



# Universiteit Leiden

## Computer Science

Image Retrieval Using  
Local Feature Correspondences

Ran Tao

MASTER'S THESIS

Supervisor: Dr. Michael S. Lew

Leiden Institute of Advanced Computer Science (LIACS)  
Leiden University  
Niels Bohrweg 1  
2333 CA Leiden  
The Netherlands

# Image Retrieval Using Local Feature Correspondences

Ran Tao, and Michael S. Lew, *Member, IEEE*

**Abstract**—This paper investigates the problem of object-based image retrieval using local feature correspondences, under the mild assumption that more correspondences can be established between two images that contain the same object than two irrelevant ones. First, tentative keypoint correspondences are generated by comparing the SIFT descriptors. Then a two-view geometric relation is estimated using RANSAC to detect the mismatches. The similarity of two images is determined by the number of tentative matches and true correspondences, with the true matches being more important. Different techniques are explored to make the approach accurate and efficient. Particularly, a significant speedup is achieved by limiting the number of samples drawn in the hypothesize-and-verify procedure while the searching accuracy is minimally impacted due to the excellent property of early termination on relevant images of the locally optimized progressive sample consensus algorithm (LO-PROSAC). Promising results are obtained in the experiments on our 1000-image dataset.

**Index Terms**—Feature correspondence, content-based image retrieval, RANSAC, two-view geometry

## 1 INTRODUCTION

WITH the explosive growth of digital images, content-based image retrieval (CBIR) has become one of the most interesting research topics in the past two decades. A large number of techniques and methods have been developed, covering various fields related to CBIR. Comprehensive surveys are available [1], [2], [3]. Summarizations also exist on sub-topics such as image features [4], relevance feedback in image retrieval [5] and performance evaluation of CBIR systems [6]. Despite considerable progress has been made, CBIR, as a general problem, remains unsolved. This article focuses on the problem of object-based image retrieval, searching images of a particular object by an image query among an unorganized set of images.

In cluttered real world images, the same object often has different appearances, due to the phenomena like partial occlusion and deformation. Factors, such as illumination or 3D viewpoint change, make the retrieval task even more challenging. In this work, we try to do image retrieval by establishing local feature correspondences between two images, under the mild assumption that more feature correspondences can be generated between two images that contain the same object than two irrelevant ones.

Feature correspondence is an essential issue in computer vision. It is widely formulated as a graph matching task, trying to optimize a complex objective function based on the feature vectors and spatial constraints [7], [8].

Although good matching accuracy can be achieved, the fact that the optimization procedure is computationally complicated makes these algorithms unsuitable for image retrieval task.

An alternative matching paradigm is applied in this work, which is a two-step process. First, image keypoints are extracted and preliminarily matched by comparing the feature descriptors. Then spatial constraints are robustly estimated using the RANdom SAMple Consensus (RANSAC) algorithm [9] on the error-prone tentative correspondence set in order to detect the mismatches. The similarity measure of two images is based on the number of tentative matches created in the primary comparison and true correspondences consistent with the estimated spatial relation.

Different techniques are applied and compared both in the tentative correspondences generation step and the mismatch detection phase. Particularly, several RANSAC variations [15], [17], [19] are deeply studied in order to utilize their distinctive advantages to speed up the image retrieval procedure while keeping the performance minimally impacted in the aspect of searching accuracy.

The rest of the article is structured as follows. It begins with a revisit of the correspondence-based approach in Section 2. Then three two-view geometric relations are described in Section 3. Section 4 introduces the standard RANSAC algorithm and four variations in detail. Experimental results are presented in Section 5. We conclude in Section 6.

## 2 CORRESPONDENCE-BASED APPROACH

The core of this approach to image retrieval lies in local feature correspondence. We assume that two relevant images have more correspondences than two irrelevant ones. The whole approach proceeds as follows. First, local features are extracted and tentative correspondences be-

• Ran Tao is with Leiden Institute of Advanced Computer Science, Niels Bohrweg 1, 2333 CA, Leiden, The Netherlands.

E-mail: rantao.mail@gmail.com.

• Michael S. Lew is with Leiden Institute of Advanced Computer Science, Niels Bohrweg 1, 2333 CA, Leiden, The Netherlands.

E-mail: mlew@liacs.nl.

tween the query image and the target image are generated by comparing their local image descriptors. Incorrect matches, e.g., from the object of interest in one image to the cluttered background of the other, are inevitable in the preliminary comparison, since local feature is semantically meaningless and merely represents the appearance of a relatively small image patch. Therefore, spatial verification is then applied to detect and remove the incorrect correspondences. The similarity of the two images is obtained from the number of the tentative correspondences and true matches. The resulting images are returned in a descending order of the similarity measure. The diagram of the method is shown in Fig. 1. The following subsections review the techniques applied in each component of our method.

## 2.1 Local Feature

Local feature extraction is divided into two steps. It begins with detecting distinctive small regions in the image. Then from the selected regions, descriptors are extracted to represent the properties of these patches. In this work, the difference-of-Gaussian (DoG) approach [10] is applied for interest point detection and the SIFT descriptor [10] for representation.

The DoG method identifies potential keypoints by finding the local extrema of the DoG images created in Gaussian scale space [21], followed by a finer keypoint localization and rejection of edge points and those with low contrast. Each keypoint is then assigned a dominant orientation in order to achieve rotation invariance.

A SIFT descriptor is created over the  $4 \times 4$  subregions around the keypoint location. Within each subregion, the pixel-wise oriented gradients are accumulated into orientation histograms with 8 bins. Therefore, the resulting descriptor is of dimension 128.

Using the DoG detector and SIFT descriptor, each image is represented by a set of 128-D vectors.

## 2.2 Tentative Correspondences Generation

The correspondences of the keypoints are established between the query and the target image by comparing the SIFT descriptors. For each keypoint, its nearest neighbor in the set of keypoints from the other image is identified as its candidate match, which is accepted only when passing Lowe’s distance ratio test [10], i.e., the ratio of the distance from the nearest neighbor to that of the second nearest is below a certain threshold. The nearest neighbor is defined as the keypoint with minimum Euclidean distance in the SIFT feature space. Both the Best-Bin-First (BBF) search scheme [11] and the randomized kd-trees approach (RKD) [12], [13] are tried in this work to efficiently identify the approximate nearest neighbors in the 128-D space.

### 2.2.1 Best-Bin-First

BBF is a new form of the kd-tree search [22]. The superiority of BBF over the standard version of kd-tree is the priority search [11] which checks the bins in the order of increasing distance from the query point in the process of backtracking. The distance of a bin from the query point is defined as the minimal distance between the query and

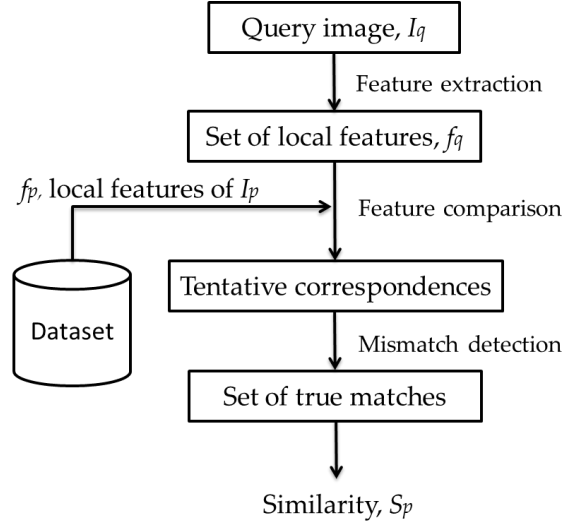


Fig. 1. Overall structure of the correspondence-based approach.

any point on the boundary of the bin.

Given a query point, the priority search in a kd-tree proceeds as follows. It first descends the tree and finds the bin containing the query point. Then it examines the bins in the order of their distance from the query point. The search terminates when no more bins are within the distance between the query and the best candidate.

By limiting the number of bins to examine, BBF turns into an approximate version, which returns the best neighbor found up to that point as the approximate nearest neighbor.

### 2.2.2 Randomized Kd-trees

RKD is an improved version of BBF in which multiple randomized kd-trees are created. BBF builds the single kd-tree by splitting the data in half on the dimension where the data have the greatest variance. In a different manner, randomized kd-trees are created by selecting the split dimension randomly from the top  $N$  dimensions with the highest variance.  $N$  is set to be 5 in the implementation.

In the searching phase, only one priority queue is maintained across multiple trees. After descending each of the trees, the priority search scheme repeatedly picks the closest bin from the queue. An approximate nearest neighbor is returned after examining a fixed number of bins.

By splitting the data space in different ways using multiple kd-trees, RKD has a larger chance of finding the optimal solution than BBF within a fixed number of examinations.

## 2.3 Mismatch Detection and Elimination

A certain global geometric model is drawn from the set of tentative correspondences, and those matches inconsistent with the model are recognized as incorrect correspondences. Considering its robust performance in the presence of large fraction of outliers, RANSAC is applied to estimate the geometric relation.

The adopted geometric models and RANSAC varia-

tions are described in detail in Section 3 and Section 4 respectively.

## 2.4 Similarity Measure

Since only for a pair of relevant images, a large number of true correspondences can be obtained after the mismatch elimination process, the number of inliers is a significant indicator of two images' similarity. Though less important, the number of tentative matches is also a factor in determining the similarity, under the assumption that two relevant images that contain the same object tend to have more local feature correspondences established in the primary comparison. Hence, the similarity of the query and the target image is calculated on the basis of the number of tentative matches and true correspondences, with the correct matches being more important. The similarity score  $Sim$  is defined as

$$Sim = 0.8 * \#true + 0.2 * \#tentative, \quad (1)$$

where  $\#true$  is the number of true correspondences and  $\#tentative$  is the number of tentative matches.

## 3 GEOMETRIC RELATIONS

This section describes the geometric relations which can constrain the motion of points between two views. These relations are robustly estimated in order to detect the incorrect matches in the set of tentative correspondences between the query image and the target image. Three two-view geometric relations are employed and compared in our work: fundamental matrix, homography and affinity [14].

**Fundamental matrix.** Fundamental matrix is the algebraic representation of epipolar geometry [14], which is the intrinsic projective geometry between two views. Suppose the two keypoints of a correspondence  $(\bar{x}; \bar{x}')$  arise from the same 3D point in real world, then the image points  $\bar{x}$  and  $\bar{x}'$  satisfy the relation  $\bar{x}'^T F \bar{x} = 0$ , where  $F$  is the fundamental matrix. Epipolar geometry is independent of the scene structure.

Fundamental matrix  $F$  is a 3\*3 matrix, with the rank of 2. It can be derived from the image correspondences alone, without reference to the camera matrices. At least 7 correspondences are required to compute  $F$ .

**Homography.** Homography can be considered as a special case of epipolar geometry. For a true match  $(\bar{x}; \bar{x}')$ , the epipolar geometry restricts the position of  $\bar{x}'$  to be on the epipolar line defined by  $F\bar{x}$ , while the homography constrains  $\bar{x}'$  to be fixed at point defined by  $H\bar{x}$ , where  $H$  is the representation of homography. Homography can arise for planar scenes, i.e., two images of the same planar surface in 3D world are related by a homography. It also occurs for non-planar scenes when two images are taken by the camera that only undergoes a rotation about its optic center.

Homography  $H$  is a 3\*3 invertible matrix. It can also be computed merely from the image correspondences. The minimal number of required correspondences is 4.

**Affinity.** The image-image affinity is a specialization

TABLE 1  
SUMMARIZATION OF THE THREE GEOMETRIC RELATIONS

Model	C	P	Constraint	Parameters
Fundamental Matrix	7	7	$\bar{x}'^T F \bar{x} = 0$	$F = \begin{pmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{pmatrix}$
Homography	4	8	$\bar{x}' = H \bar{x}$	$H = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}$
Affinity	3	6	$\bar{x}' = H_A \bar{x}$	$H_A = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & a_7 \end{pmatrix}$

$C$  is the minimal number of correspondences required to compute the model.  
 $P$  is the number of parameters in the model.

of homography. It occurs in affine image condition [14], i.e., the field of view is small and the distance from the camera center to the viewed object is relatively large compared to the depth relief of the object.

Affinity  $H_A$  is a 3\*3 matrix, with the first two elements of the last row being zero. It can be computed from at least 3 correspondences.

The properties of the three geometric relations are summarized in Table 1.

## 4 RANSAC FAMILY

The original RANSAC algorithm was proposed by Fishler and Bolles in 1981 [9]. Afterwards, numerous algorithms have been derived from RANSAC. In this section, we first review the standard RANSAC algorithm. Then three RANSAC variations are introduced, which have properties suitable for image retrieval. Finally, a new variation is proposed.

### 4.1 Standard RANSAC

RANSAC is widely applied to estimate the parameters of a model using data which are contaminated by outliers. The data that are consistent with the desired model are called inliers and the rest are outliers.

Generally speaking, the input of the RANSAC algorithm is a set  $\Omega$  of data points. The goal is to find the optimal parameters  $\theta'$  of a model that maximize the number of inliers. RANSAC simply iterates two steps: generating a hypothesis of the parameters and verifying it to the data. (i) In the hypothesis generation step, a sample  $S_k$  of  $m$  data points is randomly drawn from  $\Omega$  and a hypothesis  $\theta^k$  of the parameters is computed from  $S_k$ . (ii) In the hypothesis verification step, the support set  $I_k$ , i.e., the set of inliers consistent with  $\theta^k$  is calculated and the quality of  $\theta^k$  is indicated by the cardinality of  $I_k$ ,  $|I_k|$ . The two steps are repeated until the probability  $\eta$  of finding a better model falls under a threshold  $\eta_0$ . Under

the assumption that a model computed from an all-inlier sample is consistent with all inliers in the whole set  $\Omega$  [15], the termination criterion can be expressed in an alternative way: after  $k$  samples, the probability  $\eta$  of all these  $k$  samples being contaminated by at least one outlier is below a threshold  $\eta_0$ ,

$$\eta = (1 - \varepsilon^m)^k \leq \eta_0, \quad (2)$$

where  $\varepsilon$  is the fraction of inliers in  $\Omega$ . Since  $\varepsilon$  is unknown beforehand, it is updated once a new maximum is reached in the iteration.

Specifically, RANSAC is applied in our work to estimate the geometric relation that constrains the motion of points in two images. In this section, we use the fundamental matrix to illustrate how the RANSAC algorithm works. The input of RANSAC is a set  $\Omega$  of tentative keypoint correspondences,  $\Omega = \{(\bar{x}_i; \bar{x}'_i) \mid i = 1, 2, \dots, n\}$ , where  $\bar{x}_i$  belongs to the query image and  $\bar{x}'_i$  is the corresponding point in the target image, and  $n$  is the number of tentative matches. The output is the optimal number of inliers that are consistent with the estimated geometric model. The error function representing the distance of a correspondence to the estimated fundamental matrix is simply defined as

$$\rho(F, (\bar{x}; \bar{x}')) = d(\bar{x}, F\bar{x}), \quad (3)$$

where  $d(\bar{x}, \bar{l})$  is the distance in pixel between a point  $\bar{x}$  and a line  $\bar{l}$ . Correspondences with the error smaller than  $\Delta$  are considered to be inliers. The RANSAC framework for estimating the fundamental matrix is as follows.

**Algorithm 1:** The structure of RANSAC for estimating fundamental matrix.

**Input:**  $\Omega$ ,  $\eta_0 (= 0.01)$ ,  $\Delta (= 3)$

**Repeat** until  $\eta \leq \eta_0$

1. Hypothesis generation
  - a) Randomly select a sample  $S_k$  of  $m (= 7)$  from  $\Omega$ .
  - b) Calculate the fundamental matrix  $F_k$  using the 7-point algorithm [14], [16]. There will be one or three real solutions.
2. Hypothesis verification
  - a) Calculate the support set  $I_k$  of  $F_k$ . If there are three solutions, keep the solution with the largest support set.
  - b) If  $I_k$  is larger than any previous set, i.e., a new maximum is reached, store  $I_k$ ,  $I^* = I_k$ , and update the fraction of inliers,  $\varepsilon = I^* / n$ .

**Output:** If the non-randomness requirement [17] is satisfied, calculate  $F^*$  from  $I^*$  using the normalized 8-point algorithm [14], [18] and output the cardinality of the support set of  $F^*$ . Otherwise, output zero.

As pointed out in [14], using 7-point algorithm in the iteration has two advantages. One is that the fundamental matrix obtained is rank 2 and it is not necessary to force it to be rank 2 in an extra step as in the 8-point algorithm. The second is that using 7 correspondences instead of 8

can decrease the probability of selecting a contaminated sample in the hypothesis generation step. Normalized 8-point algorithm is applied in the final step since it performs better than the 7-point algorithm when nine or more correspondences are available, which is shown in [15]. The non-randomness requirement [17] prevents the RANSAC algorithm from choosing an incorrect model which by chance gains a large support.

## 4.2 Randomized RANSAC

Randomized RANSAC with  $T_{d,d}$  test (R-RANSAC) [19] is a RANSAC variation aiming to speed up the standard RANSAC. The running time  $T$  of the standard RANSAC can roughly be expressed as

$$T = k(T_G + nT_v) \quad (4)$$

where  $k$  is the number of samples,  $T_G$  is the time for generating a hypothesis from the sample,  $T_v$  is the average time needed to evaluate the hypothesis on each datum, and  $n$  is number of data points in the whole set  $\Omega$ .

There are usually two ways to make RANSAC faster: (i) reducing the necessary number of samples drawn, i.e.,  $k$  in (4) and (ii) reducing the number of data points that are evaluated in the hypothesis verification step. The scheme of R-RANSAC belongs to the latter one.

At the very beginning of the hypothesis verification step, R-RANSAC performs the  $T_{d,d}$  test, and full evaluation on the whole data is carried out only for those promising hypotheses that pass the  $T_{d,d}$  test. As defined in [19],  $T_{d,d}$  test is passed if all  $d$  randomly selected points are consistent with the hypothesized model. In this manner, those erroneous hypotheses generated from contaminated samples are probably rejected by the  $T_{d,d}$  test and the average number of data evaluated for each hypothesis is smaller than  $n$ .

R-RANSAC is very suitable for the image retrieval task, because most generated hypotheses are erroneous, especially between two irrelevant images. By rejecting the incorrect models in the preverification stage, R-RANSAC can achieve a speedup. In our implementation,  $d$  is set to be 1 as suggested in [19].

**Algorithm 2:** The structure of R-RANSAC with  $T_{1,1}$  test for estimating the fundamental matrix.

**Input:**  $\Omega$ ,  $\eta_0 (= 0.01)$ ,  $\Delta (= 3)$

**Repeat** until  $\eta \leq \eta_0$

1. Hypothesis generation
  - a) Randomly select a sample  $S_k$  of  $m (= 7)$  from  $\Omega$ .
  - b) Calculate the fundamental matrix  $F_k$  using the 7-point algorithm. There will be one or three real solutions.
2. Hypothesis verification
  - a) Perform the  $T_{1,1}$  test. If no solution passes the test, jump to the hypothesis generation step. Otherwise, calculate the support set  $I_k$  of  $F_k$ . If two or three solutions are available, keep the one with largest support set.
  - b) If  $I_k$  is larger than any previous set, i.e., a

new maximum is reached, store  $I_k$ ,  $I^* = I_k$ , and update the fraction of inliers,  $\varepsilon = |I^*|/n$ .

**Output:** If the non-randomness requirement is satisfied, calculate  $F^*$  from  $I^*$  using the normalized 8-point algorithm and output the cardinality of the support set of  $F^*$ . Otherwise, output zero.

### 4.3 Locally Optimized RANSAC

The locally optimized RANSAC (LO-RANSAC) [15] improves the standard RANSAC by adding a local optimization step (LO step), which is carried out only if a new maximum in the number of inliers is reached. The brief structure of LO-RANSAC for computing fundamental matrix is presented below.

Algorithm 3: The structure of LO-RANSAC for estimating fundamental matrix.

**Input:**  $\Omega$ ,  $\eta_0 (=0.01)$ ,  $\Delta (=3)$

**Repeat** until  $\eta \leq \eta_0$

1. Hypothesis generation
  - a) Randomly select a sample  $S_k$  of  $m (=7)$  from  $\Omega$ .
  - b) Calculate the fundamental matrix  $F_k$  using the 7-point algorithm. There will be one or three real solutions.
2. Hypothesis verification
  - a) Calculate the support set  $I_k$  of  $F_k$ , i.e., the correspondences with error smaller than  $\Delta$ . If there are three solutions, keep the solution with the largest support set.
  - b) If  $I_k$  is larger than any previous set, i.e.,  $|I_k| > |I_j|$  for all  $j < k$ , then run **LO step**. If the model  $F_{LO}$  obtained in the LO step is better than  $F_k$ , then  $I^* = I_{LO}$ ; otherwise,  $I^* = I_k$ . Update the fraction of inliers,  $\varepsilon = |I^*|/n$ , where  $n = |\Omega|$ .

**Output:** If the non-randomness requirement is satisfied, calculate  $F^*$  from  $I^*$  using the normalized 8-point algorithm and output the cardinality of the support set of  $F^*$ . Otherwise, output zero.

Note, the termination criterion of the algorithm is on the basis of the optimal support set  $I^*$  while the execution of the local optimization step depends on  $I_k$ . The LO step applied is the so-called *inner RANSAC with iteration* [15].

Due to the LO step, more inliers are detected, and consequently the number of samples drawn decreases. Comparing with the standard RANSAC, LO-RANSAC speeds up the hypothesize-and-verify iteration by generating fewer samples. This decent property of early termination<sup>1</sup> on relevant images is explored, which is discussed in the experiments.

### 4.4 Progressive Sample Consensus

In standard RANSAC, samples are drawn randomly from the whole data. The progressive sample consensus (PROSAC) [17] uses a new sampling strategy. Under the as-

sumption that tentative correspondences with higher similarity are more likely to be inliers, PROSAC first sorts the correspondences in the descending order of their similarity and then draws samples among the top-ranked data. The size of the subset from which samples are generated is increased gradually.

Besides, there is a termination length optimization step in PROSAC. The optimal termination length  $n^*$  is defined as

$$n^* = \arg \max_{m < n' \leq n} (|I_{n'}|/n') \quad (5)$$

where  $|I_{n'}|$  is the number of inliers in the top  $n'$  data points. The fraction of inliers  $\varepsilon$  in (2) is then updated if  $|I_{n'}|/n'$  is a new maximum.

Algorithm 4: The structure of PROSAC for estimating fundamental matrix.

**Input:**  $\Omega$ ,  $\eta_0 (=0.01)$ ,  $\Delta (=3)$

**Sort** the tentative correspondences in ascending order of the Euclidean distance of their SIFT descriptors.

**Repeat** until  $\eta \leq \eta_0$

1. Hypothesis generation
  - a) Draw a sample  $S_k$  of size  $m (=7)$  from  $\Omega$  using the guided sampling strategy.
  - b) Calculate the fundamental matrix  $F_k$  using the 7-point algorithm. There will be one or three real solutions.
2. Hypothesis verification
  - a) Calculate the support set  $I_k$  of  $F_k$ . If there are three solutions, keep the solution with the largest support set.
  - b) If  $I_k$  is larger than any previous set, i.e., a new maximum is reached, then  $I^* = I_k$ , and run the termination length optimization step to get the optimal length  $n^*$ . If  $|I_{n'}|/n'$  is a new maximum, then update the fraction of inliers, i.e.,  $\varepsilon = |I_{n'}|/n'$ .

**Output:** If the non-randomness requirement is satisfied, calculate  $F^*$  from  $I^*$  using the normalized 8-point algorithm and output the cardinality of the support set of  $F^*$ . Otherwise, output zero.

The guided sampling strategy enables the most promising samples being examined early, which means a relatively large  $\varepsilon$  can be obtained in early stage and the algorithm can hence terminate early according to (2). The termination length optimization can further speed up the algorithm by finding an optimal fraction of inliers  $|I_{n'}|/n'$ . The property of early termination of PROSAC only holds for relevant images, since the fundamental assumption of PROSAC is valid only for two views of the same scene.

### 4.5 Locally Optimized PROSAC

By adding the local optimization step of LO-RANSAC into PROSAC, we propose a new RANSAC variation, named locally optimized progressive sample consensus (LO-PROSAC). The target is to further strengthen the advantage of early termination on two relevant images. The

<sup>1</sup> In this article, the property of early termination of a RANSAC variation refers in particular to the fact that the variation draws fewer samples than the standard RANSAC to meet the termination criterion.

structure of LO-PROSAC is as follows.

Algorithm 5: The structure of LO-PROSAC for estimating fundamental matrix.

**Input:**  $\Omega$ ,  $\eta_0(=0.01)$ ,  $\Delta(=3)$

**Sort** the tentative correspondences in ascending order of the Euclidean distance of their SIFT descriptors.

**Repeat** until  $\eta \leq \eta_0$

1. Hypothesis generation
  - a) Draw a sample  $S_k$  of size  $m(=7)$  from  $\Omega$  using the guided sampling strategy.
  - b) Calculate the fundamental matrix  $F_k$  using the 7-point algorithm. There will be one or three real solutions.
2. Hypothesis verification
  - a) Calculate the support set  $I_k$  of  $F_k$ . If there are three solutions, keep the solution with the largest support set.
  - b) If  $I_k$  is larger than any previous set, i.e.,  $|I_k| > |I_j|$  for all  $j < k$ , then run **LO step**. If the model  $F_{LO}$  obtained in the LO step is better than  $F_k$ , then  $I^* = I_{LO}$ ; otherwise,  $I^* = I_k$ . Based on  $I^*$ , run the termination length optimization to get  $n^*$ . If  $|I_n|/n^*$  is a new maximum, update the fraction of inliers,  $\varepsilon = |I_n|/n^*$ .

**Output:** If the non-randomness requirement is satisfied, calculate  $F^*$  from  $I^*$  using the normalized 8-point algorithm and output the cardinality of the support set of  $F^*$ . Otherwise, output zero.

Combining local optimization step, guided sampling strategy and termination length optimization, LO-PROSAC is supposed to terminate earlier on relevant images than LO-RANSAC and PROSAC by using fewer samples to obtain a support set that is large enough to meet the termination criterion specified by (2).

## 5 EXPERIMENTAL RESULTS

Various experiments were carried out for a comprehensive empirical study of our correspondence-based approach. The effectiveness of local feature correspondences in object-based image retrieval is shown by the comparison with three baseline methods applying global color and texture image features (in experiment #1). The three two-view geometric relations, i.e., fundamental matrix, homography and affinity, are compared (in experiment #2). Besides, the advantage of RKD over BBF is studied (in experiment #3). And the properties of different RANSAC variations are investigated and exploited to speed up the image retrieval procedure (in experiment #4). All the experiments were done on a desktop with a mainstream processor (a single core of a 3.16 GHz Intel Core Duo E8500 CPU).

**Data collection.** A dataset of 1000 images was collected using a common digital camera, including both outdoor scenes, such as windmill, and indoor objects like fridge or sofa. All the images were resized to be 512 (in pixel) in the longer edge. The collection was manually

annotated and a ground truth with 30 queries was generated. Exemplar images are shown in Fig. 2.

**Evaluation measure.** The accuracy is measured using  $R(25)$ ,  $R(50)$ ,  $R(75)$  and  $R(100)$ , i.e., *recall* of the top 25, 50, 75 and 100 resulting images. The number of 25 is determined by two reasons. The first reason is that we assume 25 images are shown per page. The other one is based on the fact that the maximal number of relevant image for a query in our ground truth is 23 ( $<25$ ). 100 is an estimation of the maximal number of resulting images that a user is willing to check. In the aspect of speed, four measures are used: *kdtime*, *ransactime*, *#sample* and *totaltime*. The metric, *kdtime*, is the average time of establishing tentative correspondences between two images and *ransactime* is the time the RANSAC-like algorithm spends on each image pair. We use the term of ‘‘RANSAC-like algorithm’’ to generally refer to the standard RANSAC or any of the four variations described in the previous section. The average number of samples drawn in the hypothesize-and-verify procedure is represented by *#sample*. The measure, *totaltime*, is the sum of *kdtime* and *ransactime*. The unit of *kdtime*, *ransactime* and *totaltime* is millisecond (ms).

### 5.1 Experiment #1

As pointed out, systems using color histogram often serve as a simple baseline for image retrieval [4]. This subsection compares the correspondence-based approach with three systems created using HSV color histogram and local binary pattern (LBP) [20], which is a widely applied texture feature. The four systems are described as follows.

**HSVsys.** *HSVsys* uses color histogram in the HSV color space. Bin size is set as 16 for *hue*, 4 for *saturation* and 4 for *value*, resulting in a 256-D feature vector. The 256-D vector is then normalized to unit length.

**LBPsys.** *LBPsys* applies LBP in its simplest form. The local binary pattern for each pixel is determined by comparing the pixel with each of its 8 neighbors in a certain order, which gives an 8-digit binary number. The LBP feature vector is the histogram of the frequencies of the occurring binary patterns. The 256-D LBP feature vector is normalized to unit length.

**HSVLBPsys.** *HSVLBPsys* adopts both HSV color histogram and LBP histogram.

**CORRESsys.** For the correspondence-based approach, named as *CORRESsys* within this subsection, BBF is applied for approximate nearest neighbor search in the tentative correspondences generation step and standard RANSAC is employed to estimate a homography relation between two images to detect the mismatches.

The average results on 30 queries are shown in Table 2. And due to the randomness of the RANSAC algorithm, the correspondence-based method was executed 10 times and the results were averaged.

The decent result of *HSVsys* explains the reason that chooses HSV color histogram to develop a baseline system. The much better result achieved by *CORRESsys* demonstrates the effectiveness of local feature correspondences for object-based image retrieval. Color histogram and LBP, as global features, are vulnerable to the phenomena like partial occlusion and cluttered back-

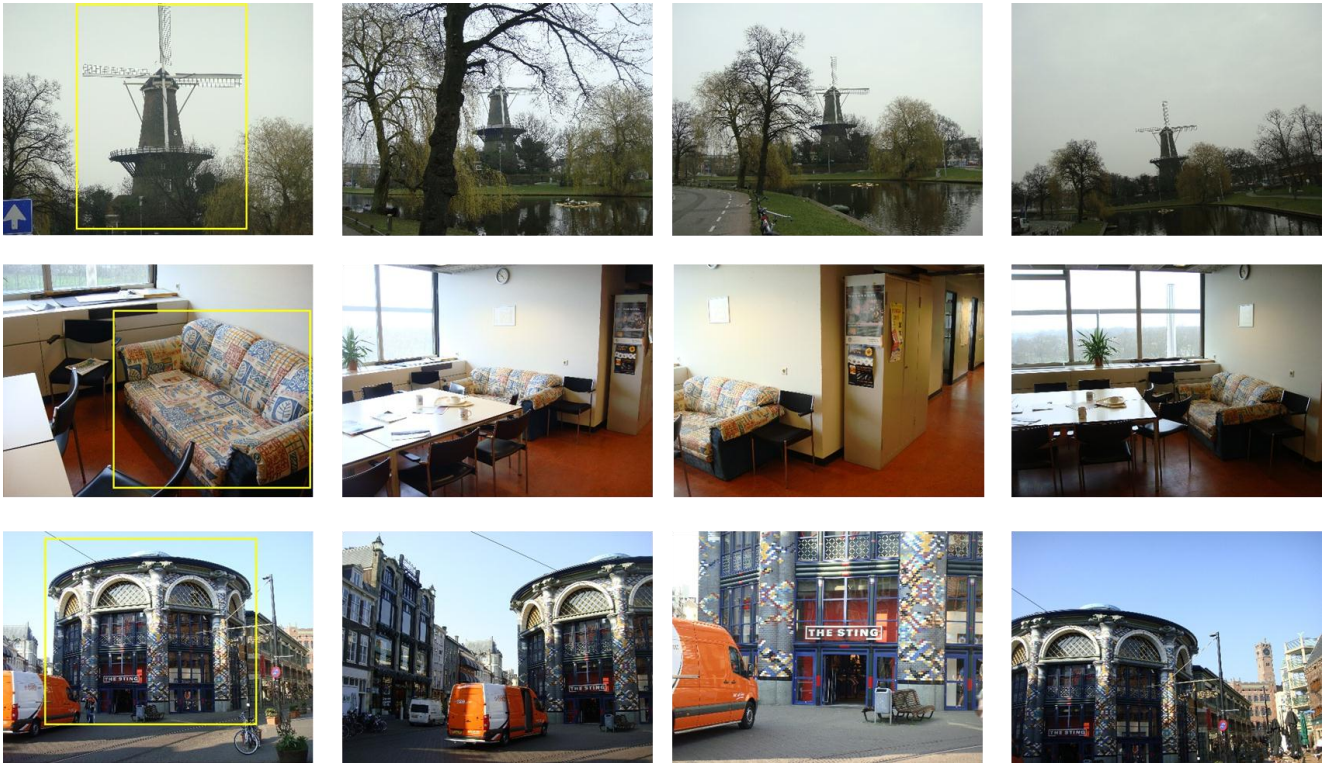


Fig. 2. Exemplar images. The images in the first column are query images with the object of interest highlighted. The other three images in each row are the relevant images of the query.

TABLE 2  
THE ACCURACY OF  $HSV_{sys}$ ,  $LBP_{sys}$ ,  $HSVLP_{sys}$  AND  $CORRES_{sys}$

	R(25)	R(50)	R(75)	R(100)
$HSV_{sys}$	0.3205	0.4136	0.4784	0.5266
$LBP_{sys}$	0.1965	0.3073	0.3491	0.3913
$HSVLP_{sys}$	0.3159	0.4163	0.4816	0.5369
$CORRES_{sys}$	<b>0.5750</b>	<b>0.6396</b>	<b>0.6811</b>	<b>0.7034</b>

grounds, making the three systems inferior to the correspondence-based approach, which is, by comparison, robust against occlusion, clutter and geometric transformations like scale and rotation.

## 5.2 Experiment #2

In order to compare the performance of the three two-view geometric relations, three correspondence-based approaches,  $F_{sys}$ ,  $H_{sys}$  and  $A_{sys}$ , were constructed and evaluated, which use fundamental matrix, homography and affinity respectively in the mismatch detection step. The other parts are the same, i.e., standard RANSAC as the robust estimator and BBF for approximate nearest neighbor search. Each of these three approaches ran 10 times on the 30 queries. The average result is presented in Table 3.

Both homography and affinity impose a strict constraint on the motion of two views since these two relations only arise for planar scenes (assuming the images

TABLE 3  
THE ACCURACY OF  $F_{sys}$ ,  $H_{sys}$  AND  $A_{sys}$  WHICH RESPECTIVELY USE FUNDAMENTAL MATRIX, HOMOGRAPHY AND AFFINITY AS THE GEOMETRIC RELATION

	R(25)	R(50)	R(75)	R(100)
$F_{sys}$	0.6745	0.7043	0.7301	0.7546
$H_{sys}$	0.5750	0.6396	0.6811	0.7034
$A_{sys}$	0.5839	0.6421	0.6870	0.7116

are not captured by rotating cameras), theoretically. Few inliers can survive from the mismatches elimination process using either homography or affinity, as tentative matches that are not on or near the plane specified by the estimated relation are all considered to be outliers. Therefore, for non-planar scenes, the approach applying homography or affinity loses its ability of distinguishing relevant images and irrelevant ones in terms of the number of inliers. By comparison, fundamental matrix, independent of the scene structure, is a more general geometric model. Applying fundamental matrix gives the highest accuracy.

Similar performance in the aspect of searching accuracy is achieved on our dataset using homography and affinity. Considering the computation of homography from correspondences is much more complex, affinity outperforms homography.

## 5.3 Experiment #3

Silpa-Anan and Hartley [12] have shown that using mul-

TABLE 4  
THE ACCURACY OF THE FIVE APPROACHES THAT USE 1, 3, 5, 8, 10 KD-TREES RESPECTIVELY

	R(25)	R(50)	R(75)	R(100)
BBFsys	0.6749	0.7043	0.7282	0.7535
RKD3sys	0.6911	0.7264	0.7551	0.7821
RKD5sys	0.6918	0.7278	0.7569	0.7860
RKD8sys	0.6926	0.7280	0.7571	0.7869
RKD10sys	0.6915	0.7269	0.7580	0.7896

TABLE 5  
THE PERFORMANCE ON SPEED OF THE FIVE APPROACHES THAT USE 1, 3, 5, 8, 10 KD-TREES RESPECTIVELY

	kdtime(ms)	ransactime(ms)	#sample	totaltime(ms)
BBFsys	12.03	56.05	1576.59	68.08
RKD3sys	15.52	43.29	1215.95	58.81
RKD5sys	17.53	37.57	1053.38	55.10
RKD8sys	19.82	33.66	942.08	53.48
RKD10sys	21.11	32.06	896.27	53.17

multiple randomized kd-trees can increase the successful rate of finding the nearest neighbor in high dimensional space. The effect of RKD in our image retrieval task is investigated in this subsection by carrying out experiments that compare the performance of approaches using different number of trees.

Five correspondence-based approaches were created, named *BBFsys*, *RKD3sys*, *RKD5sys*, *RKD8sys* and *RKD10sys*. *BBFsys* uses a single non-randomized kd-tree while *RKD3sys*, *RKD5sys*, *RKD8sys* and *RKD10sys* build 3, 5, 8 and 10 randomized trees respectively. The maximal number of bins to examine is 50. In the mismatch detection step, all methods apply standard RANSAC to estimate the fundamental matrix.

The algorithms were executed 50 times. Apart from the accuracy, the speed was also evaluated with four metrics: *kdtime*, *ransactime*, *totaltime* and *#sample*. Table 4 presents the results of the five approaches in terms of accuracy. And the performance on speed is shown in Table 5.

Table 4 shows that approaches with multiple trees perform a little better in the aspect of accuracy. The advantage of using more trees is reflected by the fact that fewer samples are drawn in the RANSAC procedure when more trees are built. That is, due to the higher successful rate of searching the nearest neighbor in the SIFT feature space, algorithms using more trees can generate a better set of tentative matches with a larger fraction of inliers. Though more time is needed in the tentative correspondences generation step when applying more kd-trees, the whole process of comparing the query image with the target image is faster. Besides, as more trees are created, the advantage in speed becomes increasingly

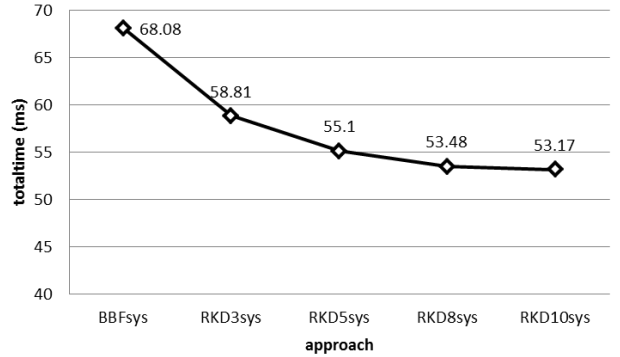


Fig. 3. The speed performance of five approaches, measured in *totaltime*.

TABLE 6  
THE PERFORMANCE OF THE TWO APPROACHES THAT APPLY STANDARD RANSAC AND R-RANSAC RESPECTIVELY AS THE ROBUST ESTIMATOR

	R(25)	R(50)	R(75)	R(100)	ransactime(ms)
RANSACsys	0.6918	0.7278	0.7569	0.7860	37.57
R-RANSACsys	0.6771	0.7169	0.7563	0.7782	31.02

smaller, which is displayed in Fig. 3.

#### 5.4 Experiment #4

The properties of the RANSAC variations are fully utilized to speed up the image retrieval procedure. Two different strategies are proposed.

One strategy is to use the R-RANSAC instead of the standard RANSAC to estimate the geometric relation. The fraction of wrong matches in the set of tentative correspondences is usually high, particularly in the set created between two irrelevant images. Many hypotheses are believed to be incorrect models generated from contaminated samples. By rejecting wrong models through the  $T_{d,d}$  test, R-RANSAC is expected to achieve a speedup. This was verified by the experiment.

Each of the two approaches (*RANSACsys* and *R-RANSACsys*) that respectively use standard RANSAC and R-RANSAC with  $T_{1,1}$  test ran 50 times. The average result is shown in Table 6. At the cost of a small decrease in accuracy, *R-RANSACsys* saves about 17.43% of the time that *RANSACsys* spends in the mismatch detection step.

Another strategy is based on the fact that the hypothesize-and-verify iteration usually draws fewer samples on two relevant images than on two irrelevant ones because the fraction of true matches between two relevant images is much higher. By limiting the number of samples drawn, the large amount of time spent on irrelevant images can be saved. Making use of the early termination property of LO-RANSAC, PROSAC or LO-PROSAC on relevant images, the number of samples can be further constrained.

The approaches, *SRANSACsys*, *LORANSACsys*, *PROSACsys* and *LOPROSACsys*, which respectively apply standard RANSAC, LO-RANSAC, PROSAC and LO-

TABLE 7

THE PERFORMANCE OF FOUR APPROACHES THAT APPLY STANDARD RANSAC, LO-RANSAC, PROSAC AND LO-PROSAC RESPECTIVELY AS THE ROBUST ESTIMATOR, WITH NO LIMITATION ON THE NUMBER OF SAMPLES DRAWN

	R(25)	R(50)	R(75)	R(100)	ransactime(ms)	#sample
SRANSAC <sub>sys</sub>	0.6918	0.7278	0.7569	0.7860	37.57	1053.38
LORANSAC <sub>sys</sub>	0.6889	0.7245	0.7504	0.7812	35.71	1038.57
PROSAC <sub>sys</sub>	0.6824	0.7254	0.7516	0.7769	35.67	1074.70
LOPROSAC <sub>sys</sub>	0.6822	0.7236	0.7515	0.7750	35.18	1058.88

TABLE 8

THE PERFORMANCE OF FOUR APPROACHES THAT APPLY STANDARD RANSAC, LO-RANSAC, PROSAC AND LO-PROSAC RESPECTIVELY AS THE ROBUST ESTIMATOR, WITH THE MAXIMAL NUMBER OF SAMPLES ALLOWED BEING 500

	R(25)	R(50)	R(75)	R(100)	ransactime(ms)	#sample
SRANSAC <sub>sys</sub>	0.6765	0.7115	0.7547	0.7845	10.56	276.20
LORANSAC <sub>sys</sub>	0.6757	0.7120	0.7499	0.7837	10.57	274.65
PROSAC <sub>sys</sub>	0.6683	0.7193	0.7545	0.7776	10.39	287.00
LOPROSAC <sub>sys</sub>	0.6748	0.7187	0.7566	0.7782	10.54	285.57

TABLE 9

THE PERFORMANCE OF FOUR APPROACHES THAT APPLY STANDARD RANSAC, LO-RANSAC, PROSAC AND LO-PROSAC RESPECTIVELY AS THE ROBUST ESTIMATOR, WITH THE MAXIMAL NUMBER OF SAMPLES ALLOWED BEING 200

	R(25)	R(50)	R(75)	R(100)	ransactime(ms)	#sample
SRANSAC <sub>sys</sub>	0.6586	0.7076	0.7556	0.7797	6.08	141.63
LORANSAC <sub>sys</sub>	0.6620	0.7011	0.7492	0.7755	6.23	141.32
PROSAC <sub>sys</sub>	0.6641	0.7166	0.7478	0.7714	5.99	147.64
LOPROSAC <sub>sys</sub>	0.6672	0.7226	0.7512	0.7744	6.19	147.12

TABLE 10

THE PERFORMANCE OF FOUR APPROACHES THAT APPLY STANDARD RANSAC, LO-RANSAC, PROSAC AND LO-PROSAC RESPECTIVELY AS THE ROBUST ESTIMATOR, WITH THE MAXIMAL NUMBER OF SAMPLES ALLOWED BEING 100

	R(25)	R(50)	R(75)	R(100)	ransactime(ms)	#sample
SRANSAC <sub>sys</sub>	0.6447	0.7053	0.7492	0.7740	4.07	80.24
LORANSAC <sub>sys</sub>	0.6522	0.7032	0.7494	0.7728	4.23	80.07
PROSAC <sub>sys</sub>	0.6625	0.7145	0.7439	0.7667	3.96	82.98
LOPROSAC <sub>sys</sub>	0.6639	0.7162	0.7448	0.7675	4.16	82.74

TABLE 11

THE PERFORMANCE OF FOUR APPROACHES THAT APPLY STANDARD RANSAC, LO-RANSAC, PROSAC AND LO-PROSAC RESPECTIVELY AS THE ROBUST ESTIMATOR, WITH THE MAXIMAL NUMBER OF SAMPLES ALLOWED BEING 50

	R(25)	R(50)	R(75)	R(100)	ransactime(ms)	#sample
SRANSAC <sub>sys</sub>	0.6375	0.7091	0.7441	0.7650	2.84	43.37
LORANSAC <sub>sys</sub>	0.6407	0.7042	0.7452	0.7668	3.01	43.34
PROSAC <sub>sys</sub>	0.6617	0.7101	0.7413	0.7603	2.73	44.16
LOPROSAC <sub>sys</sub>	0.6640	0.7131	0.7434	0.7626	2.92	44.15

TABLE 12

THE PERFORMANCE OF FOUR APPROACHES THAT APPLY STANDARD RANSAC, LO-RANSAC, PROSAC AND LO-PROSAC RESPECTIVELY AS THE ROBUST ESTIMATOR, WITH THE MAXIMAL NUMBER OF SAMPLES ALLOWED BEING 20

	R(25)	R(50)	R(75)	R(100)	ransactime(ms)	#sample
SRANSAC <sub>sys</sub>	0.6276	0.6981	0.7279	0.7541	2.03	18.49
LORANSAC <sub>sys</sub>	0.6368	0.7039	0.7351	0.7592	2.15	18.48
PROSAC <sub>sys</sub>	0.6463	0.6985	0.7302	0.7476	1.93	18.48
LOPROSAC <sub>sys</sub>	0.6585	0.7040	0.7356	0.7526	2.08	18.47

TABLE 13

THE PERFORMANCE OF FOUR APPROACHES THAT APPLY STANDARD RANSAC, LO-RANSAC, PROSAC AND LO-PROSAC RESPECTIVELY AS THE ROBUST ESTIMATOR, WITH THE MAXIMAL NUMBER OF SAMPLES ALLOWED BEING 10

	R(25)	R(50)	R(75)	R(100)	ransactime(ms)	#sample
SRANSAC <sub>sys</sub>	0.6205	0.6843	0.7218	0.7418	1.73	9.44
LORANSAC <sub>sys</sub>	0.6246	0.6896	0.7233	0.7453	1.81	9.44
PROSAC <sub>sys</sub>	0.6335	0.6866	0.7171	0.7410	1.64	9.39
LOPROSAC <sub>sys</sub>	0.6455	0.6950	0.7250	0.7455	1.76	9.39

be infinity, 500, 200, 100, 50, 20 and 10.

Table 7 shows that when no constraint is imposed on the number of samples, SRANSAC<sub>sys</sub> performs best with a tiny advantage in accuracy and the average number of samples generated per image pair is over 1000. By decreasing the number of samples allowed from infinity to 10, a plunge in *ransactime* is observed, e.g., from 37.57ms to 1.73ms for SRANSAC<sub>sys</sub>, which is displayed in Fig. 4. As expected, the accuracy of SRANSAC<sub>sys</sub> undergoes a more significant decline than that of LORANSAC<sub>sys</sub>, PROSAC<sub>sys</sub> and LOPROSAC<sub>sys</sub>. Fig. 5 shows the decrease in measure *R(25)*. LOPROSAC<sub>sys</sub> that uses LO-PROSAC has the smallest decrease in accuracy due to its excellent property of early termination which enables a large number of inliers being found within a small number of iterations on relevant images. But even for LOPROSAC<sub>sys</sub>, a big drop is observed when the number of samples is less than 50. Hence, a decent setting is applying LO-PROSAC as the estimator and limiting the maxi-

PROSAC, were evaluated. Fundamental matrix is used as the geometric model and RDK with 5 kd-trees for nearest neighbor search. The 7 tables, from Table 7 to Table 13, present the average result of 50 runs, with the maximal number of samples allowed being limited respectively to

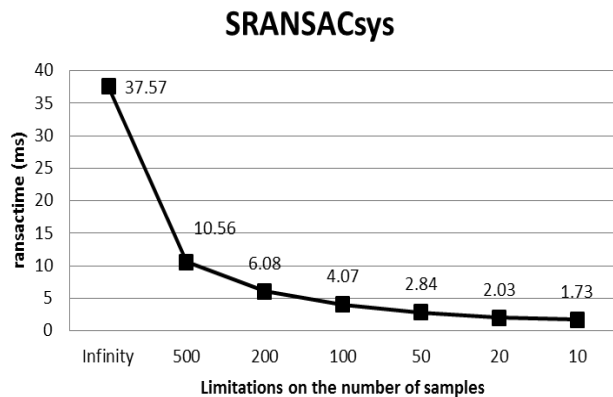


Fig. 4. The plunge in *ransactime* as the number of samples allowed decreases from infinity to 10.

mal number of samples to be 50, which gives the result of 66.40%, 71.31%, 74.34% and 76.26% on measures  $R(25)$ ,  $R(50)$ ,  $R(75)$  and  $R(100)$  at the expense of about 2.92s in the mismatch detection step for a query on a 1000-image dataset. And taking the time of creating tentative matches into consideration, the average time for each query is about 20.45s (17.53s + 2.92s), which is really promising.

## 6 CONCLUSION

This work explores the problem of object-based image retrieval using local feature correspondences. Fundamental matrix is more appropriate than homography and affinity to be the two-view geometric relation, since it works well for both 2D and 3D scenes. Multiple kd-trees are preferred in the tentative correspondences generation step for the reason that using RKD not only gives higher searching accuracy but also speeds up the hypothesize-and-verify iteration in RANSAC-like algorithms by providing better tentative match set. R-RANSAC is more efficient than the standard RANSAC as the principle of R-RANSAC coincides with the nature of the image retrieval task, i.e., most images in the dataset are irrelevant to the query image and a large number of hypotheses are generated from samples contaminated by incorrect matches. Limiting the number of samples drawn in the hypothesize-and-verify procedure tremendously speeds up the RANSAC-like algorithms while the searching accuracy is minimally impacted by making the best of the early termination property of LO-PROSAC on relevant images.

In this work, the techniques developed originally for matching task are successfully employed to solve the image retrieval problem. Particularly, the distinctive properties of several RANSAC variations are deeply investigated and utilized to speed up the image retrieval procedure.

Future work will focus on the tentative correspondences generation step as the experimental results show that it costs much more time than the mismatch detection process and we believe it becomes the bottleneck both in speed and accuracy.

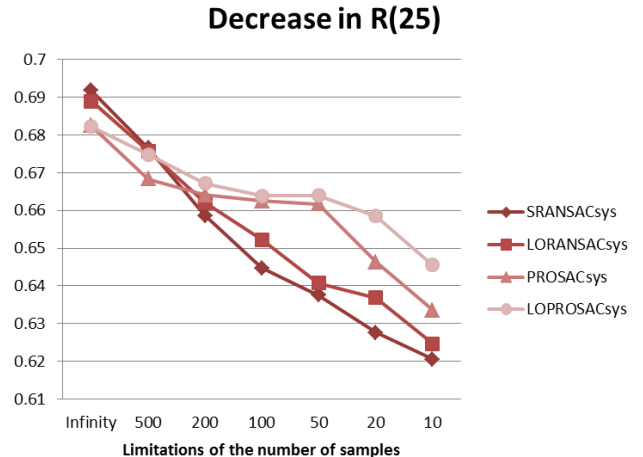


Fig. 5. The decrease in measure  $R(25)$  as the number of samples allowed decreases from infinity to 10.

## REFERENCES

- [1] A.W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-Based Image Retrieval at the End of the Early Years," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349-1380, Dec. 2000.
- [2] R. Datta, D. Joshi, J. Li, and J.Z. Wang, "Image Retrieval: Ideas, Influences, and Trends of the New Age," *ACM Comput. Surv.*, vol. 2, no. 2, pp. 1-60, May 2008.
- [3] M.S. Lew, N. Sebe, C. Djeraba and R. Jain, "Content-based Multimedia Information Retrieval: State of the Art and Challenges," *ACM Trans. Multimedia Computing, Communication, and Application*, vol. 2, no. 1, pp. 1-19, Feb. 2006.
- [4] T. Deselaers, D. Keysers, and H. Ney, "Features for Image Retrieval: An Experimental Comparison," *Information Retrieval*, vol. 11, no. 2, pp. 77-107, Apr. 2008.
- [5] X.S. Zhou and T.S. Huang, "Relevance Feedback in Image Retrieval: A Comprehensive Review," *Multimedia Systems*, vol. 8, no. 6, pp. 536-544, Apr. 2003.
- [6] H. Müller, W. Müller, D. M. Squire, S. Marchand-Maillet and T. Pun, "Performance Evaluation in Content-based Image Retrieval: Overview and Proposals," *Pattern Recognition Letters*, vol. 22, no. 5, pp. 593-601, Apr. 2001.
- [7] L. Torresani, V. Kolmogorov and C. Rother, "Feature Correspondence via Graph Matching: Models and Global Optimization," *Proc. Tenth European Conf. Computer Vision (ECCV '08)*, pp. 596-609, 2008.
- [8] Z. Zhang, Z. Li and M. Drew, "Feature Correspondence with Constrained Global Spatial Structures", *IEEE Int'l Conf. Image Processing*, pp. 177-180, 2009.
- [9] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. ACM*, vol. 24, no. 6, pp. 381-395, Jun. 1981.
- [10] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.
- [11] J.S. Beis and D.G. Lowe, "Shaping Indexing Using Approximate Nearest-Neighbor Search in High-Dimensional Spaces," *IEEE*

- Conf. Computer Vision and Pattern Recognition*, pp. 1000-1006, 1997.
- [12] C. Silpa-Anan and R. Hartley, "Optimized KD-trees for Fast Image Descriptor Matching," *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1-8, 2008.
- [13] M. Muja and D.G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration," *Int'l Conf. Computer Vision Theory and Applications*, pp. 331-340, 2009.
- [14] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, second ed. Cambridge Univ., 2003.
- [15] O. Chum, J. Matas, and J. Kittler, "Locally Optimized RANSAC," *Proc. Ann. Symp. German Assoc. for Pattern Recognition (DAGM '03)*, pp. 236-243, 2003.
- [16] R. Hartley, "Projective Reconstruction and Invariants from Multiple Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 10, pp. 1036-1041, Oct. 1994.
- [17] O. Chum and J. Matas, "Matching with PROSAC - Progressive Sample Consensus," *IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 220-226, Jun. 2005.
- [18] R. Hartley, "In Defence of the 8-point Algorithm," *Proc. Fifth Int'l Conf. Computer Vision (ICCV '95)*, pp. 1064-1070, Jun. 1995.
- [19] O. Chum and J. Matas, "Randomized RANSAC with  $T_{d,d}$  Test," *Proc. British Machine Vision Conf. (BMVC '02)*, vol. 2, pp. 448-457, 2002.
- [20] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, Jul. 2002.
- [21] T. Lindeberg, "Scale-space Theory: A Basic Tool for Analysing Structures at Different Scales," *J. Applied Statistics*, vol. 21, no. 2, pp. 224-270, 1994.
- [22] J.H. Friedman, J.L. Bentley, and R.A. Finkel, "An Algorithm for Finding Best Matches in Logarithmic Expected Time," *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 209-226, Sep. 1977.