Real-Time Image Matching Based on Multiple View Kernel Projection

Quan Wang and Suya You
Integrated Media Systems Center
Computer Science Department, University of Southern California
{quanwang, suyay}@usc.edu

Abstract

This paper proposes a novel matching method for real-time finding the correspondences among different images containing the same object.

The method utilizes an efficient Kernel Projection scheme to descript the image patch around a detected feature point. In order to achieve invariance and tolerance to geometric distortions, it combines a training stage based on generated synthetic views of the object. The two reliable and efficient methods cooperate together, resulting the core part of our novel Multiple View Kernel Projection method (MVKP). Finally, considering the properties and distribution of the described feature vectors, we search for the best correspondence between two sets of features using a Fast Filtering Vector Approximation (FFVA) algorithm, which can be viewed as a fast lower-bound rejection scheme. Extensive experimental results on both synthetic and real data have demonstrated the effectiveness of the proposed approach.

1. Introduction

Image matching is a fundamental task in computer vision, used to align two or more images taken, for example, at different times, from different sensors, or from different aspects. Virtually all the intelligent vision processing and understanding systems require image matching, or closely related operations as intermediate steps. Examples of systems in which image matching is a significant component include automated image registration, object recognition, image database retrieval, 3D scene reconstruction, and vision-based autonomous navigation.

Technically, the image matching process generally consists of three components: (1) **Feature detection** finds stable matching primitives over spatial scale space to achieve scale and pose invariance. Recently, local features have been widely employed due to their distinctiveness and ability to handling complex imaging conditions such as occlusions and cluttered backgrounds [1, 2, 6, 7, 14]. The approach proposed in this paper extracts highly-distinctive local features, i.e. interest points, as basic primitives to represent image information and perform image matching. (2) **Feature description** represents the detected features into a compact, robust and stable structure for image matching. Among the various proposed approaches, Kernel Projection using Walsh-Hadamard kernels has demonstrated better performance in terms of robustness and processing time [4]. Kernel Projection, however, does not naturally have the important property of geometry invariant. Therefore can not handle geometric distortions caused by viewpoint or pose changes. We solved this problem by introducing a novel approach called Multiple View Kernel Projection (MVKP) to represent and describe the detected local features. The unique feature representations are compact and show superior advantages in terms of distinctiveness, robustness to occlusions, and tolerance of geometric distortions. (3) **Optimal matching and search** uses the feature descriptions and additional constrains to locate, index, and recognize the targets and scenes of interest. Local feature based approaches typically produces a large number of features needed to match [1, 2, 7, 13]. Methods that search exhaustively are highly computationally expensive, unsuitable for real-time applications. To resolve this problem, we use an effective approach, Fast Filtering Vector Approximation (FFVA) that can efficiently match a very large high-dimensional database of image features in real-time [13].

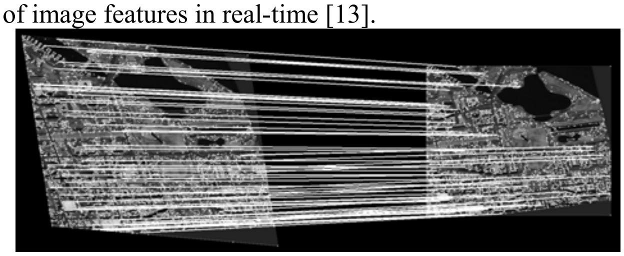


Figure 1(a), synthetic image matching, without consistent check

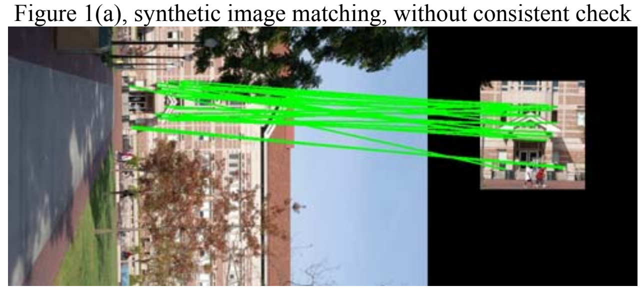


Figure 1(b): real-time real-world image matching result

We integrated all above components to produce a complete image matching system for a range of applications including automated object recognition, non-text image retrieval, and wide-baseline image matching and registration. We have extensively tested the system with both synthetic and real datasets. Comparing with several existing methods, our real-time MVKP system demonstrates both effectiveness and robustness.

2. Related work

Among the image matching approaches based on local features, early works mostly focused on the information provided by one single view of the object. Schmid and Mohr [5] introduced a rotationally invariant descriptor for local image patch based on local greylevel invariants. The ground-breaking work of D.G. Lowe [1] demonstrated that rotation as well as scale invariance can be achieved by first using difference-of-Gaussian function to detect stable interest points, then construct the local region descriptor using assigned orientation and several histograms. The proposed SIFT method produced significant influence on later works. For example, Ke and Sukthankar [6] applied PCA to image gradient patch in order to reduce the descriptor's dimensionality. GLOH [7] is an extension of the SIFT descriptor by computing it for a log-polar location grid with 3 bins in radial direction.

To achieve viewpoint invariant, another line of research is to combine the information of multiple views and train the system in an offline stage so that it will learn the main characters of the same object under different viewing conditions. Consequently, the online matching process can be much faster, even real-time.

Concerning the data source of multiple views, some works use affine transformation to synthesize a number of views from one single input view [2, 8, 9, 10] while others take real images captured from camera as input [11, 12]. Our approach also employed the similar idea by introducing a multiple view training stage to generate a number of synthetic views from single image. We choose the synthesized-view approach due to the ground truth of training images' correspondences it can provide. We use Kernel Projection scheme to extract the significant components containing in the synthesized images and to establish compact feature descriptors.

Lepetit, et al. [2] treat the multiple-view point matching problems as a classification problem. They synthesize small patches (called view set) of each individual feature point served as training input. PCA and k-mean algorithms are applied to those patches to provide the local descriptor. After the offline training stage, the same keypoint detector and PCA projection matrix are used on the query image patches. Eventually, the feature vectors of training and query images are matched by simple linear scan. Later on in their continuous work [8], classification tree is used to replace the PCA and k-mean as well as the final nearest neighbor search. The branching of the trees is decided by simple comparison of nearby intensity values and the final classification is determined by statistic analysis at the leaf node. The online matching process is fast enough for real-time application. However, the forest construction is very slow (10-15 minutes) and it is pointed out that their actual results can vary depending on the viewpoint and illumination conditions [9].

[11] is an extension of the randomized tree (RT) focusing on non-planar object tracking without 3-D model. With the help of RT structure, features can be updated and selected dynamically, called "harvest". The training views are obtained by moving the object slowly in front of the camera. Tuzel, et at. [14] proposed the covariance of d-features as a new descriptor, computed from a set of integral images. A distance metric for the new descriptor is also given. Boffy, et at. [9] use additional information about the appearance of the object under the actual viewing condition to update the classification trees at run-time. They also use special designed spatially distributed trees to enhance the reliability and speed.

Projection and rejection scheme has long been proved to be efficient for pattern matching and general classification problems. Various projection vectors have been studied. Among the previous works, researchers emphasized on the discrimination abilities of the projection kernels [15, 16], while Y. Hel-Or, et al. [4] argued that besides the discrimination, it is also important to choose projection kernels that are "very fast to apply". For this purpose, they choose Walsh-hadamard (WH) kernels and achieved a speed enhancement by almost two orders of magnitude. Further more, experimental results indicate their Projection Kernel method is robust to noise and lighting changes. However, as a fast window matching technique, the method can not handle geometric distortion brought by view angle changes.

3. The Walsh-Hadamard kernels projection

The projection scheme in our MVKP method is based on WH kernels, which is a special case of Gray-Code Kernels [17] and general projection kernels in Euclidean space.

3.1. General projections in Euclidean space

Suppose there are two sets of image patches with size $k \times k$. Each patch can be directly expressed as a k^2-dimensional vector. Therefore the similarity between two patches can be measured as the Euclidean distance between the two corresponding vectors. Obviously, such similarity is impractical to compute especially when the number of patches to be measured is large. The projection strategy is to project the original vectors onto a smaller set of projection kernels, which are fast to compute and still maintain the distance relationship.

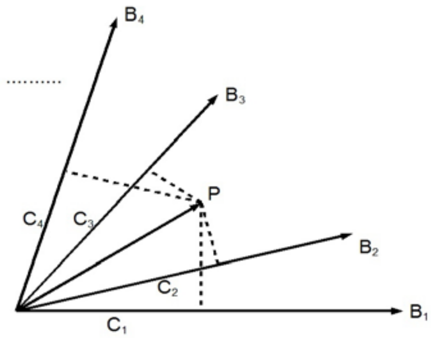


Figure 2: General projection

Assume $\vec{b}_1, \vec{b}_2, \vec{b}_3 \ldots$ are orthonormal projection basises in k^2-dimension Euclidean space (figure 2). P is a point in the k^2-dimension space with projected components $\vec{v}_1, \vec{v}_2, \vec{v}_3 \ldots$ respectively. Scalars $c_i = \vec{v}_i^T \vec{v}_i$. Let d(P) represents the squared Euclidean distance from P to the origin O, then we have:

$$d(P) = \sum_{i=1}^{k^2} c_i^2 \quad (1)$$

It is trivial from the above equation or followed from the *Cauchy-Schwartz inequality* since the Euclidean distance is a norm, that lower bounds of d(P) can be calculated using a number of projection scalars. The lower bounds layers can be expressed as the following:

$$\sum_{i=1}^{1} c_i^2 \le \sum_{i=1}^{2} c_i^2 \le \sum_{i=1}^{3} c_i^2 \le \cdots \le \sum_{i=1}^{k^2} c_i^2 = d(P) \quad (2)$$

With the increasing number of projections kernels involved in the calculation, the lower bound becomes tighter. When all the k^2 kernels are involved, the lower bound becomes the actual squared Euclidean distance. For the projection scheme to be efficient, there are two factors need to be considered: On the one hand, the projection basises should be ordered in a way such that the lower bound can become tight after only a small number of projections. On the other hand, equally important is the requirement that "the kernels should be efficient to apply enabling real-time performance" [4].

3.2. The Walsh-Hadamard kernel

The WH kernel is one special case of the Gray-Code projection kernels satisfying the above two requirements.

First, the WH projection kernels are very efficient to generate and apply. One-dimensional kernels can be generated using binary tree while consecutive kernels are α-related [17]. In the context of 2-D image processing, two-dimensional kernels can be generated as the outer product of one-dimensional kernels. All the coordinates of WH kernel's basis vectors are either +1 or –1. Consequently, projection onto WH kernels involves only dimensionality number of additions or subtractions, which can be performed very fast.

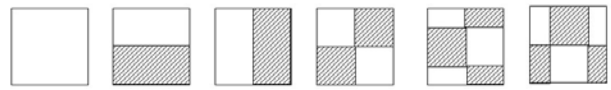


Figure 3: 2-dimensional 4×4 WH kernels in increasing order of frequency. Blank represents value 1 and shadow represents -1.

Second, when the kernels are ordered according to increasing frequency of sign changes, experimental results in section 5 show that a tight lower bound can be achieved using only a small number of kernels. Thus, we can greatly reduce the complexity of similarity computation while still captures the major difference between feature vectors. Figure 3 shows a list of two-dimensional WH kernels in increasing order of frequency. [17] introduced an efficient algorithm to compute the ordering of the kernels, which captures the increase in spatial frequency.

4. MVKP for real-time feature matching

Kernel projection using WH kernels is able to measure the similarity between two large sets of image patterns in real-time, however, it can not handle geometric variance caused by view angle changes. In order to achieve invariance and tolerance to geometric distortions, we combine the WH kernel projection method with a multiple view training stage. The training stage is aimed at providing the system with additional information concerning affine distortions, such that the same object can still be matched under different view angles. Figure 4 illustrates the structure of MVKP. The following sections detail major components.

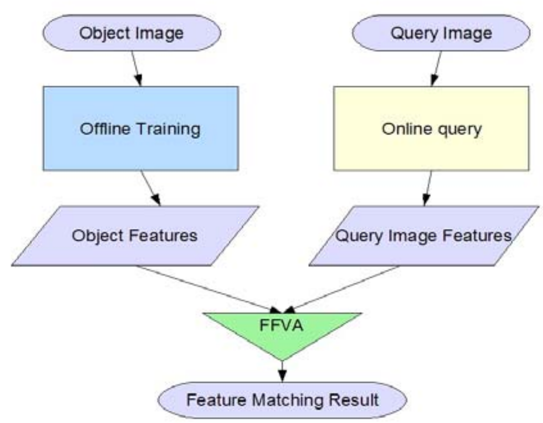


Figure 4: System overview of MVKP

4.1. Offline training stage

During the offline training stage, the MVKP method takes one object image as input, smooth it using Gaussian filter, generate 50-100 synthetic training images from it and then describe the main characters for each selected object location. The output of the training stage is a set of feature vectors, subsets of which corresponds to each selected object location. Figure 5 below illustrates the major components of the training stage.

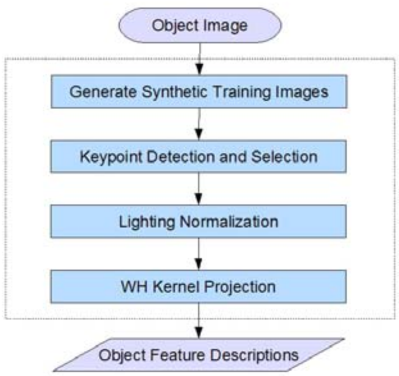


Figure 5: Major components of training stage

The method first synthesizes a number of training views of the input object image using affine transformation. A general affine transformation can be expressed as the following [18]:

$$x' = H_A x = \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix} x \qquad (3)$$

$$A = R(\theta)R(-\phi)DR(\phi) \text{ and } D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \qquad (4)$$

where R is the rotation matrix and t is a translation with components t_1 and t_2. Matrix A corresponds to a rotation of θ first, followed by a rotation of $-\phi$ then scale changes of λ_1 and λ_2 in horizontal and vertical direction respectively. At last, the image is rotated back by ϕ. The six affine transformation parameters are generated randomly to cover the whole parameter space for rotation and shear angles. So we choose the ranges $\theta \in [-\pi, \pi]$, $\phi \in [-\pi/2, \pi/2]$, $\lambda_1, \lambda_2 \in [0.4, 1.6]$, $t_1, t_2 =$ 0, 1, 2, or 3.

Local feature points are identified by searching for local maximum eigenvalues within 3×3 patches. The patches with a local minimum smaller than a threshold are discarded. The detector is designed to guarantee that one feature point will not be too close (for example, 3 pixels) to one another. Otherwise, two features might have a very similar description and consequently fail the *distance ratio criteria* introduced in section 4.3. After all the keypoints in all the synthetic views are detected, we can tell how many of them belong to the same object location in the object image, since all the affine transformations are synthesized. It is assumed that the object locations that appeared more often on the synthetic views have a higher probability to be detected in the query image containing the same object [8]. Therefore, we select 100-200 "mostly common appeared" object locations for future feature matching use. Each object location is represented as a link list containing the coordinates in the corresponding views.

Within each synthetic view, we extract a 32×32 patch around each detected and selected feature point. Because the projection of the image patch onto the first WH kernel conveniently gives its DC value. Robustness to lighting changes can be achieved by simply disregarding the first projection kernel. In addition to that, we normalize (translate and rescale) each patch's intensity values to the same range in order to enhance the performance against different lighting conditions.

The lists of extracted image patches contain the information of various possible appearances for all feature locations. The last step of the training stage is to describe the extracted patches into feature vectors. Each patch's intensity values, forming a very-high-dimension vector, are provided to the kernel projection method so that the final descriptors belongs to the same object location can be more effective, compact and contain the information under various viewing situations. WH kernels are used for the kernel projection. In our experiments, we found that typically the first 20 WH kernels are enough for a reliable feature description. After kernel projection, k-mean can be used to further reduce the size of the feature set. For all the feature vectors representing the same object location, 10-20 clusters are formed and the center vector of each cluster is used to represent the whole cluster.

4.2. Feature set construction for query image

Given a query image containing the same object, our goal is to find the correspondences between the query image and the object image. After the offline training stage, we have lists of object feature vectors. Each of them corresponds to an interest selected object location. Now we need to construct a similar feature set for query image. Figure 6 below shows the major steps of this stage.

After the query image is read and smoothed, the same feature point detector is applied. Because this is an online stage desired to be as fast as possible, we only select a number of "strongest" feature points reported by the detector. Let x_1 be the number of selected stable object locations in the training stage and x_2 is the number of selected feature points in this stage, y is the number of final reported correspondence (*NFRC*), then we have:

$$y \le \min\{x_1, x_2\} \qquad (6)$$

Typically, x_2 is around 500, assume x_1 is 100, then the

NFRC will be no larger than 100.

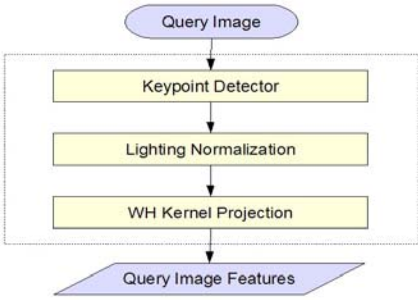


Figure 6: Feature set construction for query image

After the keypoint detection, the intensity values of the image patch around each keypoint give us original vector description. Those original vectors are normalized to the same intensity range to enhance the robustness against lighting changes. The normalized vectors are projected onto a number of (the same number as in the training stage) WH kernels resulting in compact final descriptors.

As the first part of online query process, the feature set construction is comparatively much faster. The really time-consuming part is to find the correspondence between feature sets.

4.3. Establishing feature correspondences

Give the feature descriptors covering various viewing conditions for each object location and the feature descriptors for the query image, the final task is to establish the correct correspondence between two features sets efficiently. The rejection scheme in [4] can't be directly adapted to our problem because it requires all the query image patches be continuously distributed. Thus we use a different technique based on lower bound rejections to accomplish the task.

We employ Euclidean distance as similarity metric due to its simplicity and low computational cost. Diverse Nearest Neighbor (NN) search techniques have been studied under this context. The authors of [2] use linear scan because of its simplicity and accuracy, while in [19], an approximate NN-search method over traditional KD-tree structure is introduced in order to efficiently index the high-dimensional (128-D) feature vectors.

To decide the proper NN-search technique for our MVKP method, first we investigated the feature properties generated by WH kernel projection. The following is an example of three feature vectors generated by kernel projection at the training stage:

Feature vector #1: *22875, 2962, -1843, -935, 1037…*
Feature vector #2: *17886, -2797, 1175, 315, -1008…*
Feature vector #3: *19568, -3567, 1338, 347, 1572 …*

The dimensionality of our feature vector typically ranges from 20 to 100 (depending on how many kernels are used) while the magnitude is comparatively large. It can also be seen from the experiments that, our features vectors are sparser distributed in the space compared with feature vectors in [2], where features are more likely to cluster together. Our feature vectors are more distinctive and further away one from another. Accordingly, we use fast FFVA method [13] to perform the NN-search.

FFVA was proposed for fast indexing and matching high-dimensional features for large databases. It has been proved to be efficient at dealing with large-magnitude, semi-uniform and sparse distributed, high-dimensional feature set with an accuracy close to exact linear scan. Experimental results showed that when the dimension is within 100 and the number of database vectors is within 4000, FFVA demonstrates faster query speed even when compared with tree-based approximated NN-search method. Figure 7 below shows the system overview of FFVA for an efficient nearest neighbor search.

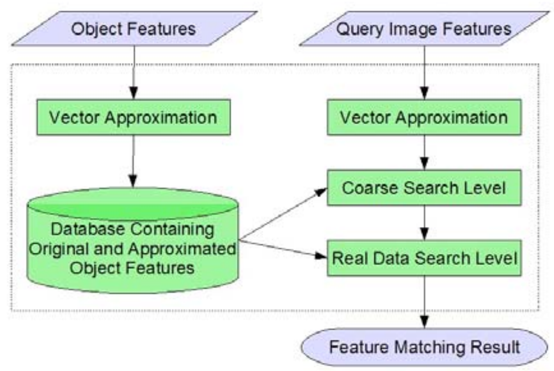


Figure 7: System overview of FFVA

Two major levels are involved in FFVA NN-search: 1) *coarse search level* to sequentially scan the approximations list and eliminate a large portion of data using *block distance* as lower bound, and 2) *real data search level* to calculate accurate distances of residual candidates and decide the final k- nearest neighbors.

Like all the vector approximation based NN-search techniques with static partition length, FFVA works better when the feature vectors are semi-uniform or sparse distributed [13, 20], in which case, the majority of the data will never have the chance to enter the second level. The vector approximation strategy, which provides the compact representations of original vectors, is especially efficient for large magnitude vectors.

After NN-search using FFVA, there are two additional layers to further refine the matching result. The first layer is to remove the false alarms from complex background. We used the *distance ratio* as evaluation criteria, that is:

“the second closest neighbor should be significant far away from the closest one”[1]. The threshold value of α=0.5~0.9 was used in our experiments. Only those correspondences past the first layer will enter the last layer, which is a consistent check using RANSAC. Experimental results show the last layer is only necessary for challenging real image tests with complex background. For the relatively easy synthetic test, our method is reliable enough to skip the consistent check layer.

5. Experimental result

The proposed MVKP method was tested using synthetic and real images as well as the combination of both. Real images are captured using a DSLR camera with high light-sensitivity settings (ISO=800~1600) and in-camera noise reduction off, which gives the input pictures hardware (CCD) generated randomly distributed noise points with random intensity. To obtain the synthetic images, we either performed synthetic viewpoint and lighting changes on the real images or download computer generated images from the Internet.

We compared our method with SIFT method, classification method using PCA [2] (represented by CPCA) and randomized tree based method [8] (repressented by CRTR) to demonstrate our method’s effectiveness. In all the experiments the number of generated training views for MVKP is 100, for CRTR is 1000, the number of selected objection feature location is 200, the maximum keypoint number returned by the detector is 500, image patch size is 32 by 32 and the number of k-mean kernels to represent each object location is 20. The test computer is a desktop PC with Pentium VI 1.4G CPU.

5.1. Effect of projection kernels

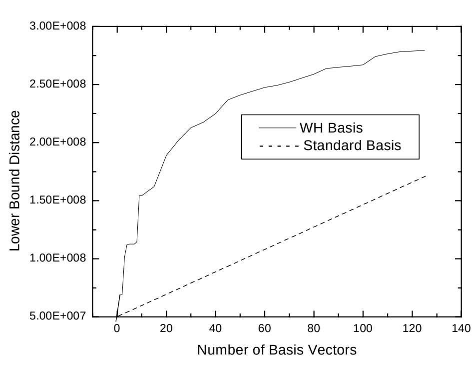


Figure 8: Lower bound distance obtained versus number of basis vectors used. WH projection kernels versus standard basis.

In the first test, we use two original 256-dimensional feature vectors (one from training stage and the other from the query image) and project them onto the first 5, 10, … , 125 WH kernels respectively. Each time, we calculate a lower bound of the squared Euclidean distance (around 3×10^8 for this test) between the projected vectors.

Figure 8 demonstrates the kernels’ effectiveness. Compared with the result of standard basis vectors, the projections onto the first 20~50 WH kernels already captures the majority difference between the two vectors.

5.2. Feature distinctiveness

This experiment shows that our feature vectors generated from WH kernel projections are sparser distributed in the space compared with CPCA method. In other words, our feature vectors are more distinctive one from the other, resulting more reliable feature matching and allowing vector approximation based NN-search technique like FFVA working more efficiently.

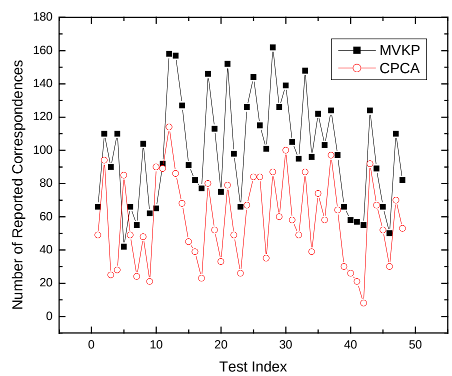


Figure 9: Number of returned correspondences after distance ratio (α=0.5) layer. The number of projection kernels is 50.

Figure 9 shows that for CPCA and MVKP method, the number of reported correspondences under the same distance ratio α=0.5. For the same feature sets size, MVKP has a much higher reported correspondence number indicating MVKP’s feature vectors are less likely to cluster together than CPCA’s. Therefore, it is easier to find a distinctive matching in MVKP’s feature space.

5.3. Matching accuracy and robustness

For synthetic image test, even without the consistent check step like RANSAC, our MVKP method is able to return a large number of correct matching in real-time. The really challenging experiments are those using real images or the combination of real-world and computer-generated images. For pure real image tests, the pictures of the same object are captured from different view angle, under different lighting condition and maybe partial occluded. We also carried out some “extremely difficult” tests, in which the training images are the logos of popular local stores downloaded from the Internet, while the query images are captured from the real-world posters, signs of those stores. Because the training images are computer-generated images, they typical have different shape, color, color distribution or even different layout from the real-world logos, which makes the testing cases very challenging. We also had live-demo comparison when query images arriving in real-time captured from a web-camera.

Overall, Our MVKP method (although only 100 views are used for training) shows better accuracy, comparable number of reported correspondences and faster query speed compared with CRTR (1000 views training). In some very difficult testing cases, CRTR gives no output at all while our method still have rather good results. The following pictures show some of those testing results.

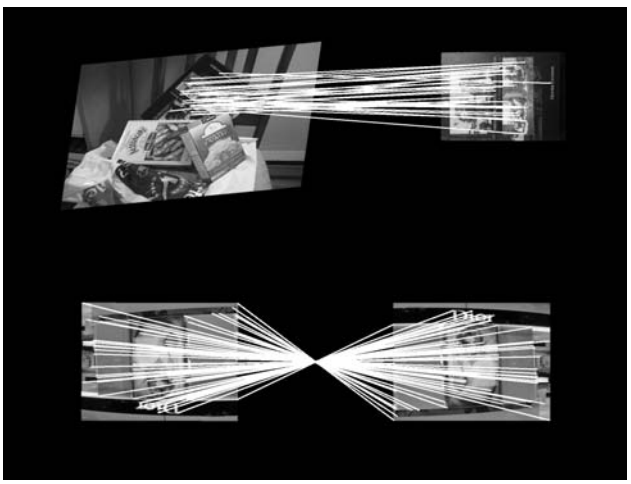
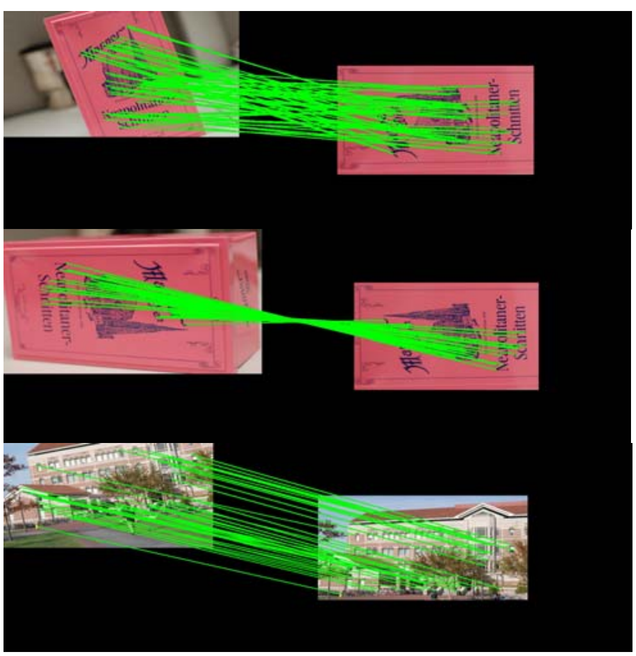


Figure 10(a): Synthetic images test result



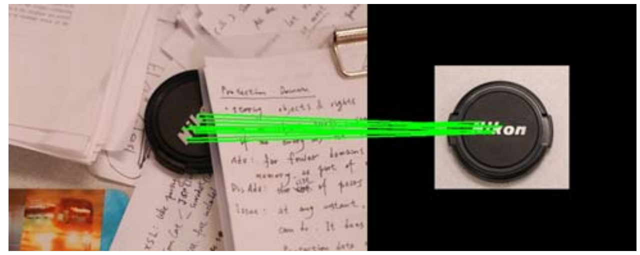


Figure 10(b): Real images test result

Figure 10(c): Real and computer-generated images test result. Right side training images are computer-generated images downloaded from the Internet. In the "Burger King" example, red (dark dash) lines are outliers removed by consistent check.

5.4. Matching speed

CPCA treats feature matching as a classification problem and achieves an online matching speed 5 times faster than SIFT. It is fast enough for many application areas but still not real-time. The introduction of randomized tree (CRTR method) brought the performance into the real-time range and also obtained more robust performance. Figure 11 shows the results of our query speed test using large-size synthetic images.

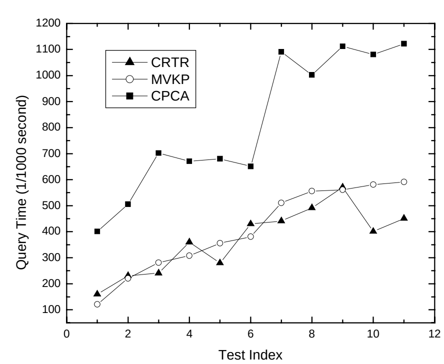


Figure 11: Query speed comparison. 11 tests are included. Index is ordered according to the query time of MVKP. Exact linear scan is used for both MVKP and CPCA as NN-search method.

The number of feature vectors generated from training stage is 4,000 while the number of query image feature vectors ranges from several hundreds to 5,000. The original real images include aerial image, ad poster, small indoor item and complex scene. Synthetic scale, rotation, shear and lighting changes are applied to those real images to generate query images. Linear scan is used for both CPCA and our MVKP methods. Even when the simple and slow linear scan is used, MVKP's query speed is much faster than CPCA and comparable with CRTR. The typical training time for CRTR is around 15 minutes, for CPCA is around 1 minute and for MVKP is only around 30 seconds.

Analyzing the matching time composition for our MVKP method we found that the time spent for NN-search using linear scan is more than 70% of the total online matching time. The description step is within 0.1 seconds thanks to the fast applicable WH projection kernels. When the feature sets size is large or the application demands a large number of correspondences, a more efficient NN-search technique is the key to the whole system performance. We choose FFVA method due to the feature vector's two inherent properties: large magnitude and sparse distribution. Our experimental result shows a significant speed enhancement (more than double the speed) over the exact linear scan method. Furthermore, although an approximated NN-search technique, FFVA has accuracy close to exact linear scan [13].

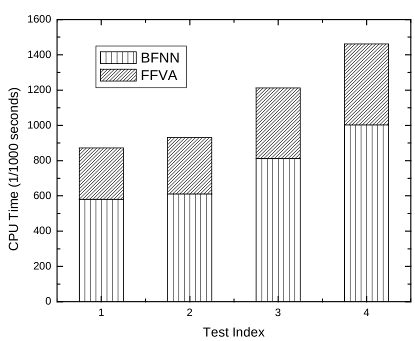


Figure 12: Search speed comparison, BFNN versus FFVA

The above results are obtained during synthetic image tests. Linear scan and FFVA runs on the same feature vector sets and their CPU time is recorded.

References

[1] D. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 2004

[2] V. Lepetit J. Pilet P. Fua. Point matching as a classification problem for fast and robust object pose estimation. In Conference on Computer Vision and Pattern Recognition, 2004

[3] P. Viola and M. Jones. Robust real-time object detection. In ICCV Workshop on Statistical and Computation Theories of Vision, July 2001.

[4] Y. Hel-Or and H. Hel-Or. Real-time pattern recognition using projection kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence, Sep 2005.

[5] C. Schmid and R. Mohr, Local grayvalue invariants for image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 530–534, May 1997.

[6] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In Proceedings of the Conference on Computer Vision and Pattern Recognition, pages 511.517, 2004.

[7] K. Mikolajczik and C. Schmid, A performance evaluation of local descriptors, in Conference on Computer Vision and Pattern Recognition, June 2003, pp. 257–263.

[8] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In Conference on Computer Vision and Pattern Recognition, 2005.

[9] A. Boffy, Y. Tsin and Y. Genc. Real-Time Feature Matching using Adaptive and Spatially Distributed Classification Trees. In British Machine Vision Conference, 2006

[10] V. Ferrari, T. Tuytelaars, Luc Van Gool. Wide-baseline Multiple-view Correspondences. In Conference on Computer Vision and Pattern Recognition, 2003.

[11] M. Ozuysal, V. Lepetit, F. Fleuret, and P. Fua. Feature Harvesting for Tracking-by-Detection. European Conference on Computer Vision, 2006

[12] J. Winn and A. Criminisi. Object Class Recognition at a Glance. Video demo in Conference on Computer Vision and Pattern Recognition, 2006.

[13] Q. Wang and S. You. Fast Similarity Search for High-Dimensional Dataset. IEEE International Workshop on Multimedia Information Processing and Retrieval, 2006. Available at: http://www-scf.usc.edu/~quanwang/FFVA.pdf

[14] O. Tuzel, F. Porikli, P. Meer. Region Covariance: A Fast Descriptor for Detection and Classification, European Conference on Computer Vision, May 2006

[15] M. Elad, Y. Hel-Or, and R. Keshet. Pattern detection using maximal rejection classifier. In International Workshop on Visual Form, pages 28–30, May 2000.

[16] D. Keren, M. Osadchy, and C. Gotsman. Antifaces: A novel, fast method for image detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(7):747–761, 2001.

[17] G. Ben-Artzi, H. Hel-Or, Y. Hel-Or. Filtering with Gray-code kernels, International Conference on Pattern Recognition, 2004.

[18] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2000.

[19] Jeffrey S. Beis and David G. Lowe, Shape indexing using approximate nearest-neighbor search in high-dimensional spaces, In Conference on Computer Vision and Pattern Recognition, 1997.

[20] H. Ferhatosmanoglu, E. Tuncel, D. Agrawal, A. El Abbadi, Vector Approximation based Indexing for Non-uniform High Dimensional Data Sets, ACM: Conference on Information and Knowledge Management, 2000.