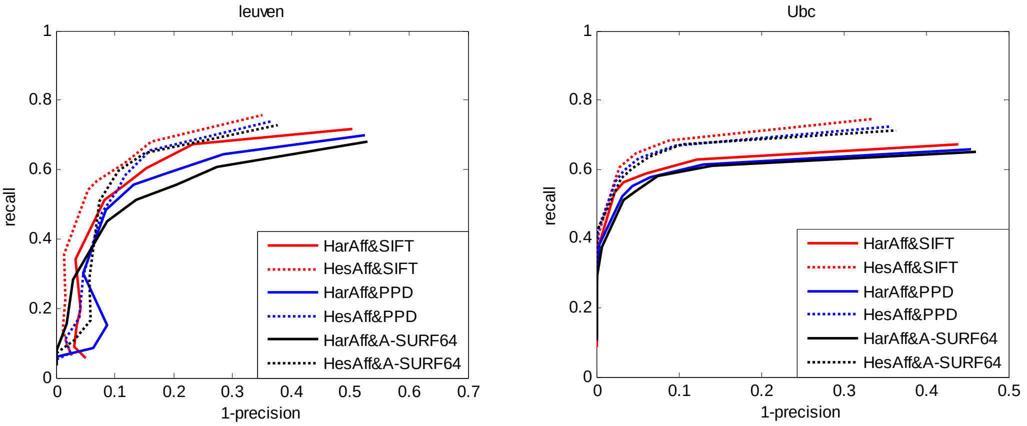
(a) (b)

Fig. 11. Blur for structured scene (a), textured scene (b)

Illumination change and JPEG compression:

Finally, under illumination change and JPEG compression, the difference in performance between PPD, A-SURF and SIFT is not quite apparent, as shown in **Fig. 12a** and **Fig. 12b**. Due to no significant geometric transformations in these images, the locations and shapes of local image regions are more accurate. Thus, all the three algorithms obtain much better scores in these scenes than in the other scenes. The performance differences between them are also reduced to the minimum level.

To sum up, SIFT representation is superior to PPD64 in most cases, but the differences between them are small. We can also observe that PPD sometimes locally outperforms SIFT when 1-precision is small (shown in **Figs. 9c**, **11a**, and **12b**). Additionally, in most of the experiments, PPD64 obtains slightly better scores than A-SURF64, especially on the Hessian-Affine regions. This is because that the accurate regions favor our method more as compared with A-SURF64 (A-SURF64 is relatively more robust). Such performance differences, however, seem to have no significant impact on image retrieval results, which is verified in the last experiment.



(a) (b)

Fig. 12. Illumination change for (a), JPEG compression (b)

4.4.3 Computation Cost Comparison

Table 1 compares the three algorithms in terms of the running time. In our test, the local image region detection algorithm and description algorithm are independent, therefore, the running time of detection and normalization for PPD64, A-SURF, and SIFT is identical. Since we focus on the comparison of the time spent on building descriptor, the running time is omitted for the first row. The second row shows the time for the three algorithms to compute descriptor representation. The final row shows their matching time respectively.

Table 1. Comparison of the average time consumption with PPD64, A-SURF64, and SIFT

	PPD64	**A-SURF64**	**SIFT**
Detection and normalization	—	—	—
Descriptor	0.4092s	0.4028s	1.6098s
Matching	0.5336s	0.5336s	1.0198s

From **Table 1**, we can observe that PPD64 is roughly four times faster in the descriptor construction and two times faster in the point pair matching than SIFT, but slightly slower than A-SURF64. Note that the time values in **Table 1** are the average of 5 times of tests, which are computed from 1758 local image regions obtained with Harris-Affine detector in the first image of the Graffiti set.

4.4.4 Mobile Image Retrieval

In this experiment, we integrate PPD, A-SURF, and SIFT in a real-application mobile image retrieval system, and compare their performance. The detail is described below. First, 5,000 images of different scenes have been scanned from the Magazine "Business Week" to create a reference image database for later retrieval. Then, 2,000 testing images are captured from the same scenes using different mobile phones by different people. Some testing images are showed in **Fig. 13**. As can be seen, there are significant image degradations in testing images mainly arising from non-linear illumination change, blur, and noise contamination. Also, viewpoint change, scale, and rotation change further increase the degradations. Overall these images can be approximately looked as planar surfaces; however, due to the disturbances coming from various light sources, there are apparent reflective phenomena of planar surfaces in testing images, as found on the “car” image in Fig. 13.

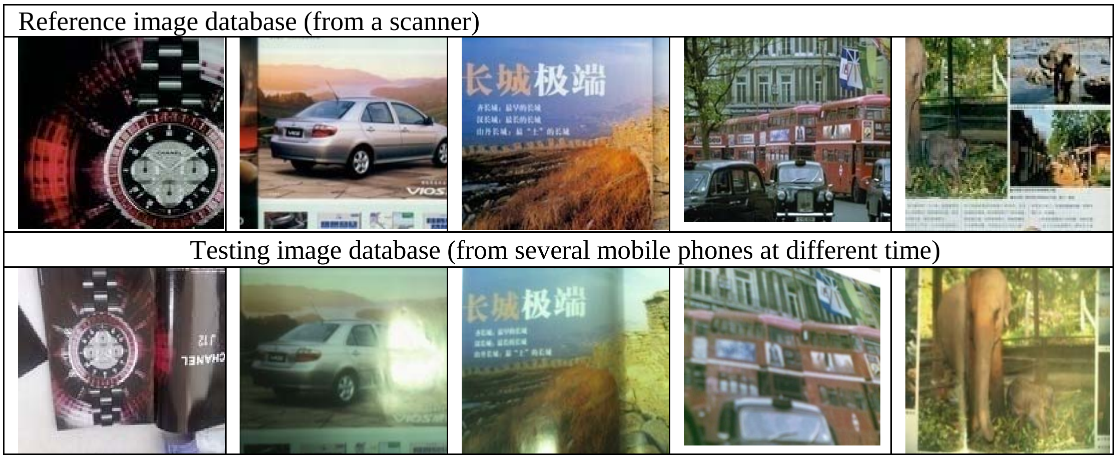

Reference image database (from a scanner)
Testing image database (from several mobile phones at different time)
长城极端
CHANEL
VIOS

Fig. 13. Data set 2, example images for mobile image retrieval

The image retrieval using SIFT descriptors (or A-SURF, PPD descriptors) is formulated as follows:

(1) For each image in the reference image database, we detect DOG interest points.
(2) For each interest region, we compute an SIFT descriptor (or A-SURF, PPD descriptor).
(3) Vector quantizes all of the descriptors in the reference image database into clusters, which compose a visual vocabulary.
(4) All the images in the reference image database are encoded using “TF-IDF” [2] to obtain respective high-dimensional content metric (a document vector).
(5) Each image in the testing image database is used as a query into the reference database. The above processing steps other than (3) are re-executed for each testing image to obtain the content metric.
(6) At the retrieval stage, images in the reference image database are ranked by their normalized scalar product (cosine of angle) between the query vector and all the document vectors in the reference database, and the first one is considered as the candidate image. Further, if the candidate image is the expected target image, the algorithm is awarded 1 point, and the retrieval accuracy rate of the algorithm is defined as its total score divided by 2000. The final retrieval results are illustrated in Table 2 where the three algorithms demonstrate comparable performance. Note that all of the images are first pre-processed by the saliency detection algorithm [30] to acquire high informational regions which we usually focus on and lessen computation demands.

Table 2. The practical results of image retrieval with PPD64, A-SURF64, and SIFT

	PPD64	**A-SURF64**	**SIFT**
Cluster algorithm	AKM [3]	AKM	AKM
Quantized Cluster centers	150,000	150,000	150,000
Accuracy rate	94.95%	94.50%	96.20%

The explanations for the retrieval results are as follows. First, the main reason of the results is due to the minor differences in quality of local features. In Section 4.4.2, the experiment results have demonstrated that the distinctiveness of local features created by the three algorithms is similar in the scenes of light change, blur, scale change, and rotation. Second, another reason is the approximate computations in the steps of clustering and encoding which significantly reduce requirements in terms of discriminative power for descriptors. The distinctiveness of the descriptor, which is very crucial to find exact correspondences among images, is less important in image retrieval, whose purpose is to find the most similar image as a whole. Third, the saliency detection algorithm [30] used for pre-processing also contributes much to the high scores for the three algorithms in terms of accuracy rate. Too many characters in images result in excessive feature points. This heavily increases the computation burden and makes the cluster computation impractical. The saliency detection helps us to obtain high informational regions while filtering out much background noise. As a result, 85% of the feature points are removed and a resultant increase in performance about 20-30% occurs.

In addition, the local intensively reflective phenomena of planar surfaces do not exert significant negative impact on the experiment results. This is because that the phenomenon can be regarded as a partial occlusion while local invariant features in themselves are not sensitive to it. Thus, the experiment results also confirm the robustness of the image retrieval based on local invariant features.

5. Conclusions

In this paper, we have presented a robust low-computation local descriptor called PPD. It gets inspiration from SIFT and uses the SIFT-like grid to capture the spatial information as well. However, instead of employing the discrete orientation based histogram, we introduce the regional statistics in the phase space as a feature representation, which has more robustness. PPD circumvents the time-consuming computation of orientation and can be implemented quite efficiently with a simple region partition rule. Experiment results show that PPD illustrates a favorable discrimination power and a significant reduction in computation requirements as compared to SIFT. Compared with A-SURF64, PPD obtains slightly better scores in most cases with comparable computation cost. In the practical mobile image retrieval experiment, PPD also yields comparable results in terms of the accuracy rate to SIFT.Genearlly, PPD offers an appropriate trade-off between performance and computation burden. Thus, we believe that PPD holds a great promise for such applications where low computation requirements are necessary. We will explore simplified alternative interpolation strategies to further enhance PPD in discriminance power and apply the ideas behind phase space partition to other description representations.

References

[1] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Proc. IEEE International Conference on Computer Vision*, vol. 1, pp. 525-531, 2001.

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

[3] 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.

[4] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: a comprehensive study," *Int. J. Comput. Vision*, vol. 73, no. 2, pp. 213-238, 2007.

[5] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *Int. J. Comput. Vision*, vol. 60, no. 1, pp. 63-86, 2004.

[6] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[7] T. Lindeberg, "Feature Detection with Automatic Scale Selection," *Int. J. Comput. Vision*, vol. 30, no. 2, pp. 79-116, 1998.

[8] J. Koenderink and A. J. van Doorn, "Representation of local geometry in the visual system," *Biological Cybernetics,* vol. 55, pp. 367-375, 1987.

[9] W. Freeman and E. Adelson, "The design and use of steerable filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 9, pp. 891-906, 1991.

[10] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets," in *Proc. European Conference on Computer Vision*, pp. 414-431, 2002.

[11] L.J.V. Gool, T. Moons, and D. Ungureanu, "Affine/photometric invariants for planar intensity patterns," in *Proc. European Conference on Computer Vision*, pp. 642-651, 1996.

[12] S. Lazebnik, C. Schmid, and J. Ponce, "Sparse Texture Representation Using Affine-Invariant Neighborhoods," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[13] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509-522, 2002.

[14] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615-1630, 2005.

[15] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3D objects," in *Proc. IEEE International Conference on Computer Vision*, vol. 1, pp. 800-807, 2005.

[16] Matthew Toews and William Wells III, "SIFT-Rank: Ordinal Description for Invariant Feature Correspondence," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[17] Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[18] E. Tola, V. Lepetit, and P. Fua, "A Fast Local Descriptor for Dense Matching," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[19] Jie Chen, Shiguang Shan, Chu He, Guoying Zhao, Matti Pietikäinen, Xilin Chen and Wen Gao, "WLD: A Robust Local Image Descriptor," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009.

[20] Yoon-Sik Tak and Eenjun Hwang, "Pruning and Matching Scheme for Rotation Invariant Leaf Image Retrieval," *KSII Transactions on Internet and Information Systems*, vol. 2, no. 6, 2008.

[21] H. Bay, T. Tuytelaars, and L.V. Gool, "SURF: speeded up robust features," *Computer Vision and Image Understanding*, pp. 346-359, 2008.

[22] M. Heikkilä, M. Pietikäinen and C. Schmid, "Description of Interest Regions with Local Binary Patterns," *Pattern Recognition*, vol. 42, no. 3, pp. 425-436, 2009.

[23] Chun-Rong Huang, Chu-Song Chen, and Pau-Choo Chung, "Contrast context histogram – An efficient discriminating local descriptor for object recognition and image matching," *Pattern Recognition,* vol. 42, no. 3, pp. 425-436, 2008.

[24] http://www.robots.ox.ac.uk/~vgg/research/affine/

[25] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions," *IVC,* September 10, pp. 761-767, 2004.

[26] T. Tuytelaars and L.V. Gool, "Matching widely separated views based on affine invariant regions," *Int. J. Comput. Vision*, vol. 59, no. 1, pp. 61-85, 2004.

[27] Timor Kadir, Andrew Zisserman, and Michael Brady, "An affine invariant salient region detector," in *Proc. European Conference on Computer Vision*, 2004.

[28] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L.V. Gool, "A comparison of affine region detectors," *Int. J. Comput. Vision*, vol. 65, no. 1/2, pp. 43-72, 2005.

[29] http://www.vlfeat.org/~vedaldi/code/sift.html

[30] X. Hou and L. Zhang, "Saliency detection: a spectral residual approach," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

Congxin Liu received a B.S. degree from Wuhan University of Hydraulic and Electric Engineering (Yi Chang), China, in 1997, an M.S. degree from Three Gorges University, China, in 2004. He is currently a Ph.D. student in the Department of Electronic and Electrical Engineering in Shanghai Jiao Tong University. His research interests include local invariant feature and image matching.

Jie Yang received a Ph.D. degree in computer science from the University of Hamburg, Germany in 1994. Dr Yang is now the professor of Institute of Image Processing & Pattern Recognition in Shanghai Jiao Tong University, China. He has taken charge of many research projects (e.g. National Science Foundation, 863 National High Tech. Plan) and published one book in Germany and more than 200 journal papers. His major research interests are image retrieval, object detection and recognition, data mining, and medical image processing.

Deying Feng received a B.S. degree from Shandong University of Technology, China, in 2005, an M.S. degree from Shanghai Maritime University, China, in 2008. He is now a Ph.D. student in Shanghai Jiao Tong University, China. His research interests include image retrieval and similarity search.