International Journal of
Intelligent Systems and
Applications in Engineering

ISSN:2147-6799 www.atscience.org/IJISAE

Advanced Technology and Science

Original Research Paper

AIR: An Agent for Robust Image Matching and Retrieval

Jimmy Addison Lee*[a], Attila Szabó [a], Yiqun Li [a]

Received 1st April 2013, Accepted 11th June 2013

Abstract: *This paper presents a novel scheme coined AIR (Agent for Image Recognition), acting as an agent, to oversee the image matching and retrieval processes. Firstly, neighboring keypoints within close spatial proximity are examined and used to hypothesize true keypoint matches. While this approach is robust to noise (e.g. a tree) since spatial relation is considered, missing (undetected) keypoints in one image can also be recovered resulting in more keypoint matches. Secondly, the agent is able to recognize instability of projective transformations in certain cases (e.g. non-planar scenes). The geometric approach is substituted with LIS (Longest Increasing Subsequence) approach which does not require any complex geometric transformations. The effectiveness of AIR is substantiated by an image retrieval experiment which demonstrates that it achieves a twofold increase in true matches and higher matching accuracy when compared to RANSAC homography approach.*

Keywords: *Image recognition, Image matching, Image retrieval, Spatial relation approach, longest increasing subsequence*

1. Introduction

The brute force method of comparing every pixel in two images is computationally prohibitive. Intuitively, one can relate the two images by matching only regions in the images that are in some way interesting. They are local as they are related to small regions on objects instead of the whole object itself. This property makes them distinctive as well as robust to occlusion and clutter. These regions are referred to as local features, and sometimes known as interest points or keypoints. Today, the use of keypoints to find correspondences across multiple images is a key step in many image processing and computer vision applications. Some of the most notable examples are panorama stitching [1-3], wide baseline matching [4-7], image retrieval [8, 9], object recognition [10-12], and object class recognition [13-15]. Differences between the images can be a substantial range of affine distortion, noise level, change in illumination, scaling, rotation, and viewpoint. The keypoints should be invariant to these differences in order to robustly match two images of the same object or scene. A good keypoint should be highly distinctive in the way that a single keypoint can be correctly matched with high probability against a large database of keypoints from many images. Nevertheless, the more invariant it is, the less distinctive it will be, which the trade-off between invariance and distinctiveness is. Typically, there are one or more follow-up verification steps to verify the keypoint matches. Without prior knowledge what types of images we are receiving, we often assume that the scenes are composed entirely of planes; and that all planes can be detected whereby planar homographies can be derived. When this assumption is invalid, the matching fails.

In this paper, we propose an intelligent scheme coined AIR (Agent for Image Recognition), acting as an agent, to hypothesize true keypoint matches, or in fact overseeing the keypoint matching process. Neighboring keypoints within close spatial proximity are examined and used to hypothesize true keypoint matches. The fundamental idea behind this approach is that if two keypoints are true corresponding keypoints in the two images, at least some of their neighboring keypoints should be corresponded. By building a relationship between each keypoint and its neighboring keypoints, our approach can robustly deal with two common problems.

1) Asymmetric numbers of keypoints detected in the two images, since a keypoint detected in one image may not appear in the other image and therefore results in a lesser number of keypoint matches. These missing (undetected) keypoints can never be recovered.

2) False corresponding keypoints found in the two images after projectivity due to noise, e.g., a tree in one of the images will comprise a massive number of keypoints which can be easily mismatched after projectivity.

AIR is also able to recognize instability of the approach in some cases (e.g. non-planar scenes) after projectivity from the low number of keypoint matches. It substitutes with LIS (Longest Increasing Subsequence) approach which allows less rigid correspondence between the matched image pairs. This approach finds a subsequence (of keypoints in the first image) of a sorted sequence (of corresponding keypoints in the second image), in which the subsequence elements are in sorted order and is as long as possible. The subsequence is not necessarily contiguous, or unique. The concept is that an image pair is geometrically consistent if the geometric order of their corresponding keypoints is consistent. The rest of the paper is organized as follows. Section 2 discusses related work. In Section 3, our image recognition methodology is presented whereby AIR is described. Section 4 provides the experimental results. Section 5 concludes the paper.

2. Related Work

[a] *Institute for Infocomm Research, Singapore*
* *Corresponding Author: Email: jalee@i2r.a-star.edu.sg,*

Mikolajczyk and Schmid [16] evaluated a variety of object recognition algorithms and identified that the SIFT [12] (Scale-Invariant Feature Transform) and SIFT-based algorithms such as SURF [17] (Speeded Up Robust Features) are the most resistant to common image deformations and have achieved the best performance. SIFT-based features are invariant to image scale, translation, rotation, and partially invariant to illumination and viewpoint changes. Details on application of these features can be found in [3, 18, 19]. In its original matching scheme, a pair of keypoints is considered a match if the distance ratio between the closest match and the second closest match is below a certain threshold. While the distance ratio can eliminate some of the false keypoint matches, we often still need to identify correct subsets of keypoints containing less than 1% inliers.

To solve the outlier problem, the RANSAC [20] (Random Sample Consensus) algorithm and other similar hypothesize-and-verify methods have been proposed in the literature. The RANSAC algorithm is a robust method based on random sampling and rejects all keypoint matches not conforming to the found homography model [21] or epipolar geometry [21]. Although this method works fine in many applications, they perform poorly when the number of false keypoint matches outnumbers the number of true keypoint matches; or when the number of keypoint matches is modest (limited). The RANSAC idea was modified by Nister and Stewenius [22] to include competitive verification of models. The algorithm named "Preemptive RANSAC" was demonstrated to perform well in a real-time structure-from-motion system. The limitation is that only a fixed number of models are evaluated, which is equivalent to a priori assumption that a lower bound on the fraction of inliers is known. This limits the applicability of preemptive RANSAC in wide baseline stereo where the fraction of inliers varies widely.

Another popular method is the Hough Transform [23-25], which clusters keypoints in pose space. The Hough Transform identifies clusters of keypoints with a consistent interpretation by using each keypoint to vote for all object poses that are consistent with the keypoint. However, in [12], it was shown that follow-ups are required after Hough Transform is performed in order to eliminate more false keypoint matches, e.g., least-squares pose determination, followed by a probabilistic model given in [18]. Moreover, Hough Transform requires a huge computation load in pixel transformation and a large storage (or memory) is also required for the voted Hough space. Without proper parallelization, it will be very difficult for Hough Transform to achieve real-time performance.

3. Image Recognition Methodology

This section describes our image recognition technique used to identify objects in different images. Keypoints between two images of the same scene or object must be robustly detected, described, matched and verified. We exploit Fast-Hessian detector and SURF descriptor proposed by Bay et al. [17] due to its speed and accuracy. We find the best match between a query image and the database images by Euclidean distance, using the k-d data structure and search algorithm [25]. The algorithm generalizes classical binary trees to higher dimensional spaces so that one can locate nearest neighbors to a descriptor vector in O (log N) time instead of the brute-force O (N) time, with N being the size of the images in the image database.

The Agent (AIR) as illustrated in Fig. 1 inspects the keypoint matches based on spatial relations. Once the keypoints passed the inspection, the agent examines the reliability of the matched image. If the match is not satisfactory (unreliable), it will automatically switch to the LIS approach. The two tasks will be discussed in details in the following sections.

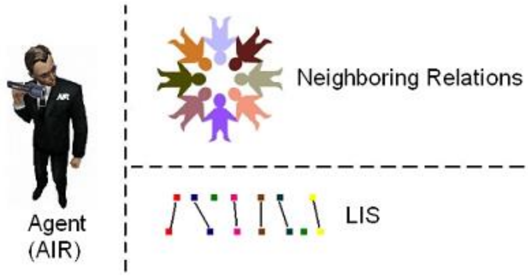


Figure 1. Agent (AIR) is responsible for two main tasks

3.1. Spatial Relations

For each image, we save each detected keypoint and its nearest 12 neighboring points. Let us consider a 3D coordinate frame and two planar surfaces of the same scene but with different camera angles. Let's call the two planar surfaces R1 and R2 as shown in Fig. 2. R2 is defined by the point $\overrightarrow{b_0}$ and two linearly independent vectors $\overrightarrow{b_1}$ and $\overrightarrow{b_2}$ contained in the region. Let us consider a keypoint $\overrightarrow{P_2}$ in R2. Since the vectors $\overrightarrow{b_1}$ and $\overrightarrow{b_2}$ form a basis in R2, we can express $\overrightarrow{P_2}$ as

$$q1\overrightarrow{b_1} + q2\overrightarrow{b_2} + \overrightarrow{b_0} = \left(\overrightarrow{b_1}, \overrightarrow{b_2}, \overrightarrow{b_0}\right)\begin{pmatrix} q1 \\ q2 \\ 1 \end{pmatrix} = B\vec{q}, \tag{1}$$

where $B = \left(\overrightarrow{\mathrm{b}_1}, \overrightarrow{\mathrm{b}_2}, \overrightarrow{\mathrm{b}_0}\right) \in \Re^{3\times3}$ defines the planar surface R2, and $\vec{\mathrm{q}} = (q1, q2, 1)^T$ defines the 2D coordinates of $\overrightarrow{P_2}$ with respect to the basis $\left(\overrightarrow{b_1}, \overrightarrow{b_2}\right)$. We can compute a similar identity for planar surface R1 as

$$\overrightarrow{P_1} = A\vec{s}, \tag{2}$$

where $A = (\overrightarrow{\mathrm{a}_1}, \overrightarrow{\mathrm{a}_2}, \overrightarrow{\mathrm{a}_0}) \in \Re^{3\times3}$ defines *R1*, and $\vec{s} = (s1, s2, 1)^T$ defines the 2D coordinates of $\overrightarrow{P_1}$ with respect to the basis $(\overrightarrow{\mathrm{a}_1}, \overrightarrow{\mathrm{a}_2})$ We impose the constraint that point $\overrightarrow{P_1}$ maps to point $\overrightarrow{P_2}$ under perspective projection centered at the origin:

$$\overrightarrow{\mathrm{P}_1} = \propto (\vec{\mathrm{q}})\overrightarrow{\mathrm{P}_2}\,, \tag{3}$$

where $\propto (\vec{\mathrm{q}})$ is a scalar that depends on $\overrightarrow{\mathrm{P}_2}$, and consequently on $\vec{\mathrm{q}}$. By combining the equation above with the constraint that $\overrightarrow{\mathrm{P}_1}$, and $\overrightarrow{\mathrm{P}_2}$, must be situated in its corresponding planar region, we obtain the relationship between the 2D coordinates of these points:

$$\vec{\mathrm{s}} = \propto (\vec{\mathrm{q}})\mathrm{A}^{-1}\mathrm{B}\vec{\mathrm{q}}\,, \tag{4}$$

where the role of $\propto (\vec{\mathrm{q}})$ is to simply scale the term $\propto (\vec{\mathrm{q}})\mathrm{A}^{-1}\mathrm{B}\vec{\mathrm{q}}$ such that its third coordinate is 1. We can represent $\propto (\vec{\mathrm{q}})\mathrm{A}^{-1}$ B as a homography matrix Hm and compute the above equation as

$$\vec{\mathrm{s}} = \mathrm{H_m}\vec{\mathrm{q}}\,, \tag{5}$$

If R1 and R2 are true corresponding planars, the keypoint $\overrightarrow{\mathrm{P}_2}$ and its 6 nearest neighboring points (shaded in different colors in Fig. 2) in R2 should fit the homography matrix $\mathrm{H_m}$ to correctly locate the 7 corresponding keypoints in R1. There are more than 6 nearest neighbors (total of 12) stored in a descriptor vector although only 6 are used.

This is to solve the asymmetric problem where a point can be detected in one quadrilateral region but not in the other, and thus the nearest 6 neighbors may be slightly different in this case. E.g., the *6th* nearest neighbor for $\overrightarrow{\mathrm{P}_2}$ in R2 may be the *7th* nearest neighbor for $\overrightarrow{\mathrm{P}_1}$ in R_1. Therefore, we stored slightly more than 6 nearest neighbors to overcome this problem.

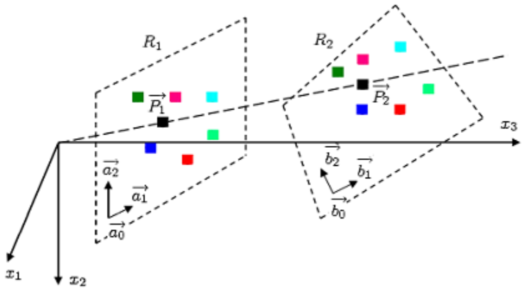


Figure 2. Point projectivity between two planar surfaces R_1 and R_2 of the same scene but different camera angles

Observations after various experiments summarized the point-to-point homography matching results into different cases. Given two planar surfaces (may be true or false corresponding pair) R_3. and R_4., a keypoint (let's call it $\overrightarrow{P_3}$), its 6 nearest neighboring points $\{\overrightarrow{P_{3a}}, \overrightarrow{P_{3b}}, \ldots, \overrightarrow{P_{3f}}\}$, and their corresponding points in R_4 (may be true or false corresponding points) denoted as $\overrightarrow{P_4}$), $\{\overrightarrow{P_{4a}}, \overrightarrow{P_{4b}}, \ldots, \overrightarrow{P_{4f}}\}$, the 7 frequent cases after projectivity of the 7 points by H_m from R_3 to R_4 can be categorized as:

All 7 points $\overrightarrow{P_4}$, $\overrightarrow{P_{4a}}$ to $\overrightarrow{P_{4f}}$ found.
$\overrightarrow{P_4}$, found, and most neighboring points (≥ 4) found.
$\overrightarrow{P_4}$, not found, but most neighboring points (≥ 5) found.
$\overrightarrow{P_4}$, found, and few neighboring points (< 4) found.
$\overrightarrow{P_4}$, not found, and few neighboring points (< 5) found.
$\overrightarrow{P_4}$, found, but no neighboring points (< 1) found.
All 7 points $\overrightarrow{P_4}$, $\overrightarrow{P_{4a}}$ to $\overrightarrow{P_{4f}}$ not found.

We determine from experiments that candidate points falling within cases 1-3 give very high accuracy of correct matches. As a result, we only consider a candidate point as valid if it falls within these three cases. This approach is very robust since it does not evaluate the pair of candidate points in the two planar surfaces individually, but as a whole with the neighbors in order to adjudicate if the pair is a correct match. Consequently, keypoints coming from other objects or structures, e.g., the dense keypoints detected on a tree, will have very low chances of being matched. We are also the first to recover keypoints that are not detected in one image but in the other. This can be easily understood from case 3 where $\overrightarrow{P_4}$, is not found but most of the neighboring points (at least 5) are found. We can recover such missing (undetected) keypoint in an image if the point exists in the other image and most of their neighboring points are corresponded. This solves the common problem in keypoint detection where the numbers of detected keypoints in two images are asymmetric. Given that the numbers of keypoints are asymmetric in the pair of planar surfaces, in this case R_3 and R_4, we apply an inverse homography matrix H_m^{-1} to the unmatched keypoints in R_4 to locate any corresponding keypoints in R_3 in the same manner as before. Eventually, the best pair of corresponding planar surfaces in the two images will obtain the largest list of corresponding keypoints.

However, homography based method only works well on images comprising planar rectangular structures or quadrilateral regions as described above. When it comes to non-planar scenes, the approach is unstable. AIR recognizes this failure by examining the number of keypoint matches, e.g. low number of matches indicates unreliability of matched image. Alternate approach (LIS) is substituted, which will be discussed next.

3.2. Longest Increasing Subsequence (LIS)

In this method, we find true keypoint matches by imposing a geometrical constraint on the matches. This constraint applies to a set of matched keypoint-pairs. Our assumption is that true keypoint matches are the elements of the largest subset of corresponding keypoints which are consistent. The set of matched keypoint-pairs are consistent if the keypoints order is the same in both images. We can find the largest subset by calculating the longest increasing subsequence [26]. The pseudo-code for the algorithm is provided in Algorithm 1. First we sort the keypoints according to their x coordinate in the first image. Then we create a sequence from the x coordinates of the keypoints in the second image (keeping the order after sorting). The longest increasing subsequence of this sequence will give us the indexes of the keypoints we want. They will be geometrically consistent and they will be the largest subset of such. Fig. 3 illustrates this.

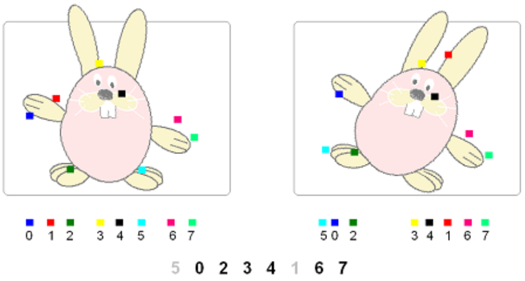


Figure 3. An illustration of Longest Increasing Subsequence (LIS) on two images with detected keypoints as shown. The LIS output in this case is 0, 2, 3, 4, 6, 7

This geometrical constraint is invariant to translation and scaling. Those transformations do not have effect on the geometrical orders of the keypoints. It also allows a little elasticity. However, it is not invariant to rotation, affine transformation, and homographic transformations. Those can change the relative order of the keypoints. We can address this problem by calculating the LIS on rotated images. We only have to apply the rotation on the keypoint coordinates, so we do not rotate the whole image and extract the keypoints again. Assuming we have found the "right" angles of rotation, we can prove that LIS is invariant to affine transformations.

Algorithm 1 Find LIS in Array A

```
n := A.length
m-idx : = newArray(n)
m-val := newArray(n)
previous : = newArray(n)
maxJength : = 0
for i from 0 to n-1: do
  //Binary search in the m-val array
  idx := lowerJbound(m-val[0 : max-length], A[i])
  mJdx[idx] := i
  mval[idx] : = A[i]
  if idx > 0 then
  previous[i] : = mJdx[idx - 1] end if
  if idx = maxJength then
  maxJength : = maxJength + 1
  end if
end for
  LIS := newArray(maxJength) idx := mJdx[maxJength - 1]
for i from maxJength-i to 0 do
  LIS [i] := idx
  idx := previous[idx]
end for
return LIS
```

The proof of affine invariance is as follows: Let Affine : $\Re^2 \rightarrow \Re^2$ be an affine transformation. A f fine $(\vec{p}) = A\vec{p} + \vec{t}$ where $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is the affine transformation matrix, and

$\vec{t} = \binom{e}{f}$ is the translation vector. Let Rotate : $\Re^2 \to \Re^2$be a rotation. Rotate$(\vec{p}) = R\vec{p}$ where

$$R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

and θ is the angle of the rotation. If we apply an affine transformation on a set of points $P = \{\vec{p_i} \in \Re^2\}$, and also apply the rotation, we will get $P' = \{\vec{p_i'} \in \Re^2\}$ where $\vec{p_i'} =$ Rotate$(A\,f\,fine(\vec{p_i}))$. The detailed equation is as follows:

$$\begin{pmatrix} x_i' \\ y_i' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}\left(\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}\right) \quad (6)$$

where

$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \vec{p_i}$ and $\begin{pmatrix} x_i' \\ y_i' \end{pmatrix} = \vec{p_i'}$.

Our only concern is the difference between the x coordinates of two points (P_0) ⃗ and (P_1) ⃗ (the original points) and the difference between the x coordinates of (P_1^') ⃗ and -(P_0^') ⃗ (the transformed points). From eq. 6 we can derive the following:

$$x_1' - x_0' = (a\cos\theta - c\sin\theta)(x_1 - x_0) + (b\sin\theta - d\cos\theta)(y_1 - y_0)$$

(7)

(7)

We can choose θ (and also r) from the equations b = r sin θ and d = r cos θ, where $\theta, r \in \Re$ and r det(A) > 0 assuming $det(A) \neq 0$. This is always true in practice. The θ will be the "right" angle we should choose so that $x_1' - x_0'$ will not be dependent on the y coordinates y1 and y0. After substitution, eq. 7 can be rewritten as:

$$x_1' - x_0' = \left(\frac{ad-bc}{r}\right)(x_1 - x_0)\,. \quad (8)$$

$As \left(\frac{ad-bc}{r}\right) = \frac{\det(A)}{r} > 0$, we showed that we can preserve the relative order of points by applying a carefully chosen rotation.

In practice we do not know θ, because we do not know the parameters of the affine transformation. But computing LIS is very fast, so we can afford to do it multiple times. We randomly choose several angles for the first and for the second image. We compute the LIS for all possible angle-pairs (KxL times if the numbers of angles are K and L for the images respectively). We keep the true matches from the largest subset we obtained. We create the angles for the first image by the following steps. First we choose a random angle $\theta = rand(0,2\pi)$. Then the set of angles becomes

$\theta_k = \left\{\theta + \frac{k.2\pi}{K}\right\}$,

where k=0,1,….,K-1. We choose the θ_L angles for the second images similarly. K = 3 and L=7 are good choices based on our experiments.

4. Experiment

We validate the AIR approach in an image retrieval experiment. We compare it with three other schemes. Thus, the four schemes in our evaluations are as follows: (1) SURF + RANSAC Homography [21, 27], (2) SURF + LIS standalone, (3) SURF + Spatial Relations standalone, and (4) SURF + AIR. We include SURF + LIS standalone and SURF + Spatial Relations standalone in our evaluation although AIR is comprising both LIS and Spatial Relations. The reason is because we want to determine from the results whether the agent (AIR) is intelligent enough to switch between the two approaches for different images. The results should be improved with AIR.

4.1. Data set and Evaluation measures

We use the Stanford Mobile Visual Search data set proposed in [28] for our evaluation. This data set has several key characteristics that are lacking in existing data sets: rigid objects, widely varying lighting conditions, perspective distortion, typical foreground and background clutter, realistic ground-truth reference data, and query data collected from heterogeneous low and high- end camera phones. The data are in several different categories: CDs, DVDs, books, business cards, text documents, video clips, and museum paintings. Some sample query and database images are shown in Fig. 4. The number of database and query images for different categories is shown in Table 1. There are a total of 2800 query images for 700 distinct classes across 7 image categories used in the evaluation. The original resolution of the images varies for all categories, and we deliberately reduce the size of the images to 320 x 240 to make them more compact for efficient transmission and storage, well-suited for mobile visual search applications. This also makes the evaluation more challenging (dealing with low resolution images).

The evaluation measures are straightforward. We report the percentage of correct images retrieved and the average number of matched keypoints for each category. These measurements are similar to the ones used in [28].

Table 1. Number of query and database images for different categories used in the evaluation.

Category	Database	Query
CDs	100	400
DVDs	100	400
Books	100	400
Video Clips	100	400
Business Cards	100	400
Text Documents	100	400
Paintings	100	400

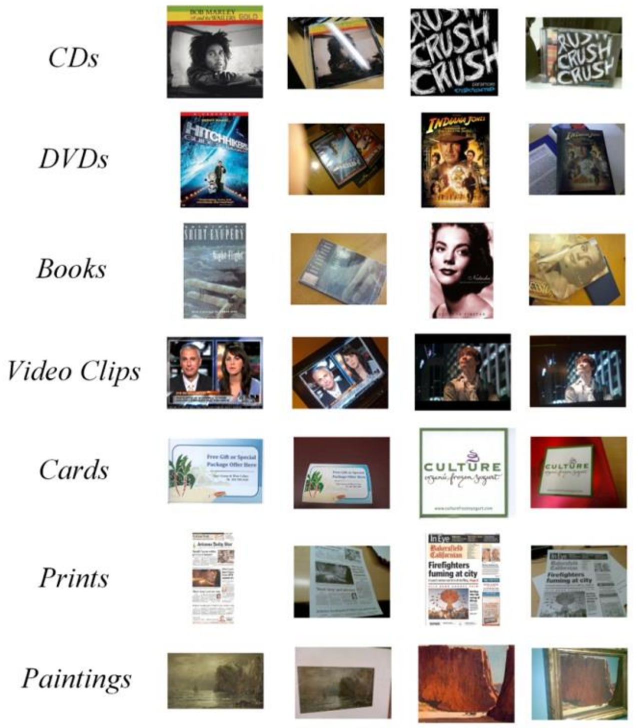
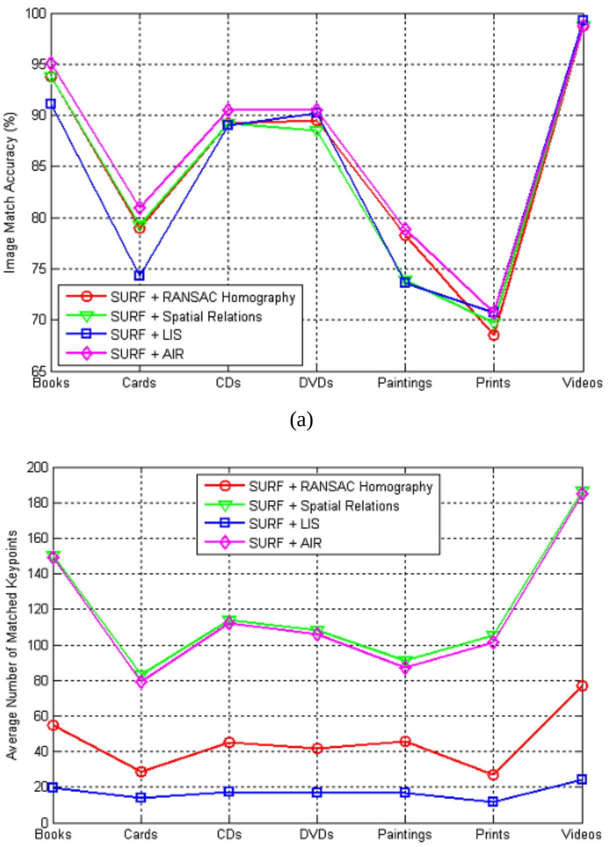


Figure 4. Stanford Mobile Visual Search data set (Chandrasekhar et al., 2011)used for our evaluation. We used a total of 7 categories as shown. The images are captured with a variety of camera-phones, and under widely varying lighting conditions



(a)

(b)

Figure 5. Results of the four approaches for each data set's category (total of 7). (a) shows the image matching accuracy (correct retrieval) in percentage. (b) shows the average number of matched keypoints

4.2. Results

In Fig. 5, we report results for the four schemes as described above. Firstly, we observe that SURF + Spatial Relations standalone and SURF + LIS standalone do not give the best results. However, when they are combined into the AIR approach, it translates into better retrieval results. Out of the 7 categories in the data set, SURF + AIR dominates 6 categories as shown in Fig. 5(a), with 1 category (videos) having the same matching accuracy with SURF + RANSAC homography. LIS achieves highest matching accuracy among all in the video clips.

Secondly, we note that SURF + AIR and SURF + Spatial Relations both give very high average number of matched keypoints as shown in Fig. 5(b). The average number of matched keypoints in each category is about twofold and more of SURF + RANSAC homography. This is predictable as the spatial relations approach can recover missing (undetected) keypoints based on neighboring relations as explained in Section 3.1.

5. Conclusions

In this paper, we have proposed a novel design of an image recognition agent called AIR (Agent for Image Recognition) showing high potential in image matching and image retrieval applications. AIR is able to verify true keypoint matches while recovering missing (undetected) keypoints in one image by exploiting the spatial relations approach as described in Section 3.1. It is more robust to false keypoint matches or noise as it does not only evaluate each pair of candidate keypoints in the two images, but also on each of their neighboring keypoints based on spatial proximity.

AIR is also able to recognize instability of the homography-based approach in certain images, and automatically switches to the LIS (Longest Increasing Subsequence) approach as proposed in Section 3.2. The LIS approach allows less rigid correspondence between the matched image pairs.

We have demonstrated AIR in an image retrieval experiment on the Stanford Mobile Visual Search data set, where the results favored AIR for its increased accuracy and larger number of matched keypoints. It achieved a twofold more matched keypoints when compared to the state- of-the-art approach (SURF + RANSAC homography).

References

[1] Agarwala A, Agrawala M, Cohen M, Salesin D, Szeliski R (2006). Photographing long scenes with multi-viewpoint panoramas. ACM Trans. on Graphics (SIGGRAPH). 25(3): 853-861.

[2] Brown M, Lowe DG (2003). Recognizing panoramas. Proc. 9th IEEE Int'l Conference on Computer Vision (ICCV). 2: 1218-1225.

[3] Brown M, Lowe DG (2007). Automatic panoramic image stitching using invariant features. Int'l Journal of Computer Vision (IJCV). 74(1): 59-73.

[4] Baumberg A (2000). Reliable feature matching across widely separated views. Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR). 1: 774-781.

[5] Goedeme T, Tuytelaars T, Van-Gool L (2004). Fast wide baseline matching for visual navigation. Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR). 1: 24-29.

[6] Kannala J, Brandt SS (2007). Quasi-dense wide baseline matching using match propagation. Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR). 1-8.

[7] Lee JA, Yow KC, Chia YS (2009). Robust matching of building facades under large viewpoint changes. Proc. 12th IEEE Int'l Conference on Computer Vision (ICCV). 1258-1264

[8] Katare A, Mitra SK, Banerjee A (2007). Content based image retrieval system for multi object images using combined features. Proc. IEEE Computer Society Int'l Conference on Computer Vision and Pattern Recognition (CVPR). 595-599

[9] Wang J, Zha H, Cipolla R (2005). Combining interest points and edges for content-based image retrieval. Proc. 12th IEEE Int'l Conference on Image Processing (ICIP). 3: 1256-1259

[10] Belongie S, Malik J, Puzicha J (2002). Shape matching and object recognition using shape contexts. IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI). 23(4): 509-522

[11] Frome A, Huber D, Kolluri R, Billow T, Malik J (2004). Recognizing objects in range data using regional point descriptors. Proc. 8th European Conference on Computer Vision (ECCV). 3: 224-237

[12] Lowe DG (2004). Distinctive image features from scale-invariant keypoints. Int'l Journal of Computer Vision (IJCV). 60(2): 91-110

[13] Leordeanu M, Hebert M, Sukthankar R (2007). Beyond local appearance: Category recognition from pairwise interactions

of simple features. Proc. IEEE Computer Society Int’l Conference on Computer Vision and Pattern Recognition (CVPR). 1-8

[14] Mikolajczyk K, Leibe B, Schiele B (2006). Multiple object class detection with a generative model. Proc. IEEE Computer Society Int’l Conference on Computer Vision and Pattern Recognition (CVPR). 1: 26-36

[15] Mutch J, Lowe DG (2006). Multiclass object recognition with sparse, localized features. Proc. IEEE Computer Society Int’l Conference on Computer Vision and Pattern Recognition (CVPR). 1: 11-18.

[16] Mikolajczyk K, Schmid C (2005). A performance evaluation of local descriptors. IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI). 27(10): 1615-1630

[17] Bay H, Tuytelaars T, Van-Gool L (2006). SURF: Speeded Up Robust Features. Proc. 9th European Conference on Computer Vision (ECCV). 1: 404-417

[18] Lowe DG (2001). Local feature view clustering for 3D object recognition. Proc. IEEE Computer Society Int’l Conference on Computer Vision and Pattern Recognition (CVPR). 1: 682-688

[19] Se S, Lowe DG, Little J (2002). Global localization using distinctive visual features. Proc. IEEE/RSJ Int’l Conference on Intelligent Robots and Systems (IROS). 226-231

[20] Fischler MA., Bolles RC. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Comm. of the ACM. 24 (6). 381-395

[21] Hartley RI, Zisserman A (2000). Multiple View Geometry in Computer Vision. Cambridge University Press UK

[22] Nister D., Stewenius H (2006). Scale recognition with a vocabulary tree. Proc. IEEE Computer Society Int’l Conference on Computer Vision and Pattern Recognition (CVPR). Vol. 2. pp. 2161-2168

[23] Ballard DH (1981). Generalizing the hough transform to detect arbitrary shapes. Pattern Recognition (PR). 13 (2). 111-122

[24] Grimson WEL (1990). Object Recognition by Computer: The Role of Geometric Constraints. The MIT Press Cambridge. 263-284

[25] Hough PVC (1962). Method and means for recognizing complex patterns. U.S. Patent 3069654

[26] Fredman M (1975). On computing the length of longest increasing subsequences. Discrete Mathematics. 11 (1). 29-35

[27] Faugeras O (1993). Three-Dimensional Computer Vision: A Geometric Viewpoint. The MIT Press Cambridge

[28] Chandrasekhar V, Chen DM, Tsai SS, Cheung NM, Chen H, Takacs G, Reznik Y, Vedantham R, Grzeszczuk R, Bach J, Girod B (2011). The stanford mobile visual search data set. Proc. 2nd Annual ACM SIGMM Conference on Multimedia Systems (MMSys). 117-122