Keypoint Induced Distance Profiles for Visual Recognition

Tat-Jun Chin* and David Suter
School of Computer Science,
The University of Adelaide, South Australia
{tjchin, dsuter}@cs.adelaide.edu.au

Abstract

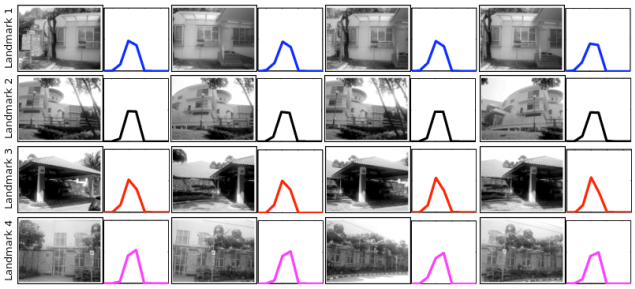
We show that histograms of keypoint descriptor distances can make useful features for visual recognition. Descriptor distances are often exhaustively computed between sets of keypoints, but besides finding the k-smallest distances the structure of the distribution of these distances has been largely overlooked. We highlight the potential of such information in the task of particular scene recognition. Discriminative scene signatures in the form of histograms of keypoint descriptor distances are constructed in a supervised manner. The distances are computed between properly selected reference keypoints and the keypoints detected in the input image. The signature is low dimensional, computationally cheap to obtain, and can distinguish a large number of scenes. We introduce a scheme based on Multiclass AdaBoost to select the appropriate reference keypoints.

The resulting system is capable of handling a large number of scene classes at a fraction of the time required for exhaustively matching sets of keypoints. This supports supports a coarse-to-fine search strategy for approaches reliant on keypoint matching. We test the idea on 3 datasets for particular scene recognition and report the obtained results.

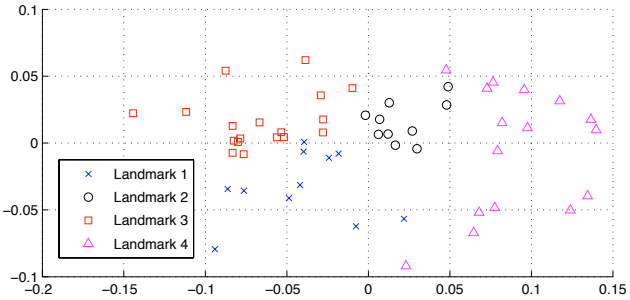
1. Introduction

Methods for detecting local interest points or keypoints in images have become reliable workhorses in many scene and object recognition tasks. Keypoints are designed to be consistently detectable under large perspective, scale and illumination variations. A keypoint is usually endowed with a descriptor which serves as a robust feature to describe the local patch which motivated the keypoint detection. Due to their effectiveness, keypoint detectors are frequently used as low-level feature extractors in popular visual recognition paradigms such as the bag-of-words representation and wide-baseline-matching. Examples include [17, 33].

*Part of this work was done when Chin was at the Institute for Infocomm Research, Singapore.



(a) Images of landmarks and distance profiles (actual results shown).



(b) The distances profiles projected onto the first-2 principal components.

Figure 1. In (a) SIFT keypoints are detected on images of 4 landmarks and a suitably chosen reference keypoint (not shown here) is used to generate 10 bin-histograms of descriptor distances or "distance profiles". It can be seen in the PCA space of (b) that the representation is sufficiently distinct and robust for distinguishing the landmarks which were captured from varying viewpoints.

Often when utilizing keypoints, the computation of distances between the descriptors of sets of keypoints is carried out. For example, when quantizing the set of keypoints of an image $\mathcal{A}$ into a bag-of-words representation using a visual vocabulary $\mathcal{B}$ (each cluster center is essentially a descriptor), or when matching the sets of keypoints $\mathcal{A}$ and $\mathcal{B}$ of two adjacent scenes for image registration. However, in such scenarios, usually only the minimum value of a set of distances receives attention, e.g. find the cluster center in the visual vocabulary closest to a keypoint, or find the closest matching keypoint pair to establish a correspondence.

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

The rest of the distances, although having been computed already, are often ignored. We argue that the "side information" which occurs in the main feature extraction pipeline can serve useful purposes as well.

We demonstrate the potential of this idea on the task of "particular scene" recognition by introducing a novel image representation in the form of a histogram of keypoint descriptor distances. The distances are computed between a properly selected reference keypoint and the keypoints detected in an image. We refer to the resulting histograms as "distance profiles." Fig. 1 demonstrates the idea. Note that the reference keypoint did not arise from the input images, but is a member of a previously accumulated keypoint library, i.e. the discriminative keypoints of the set of scenes.

The premise is that a suitably chosen reference keypoint can induce noticeably different distance profiles for different scenes. Moreover, if more or less the same collection of keypoints are detected in a particular scene despite varying imaging conditions, the generated distance profiles can act as a robust scene signature. In this paper, we propose to apply a multiclass extension of AdaBoost to select useful reference keypoints from scene images in a database. A classifier which takes the distance profile of an image as input is also constructed. Given a new image, the classifier is very fast to evaluate, and the output presents a ranked list of possible matches in the scene database.

Our work is part of the growing trend of harnessing conventionally ignored side information to aid in the main vision task. Other examples include [24], where if we realize that the k-nearest distances are computed anyway in pursuit of the nearest visterm, we can actually quantize a keypoint to the k-nearest visterms as a form of query expansion. Another example is [6], where instead of just finding the best match for a test keypoint among a set of library keypoints, the list of 0-1 matching results against the library keypoints can be used as a descriptor for the test keypoint.

The rest of the paper is organized as follows: After surveying related work in Section 2, we introduce the proposed image representation and show how to obtain it in Section 3. In Section 4 we explain how to build an ensemble of CART trees grown from the distance histograms. The proposed method is then applied on three datasets for particular scene recognition and the results are reported in Section 5. Finally we discuss and draw conclusions in Section 6.

2. Related Work and Motivation

"Particular scene" recognition differs from "scene category" recognition in that the former is interested in finding the precise location or place from which the input image is taken, while the the latter aims to identify the rough category or type of scene captured in the input image. Particular scene recognition or "place" recognition has been widely studied in robotics (e.g. [29, 25, 38]) whereby the goal is to use visual information to establish the position of a robot in a global reference frame. Place recognition also receives attention more widely (e.g. [10, 35, 28]), where the objective is usually to "geo-tag" or augment images of places and landmarks with further information.

It has been observed [35] that "scene category" recognition works best using features with high invariance which can smooth out intraclass differences. For example, current research efforts concentrate on the bag-of-visterms representation [26, 23, 36] and kernels between sets of keypoints for *partial* matching [11, 14, 16]. On the other hand, "particular scene" recognition thrives on features with high discriminative power [35], such as matching SIFT [18] keypoints. Examples include [29, 9, 21, 25, 13, 10, 7]. These approaches usually contain a learning step where the more discriminatory keypoints for a particular scene are extracted to build scene-specific models or classifiers. To name a few, these include Support Vector Machines [25], local entropy estimations [10] and AdaBoost [21, 7]. Nonetheless, given a new image, evaluating the models or classifiers will generally involve steps equivalent to keypoint matching.

Other methods not relying on interest points for particular scene recognition exist. An approach to detect windows in building facades is proposed in [1], and place recognition can be achieved assuming that the buildings in an area have unique window types and configurations. This is closely related to [28] where repetitive facades of urban buildings are considered as unique textures which can be used for recognition. A new image feature based on the PCA of Census Transform histograms is proposed in [35] and is found to be effective for the task of particular scene recognition. Recently, image epitomes [20] are used to concisely capture the appearance and geometric structure of a physical environment. It is claimed that the representation supports very efficient and robust location recognition. Another work [2] exploited GPS information to aid in building recognition.

Although highly effective, keypoint matching in general is a computationally expensive process. Many ideas have been proposed to speed up keypoint matching. Keypoint descriptors can be indexed in a kd-tree data structure with a best-bin-first retrieval strategy [4] to enable fast approximate nearest neighbour operations. Recently, randomization for kd-trees have been proposed [32] for further speed-ups at the expense of more storage space. Ke and Sukthankar [12] performed PCA on the orientation histograms to arrive at more concise SIFT descriptors with improved accuracy. Considerable speed-up is achieved by introducing a training stage and casting keypoint matching as a classification problem [15, 22]. In these methods a new keypoint is matched directly based on the appearance of its local patch without building a descriptor.

Although the methods above significantly increased the efficiency of keypoint matching, they invariably deal with

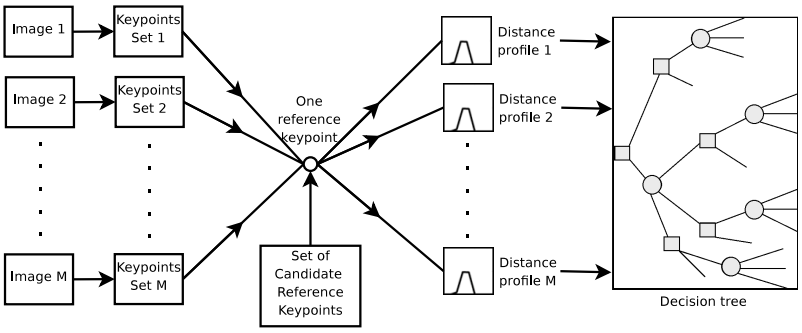


Figure 2. A reference keypoint is picked from a set of candidates and is used to induce distance profiles from the training images. Based on the class/scene label of each image, a decision tree is grown from the distance profiles. Boosting is used to simultaneously select a set of useful reference keypoints and combine their resulting decision trees into a classifier for particular scene recognition.

a single or a small number of objects or scenes (with the exception of [15] where multiclass extension is possible but not thoroughly explored). When there are many classes to be matched, repeating the above methods multiple times is computationally inefficient and also detrimental in terms of memory usage since multiple classifiers or data structures are repeatedly invoked or accessed.

We propose a method to improve the scalability of keypoint matching for *particular scene recognition*. Our idea is to perform true multiclass classification based on using distance profiles as informative scene signatures. If induced by appropriately selected reference keypoints, the distance profiles can distinguish multiple scenes, thus supporting a multiclass framework; refer to Fig. 1. From the distance profiles, we grow CART trees and combine them in a boosted ensemble, producing a single overall multiclass classifier. Fig. 2 illustrates the general idea. Given a new image the classifier outputs a ranked list of scene matches. Guided by the list a more precise search based on keypoint matching proceeds. Besides improving the scalability of particular scene recognition, we demonstrate that distance profiles and side information in general can be highly useful.

Our work bears the most similarity with [34, 27] where a simple global image signature obtained from compressed gabor filter outputs, called "gist" of the scene, is used to quickly retrieve a list of possible results for a query image. A more refined search is then conducted on the retrieved list for accurate recognition. While the gist can be used to separate widely divergent image categories, e.g. indoor, outdoor, nature, mountain, it will most likely fail in particular scene recognition since the images invariably belong to the same rough category, i.e. outdoor with a building centered. This is not an issue in [34, 27] since they are dealing with *category level* recognition. In contrast, our work builds image signatures from detected keypoints in a supervised manner. The feature is used as an input to a fast multiclass classifier which outputs a short list of possible places on which more careful discrimination can proceed.

3. Generating Distance Profiles

Let I be an image and $\mathcal{X} = \{x_i\}_{i=1,\ldots,F}$ be the set of F local features of I, e.g. the x_i's are the SIFT descriptors of keypoints detected in I. Let ϕ be the feature of a reference keypoint. The set of distances induced by ϕ on $\mathcal{X}$ is

$$\Delta = \{d_i\}_{i=1,\ldots,F} \ , \tag{1}$$

$$\text{where} \quad d_i = \|x_i - \phi\|^2 \ . \tag{2}$$

The distance profile $P_{I,\phi}$ of image I as induced by ϕ is obtained by computing a histogram on Δ. More formally,

$$P_{I,\phi}(k) = \#\{d_j \,|\, d_j \in \Delta, \alpha_k \leq d_j < \beta_k\}/F \ . \tag{3}$$

Respectively α_k and β_k are the lower and upper edges of the k-th bin. Note that the histogram is normalized by F. The idea is that when ϕ is suitably selected, $P_{I,\phi}$ can serve as an effective representation for the scene in image I.

Without a priori restricting the type of feature in $\mathcal{X}$ or resorting to ad-hoc scaling, it is hard to guarantee that the d_i's always stay within bounds of the binning range such that $P_{I,\phi}(k)$ are consistently scaled for all I and ϕ. Fortunately, for methods such as SIFT [18] and SURF [3] where the descriptors are normalized histograms by definition, this is not a concern as long as ϕ is also an instance from the same feature space, i.e. Eq. (2) will be bounded within $[0, 2]$.

In the following "keypoint" and "descriptor" are used interchangeably since we ignore the spatial position of local features. Unless stated otherwise both words refer to the descriptor vector of an interest point.

4. Training an Ensemble of Distance Profiles

A dataset of M images of N particular scenes is first collected as $\{(I_i, y_i)\}_{i=1,\ldots,M}$, where $y_i \in \{1, \ldots, N\}$ is the scene label of I_i. We assume that M_n sample images are available for the n-th scene, i.e. $M = \sum_{n=1}^{N} M_n$. This is augmented with sets of keypoints $\{\mathcal{X}_i\}_{i=1,\ldots,M}$ detected in each image. Note that the size of each $\mathcal{X}_i$ depends on

the visual contents of I_i. Methods such as [25, 21, 10, 7] are then applied on the dataset to extract the informative keypoints $\mathcal{B}_1, \ldots, \mathcal{B}_N$ of each scene, where

$$\mathcal{B}_n \subset \bigcup_{s=s_1,\ldots,s_{M_n}} \mathcal{X}_s \ , \ s_p \neq s_q \text{ for } p \neq q \ , \tag{4}$$

and $y_s = n$ for all s. Depending on the specific method, scene-specific models or classifiers are built using $\mathcal{B}_1, \ldots, \mathcal{B}_N$. However, given a set of keypoints $\mathcal{A}$ at runtime, invoking the methods involves a process equivalent to exhaustively matching $\mathcal{A}$ to $\mathcal{B}_1, \ldots, \mathcal{B}_N$.

In this paper we select the reference keypoints from the superset $\mathcal{B} = \mathcal{B}_1 \cup \cdots \cup \mathcal{B}_N$ to generate distance profiles. This is motivated by two reasons: (1) Keypoints in $\mathcal{B}$ have been determined to be highly discriminative, i.e. existing in a scene but not others, thus they have a higher probability in inducing useful signatures, and (2) there is no need to store two different libraries of keypoints if a coarse-to-fine search strategy is adopted since those in $\mathcal{B}$ support both component recognition algorithms. The following subsections describe how to carry out the selection of reference keypoints.

4.1. Growing Weak Learners

Let there be R candidate reference keypoints ϕ_j, where $j = 1, \ldots, R$ and ϕ_j is from the set $\mathcal{B}$. As a preprocessing step, for all ϕ_j we generate K-bin distance profiles for the training images according to Section 3, yielding

$$\{P_{I_i,\phi_j}(k)\}_{i=1,\ldots,M,\ j=1,\ldots,R,\ k=1,\ldots,K} \ . \tag{5}$$

Without forgetting that each distance profile is a histogram of values, we abbreviate the above to $P_{i,j}$. For each candidate ϕ_j, we grow CART trees [5] which are well suited for multiclass classification to classify $P_{i,j}$ based on labels y_i. The trees will be used as weak learners in boosting.

Tree nodes are split along the dimensions (bins) of the distance profiles. At each node, the best split is defined to be the one which allows the maximum reduction in Gini impurity. The impurity at node e is defined as

$$im(e) = \sum_{p \neq q} f(e,p) f(e,q) = 1 - \sum_{n=1,\ldots,N} f(e,n)^2 \ , \tag{6}$$

where $f(e,.)$ is the probability mass function of class labels at node e. This is approximated by an empirical value based on the samples which arrive at e, e.g. at the root node,

$$f(0,n) = \frac{\sum_{i=1}^{M} \delta(y_i - n)}{M} \ , \tag{7}$$

where δ is the Kronecker delta function. For a candidate split at node e, the reduction in impurity achieved is

$$\Delta_{im}(e) = im(e) - im(e_l) - im(e_r) \ , \tag{8}$$

where e_l and e_r are the left and right child nodes of e according to the split. Searching for the optimal split at each node requires $\mathcal{O}(KM)$ computations, therefore it is beneficial to restrict K to a small value. Experiments in Section 5 show that $K \in [5, 15]$ is sufficient for good performance.

For each node, we adopt the usual stopping criterion of halting subsequent splits when no further reduction in impurity is achievable. In the interest of creating a fast scene recognition system, we also impose an additional criterion which restricts a tree to a maximum depth— a value we set to $\lceil \log_2 N \rceil$, i.e. grow the smallest tree possible for separating N classes. For example, for $N = 10$, the maximum depth is 4. Since we do not perform pruning for the trees, this also prevents overfitting.

4.2. Selecting Reference Keypoints with Boosting

For each reference keypoint ϕ_j, the previous operations train a tree classifier $\tau_j(\cdot)$ which takes a distance profile as input and predicts one of the possible N class labels for it. The trees form a pool of weak learners which, along with the dataset, we subject to a boosting procedure. Boosting will select and combine a T number of trees (e.g. $T = 100$) based on empirical accuracy to form a strong overall classifier. Naturally this also yields a list of reference keypoints useful for inducing highly discriminative distance profiles.

The SAMME algorithm [37] was proposed to extend the popular AdaBoost algorithm [8] to multiclass problems. It overcomes the limitation of AdaBoost which requires each *weak* learner to have more than 50% empirical accuracy— a relatively easy task in binary classification but a tall order in an N class problem. Table 1 lists the SAMME algorithm used in conjunction with trees as weak learners trained from distance profiles for particular scene recognition. A major difference between SAMME and AdaBoost is the addition of the $log(N-1)$ term in Eq. (10) which allows a weak learner's accuracy, given by $(1 - \epsilon^t_{j*})$, to be more than just $1/N$ (i.e. random guessing for an N class problem) to ensure a non-negative contribution α^t. This seemingly heuristical modification makes the algorithm correspond to fitting a forward stagewise additive model using a multi-class exponential loss function [37].

We fit each CART tree τ_j to the sample weights (Step 3 in Table 1) by reweighting the labels associated with the leaf nodes of τ_j. More specifically, at a particular iteration t let w_i reflect the current set of weights. When fitting tree τ_j, let $\mathbb{I}(P_{i,j}|\psi)$ indicate whether a particular leaf node ψ of τ_j contains the distance profile of sample I_i, i.e. $\mathbb{I}(P_{i,j}|\psi) = 1$ if $P_{i,j}$ arrived at ψ, otherwise 0. The label assigned to ψ is

$$l^* = \arg\max_l \sum_{i=1}^{M} w_i \cdot \mathbb{I}(P_{i,j}|\psi) \delta(y_i - l) \tag{12}$$

or $\tau^t_j(P_{i,j}) = l^*$ if $\mathbb{I}(P_{i,j}|\psi) = 1$. In other words, assign

1. Initialize M sample weights $w_i = 1/M$.
2. **For** $t = 1, \ldots, T$ **do**
3. Fit each tree τ_j to the weights w_i (see text).
4. For each fitted tree τ_j^t, evaluate error

$$\epsilon_j^t = \sum_{i=1}^{M} w_i(1 - \delta(\tau_j^t(P_{i,j}) - y_i))/\sum_{i=1}^{M} w_i \ . \quad (9)$$

5. Find $j^* = \arg\min_j \epsilon_j^t$ and set $h^t = \tau_{j^*}^t$.
6. Store current weights w_i along with h^t.
7. Compute t-th weak learner contribution

$$\alpha^t = \log \frac{1 - \epsilon_{j^*}^t}{\epsilon_{j^*}^t} + \log(N-1) \ . \quad (10)$$

8. Update sample weights as

$$w_i \leftarrow w_i \cdot \exp^{\alpha^t(1-\delta(\tau_{j^*}^t(P_{i,j^*})-y_i))} \ . \quad (11)$$

9. **end for**
10. Output overall classifier with T weak learners h^t (see Section 4.3).

Table 1. Multiclass AdaBoost on distance profiles.

label l^* to distance profiles $P_{i,j}$ which descend into ψ. This means that samples wrongly classified in the previous iteration have more say in determining the leaf labels, i.e. the sample weights affect the behaviour of the fitted tree τ_j^t.

4.3. An Ensemble of Trees for Scene Recognition

For the task of multiclass particular scene recognition, we design a novel ensemble classifier based on the selected weak learners. Instead of adopting a winner takes all principle as in a conventional boosted ensemble, we allow each component weak learner to cast weighted votes for more than one class label. This contributes towards the stability of the classifier in a multiclass setting especially when the number of classes N is large.

Let function $\mathbb{C}(h^t, P)$ return an N-vector, where the n-th component is 1 if the leaf node of h^t into which distance profile P has descended contains training samples[1] of class n, and zero if otherwise. Then, the proposed classifier ensemble outputs an N-vector L defined as

$$L = \sum_{t=1}^{T} \alpha^t \cdot \mathbb{C}(h^t, P)/\sum_{t=1}^{T} \alpha^t \quad (13)$$

Note that for each t, distance profile P is generated from a testing image according to the reference keypoint associated with h^t. The form of the ensemble differs from the traditional boosted combination where each component h^t gives only one label $h^t(P) \in \{1, \ldots, N\}$ as output.

The result in L represents the confidence value of matching a query image to each of the N scenes in the database. We sort the values in L decreasingly to obtain a ranked list of possible scene given a query image. Scenes at the n-th percentile and above can then be subjected to a more careful analysis, e.g. carry out discriminative keypoint matching for scenes at the top 10% of the ranked list.

5. Results

We performed experiments to investigate the ability of the proposed method in selecting useful reference keypoints for particular scene recognition. An attribute of the datasets we chose is the availability of a large number of scene classes ($N > 40$). This allows us to test the scalability of the resulting classifiers against the number of scene classes. The experiments below were implemented in Matlab.

Tourist Sights Graz 60 (TSG-60). The TSG-60 dataset[2] contains images of 60 tourist spots in the city of Graz. The images are mainly frontal facades of buildings of touristic interest. Each building was taken in 3 views corresponding roughly to left, frontal and right. The images are originally in colour, of size 320×240 pixels, in both portrait and landscape orientation, and compressed in JPEG format. We convert the images to grayscale for our experiments.

We partition the dataset into a training and testing set. This was done by randomly choosing, for each building, one image for testing and keeping the other two for training. Several algorithms/settings (to be described shortly) for particular scene recognition are then applied with the results recorded. The steps of partitioning, training and testing are repeated five times and we present the averaged results.

First, SURF keypoints [3] are detected in all images. On the training images, we run the algorithm of [21] to uniformly select 100 discriminative keypoints for each scene, i.e. obtain $\mathcal{B}_1, \ldots, \mathcal{B}_{60}$. Following [21] a binary image classifier is constructed for each scene based on the discriminative keypoints (see [21] for details). Each testing image is given the label of the binary classifier which returns the highest confidence. There are thus 6000 candidate reference keypoints which we used to generate 15-bin distance profiles for the training set. These are then subjected to the proposed algorithm described in Section 4 where 100 reference keypoints or weak learners are chosen and boosted. For each testing image, we retrieve the top 10% of the ranked list (i.e. 6 scenes only) given by the boosted ensemble of trees. The result is deemed correct if the ground truth label exists in the shortlist. Finally, we combine the previous two algorithms to yield a third method. This is done by invoking the classifiers of only the 6 shortlisted scenes and comparing their output. Instead of giving a ranked list, this method assigns a single label to a testing image. Table 2 presents the obtained results.

The image classifiers provide a benchmark recognition

[1] More precisely, distance profiles of training samples.

[2] Available at http://dib.joanneum.at/cape/tsg-60/.

#	Method/Setting	% correct
1	Image classifiers [21]	98.33
2	Distance profiles (top 10% shortlist)	95.00
3	Distance profiles + Image classifiers	95.00

Table 2. Recognition results on the TSG-60 dataset.

accuracy of 98.33%. The results also show that in 95% of the queries, the correct label exists in the top 10% shortlist returned by the tree ensemble. The combination of the two algorithms did not degrade the performance as a correct rate of 95% was also returned. It does not immediately seem that using distance profiles for particular scene recognition is useful. The true advantage, however, lies in a big increase in computational efficiency as we show next.

#	Method/Setting	Avg time (s)
1	Image classifiers [21]	0.9047
2	Img classifiers + kd-tree	0.5049
3	Distance profiles (10% shortlist)	0.2260
4	Dist. prof. + Img classifiers	0.3158
5	Dist. prof. + Img class. + kd-tree	0.2759

Table 3. Testing duration per image on the TSG-60 dataset (excluding keypoint/descriptor computation time). Note that the time for methods 3–5 includes distance profile generation.

Table 3 depicts the average duration of processing a testing image of various algorithms/settings. Invoking the image classifier requires a step equivalent to nearest neighbour matching (see [21] and Section 4), and when done via exhaustive searches it requires $\approx 0.9s$ for all 60 scenes. This can be improved to $\approx 0.5s$ by indexing and searching each set of discriminative keypoint in a kd-tree. Obtaining the top 6 scenes (top 10%) using the proposed method requires only $\approx 0.23s$ for 60 scenes (and as we shall see in subsequent experiments, this duration is quite independent on the size of the dataset). Invoking the corresponding 6 image classifiers with and without kd-tree on the retrieved list requires a total of only $\approx 0.28s$ and $\approx 0.32s$. This represents a three-fold improvement in speed with only minor impact to the recognition accuracy.

Zurich Buildings Database (ZuBuD). We repeat the previous experiment on a much larger dataset. The ZuBuD dataset[3] comprises of 201 buildings of historical or architectural interest in Zurich. Imaging conditions vary in viewpoint, season (lighting) and existence of occlusions. Each building has 5 images in the training set (1005 images in total) while the testing set contains 115 images. The sheer size of the dataset (in terms of the number of scene classes) and the much larger variability of imaging conditions make ZuBuD a more difficult dataset than TSG-60. Nonetheless, we retain the same parameters from the previous experiment: select 100 discriminative SURF keypoints for each building using [21], input the 20,100 candidate reference keypoints into the proposed method, generate 15-bin distance profiles, select 100 weak learners for the tree ensemble (despite a much larger number of classes), and retrieve only the top-10% from the ranked list (i.e. top 20 matches) given by the tree ensemble. Table 4 presents the our results along with several other reported results for comparisons.

[3] Available at http://www.vision.ee.ethz.ch/datasets/index.en.html.

#	Method/Setting	% correct
1	Random subwindows [19]	95.65
2	Image classifiers [21]	92.17
3	Distance profiles (top 10% shortlist)	87.83
4	Distance profiles + Image classifiers	84.35
5	Indexing local appearance [31]	40.87

Table 4. Recognition results on the ZuBuD dataset.

It can be seen that the accuracy varies greatly depending on the method. The best result of 95.65% was achieved by [19] while a poor 40.87% was given by [31]. Using distance profiles and image classifiers gave a competent result of 84.35%. The performance of the proposed method can certainly be further improved, for example, by coupling methods 1 and 3 in Table 4, or by using a larger tree ensemble size. We do not explore these options here, since the emphasis is on the potential gain in computational efficiency from using distance profiles.

#	Method/Setting	Avg time (s)
1	Image classifiers [21]	2.2017
2	Img classifiers + kdtree	1.3324
3	Distance profiles (10% shortlist)	0.2719
4	Dist. prof. + Img classifiers	0.4909
5	Dist. prof. + Img class. + kdtree	0.4032

Table 5. Testing duration per image on the ZuBuD dataset (excluding keypoint/descriptor computation time). Note that the time for methods 3–5 includes distance profile generation.

The average testing time for each method is presented in Table 5. Invoking tree ensembles based on distance profiles and executing shortlisted scene classifiers only managed to produce a more than four-fold improvement in runtime computational speed. Most importantly, we highlight that processing an image with the proposed method requires approximately the same time as for the TSG-60 dataset (compare methods 3 in Tables 3 and 5) despite a large increase in the number of scene classes (from 60 to 201). This is due to the inherent ability of the tree ensembles to handle multiple classes. Of course, a major enabling factor is the informative scene signatures that can distinguish multiple scenes induced by the chosen reference keypoints.

Campus Dataset. We built a more challenging dataset to test our ideas. Images of buildings of interest around our campus were taken on different days. Large variations in viewing angles and positions were deliberately introduced.

In total 44 buildings were captured with the number of images amounting to about 20,000. A separate testing set of 1,400 images was also collected. We call this the "Campus" dataset. Fig. 3 shows a few samples.

Figure 3. Sample images from the Campus dataset.

We use SIFT keypoints for the Campus dataset. Similar to the previous experiments, we select 100 disriminative keypoints for each scene using [21], subject the 4,400 candidate reference keypoints to the proposed algorithm, generate 15-bin distance profiles, select 100 weak learners for the tree ensemble, and return only the top-10% of the ranked list (top 5 matches) from the tree ensemble. Table 6 depicts the obtained recognition results.

#	Method/Setting	% correct
1	Image classifiers [21]	80.06
2	Distance profiles (top 10% shortlist)	91.55
3	Distance profiles + Image classifiers	81.02

Table 6. Recognition results on the Campus dataset.

The proposed method gave comparable performance to the baseline method of the image classifiers, where both scored an $\approx$ 80% correct rate. In the Campus dataset, however, the retrieval rate of the tree ensemble at $\approx$ 92% is higher than the baseline accuracy. We suspect this is due to the existence of more diversified buildings in the dataset (e.g. skyscrapers, residential houses, office complexes) which gave rise to a larger variety of candidate reference keypoints. Thus more effective distance profiles and tree ensembles could be created.

#	Method/Setting	Avg time (s)
1	Image classifiers [21]	1.3178
2	Img classifiers + kdtree	1.1382
3	Distance profiles (10% shortlist)	0.2250
4	Dist. prof. + Img classifiers	0.3637
5	Dist. prof. + Img class. + kdtree	0.3468

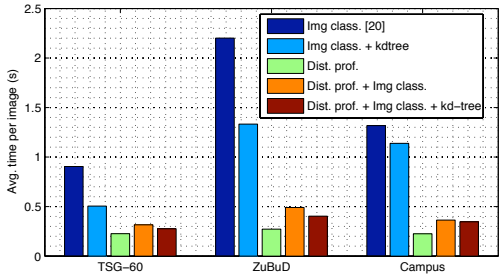
Table 7. Testing duration per image on the Campus dataset (excluding keypoint/descriptor computation time). Note that the time for methods 3–5 includes distance profile generation.

Table 7 presents a comparison on proceessing times for the Campus dataset. A straightforward application of kd-tree indexing did not produce a dramatic improvement in the testing speed of the image classifiers. This is most likely due to the fact that kd-trees scale badly against the ambient dimensionality (the SIFT descriptor has 128 dimensions while the SURF descriptor has only 64 dimensions). It is well known [30] that at high dimensions searching a kd-tree structure is only as efficient as an exhaustive search, although methods have been proposed to rectify this [4, 32]. It occurs again here that the tree ensemble required almost the same time as in the previous experiments to process a testing image— compare method 3 in Tables 3, 5 and 7. The coupling of the tree ensemble and the image classifiers produced almost a three-fold gain in computational speed.

6. Discussion and Conclusion

The results obtained from the three datasets are summarized in Fig. 4. In terms of recognition accuracy, the proposed method compares favourably to the baseline method of training classifiers from discriminative keypoints [21]. We also draw attention to the computational speed of the proposed method which is consistent across the three datasets. In contrast, since the image classifiers are evaluated in a binary classification style, their speed fluctuates heavily depending on the number of scene classes.

(a) Recognition accuracy.



(b) Computational speed.

Figure 4. Summary of experimental results.

It is easy to see why the proposed method scales more efficiently against the number of scene classes N compared to the image classifiers. The distance profiles were generated to be highly discriminative among multiple scenes and the trees grown from the distance profiles can handle multiclass classification naturally. On the other hand, the image classifiers require N evaluations of binary classification. From a

computational standpoint, given a set of F keypoints from a test image, a tree ensemble of size T requires only $T \times F$ (independent of N) computations of descriptor distances. Conversely, evaluating N binary image classifiers of size T each will need $N \times T \times F$ calculations of distances. Note that the time obtained in the experiments can be further reduced with better implementation.

The experiments show that the proposed distance profile for scene images is useful in particular place recognition. More generally, our work also demonstrates that usually ignored side information can be serve useful purposes.

References

[1] H. Ali, G. Paar, and L. Paletta. Semantic indexing for visual recognition of buildings. In *5th Int. Symp. on Mobile Mapping Technology*, 2007.

[2] K. Amlacher and L. Paletta. Geo-indexed object recognition for mobile vision tasks. In *Mobile HCI*, 2008.

[3] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *ECCV*, 2006.

[4] J. Beis and D. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *CVPR*, pages 1000–1006, 1997.

[5] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regressions Trees*. Chapman and Hall, 1984.

[6] M. Calonder, V. Lepetit, and P. Fua. Keypoint signatures for fast learning and recognition. In *ECCV*, 2008.

[7] T.-J. Chin, H. Goh, and J.-H. Lim. Using densely recorded scenes for place recognition. In *ICASSP*, 2008.

[8] Y. Freund and R. Schapire. A decision theoretic generalization of on-line learning and an application to boosting. *Journal of Comp. and Sys. Sciences*, 55(1):119–139, 1997.

[9] G. Fritz, C. Seifert, and L. Paletta. Urban object recognition from informative local features. In *ICRA*, 2005.

[10] G. Fritz, C. Seifert, and L. Paletta. A mobile vision system for urban detection with informative local descriptors. In *ICVS*, 2006.

[11] K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. In *ICCV*, 2005.

[12] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *CVPR*, 2000.

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

[14] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[15] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE TPAMI*, 28(9):1465–1479, 2006.

[16] H. Ling and S. Soatto. Proximity distribution kernels for category classification. In *ICCV*, 2007.

[17] D. Lowe. Object recognition from local scale-invariant feature. In *ICCV*, 1999.

[18] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 20(2):91–110, 2004.

[19] R. Maŕee, P. Geurts, J. Piater, and L. Wehenkel. Random subwindows for robust image classification. In *CVPR*, 2005.

[20] K. Ni, A. Kannan, A. Criminisi, and J. Winn. Epitomic location recognition. In *CVPR*, 2008.

[21] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer. Generic object recognition with boosting. *IEEE TPAMI*, 28(3):416–431, 2006.

[22] M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *CVPR*, 2007.

[23] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[24] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: improving particular object retrieval in large scale image databases. In *CVPR*, 2008.

[25] A. Pronobis, B. Caputo, P. Jensfelt, and H. Christensen. A discriminative approach to robust visual place recognition. In *IROS*, 2006.

[26] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, and T. Tuytelaars. A thousand words in a scene. *IEEE TPAMI*, 29(9):1575–1589, 2007.

[27] B. Russell, A. Torralba, C. Liu, R. Fergus, and W. Freeman. Object recognition by scene alignment. In *NIPS*, 2007.

[28] G. Schindler, P. Krishnamurthy, R. Lublinerman, Y. Liu, and F. Dellaert. Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In *CVPR*, 2008.

[29] S. Se, D. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *ICRA*, 2001.

[30] G. Shakhnarovoch, T. Darrell, and P. Indyk, editors. *Nearest-neighbor methods in learning and vision*. The MIT Press, 2006.

[31] H. Shao, T. Svoboda, V. Ferrari, T. Tuytelaars, and L. V. Gool. Fast indexing for image retrieval based on local appearance with re-ranking. In *ICIP*, 2003.

[32] C. Silpa-Anan and R. Hartley. Optimised KD-trees for fast image descriptor matching. In *CVPR*, 2008.

[33] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1470–1477, Oct. 2003.

[34] A. Torralba. Contextual priming for object detection. *IJCV*, 53(2):169–191, 2003.

[35] J. Wu and J. Rehg. Where am I: Place instance and category recognition using spatial PACT. In *CVPR*, 2008.

[36] L. Yang, R. Jin, R. Sukthankar, and F. Jurie. Unifying discriminative visual codebook generation with classifier training for object category recognition. In *CVPR*, 2008.

[37] J. Zhu, S. Rosset, H. Zou, and T. Hastie. Multi-class AdaBoost. Technical report, Dept. of Statistics, Univ. of Michigan, 2005.

[38] Z. Zivkovic, O. Booij, and B. Kröse. From images to rooms. *Robotics and Autonomous Systems*, 55(5):411–418, 2007.