We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

5

Image Matching and Recognition Techniques for Mobile Multimedia Applications

Suya You, Ulrich Neumann, Quan Wang and Jonathan Mooser
Computer Graphics and Immersive Technologies Lab
Computer Science Department
University of Southern California
USA

1. Introduction

Image spatial-temporal matching is a fundamental task in visual media processing used to comprehend two or more images taken, for example, at different times, from different sensors, or from different aspects. Many multimedia systems and applications require image matching, or closely related operations as intermediate steps, including image database classification and retrieval, Internet image search engine, and multimedia content analyzing and understanding. The rapid convergence of multimedia, computation and communication technologies with technique for device miniaturization is ushering us into a mobile, wireless and pervasively connected multimedia future. It is now quite common for mobile platform to integrate a mobile phone, digital camera, music player, and PDA into one device, heralding many exciting new applications and services, as exemplified by several new multimedia phone services recently launched, such as the Nokia Point & Find, MyClick in telecommunication markets. The services employ the image recognition and search techniques that allow users to capture designated images such as product advertisements and quickly match them to the vendor's databases to obtain detailed product information associated with the images.

While capability trend is clear, the main challenge posted by such multimedia systems is the developed image matching and recognition algorithms have to be reliable, fast, robust and capable to handle the realistic conditions in our real-world. Technically, any image matching and recognition process generally consists of three components: (1) feature selection and detection - finds stable matching primitives over spatial-temporal space to achieve scale and pose invariance, (2) feature representation and description - represents the detected features into a compact, robust and stable structure for image matching, and (3) optimal matching and search - uses the feature descriptions and additional constrains to locate, index, and recognize the targets and scenes of interest.

This chapter presents several advanced image matching and recognition technologies, of particular emphasis on mobile multimedia applications that are feasible with current or near term technology and applications. The presented work represents the latest state-of-arts where we have a strong base of techniques and knowledge in this area. In Section 2, we

present a high-performance matching technique required by real-time multimedia applications. In Section 3, a highly compact image feature description and matching technique is presented, targeting the mobile multimedia applications. Section 4 describes an efficient image search technique that can dramatically improve on previous approaches over hundred-times faster for recognition of objects from a large library of database. Finally, Section 5 presents an application system, called Augmented Museum Exhibitions that combines the mobile computation, Augmented Reality and image matching/recognition techniques to demonstrate the effectiveness and utility of the presented technologies.

2. Real-Time Image Matching Based on Multiple View Kernel Projection

Technically, any 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 (Ke & Sukthankar, 2004; Lepetit et al., 2004; Lowe, 2004; Mikolajczik & Schmid, 2003; Tuzel et al., 2006). The approach described in this Section 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 (Hel-Or Y. & Hel-Or H., 2005). Kernel Projection, however, does not naturally have the important property of geometry invariant. Therefore cannot 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 (Lowe, 2004; Mikolajczik & Schmid, 2003; Tuzel et al., 2006; Lepetit et al., 2004). 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. The FFVA technique will be described late in Section 4.

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, the real-time MVKP system demonstrates both effectiveness and robustness.

2.1 Relevant 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 [Schmid & Mohr, 1997)] introduced a rotationally invariant descriptor for local image patch based on

local greylevel invariants. The ground-breaking work of D.G. Lowe [Lowe, 2004)] 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 (Ke & Sukthankar, 2004) applied PCA to image gradient patch in order to reduce the descriptor's dimensionality. GLOH (Mikolajczik & Schmid, 2003) 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 [Lepetit et al., 2004; Lepetit et al., 2005; Boffy et al., 2006; Ferrari et al., 2005] while others take real images captured from camera as input (Ozuysal et al., 2006; Winn & Criminisi, 2006). 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 (Lepetit et al., 2004) treats the multiple-view point matching problems as a classification problem. They synthesize small patches 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 (Lepetit et al., 2005), 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 (Boffy et al., 2006).

(Ozuysal et al., 2006) 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 (Tuzel et al., 2006) proposes 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 (Boffy et al., 2006) uses 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 (Elad M. et al., 2000; Keren et al., 2001), while Y. Hel-Or, et al. (Hel-Or Y. & Hel-Or H., 2005) 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. Furthermore, experimental results indicate their Projection Kernel method is robust to noise and lighting changes. However, as a fast window matching technique, the method cannot handle geometric distortion brought by view angle changes.

2.2 The Walsh-Hadamard Kernels Projection

The projection scheme in our MVKP method is based on Walsh-Hadamard (WH) kernels, which is a special case of Gray-Code Kernels (Ben-Artzi et al., 2004) and general projection kernels in Euclidean space.

2.2.1 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. Assume $\vec{b}_1, \vec{b}_2, \vec{b}_3 \ldots$ are orthonormal projection basises in k^2 dimension Euclidean space (Figure 1(a)). *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:

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

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:

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

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" [Hel-Or Y. & Hel-Or H., 2005].

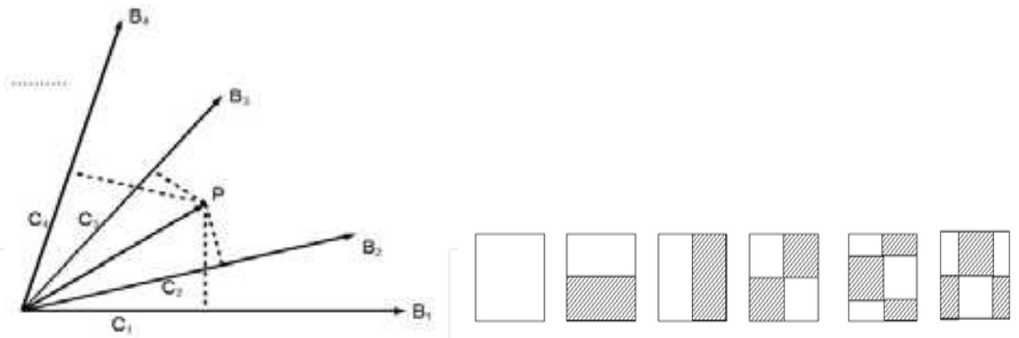
2.2.2 The Walsh-Hadamard kernel

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 (Ben-Artzi et al., 2004). 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.

Second, when the kernels are ordered according to increasing frequency of sign changes, the experimental results 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 1(b) shows a list of two-dimensional WH kernels in increasing order of frequency. (Ben-Artzi et al., 2004) introduced an efficient algorithm to compute the ordering of the kernels, which captures the increase in spatial frequency.

2.3 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.



(a) (b)

Fig. 1. Kernel projection. (a) General projection. (b) 2-dimensional 4×4 WH kernels in increasing order of frequency. Blank represents value "1" and shadow represents "-1".

2.3.1 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 2 illustrates the major components of the 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:

$$x' = H_A x = \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix} x$$
$$A = R(\theta)R(-\phi)DR(\phi) \text{ and } D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \quad (3)$$

where R is the rotation matrix and t is a translation with components $t1$ and $t2$. Matrix A corresponds to a rotation of θ first, followed by a rotation of $-\varphi$ then scale changes of $\lambda 1$ and $\lambda 2$ 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 $\theta \in [-\pi,\pi]$, $\varphi \in [-\pi/2,\pi/2]$, $\lambda 1$,$\lambda 2 \in [0.4, 1.6]$, $t1$,$t2$ = 0, 1, 2, or 3.

Searching for local maximum Eigen-values within 3×3 local patches identifies the local feature points. 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. 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 (Lepetit et al., 2005). 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 that cluster.

2.3.2 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.

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:

$$y \leq \min(x_1, x_2\} \tag{4}$$

Typically, x_2 is around 500, assume x_1 is 100, then the NFRC will be no larger than 100.

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 most time-consuming part is to find the correspondence between feature sets.

2.3.3 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 (Hel-Or Y. & Hel-Or H., 2005) 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. Nearest Neighbor (NN) search techniques have been studied under this context. The authors of (Lepetit et al., 2004) use linear scan because of its simplicity and accuracy, while in (Jeffrey & Lowe, 1997), 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 the 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 (Lepetit et al., 2004), 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 to perform the NN-search, which is described in Section 4.

2.4 Experimental Results and Evaluations

The method has been 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 incamera 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 the MVKP method with SIFT method, the classification method using PCA (Lepetit et al., 2004) represented by CPCA and the randomized tree based method (Lepetit et al., 2005) represented by CRTR. 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 1.4GHz CPU.

2.4.1 Effect of projection kernels

To evaluate the effect of projection kernels, 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 3×108 for this test) between the projected vectors.

Figure 2 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.

2.4.2 Feature distinctiveness

This experiment shows that the 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 3 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.

2.4.3 Matching accuracy and robustness

For synthetic image test, even without the consistent check step like RANSAC, the 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 computergenerated 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 had live-demo comparison when query images arriving in real-time captured from a live web camera.

Fig. 4. Evaluation with real image sets. Note that the training images (right side) are computer-generated graphics downloaded from the Internet. The testing images (left sides) are captured with digital camera.

Overall, the 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. Figure 4 shows examples of evaluation results.

2.4.4 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 5 shows the results of our query speed test using large-size synthetic images.

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 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 linear scan method. Although an approximated NN-search technique, FFVA has accuracy close to linear scan.

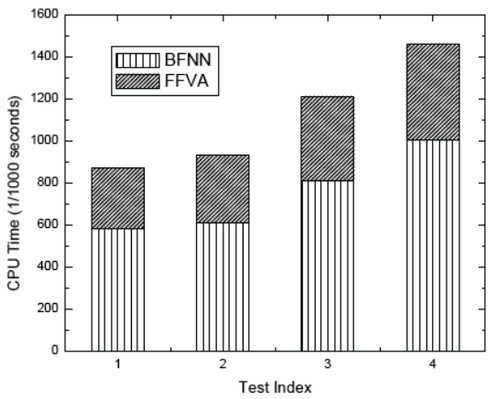


Fig. 5. Matching speed evaluation with comparisons of BFNN versus FFVA

3. Highly-Compact Image Feature Description and Matching Technique

Many methods for feature descriptions have been suggested, which can incorporate various degrees of resistance to common perturbations such as viewpoint changes, geometric deformations, and photometric transformations. Among the approaches, the SIFT descriptor has been shown to outperform other descriptors (Lowe, 2004). The SIFT descriptor is based on the gradient distribution in salient region, and constructed from a 3D histogram of gradient locations and orientations. A 128-dimension vector representing the bins of the oriented gradient histogram is used as descriptor of salient feature.

However, the high dimensionality of SIFT descriptor is a significant drawback, especially for online or large-scale dataset applications. For a typical outdoor scene, for example, the SIFT usually produces several hundreds of local features, yielding a large high-dimensional feature space needs to be searched, indexed, and matched.

Several researchers have addressed the problem of dimensionality reduction for feature descriptors. For example, Bay et al. (Bay et al., 2006) proposed an approach (SURF) that combined the Hessian matrix-based measure for the detector and Haar-wavelet responses for the descriptor, resulting in a 64-dimension feature representation. PCA-SIFT proposed in (Ke & Sukthankar, 2004) reduced the dimensionality of descriptor to the range of 36, while remaining a comparative performance to the original SIFT. The key of PCA-SIFT is to apply the standard Principal Components Analysis technique to the gradient patches extracted around local features, therefore yielding a compact feature representation. However, the PCA-SIFT needs an offline stage to train and estimate the covariance matrix used for PCA projection. This typically requires the system to collect and train a large, diverse collection of images prior to use, (it often needs to re-train and re-estimate the covariance matrix when the image database is expanded or the scenes have significant changes), thereby impeding its widespread use and benefits.

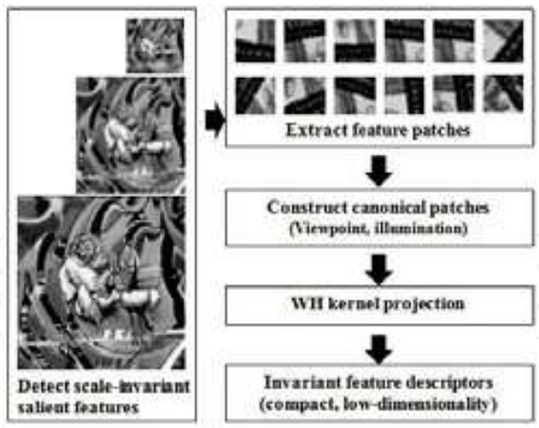


Fig. 6. CDIKP algorithm structures

This Section presents our efforts in developing an efficient local feature and its invariant descriptor for scene recognition. Our main contributions lie in a novel approach that uniquely combines the scale-invariant feature detection with a robust kernel-based representation technique to produce highly efficient feature representation. We named the approach Compact Descriptor through Invariant Kernel Projection (CDIKP). The produced feature descriptors are highly-compact (20-Dimension) in comparisons to the state-of-the-art (e.g. SIFT: 128-D, SURF: 64-D, and PCA_SIFT: 36-D), do not require any pre-training step, and show superior advantages in terms of distinctiveness, robustness to occlusions, invariance to scale, and tolerance of geometric distortions.

3.1 Approach

Figure 6 depicts the main steps of the CDIKP approach, which are detailed in following sections.

3.1.1 Scale-invariant Feature Detector

The approach selects multi-scale salient features/regions with the scale-invariant detector, similar as (Lowe, 2004) where 3D peaks are detected in a DoG scale-space. The peaks in a DoG pyramid have been shown to provide the most stable interest regions when compared to a range of other interest point detectors.

Three spatial filters are used in the detector. First, a high frequency-passed filter is employed to detect all the candidate features with local maximum responds in the DoG pyramid. The second filter is a distinctiveness filter that removes the unstable features usually lying along the object edges or linear contours. The third filter is an interpolation filter that iteratively refines the feature locations to sub-pixel accuracy. Finally, the dominant orientation and scale are computed and assigned to each detected feature. The dominant orientation and scale will be used for view normalization to achieve viewpoint invariant.

3.1.2 Scale-invariant Feature Detector

Discrimination power is an important factor required for object recognition with high data variability. We base our feature descriptor on the projection kernel scheme described in above Section 2, because the projection kernel techniques have demonstrated strong discrimination performance and they are well established analytical tools that are useful in variety of contexts including discriminative classification, scene recognition and categorization. Another attractive feature of the projection kernel techniques is their innate data compaction that can efficiently map high dimensional data to a compact representation with much lower dimensionality. This is a very attractive property for image description from which we could produce compact, lower-dimensional descriptors.

Choosing an appropriate kernel function is a key for efficient projection kernel schemes. As stated above in Section 2, two important factors have to be considered: the kernel functions should be ordered in a way such that the lower bound becomes tight after only a small number of projections, and the kernels should be efficient to enable fast computation.

We decided to use Walsh-Hadamard (WH) kernel because of its good performances in discrimination and computational efficiency. Mathematically, the WH kernel vectors can be recursively constructed as a set of orthogonal base vectors made up entirely of 1 and −1. Computation of the WH transform involves only integer additions and subtractions. Given an image patch of size $k \times k$, its WH transform is computed by projecting the patch onto k^2 WH kernel vectors. It has been shown in (Hel-Or Y. & Hel-Or H., 2005) and also confirmed by our experiments that the first few WH projection vectors can capture a high proportion of information contained in the image (Figure 2, 7). These unique properties of WH kernel projection lead us an efficient tool to build compact descriptors.

3.1.3 Generate Descriptors with WH Projections

WH kernel projection, however, does not naturally have the important property of geometry invariance, thus it cannot handle geometric distortions caused by viewpoint or pose changes. We solve this problem by performing a viewpoint normalization step on the basis of the feature's dominant orientation and scale.

Constructing the canonical views of features is relatively simple and fast. We first extract local patches centered at the feature locations from the Gaussian pyramid constructed in the above step of feature detection. The size of patch varies with the scale at which the feature was detected. Under the assumption of local planarity, a new canonical view of the local patch (with fixed size and scale) is synthesized by image warping with the feature's dominant orientation and scale. This corresponds to a regular re-sampling process in an affine space. Note that the size of the canonical patch is fixed and has to be in the power of 2, as required by WH transform. Our extensive experiments show that the size of 32x32 gives the optimal results (Figure 8).

To reduce the effect of photometric changes, we use gradient for each patch in WH transform. We have evaluated several gradient computation and forms, and found that the Gaussian weighted first order derivatives of pixel intensity along horizontal and vertical directions seemed to yield the most robust results to compensate the substantial changes of illumination. Thus, we first calculate the first-order derivatives in x and y directions within a local patch, and then weight the directional derivatives using a weighted Gaussian kernel. In this way, we obtain a pair of x-y gradient maps for each local patch that is canonically normalized to viewpoint and photometric variances.

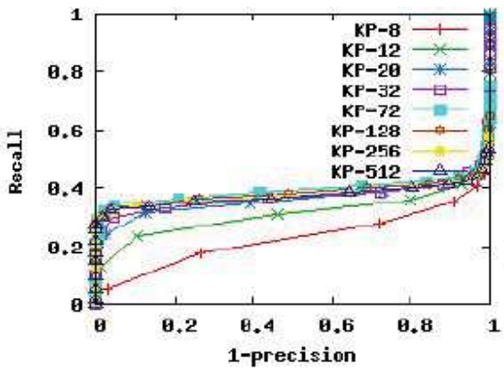


Fig. 7. Impact of the lengths of WH projection vectors on feature matching.

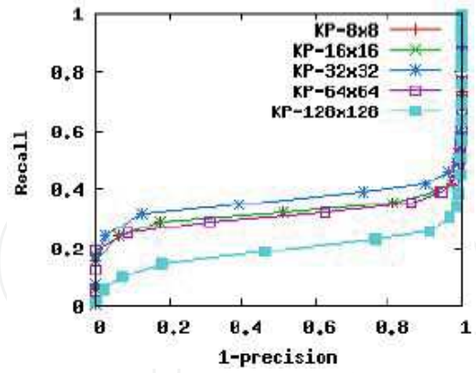


Fig. 8. Impact of patch sizes on feature match performance.

We then use the WH kernel projection to extract significant components contained in the local patches to generate feature descriptors. Since we obtain two 1024-element gradient maps for each patch/feature, we apply the WH transform twice to the gradient maps: one for x-component and one for y-component. Finally, the first 10 projection vectors of each WH transform are extracted and combined to produce a 20-dimension feature descriptor that is compact, distinctive, and viewpoint and illumination invariant.

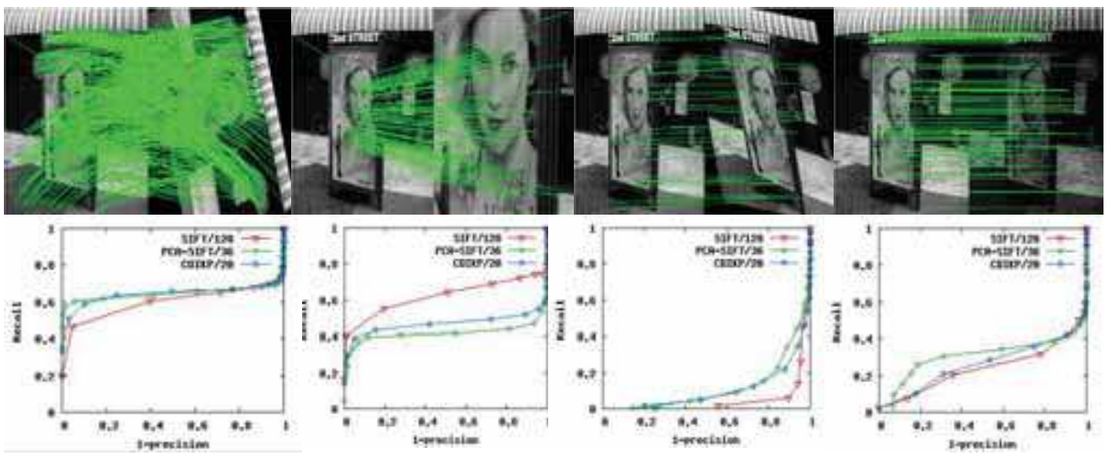
3.2 Performance Evaluation and Applications

We evaluated the proposed approach using various datasets including synthesized data, a standard evaluation set, and our own datasets acquired under varying circumstances. We evaluated the effectiveness of our approach, in comparison to other descriptors, in the terms of distinctiveness, robustness and invariance.

3.2.1 Synthesized Data Evaluation

We collected a dataset of images, and intentionally distorted them with various geometric and photometric transformations. For a pair of test images, we ran the CDIKP algorithm to automatically select distinctive features, generate descriptors, and find the feature matches. The results were evaluated using the standard metric of recall-precision graphs. We conducted performance comparisons to standard SIFT, and PCA-SIFT. In our tests, we tried to use the same set of parameters for all the three methods.

Figure 9 shows results of the CDIKP approach to a scene under different distortions, where (a) is the matched features for the original image being rotated 70 degree; (b) for 250% scaling; (c) for 0.4-x, 0.1-y shearing; and (d) for 100% illumination change and adding 15% Gaussian noise.



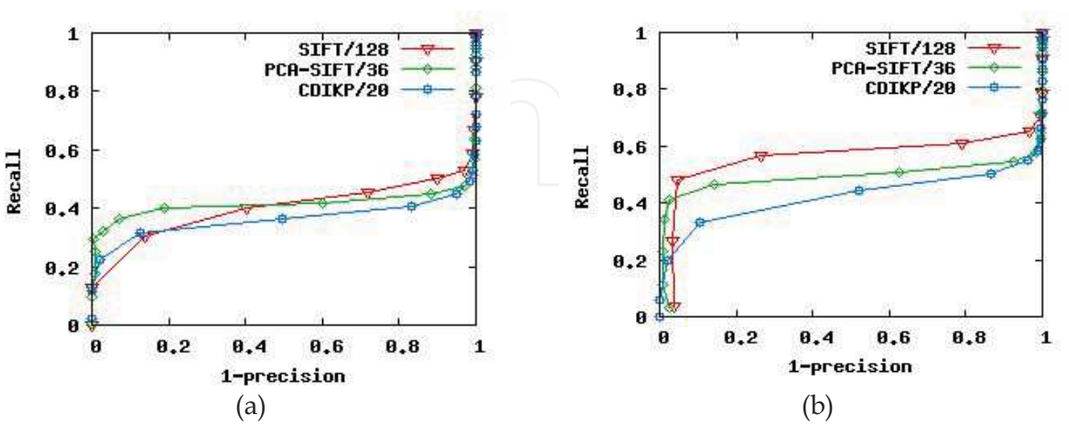
(a) (b) (c) (d)

Fig. 9. Performance evaluation under different imaging and viewpoint variances

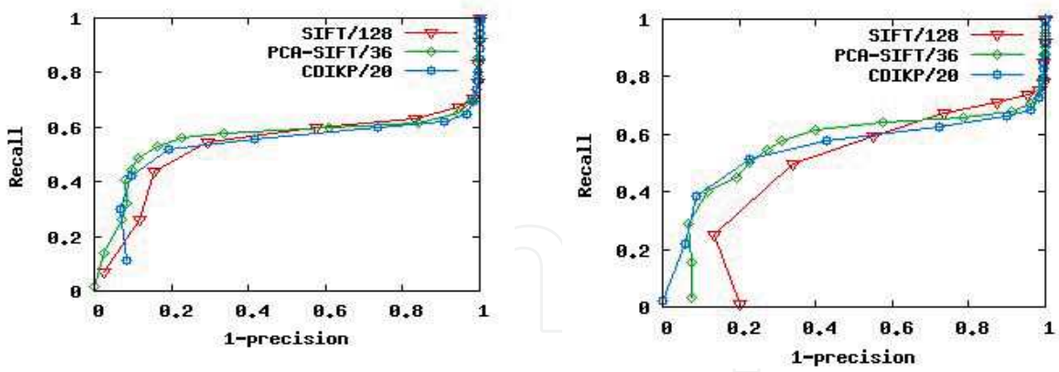
3.2.2 Standard Test Dataset with Ground Truth

Fig. 1. Evaluation with INRIA test dataset
We tested our approach using the INRIA dataset (Mikolajczik & Schmid, 2003). These are images of real scenes with recovered deformation parameters used as test ground. Figure 10 shows the results for several cases: (a) rotation and scale (Boat), (b) viewpoint changes (Wall), (c) image blur (Bikes), and (d) lighting changes (Leuven). We can see from these results that the CDIKP descriptor remains a very comparative performance, sometimes outperforms SIFT in recall for the same level of precision. Meanwhile, it is more compact and efficient to compute.

3.2.3 Scene Recognition Application



(a) (b)



(c) (d)

Fig. 10. Evaluation with INRIA test dataset

We used the approach for object recognition application intending to the content-based image retrieval on mobile-platform. Figure 11 demonstrates the scenario of applying the approach to automatically localize and recognize various commercial logos in nature mobile environments. The application uses image recognition and search techniques that allow users to capture designated images such as product advertisements and quickly match them to the vendor's databases to obtain detailed product information associated with the images. These examples demonstrate the value of the proposed approach for mobile multimedia applications such as product advertising and shopping.

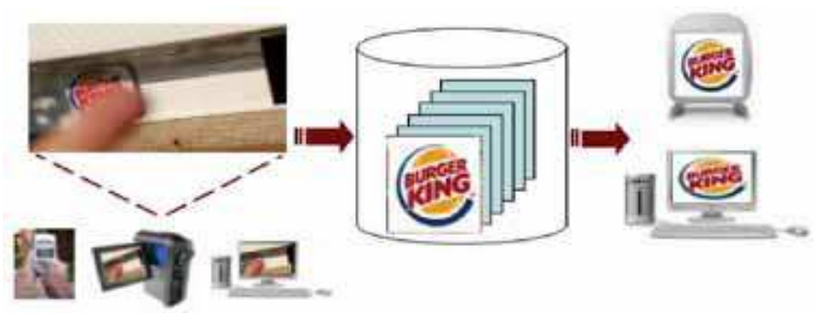


Fig. 11. Automated image matching and searching for mobile multimedia applications

4. Fast Similarity Search for High-Dimensional Dataset

Similarity search is crucial to multimedia database retrieval applications, for example, searching for correspondences between objects or retrieving multimedia contents from databases. The similarity search involves finding the most similar objects in a dataset to a query object based on a defined similarity metric. To achieve a robust and effective querying, many current multimedia systems use highly distinctive features as basic primitives to represent original data objects and perform data matching. While these feature representations have many advantages over original data including distinctiveness, robustness to noise, and invariance and tolerance to geometric and illumination distortions,

they typically produce high-dimensional feature spaces that need to be searched and processed. Methods that search exhaustively over the high-dimensional spaces are time-consuming, resulting in painfully slow evolution of such multimedia databases. Efficient search strategies are needed to rapidly and robustly screen the vast amounts of data that contain features and objects of critical interest to users' applications.

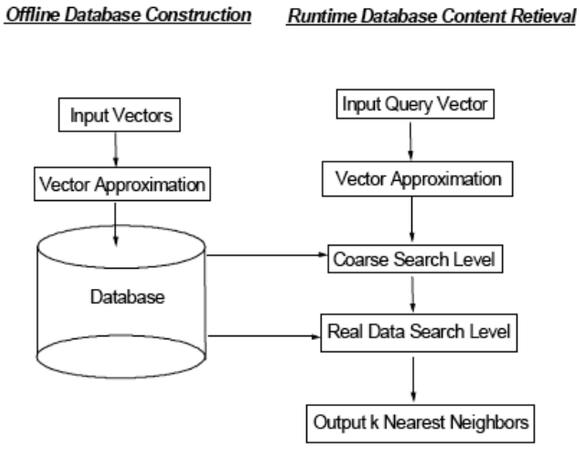


Fig. 12. Algorithmic structure FFVA approach

This session addresses the challenging problem of searching and matching high-dimensional feature sets (e.g. over 100 dimensions for each feature vector) for the applications of multimedia database retrieval and pattern recognition. Traditional tree-structure techniques hierarchically partition or cluster the entire data space into several subspaces and then use special tree structures to index objects. These types of approaches are suitable for nearest neighbor (NN) searching of datasets with low dimensionality, but their performance could rapidly degrade when directly adapting to high dimensionality. The limitations of simply modifying or adapting these techniques to highdimensional datasets are severe. This is referred to "curse of dimensionality" (Bellman ,1961) and places a practical limit on the partitioning based techniques. It has been shown in (Weber et al., 1998) and our experiments, that using the hierarchical partitioning and indexing structures for searching beyond a certain dimension becomes even worse than an exhaustive sequential-scan.

Recently, there have been great efforts in developing vector approximation (VA) techniques such as VA-File (Blott & Weber, 1997) intending to overcome the limitations of the treestructure approaches. Instead of partitioning the input data space hierarchically, the vector approximation methods directly index the objects based on linear and flat structure. While the VA approaches have several inherent problems, they demonstrate better performance in high-dimensional feature retrieval, and do not suffer from the problem of dimensionality curse.

This seccsion describes an improved vectorapproximation method, called Fast Filtering Vector Approximation (FFVA), for rapidly searching and matching high-dimensional features from large multimedia databases. FFVA is an NN-search technique that facilitates rapidly indexing and recovering the most similar matches, i.e. k-NN, in a high-dimensional database of features or spatial data. Comparing with several existing techniques including exhaustive linear scan, KD-tree (Friedman et al., 1977), Best-Bin-First (Jeffrey & Lowe, 1997),

and VA-File (Blott & Weber, 1997, Weber et al., 1998), the FFVA has demonstrated better performances in terms of query exactness, data access rate, query speed, and memory requirement.

4.1 FFVA for Similarity Search

FFVA is an improved version of VA-File method to achieve fast vector approximation and indexing. Figure 12 illustrates the structure of FFVA and its major components for an efficient nearest neighbor search.

The basic data structure of FFVA and standard VAFile method is a space-partition-table (SPT). Each dimension of input feature vectors is quantized as a number of bits used to partition it into a number of intervals on that dimension. In the whole vector space, each rectangle cell with the bit-vector representation approximates the original vectors that fall into that cell, resulting in a list of vector approximations of the original vectors. It is noteworthy that our FFVA method clusters the original vectors to the corners of SPT cells, thereby enable us to use a list of corners, instead of cells, to approximate the original vectors. Such strategy is efficient for the following fast lower bounds filtering.

There are two major levels involved in FFVA NNsearch: 1) coarse search level to sequentially scan the approximations list and eliminate a large portion of data, and 2) real data search level to calculate accurate distances of resultant candidates and decide the final knearest neighbors.

In previous works, lower-bound (the Euclidean distances from the SPT cell's nearest corner to a query point) are used for the coarse search. Experiments show that the approximation quality and computation cost of the bound-distances determinate the performance of the entire searching system (Tuncel et al., 2002; Ferhatosmanoglu et al., 2000).

Figure 13 shows the total time (sum of 900 queries) of a typical VA-file method query using real data sets (128-D features). According to the graph, the calculation of lower-bounds takes around 90% of the total querying time. This result is consistent with the experimental results in (Ciaccia & Patella, 2000). Therefore, a more efficient method for lower-bounds computation is essential for improving the entire searching performance.

In the coarse search level, only the vector approximations are accessed and *block distance* is used as similarity metric, which is calculated by employing the Manhattan distance of corresponding corner points of two SPT cells:

$$BD = \sum_{i=1}^{n} | v_{1i} - v_{2i} | \tag{6}$$

where v_{1i} and v_{2i} are the coordinates of two corresponding corner points in a n-dimension space.

Table 1: FFVA algorithmic structure

Input: query and database vectors

Output: *kNNQ* containing k-nearest neighbors

/* **Note:** *kNNQ* is a list containing the k-nearest neighbors found so far, sorted according to ascending order of exact distance from a query point */

1. Calculate or load (if pre-computed) the approximations of database vectors: *aprov* and query point: *aproq*;
2. Initialize *kNNQ*;
3. *max_db* = maximum possible value;
4. For each approximation *aprov* of database vectors

```
{
    current_bd = block distance between aprov and
     aproq;
     if (current_bd < max_db)
     {
         Calculate the corresponding actual distance;
         Insert this vector into kNNQ, if it is closer
         to query point than the last element in
         kNNQ;
         Update max_db to be the distance from
         query point to the last element of k-NN
         found so far;
     }
}
```

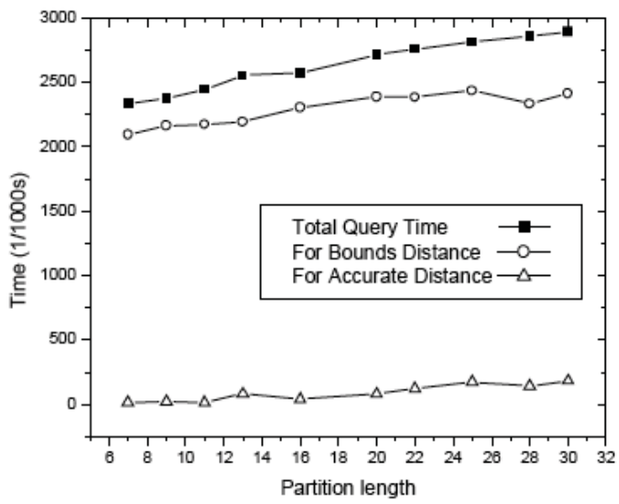


Fig. 13. Query time of VA-File

Let "*max_bd*" represents the longest exact distance (squared Euclidean distance) from a query point to current k-nearest neighbors. Since in our experiments, all the vector coordinates are integers so the exact distance is strictly lower-bounded by the block distance, whenever we encounter an approximation whose block distance from the query point is larger than *max_bd*, it can be guaranteed that at least *k* better candidates have already been found. Therefore, we can eliminate data with block distances larger than *max_bd*. Updating max_bd is also fast, as we dynamically sort and maintain the k-NN structure.
Only those candidates with block distance no larger than *max_bd* will enter the real data search level. In this level, their original vectors are accessed in order to calculate their exact distances. If the exact distance turns out to be shorter than any of current k-NN distance, the k-NN as well as *max_bd* will be updated.
Table 1 outlines the algorithmic structure of the FFVA approach.

4.2 Experimental results

In this section, we provide extensive performance evaluation and comparison of the proposed FFVA approach with four commonly used k-NN search techniques: exhaustive linear scan, standard KD-tree, BBF, and standard VA-File. The tests cover: search accuracy, data access rate, query speed, and memory requirement.
Both synthetic and real data are used in our experiments. The real data sets are large number of high-dimensional feature vectors (128-dimension for each feature) generated from a range of real images containing various object and scenes. SIFT (Scale Invariant Feature Transforms) approach (Lowe, 2004) was used to extract the image features described as 128- dimension vectors for feature matching.
In our test, the partition length for VA-File and FFVA methods is 15. For BBF, the size of buffer for "best bins" is 25 and the limit number of examined nodes is 100. During the similarity search, top two nearest neighbors (2-NNs) were returned. To evaluate the matching correctness, we used the distance ratio as evaluation criteria, that is: the second closest neighbor should be significant far away from the closest one. A threshold value of 0.6 was used throughout the experiments.

4.2.1 Search accuracy

In the accuracy test, we randomly pick up 1000 vectors from a synthetic database served as query vectors (test set). Since the test set is a subset of the database, ideally a perfect match of a query vector should have zero distance.

Dimension	60	70	80	90	100	110	115
Correct matches	1000	999	1000	1000	998	1000	1000
Dimension	120	125	130	135	140	145	150
Correct matches	1000	1000	1000	1000	1000	1000	1000

Table 2. Matching accuracy of FFVA

Table 2 summarizes the number of correct matches (out of 1000 queries) of the proposed FFVA method to the synthetic database. Gaussian noise (variance = 0.3) is added to the

uniform distributed vectors in the test set. We also conducted tests on various real data sets. Overall, the matching performance is consistent with the results of above synthetic data.

4.2.2 Search accuracy

Data access rate is an important aspect to evaluate the effectiveness of an approach for very large highdimensional database retrieval problem. KD-tree and BBF approaches utilize hierarchy tree structure to skip the nodes that are not along the searching path. FFVA and VA-File methods use compact vector approximations to avoid accessing the majority of the original database vectors.

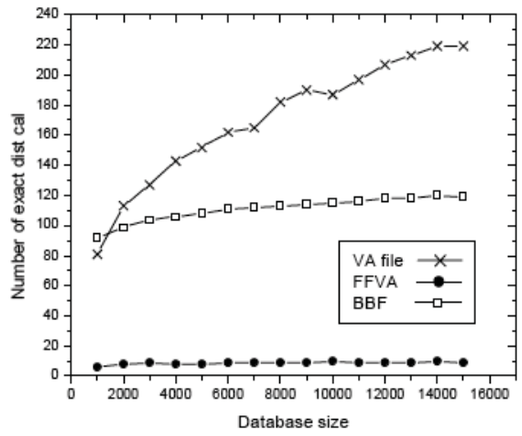


Fig. 14. Data access rate test (results are the averages of 1000 queries)

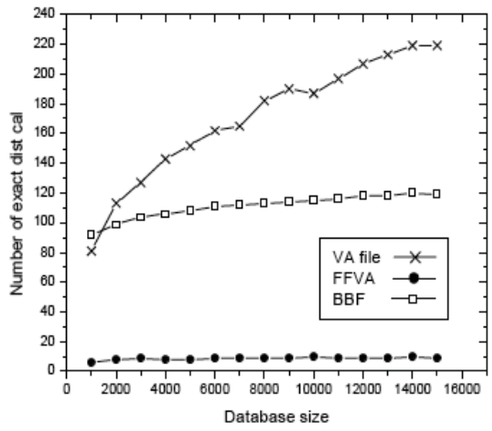


Fig. 15. Query speed test (synthetic data)

Figure 14 illustrates the test results of data access rates for three methods. The FFVA NN-search method clearly demonstrates the best performance. Its lowerbound is much tighter than that of standard VA-File method. As a result the proposed lower-bound computation based on block distance is efficient in reducing the amount of necessary data access and distance calculation. In this experiment of synthetic data, less than 10 exact distance calculations are needed for FFVA to find the 2-nearest neighbors among 15,000 120-dimension feature vectors, while other two methods (BBF and standard VA-file) spend 10-20 times more for the exact distance calculations.

4.2.3 Query speed

We tested the query speed of various data sets. In this test, we assume that all the feature vectors and their data structures are loaded into main memory (i.e. full memory access) to perform NN-search. The total query time also include the times spent on the tree construction (BBF) and vector approximation (VA-file and FFVA) processes.

Figure 15 shows the testing results (total query time of 100 queries) of synthetic data with dimension fixed at 100. Under full memory access assumption, BBF demonstrates the best performance among the three approaches when the database excesses certain size.

Figure 16 shows the test results of real data. The test data are collections of 128-dimension image features extracted from image pairs using SIFT approach. The search time per query

is total query time divided by the number of features in the query image. An exact NN-search using hierarchy structure becomes even slower than exhaustive sequential-scan when the dimensionality is high (Weber et al., 1998). Standard VA-file approach spends over much time for bounds-distances calculation when facing non-uniform real data, while FFVA and BBF demonstrate better and comparable performances.

4.2.4 Memory requirement

Memory usage has to be considered in developing an effective algorithm for search of large databases on mobile platform. BBF or similar tree-structure approach typically needs to load and store entire data sets into system memory for online processing: constructing tree structure, iteratively tracing the tree branches, and searching for the optimal tree nodes. This strategy apparently is impractical for searching very large databases when the data size easily overwhelms the limit of available memory space.

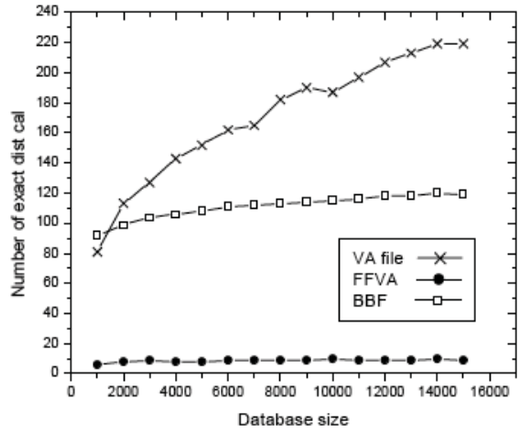


Fig. 16. Query speed test (real data)

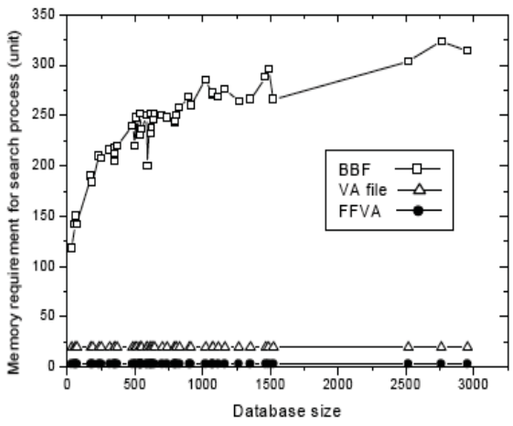


Fig. 17. Memory requirement test (note: one unit approximately equals to 512 bytes)

One of the major advantages of FFVA and VA-File is their low memory requirement provided by the flat and linear SPT data structure. Figure 17 shows the memory usages of three methods, BBF, VA-file and FFVA to query a real feature database containing 3,000 feature vectors.

We also tested the relationship among query time, data dimensionality and memory blocks (i.e. memory pages). These results clearly indicated that the memory usages of FFVA are far more effective than that of BBF. Therefore, FFVA is suitable for the application of large database retrieval or the systems that have limited memory spaces such as mobile computing devices.

5. Application: Augmented Museum Exhibitions

This Section presents an application system, called Augmented Museum Exhibitions that combines the mobile computation, Augmented Reality and image matching/recognition techniques to demonstrate the effectiveness and utility of the presented technologies.

Today, museums commonly offer a simple form of virtual annotation in the form of audio tours. Attractions ranging from New York's Museum of Modern Art to France's Palace of Versailles provide taped commentary, which visitors may listen to on headsets as they move from room to room. While a fairly popular feature, audio tours only offer a linear experience and cannot adjust to the particular interests of each visitor.

An AR museum guide is able to add a visual dimension. Pointing a handheld device at an exhibit, a user might see overlaid images and written explanations. This virtual content could include background information, schematic diagrams, or labels of individual parts, all spatially aligned with the exhibit itself (Figure 18).

The virtual content could be interactive as well. The AR content for a piece of art, for example, might allow the user to switch between background information about the artist, a description of the historical context of the piece, and a list of related works on display in the museum. An application for a science museum could display the inner workings of a machine, with the user able to adjust the level of details.

5.1 Augmented Reality and Related Works

Augmented Reality (AR) is a natural platform on which to build an interactive museum guide. Rather that relying solely on printed tags or pre-recorded audio content to aid visitors, an AR system can overlay text and graphics on top of an image of an exhibit and thus provide interactive, immersive annotations in real-time.

Graffe, et al., for example, designed an AR exhibit to demonstrate how a computer works (Grafe et al., 2002). Their system relies on a movable camera that the user can aim at various parts of a real computer. A nearby screen then displays the camera image annotated with relevant part names and graphical diagrams. Schmalstieg and Wagner presented a similar system using a handheld device (Schmalstieg & Wagner 2005). As the user walks from place to place, AR content provides information not only about the current exhibit, but also acts as a navigational tool for the entire museum.

Fig. 18. An example of how a museum visitor might use an augmented exhibit implementation on a cell phone.

Both of the above systems rely on printed markers for recognition and tracking. That means for every object to be incorporated in the AR application, a marker needs to printed and placed in the environment in such a way that it is always clearly visible. If at any time, no marker is visible inside the camera's field of view, then no AR content can be rendered. This

can lead to frustrations when a particular exhibit is partially or even fully visible but its associated marker is obscured, perhaps because another visitor is standing in the way.
Our work seeks to avoid the need for artificial markers by recognizing the target objects themselves, in this case 2D drawings and paintings. Thus, as long as an exhibit is visible to the user the application can render the associated AR content.
A number of approaches have been proposed for building natural feature based AR (Neumann & You, 1999; Coors et al., 2000; Simon et al., 2000). In this presented work, we use a modified MVKP for real-time image matching as described in Section 2.
Our information retrieval system is based on a simplified version of the multi-tier client/server architecture described in (Mooser et al., 2007). The user interacts with a client application that recognizes exhibits and sends their unique ID numbers to the server. The server then responds with all of the relevant data for that exhibit. Thus, even with a large number of clients, content for the entire application can be controlled from a single server.

5.2 System Overview

The vision-based augmented exhibition system we proposed is composed of four major components:
Acquiring query image: the system accepts query images captured from simple camera attached to a mobile device. It can also accept single JPG image or video clip.
Adapted MVKP: there are two main tasks for this component. First, given a painting, it builds the feature set for the painting. Low-resolution images (around 200x150) are enough and there is no need to extract the painting from the image in order to remove the background. Second, given a query image, it matches the painting to the database. If one of the trained paintings is matched, it establishes a feature correspondence between query image and database image. The output is the painting's ID and 3D pose with respect to the camera.
Remote Server: After the server receives the painting ID through a local Internet, it retrieves the corresponding information from its database (XML file), which it sends back to the client.
Overlaid Display: The client application, upon receiving the associated annotations from the server, displays them as overlaid virtual content on top of the current camera image. The virtual contents include the name of the painting and the artist as well as a URL pointing to related information on the Internet. The visitor can click on the URL, which will open a web browser and bring up even more information.

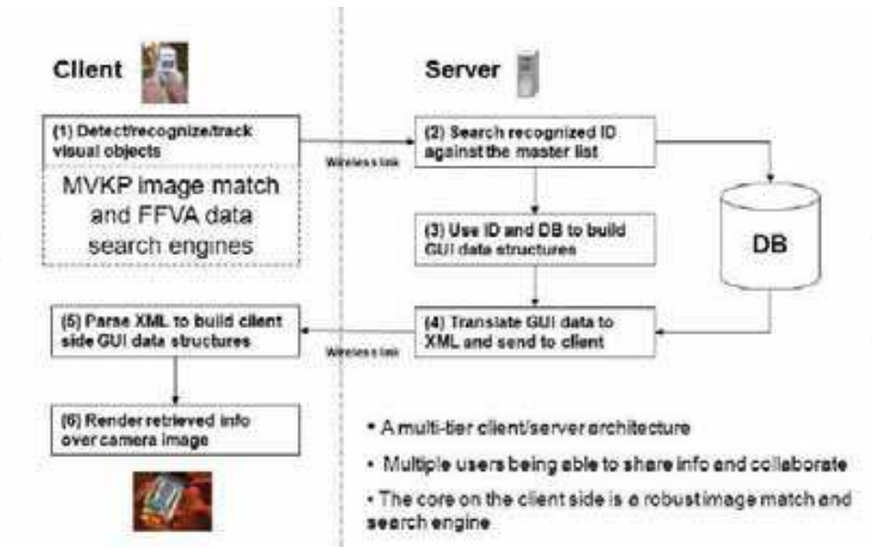


Fig. 19. System overview of the developed augmented exhibition system

Figure 19 illustrates the overall structure of our augmented exhibition system. Two major components: adapted MVKP and remote server are described in the following section.

5.3 Modified MVKP and Information Retrieval

Based on the practical requirements of the application, we chose to use MVKP as the foundation for painting recognition and 3D pose recovery. As described in Section 2, the major advantages of MVKP are: (1) robustness to lighting changes and image noises, (2) invariance to geometric distortion, (3) ability to handle complex conditions like occlusion and cluttered background, (4) sufficient accuracy for pose recovery, (5) particularly good for rigid planar objects like art paintings, (6) real-time, reliable performance, and (7) feature distinctiveness when considering a large feature database. All of these advantages make it ideal for the application of vision-based augmented exhibition system.

We also introduce several important adaptations to the MVKP method to better accommodate the requirements of AR.

In our augmented exhibition system, the outputs of MVKP method are a painting's ID and its 3D pose. The ID is then sent to a remote server through a WiFi LAN connection to retrieve the related complementary information to be displayed as virtual content on top of the painting.

5.3.1 MVKP Adaptations

Originally, the MVKP method was used to find correspondences between two input images, which means: (1) there is no need to detect the existence of interest object and there is no search among multiple objects involved, and (2) thresholds like the one in the distance ratio criteria can be set manually since user knows the query image beforehand. However, we have to make several important adaptations to the original MVKP method to meet the application requirements of augmented exhibition system.

First, for the augmented exhibition system, there can be hundreds of various painting displayed in the museum and some of them are high-textured paintings and some are not. Figure 20 shows two representative paintings. The right painting returns 50% more feature points than the left one after running the same detector. For those painting with low texture, the number of feature points returned by the detectors will also be low, which means the threshold in distance ratio criteria should also be low for it to work properly. Furthermore, there are other factors like feature distinctiveness of a specific painting that also affect the same threshold. And there are thresholds sharing the same dilemma other than distance ratio, for example, those thresholds in RanSac algorithm.

To tackle this problem, we introduce Dynamic Threshold to MVKP method. Take the threshold of distance ratio criteria for example. First we set up a global goal about how many correspondences we'd like to keep after applying the distance ratio filter. At the run time, we periodically (10 times in our experiments) check the number of correspondences the method has found so far, compare it with the global goal, and adjust the threshold accordingly. Experimental results show that, with the help of the automatic adjusted thresholds, for high textured painting we can keep the number of correspondences low and accordingly the computational cost low. For low textured painting we will still have enough correspondences to recognize the painting and recover its pose.

Second, the user of the augmented exhibition system can point the camera to anywhere inside the museum where the query image might contain no painting at all. If there is one, we need to search and decide which painting it is. Based on our experiments, we found the size of the largest consistent correspondences set after running the RanSac is the best criteria to determine which painting, if any, is contained inside the query image.

Last but not the least, for image matching methods based on local features, especially when the query image has significant view point and lighting changes, consistent check methods like RanSac are necessary in order to combine the global information. One problem involving RanSac in AR system is stability. RanSac randomly chooses three correspondences to fit an affine transformation, and for performance consideration, terminates after a limited number of iterations. Therefore, there is no guarantee that the correct affine transform can always be found. Failure of RanSac typically means one or two frames "target lost", which should be avoided for AR applications.

To solve this problem, we assume that when a certain painting is detected in one frame by the system, it is more likely that the same painting will appear in the following frames. In practice, after one painting is detected, the system will focus only on that painting's features in the following frames even after it encounters a RanSac failure. The system will revert to general search mode only when RanSac process fails a certain number of consecutive times. Through this implementation technique, we achieve stable and smooth displays for the augmented exhibition system. Besides this simple technique, every frame of the input is processed independently and there is no tracking technique involved in our current system.

5.3.2 Information Retrieval

Our system is based on a client/server architecture, where the client performs all of the visual processing and recognition and the server maintains a database of all known exhibits and their associated information. When the client positively identifies an exhibit, it sends a unique ID to the server. The server looks up the ID in its database and retrieves the relevant data, which may include the name of the work, the name of the artist, and possibly links to

related web pages. It sends this data back to the client to be displayed over the current camera image of the exhibit.

Fig. 20. Real museum test: (left) with image correspondences and recovered pose displayed under various lightings and viewpoints, and (right) with retrieved information displayed.

The advantage of using a client/server model is that changes to the underlying information can be changed in one place. Whenever a client application recognizes an exhibit that it has not seen recently, it sends a new request to the server to retrieve the latest data. Due to the ready availability of wireless LAN technologies such as WiFi, it is easy to have a mobile client make periodic request to a server. Only one send-receive round trip is needed for each exhibit, so the client and server do not need to maintain a persistent open communications channel.

5.4 Results

We implemented the system and conducted experiments with both synthesized data and real data.

Figure 20 shows sample results on real paints. In our test, we capture videos of those paints from a gallery at the USC School of Fine Art and process the videos in our augmented exhibition system with four painting trained. During the video capture, we intentionally includes many challenging cases like out-of-plane rotation of the camera, moving highlights on the painting, sudden change of illumination, intense shaking of the video camera, etc. Overall, our augmented exhibition system demonstrates fast and reliable performance in a real exhibition environment.

6. Conclusions

In this chapter, we presented several advanced image matching and recognition technologies, of particular emphasis on mobile multimedia applications that are feasible with current or near term technology and applications. The presented work represents the latest state-of-arts in this area. We introduced a high-performance matching technique, MVKP suitable for real-time

multimedia applications. Together with the MVKP, we developed an efficient image search technique that can dramatically improve on previous approaches over hundred-times faster for recognition of objects from a large library of database. Targeting the mobile multimedia applications, we also developed a highly-compact image feature description and matching technique. Finally, we presented an application system, called Augmented Museum Exhibitions that combines the mobile computation, Augmented Reality and image matching/recognition techniques to demonstrate the effectiveness and utility of the presented technologies.

7. Acknowledgments

The presented work are supported by several grants including the Center of Excellence for Research and Academic Training on Interactive Smart Oilfield Technologies (CiSoft), a joint University of Southern California-Chevron-initiative; and the Aerospace Institute for Engineering Research (AIER), a joint research initiative made up of the USC Viterbi School of Engineering, Korea Aerospace University, Inha University, and global corporations Airbus and Korean Air.

This work made use of Integrated Media Systems Center (IMSC) Shared Facilities supported by the National Science Foundation under Cooperative Agreement No. EEC-9529152. Any Opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

8. References

Bay H., Tuytelaars T. & Gool L. V. (2006). Surf: Speeded up robust features, *Proceedings of European Conference on Computer Vision,* Vol. 110, No. 3, pp. 346-359, May, 2006

Bellman R. (1961). Adaptive Control Processes: *A Guided Tour*. Princeton University Press

Ben-Artzi G., Hel-Or H. & Hel-Or Y. (2004). Filtering with Graycode kernels, *Proceedings of the* 17th *International Conference on Pattern Recognition*, vol. 1, pp.556-559, December, 2004

Blott S. & Weber R. (1997). A Simple Vector-Approximation File for Similarity Search in High-dimensional Vector paces. *Technical Report 19, ESPRIT project HERMES (no.9141)*, March 1997

Boffy A., Tsin Y. & Genc Y. (2006). Real-Time Feature Matching using Adaptive and Spatially Distributed Classification Trees. *Proceedings of the British Machine Vision Conference*, 2006

Ciaccia P. & Patella M. (2000). PAC Nearest Neighbor Queries: Approximate and Controlled Search in High-Dimensional and Metric Spaces, *Proceedings of the International Conference on Data Engineering*, 2000

Coors V., Huch T. & Kretschmer U. (2000). Matching buildings: Pose estimation in an urban environment, International Symposium on Augmented Reality, pp. 89 – 92, 2000

Elad M., Hel-Or Y. & Keshet R. (2000). Pattern detection using maximal rejection classifier. *Proceedings of the International Workshop on Visual Form*, pages 28–30, May 2000

Ferhatosmanoglu H., Tuncel E., Agrawal D. & Abbadi A. E. (2000). Vector Approximation based Indexing for Nonuniform High Dimensional Data Sets, *Proceedings of the ACM Conference on Information and Knowledge Management*, 2000

Ferrari V., Tuytelaars T. & Luc V. G. (2005). Wide-baseline Multiple-view Correspondences. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, June 2003.

Friedman J. H., Bentley J. L. & Finkel R. A. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209-226

Grafe M., Wortmann R. & Westphal H. (2002). AR-based Interactive Exploration of a Museum Exhibit, *International Workshop on Augmented Reality Toolkit*, pp. 5 – 9, 2002

Hel-Or Y. & Hel-Or H. (2005). Real-time pattern recognition using projection kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, September 2005

Jeffrey S. B. & Lowe D. (1997) Shape indexing using approximate nearest-neighbor search in high-dimensional spaces, *Proceedings of the Conference on Computer Vision and Pattern Recognition*, June 1997

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

Ke Y. & Sukthankar R. (2004) PCA-SIFT: A more distinctive representation for local image descriptors. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 511-517, June, 2004

Lepetit V., Pilet J. & Fua P. (2004). Point matching as a classification problem for fast and robust object pose estimation. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, June 2004

Lepetit V., Lagger P. & Fua P. (2005) Randomized trees for real-time keypoint recognition. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, June, 2005

Lowe D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Visions,* Vol. 60, No. 2, page numbers (91-110), ISBN 0920-5691

Mikolajczik K. & Schmid C. (2003) A performance evaluation of local descriptors, *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 257–263, June 2003

Mooser J., Wang L., You S. & Neumann U. (2007). An augmented reality interface for mobile information retrieval, *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 2226 – 2229, 2007

Neumann U. & You S. (1999). Natural feature tracking for augmented reality, *IEEE Transactions on Multimedia*, pp. 53 – 64, 1999

Ozuysal M., Lepetit V., Fleuret F. & Fua P. (2006) Feature Harvesting for Tracking-by-Detection. *Proceedings of European Conference on Computer Vision,* May 2006

Schmalstieg D. & Wagner D. (2005). A Handheld Augmented Reality Museum Guide, *IADIS Mobile Learning*, 2005.

Schmid C. & Mohr R. (1997) Local grayvalue invariants for image retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 530–534, May 1997

Simon G., Fitzgibbon A. & Zisserman A. (2000). Markerless tracking using planar structures in the scene, *International Symposium on Augmented Reality*, pp. 120– 128, 2000

Tuncel E., Ferhatosmanoglu H. & Rose K. (2002). VQ-Index: An Index Structure for Similarity Searching in Multimedia Databases, *ACM Multimedia*, 2002

Tuzel O., Porikli F. & Meer P. (2006). Region Covariance: A Fast Descriptor for Detection and Classification, *Proceedings of European Conference on Computer Vision*, pp: 589-600, May, 2006

Weber R., Schek H. & Blott S. (1998). A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Space, *Proceedings of the 24th International Conference on Very Large Data Bases*, pp: 194 – 205, 1998, ISBN:1-55860-566-5

Winn J. & Criminisi A. (2006) Object Class Recognition at a Glance. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, June, 2006

Multimedia

Edited by Kazuki Nishi

ISBN 978-953-7619-87-9

Hard cover, 452 pages

Publisher InTech

Published online 01, February, 2010

Published in print edition February, 2010

Multimedia technology will play a dominant role during the 21st century and beyond, continuously changing the world. It has been embedded in every electronic system: PC, TV, audio, mobile phone, internet application, medical electronics, traffic control, building management, financial trading, plant monitoring and other various man-machine interfaces. It improves the user satisfaction and the operational safety. It can be said that no electronic systems will be possible without multimedia technology. The aim of the book is to present the state-of-the-art research, development, and implementations of multimedia systems, technologies, and applications. All chapters represent contributions from the top researchers in this field and will serve as a valuable tool for professionals in this interdisciplinary field.

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri

Slavka Krautzeka 83/A

51000 Rijeka, Croatia

Phone: +385 (51) 770 447

Fax: +385 (51) 686 166

www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai

No.65, Yan An Road (West), Shanghai, 200040, China

中国上海市延安西路65号上海国际贵都大饭店办公楼405单元

Phone: +86-21-62489820

Fax: +86-21-62489821

IntechOpen