USING DENSELY RECORDED SCENES FOR PLACE RECOGNITION

Tat-Jun Chin, Hanlin Goh and Joo-Hwee Lim

Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
{ tjchin, hlgoh, joohwee }@i2r.a-star.edu.sg

Fig. 1. Densely recording a scene with a video sequence allows us to acquire more visual information about it.

ABSTRACT

We investigate the task of efficiently modeling a scene to build a robust *place recognition* system. We propose an approach which involves densely capturing a place with video recordings to greedily cover as many viewpoints of the place as possible. Our contribution is a framework to (1) effectively exploit the temporal continuity intrinsic in the video sequences to reduce the amount of data to process without losing the unique visual information which describes a place, and (2) train discriminative classifiers with the reduced data for place recognition. We show that our method is more efficient and effective than straightforwardly applying scene or object category recognition methods on the video frames.

Index Terms— Pattern recognition, machine vision, image recognition, image sequence analysis.

1. INTRODUCTION

Our objective is *place recognition* i.e. given an input image we wish to determine the identity of the place(s) contained in the image. By "places" we mean specific scenes, landmarks or buildings. Place recognition is instrumental in Mobile Augmented Reality (MAR) systems [1], where proponents envision an application that allows users to point their camera phone at a place to access more information about it.

Unfortunately the lack of processing power and memory on mobile platforms prohibits the use of 3D models (which are also expensive to construct). Without accurate 3D models for matching, an image-based system can recognize a place in an image only if that place was observed previously under *roughly* the same conditions (e.g. viewpoint, lighting). Consequently data collection becomes complicated: From which (and how many) viewpoints should a place be captured for the purpose of training a sufficiently robust place recognition system? This difficulty is particularly acute for large buildings and landmarks.

To alleviate the problem we propose to densely scan a place with video recordings. Instead of snapping a single image, the collector pans slowly to capture the place in video. With the same number of viewing positions, a set of video recordings can acquire much more information about a place than a set of still images. Fig. 1 illustrates the idea.

However, the deluge of visual information from video sequences presents the significant challenge of effectively processing them. Straightforwardly breaking up the video sequences into image frames and applying techniques from the scene or object recognition domain, which traditionally dealt with individual still images, will not be efficient. We propose a solution, described by the following steps, to process the video sequences to construct a place recognition system:

1. A method to *filter* and *condense* the visual information in the video sequences into a more *compact* form (§2). This is achieved by exploiting the temporal continuity intrinsic in the video sequences captured in the manner of Fig. 1.
2. Modifying an existing object recognition algorithm [2] to receive as inputs the result from Step 1 (§3). This avoids overwhelming the algorithm with the vast amount of visual information from the input video sequences (§3.2).

In our system, queries are received in the form of still images. Video processing is required only in the training phase.

1.1. Related work

Significant progress has been made in the area of *scene* or *object category* recognition (e.g. [2, 3, 4]). Though we emphasize that *place recognition* is slightly different since we aim to recognize specific places (e.g. St. Paul's Cathedral, Big Ben) rather than scene categories (e.g. church, tower), ideas from scene or object category recognition can certainly be applied. Broadly speaking, most of the current methods involve building representations or classifiers for images from features extracted from local keypoints. We modify the method of [2] and apply it to our video processing framework.

One previous research towards densely capturing scenes for place recognition is [5], where "route panoramas" are obtained by line scanning a scene with a camera mounted on a vehicle as it traverses a street in a city. Given a query image, the camera position is recovered (hence, the place is recognized) by finding it's epipole in the route panorama. This can be costly since a RANSAC-like procedure is required for each query [5]. In contrast our method requires only matching between a small number of local features (see §3 and §4).

In [6] video sequences are used as inputs for *querying* in place recognition, where video motion coders from mobile phones are exploited to aid in tracking local features across the video frames. The aim is to quickly identify seen-before and newly emerged keypoints so as to speed-up feature extraction in the *querying* phase. Our method also involves tracking keypoints, but our focus is on the *training* phase, i.e. to train a place recognition system using video sequences as samples, and hence is complementary to the work in [6].

2. PROCESSING VIDEOS OF PLACES

Given a video sequence of a place, based on the SIFT [7] framework we detect scale invariant keypoints in *every* frame and assign descriptors to them. Collectively, a massive number of keypoints are obtained, and we aim to reduce the number of keypoints to consider for subsequent processes. Since our videos are recorded in a slow panning motion, many of the keypoints in a frame will be re-occurances from the previous frames (but in slightly differing views). We can track keypoints across the video sequence to identify the overlaps.

2.1. Finding keypoint overlaps across video frames

Let $\{(\mathbf{x}_i, \mathbf{p}_i)\}$ and $\{(\mathbf{y}_j, \mathbf{q}_j)\}$, $1 \leq i \leq m$ and $1 \leq j \leq n$, be the sets of keypoints detected in two successive frames, with $\mathbf{x}_i$ and $\mathbf{y}_j$ denoting the keypoint positions and $\mathbf{p}_i$ and $\mathbf{q}_j$ their descriptors. Since the images represent two views of the same scene, a homography $\mathbf{H}$ exists between corresponding points:

$$\mathbf{H}\tilde{\mathbf{y}} \times \tilde{\mathbf{x}} = 0 \ , \tag{1}$$

where $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{x}}$ indicate that the homogenous coordinates of $\mathbf{y}$ and $\mathbf{x}$ are used. Our aim is to find the best homography $\mathbf{H}^*$.

To achieve this, we first compute a pairwise similarity matrix using the Euclidean distance between $\mathbf{p}_i$ and $\mathbf{q}_j$. All possible corresponding keypoints between the two frames are identified by considering that a pair of keypoints are matching if the distance of their SIFT descriptors are below a predefined threshold. $\mathbf{H}^*$ is determined as the $\mathbf{H}$ that allows the most number of corresponding keypoints to overlap (i.e. the distance between $\tilde{\mathbf{x}}$ and $\mathbf{H}\tilde{\mathbf{y}}$ is below a certain threshold). We perform a RANSAC procedure to estimate $\mathbf{H}^*$ (refer to [8]).

The process is repeated successively on each frame pair, and overlapping keypoints are accumulated into the same track

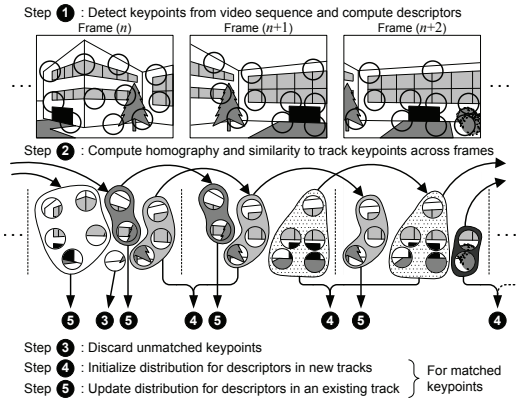

Fig. 2. Finding keypoint overlaps using temporal continuity.

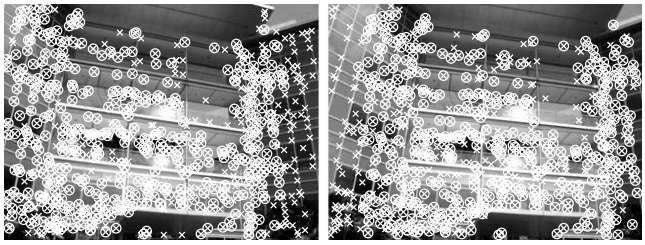

Fig. 3. Overlapping keypoints are re-occurances of the same local feature. In this pair, crosses are detected keypoints, and those with bounding circles indicate that an overlap is found.

(keypoints without matches are simply discarded). Fig. 2 illustrates the idea, and Fig. 3 shows an example result. As an indication of the effectiveness of the approach, a typical 25-frame video sequence in our database produces a total of about 30,000 SIFT keypoints. Among these only about 4,000 are determined as unique by the method. Additionally this can also be considered a *filtering* process, where only keypoints consistently detectable in multiple views are kept.

Our idea is similar in spirit with [9]. However they track each keypoint individually since there could be multiple moving objects of interest in their videos. Our idea is more suited here, since we wish to separate the static background (the place of interest) from dynamic occlusions (the keypoints of which are discarded for not obeying the global homography).

2.2. Estimating descriptor distributions incrementally

We derive a parsimonious representation for the descriptors in a particular track by representing them with a Gaussian distribution. A distribution is more expressive than a simple average, as was done in [9]. We use a diagonal instead of a full covariance since a compact representation is desired. The distributions are updated incrementally so that a large number of descriptors do not have to be maintained. For a particular track, at time t let $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$ be the mean and covariance of

its descriptor distribution of m keypoints. At time $t+1$, if a new descriptor $\mathbf{p}_{t+1}$ is added, $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$ are updated as

$$\boldsymbol{\mu}_{t+1} = \frac{m}{m+1}\boldsymbol{\mu}_t + \frac{1}{m+1}\mathbf{p}_{t+1}, \quad (2)$$

$$\boldsymbol{\Sigma}_{t+1} = \frac{m}{m+1}\boldsymbol{\Sigma}_t + \frac{m}{(m+1)^2}(\boldsymbol{\mu}_t - \mathbf{p}_{t+1})(\boldsymbol{\mu}_t - \mathbf{p}_{t+1})^T \quad (3)$$

The off-diagonal elements of $\boldsymbol{\Sigma}_{t+1}$ are then zeroed. The first two descriptors of the track are used to initialize the mean and covariance matrix. For d-dimensional descriptors, at any point in time only $2d$ unique values, corresponding to the nonzero elements of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, are kept for each track (as opposed to $d(d+3)/2$ for the full covariance case).

3. PLACE RECOGNITION VIA BOOSTING

Given a set of video recordings of different places, we apply the previous steps to obtain a set of descriptor distributions. The next step is to train classifiers (which takes *still images* as inputs) for place recognition, as the following elaborates.

3.1. Boosting descriptor distributions

We apply and modify the AdaBoost method introduced in [2] for our goal. AdaBoost aims to train classifiers of the form

$$H^c(\mathbf{I}) = \textstyle\sum_{t=1}^{T} \alpha_t^c \, h_t^c(\mathbf{I}) \ , \quad (4)$$

where $H^c(\mathbf{I})$ gives the confidence of input image $\mathbf{I}$ containing the c-th place. $H^c(\mathbf{I})$ is obtained by *boosting* a series of weak classifiers $h_t^c(\mathbf{I})$, $1 \le t \le T$, each having weight α_t^c, to become a strong classifier. A weak classifier is defined as

$$h_t^c(\mathbf{I}) = \begin{cases} 1 \ \text{if} \ \min d(\mathbf{v}_t^c, \mathbf{v}_g) \le \theta_t^c, \ \forall \ 1 \le g \le G \\ 0 \ \text{otherwise} \end{cases} , \quad (5)$$

where $\mathbf{v}_t^c$ is the defining feature of h_t^c and $\mathbf{v}_g$ is one of the G local features of $\mathbf{I}$. Function $d(\cdot,\cdot)$ is a *dissimilarity measure* used to compare $\mathbf{v}_t^c$ and $\mathbf{v}_g$ with threshold θ_t^c, and the exact form of $d(\cdot,\cdot)$ is dependent on the forms of $\mathbf{v}_t^c$ and $\mathbf{v}_g$. Given a pre-determined T by the user, the AdaBoost algorithm finds the optimal $\mathbf{v}_t^c$, θ_t^c and α_t^c successively for $1 \le t \le T$. The $\mathbf{v}_t^c$'s are the discriminative features of a set of places.

Before invoking AdaBoost to obtain the h_t^c's, a *minimum dissimilarity matrix* $\mathbf{K}$ must be computed. Let $\mathbf{e}_m$ be the m-th descriptor distribution accumulated from video sequences of the c-th place. We compute for $\mathbf{K}_{mn}$ the Kullback-Leibler Divergence (KLD) of $\mathbf{e}_m$ and $\mathbf{f}_n$ i.e. $D_{KL}(\mathbf{f}_n, \mathbf{e}_m)$, where $\mathbf{f}_n$ is the nearest neighbour of $\mathbf{e}_m$ in the n-th video sequence:

$$\mathbf{f}_n = \arg\min_{\mathbf{f}_i} D_{KL}(\mathbf{f}_i, \mathbf{e}_m) \ , \ \mathbf{f}_i \in \mathcal{F}^n \ . \quad (6)$$

$\mathcal{F}^n$ is the set of descriptor distributions from the n-th video sequence. For Gaussians, the KLD has the closed form

$$D_{KL}(\mathbf{f}_i, \mathbf{e}_m) = \frac{1}{2}\left(\log\frac{|\boldsymbol{\Sigma}_m|}{|\boldsymbol{\Sigma}_i|} + \text{tr}(\boldsymbol{\Sigma}_m^{-1}\boldsymbol{\Sigma}_i) + (\boldsymbol{\mu}_m - \boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{\mu}_m - \boldsymbol{\mu}_i) - d\right), \quad (7)$$

where $(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$ and $(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ respectively characterize the distribution of $\mathbf{e}_m$ and $\mathbf{f}_i$. Given a completed $\mathbf{K}$, we can apply the AdaBoost algorithm [2] to choose among the $\mathbf{e}_m$'s to form the set of T discriminative descriptor distributions $\mathbf{v}_t^c$.

In our application, place recognition is performed on still images. We use the Mahalanobis distance for $d(\cdot,\cdot)$ in Eq. (5):

$$d(\mathbf{v}_t^c, \mathbf{v}_g) = (\boldsymbol{\mu}_t^c - \mathbf{v}_g)^T(\boldsymbol{\Sigma}_t^c)^{-1}(\boldsymbol{\mu}_t^c - \mathbf{v}_g) \ , \quad (8)$$

where $(\boldsymbol{\mu}_t^c, \boldsymbol{\Sigma}_t^c)$ define the distribution of $\mathbf{v}_t^c$ and $\mathbf{v}_g$ is one of the G keypoint descriptors (vectors) in the query image. The AdaBoost algorithm is modified accordingly to retain only the third term of Eq. (7) to compute the threshold θ_t^c.

How we compute $\mathbf{K}$ and define the weak classifiers constitute the major differences between our work and the original idea in [2], where $\mathbf{e}_m$ and $\mathbf{f}_n$ are simply descriptors (vectors) with $\mathbf{K}_{mn} = \|\mathbf{e}_m - \mathbf{f}_n\|$, and $d(\mathbf{v}_t^c, \mathbf{v}_g) = \|\mathbf{v}_t^c - \mathbf{v}_g\|$.

3.2. The benefits of exploiting temporal continuity

The benefits of video processing, as opposed to individual treatment of each frames, is obvious by observing matrix $\mathbf{K}$. Let the size of $\mathbf{K}$ be $M \times N$. M is the total number of features from the video sequences of the positive class. By straightforwardly applying [2], i.e. by considering each keypoint from each frame of the positive class individually (there is more than one video sequence per class), M would be a massive number: For our database (see §4), M can reach 100,000! In constrast, by identifying keypoint overlaps (as in §2), M can be reduced to a much more manageable value of 12,000.

Of equal importance is the value of N, which is the total number of "samples" of places in the database. Using our framework, each video sequence is a sample (hence $N < 200$ for our database), whereas by directly applying [2], every frame in the video sequences is a sample, and N can reach up to 5000. Since the size of $\mathbf{K}$ directly impacts the efficiency of the AdaBoost algorithm (refer to [2]), by exploiting temporal continuity as described in §2, the AdaBoost procedure can be performed efficiently on a database of video sequences.

4. EXPERIMENTAL RESULTS

First, we describe our video collection procedure. Places of interest (mainly large buildings) in our campus were captured in video in the manner described in §1. The length of the videos range from 1s to 10s depending on the size of the place. Three video sequences are recorded at 30 fps from each place. Fig. 4 illustrates the types of places we have collected. We recorded 44 different places which amount to about 21,000 frames or 1.5GB of data. At each place, a separate query set of *still images* (collectively 1349 images) were also captured in an *unconstrained* manner on different days.

Several experimental settings were investigated to examine the performance of the proposed method:

Fig. 4. Samples of several places in our database.

1. Directly apply [2] on the video sequences by treating each video frame individually as single images.
2. Randomly sample a few frames from each video and directly apply [2]. This is to simulate the imaging of a place from only a few pre-selected viewpoints.
3. Apply the procedure in §2 but represent each track with just the simple average of their descriptors, as in [9]. Train classifiers using [2] with the descriptor averages.
4. Process the video sequences according to §2 and train classifiers with the method detailed in §3.

In all settings the value of T in Eq. (4) is set to 100. We implemented these settings on a randomly chosen subset of our database with 10 places. A small subset is used first since Setting 1, as explained in §3.2, can require massive amounts of memory and time to perform. Even on this subset, Setting 1 required a few days of training whereas Setting 4 required only a few hours— a clear evidence of the improved efficiency. We test the resulting classifiers on the query set. Fig. 5 illustrates the results in terms of Receiver Operating Characteristic (ROC) curves. The ROC curves were obtained by varying the threshold of the overall classifier defined in Eq. (4). Expectedly Setting 1 performed better than Setting 2, confirming that having observed a place from more viewpoints, we can produce more robust classifiers. Secondly, Setting 4 marginally outperformed Setting 3, indicating that it is beneficial to use descriptor distributions as oppposed to simple averages [9] to train classifiers. Additionally, since Setting 4 outperformed Setting 1, it can be concluded that, besides the gain in efficiency in training, the prior filtering for more consistently detectable keypoints given by our method in §2 can produce a more robust and accurate place recognition system.

Finally, we repeat Setting 4 on the whole dataset to examine the scaling capability of the proposed method. The resulting classifiers produced an Equal Error Rate (EER) of about 8% (refer to the corresponding ROC curve in Fig. 5) which is reasonably accurate considering that query images taken in an unconstrained manner must be classified into 44 classes. We are also encouraged by the fact that in real MAR systems, by exploiting GPS priming only a small number of places (< 10) need to be recognized within a locality [1].

5. CONCLUSIONS

We proposed a method to condense the local features of video recordings of places into a more compact form. The objective is to reduce the amount of data to process when training classifiers for a place recognition system. We also modified extensively an existing object recognition algorithm [2] for our purpose. Experimental results show that, apart from allowing training of classifiers to be more feasible, our method produced a more accurate and robust place recognition system compared to applying [2] directly on video sequences.

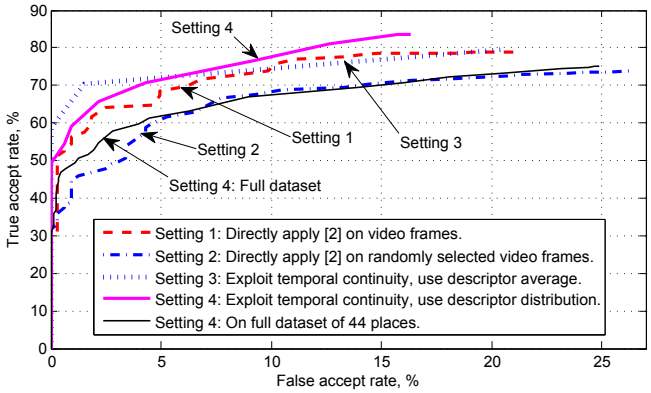

Fig. 5. Results of place recognition experiment (ROC curves).

6. REFERENCES

[1] E. Jonietz, "TR10: Augmented reality," Tech. Rep., MIT Technology Review, 2007.

[2] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer, "Generic object recognition with boosting," *PAMI*, vol. 28, no. 3, pp. 416–431, 2006.

[3] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *CVPR*, 2003.

[4] F.-F. Li and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *CVPR*, 2005.

[5] S.M. Khan, F. Rafi, and M. Shah, "Where was the picture taken: Image localization in route panoramas using epipolar geometry," in *ICME*, 2006.

[6] G. Takacs, V. Chandrasekhar, B. Girod, and R. Grzeszczuk, "Feature tracking for MAR using video coder motion vectors," in *ISMAR*, 2007.

[7] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, 2004.

[8] R. Hartley and A. Zisserman, *Multiple view geometry in comp. vision*, Cambridge Uni. Press, 2nd edition, 2003.

[9] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *CVPR*, 2003.