Mobile Visual Search from Dynamic Image Databases

Xi Chen and Markus Koskela

Department of Information and Computer Science
Aalto University School of Science, Espoo, Finland
{xi.chen,markus.koskela}@aalto.fi

Abstract. Mobile phones with integrated digital cameras provide new ways to get access to digital information and services. Images taken by the mobile phone camera can be matched to a database of objects or scenes, which enables linking of digital information to the physical world. In this paper, we describe our method for mobile image recognition, which is a part of a pilot system for linking of magazine page images to additional digital content. Such magazine databases are highly dynamic, so the recognition method needs to support addition and deletion of images without rebuilding the whole database. Meanwhile we significantly reduce the memory cost in the system without sacrificing retrieval accuracy. We present recognition results with two different databases.

Keywords: image recognition, mobile visual search, mobile augmented reality

1 Introduction

Mobile augmented reality, i.e. augmenting the user's perception of her surroundings using a mobile device, is a relatively new field of research, which has been invigorated by the current prevalence of capable mobile computing devices. These devices are becoming increasingly small and inexpensive, and they allow us to use various computing facilities while roaming in the real world. In particular, ordinary mobile phones with integrated digital cameras are ubiquitous, and already they can provide new ways to get access to digital information and services. The images or video captured by the mobile phone can be analyzed to recognize the objects [3] or scenes [18, 5] appearing in the recordings.

Consequently, the research on applicable image matching algorithms has recently been very active (e.g. [17, 10, 12, 13]), and the current state-of-the-art methods can handle recognition from databases containing millions of images. A mobile image matching algorithm should be robust against variations in illumination, background clutter, viewpoint, and scale. Mobile applications should work with stringent bandwidth, memory and computational requirements. This requires the optimization of the performance and memory usage. For example, it is possible to perform feature extraction directly on the mobile client [19], which may reduce the system latency and provide better system scalability.

In this paper, we describe our image recognition engine which is a part of a pilot system aimed at linking of images taken with a mobile phone to interactive, contextual, and short-term mobile services [4]. This kind of technologies can be used for various purposes: possible application areas include outdoor advertising, magazine and newspaper advertising, tourist applications, and shopping. We focus here on a use case with a magazine publisher as the content provider.

The rest of the paper is organized as follows. We first review briefly some relevant related work and discuss the differences to our method in Section 2. In Section 3, we describe our method for image recognition from dynamic image databases. In Section 4, we present results from experiments with two different databases. Conclusions and plans for future work are discussed in Section 5.

2 Related Work

Image and object recognition based on extracting image patches, describing each patch with a high-dimensional descriptor, and comparing the descriptors has become extremely popular and successful [10]. In particular, the *visual words* paradigm, where the descriptors are first clustered and each descriptor is then represented by a cluster identity has made it possible to recognize images from very large databases [17, 12, 13]. The visual words are however relatively noisy, as the quantization is an additional error source, so direct pair-wise matching can provide more accurate results [14], especially with very few query descriptors.

Using mobile phones to retrieve additional information related to the users' interests has been studied in a number of research projects. Many systems, such as the Nokia's MARA [9] are based on the camera's various sensors, i.e. GPS receiver, accelerometer, and magnetometer. Recently, applications based on image analysis using the mobile phone camera have also been presented. An outdoors augmented reality system for mobile phones is described in [18], where GPS location data is used to prune the image data prior to the image matching stage.

The recognition of various objects with mobile phone cameras has also raised considerable research and commercial interest. For example, an application to recognize book and CD covers from live video on mobile phones is presented in [3]. One of the most popular commercial applications with similar functionalities has been launched by Amazon / SnapTell[1]. After taking a photo with the mobile and sending it to Amazon, corresponding information about the products appearing in the photo or similar products will be sent back to the user if the object is on sale in Amazon. Further examples of similar commercial applications are *Google Goggles*[2], *kooaba*[3], and Nokia's *Point and Find*[4].

In comparison to the above applications, we present in this paper a system for retrieving extended magazine content for mobile phones. Due to page limitations, printed magazines can not include all related information on some comprehensive or interesting topics, or advertisements. This information can, however, be made accessible on the internet. The focus of the application is not on the recognition

[1] http://www.a9.com/

[2] http://www.google.com/mobile/goggles/

[3] http://www.kooaba.com/

[4] http://www.pointandfind.nokia.com/

of magazine covers or other full pages, but on the varying articles and other items within the page layout in the magazines. Therefore, the photos the users submit are not limited to whole pages, but can be of small images or details in the articles, or of some advertisements. The active database consists of a certain number of latest issues only, but is highly dynamic as new issues are constantly appearing and are added to the database. Also, the system has to work with the mainstream of mobile phones currently in use, not just with the high-quality cameras included in the high end phones. This can result e.g. in highly blurred and out-of-focus input images with very few local features due to the difficulty of the mobile phone cameras to autofocus on macro distances. Therefore, we use in this work the direct pair-wise matching of local features as our starting point and aim for real-time matching with high accuracy in this setup.

3 Mobile Image Matching from Dynamic Databases

Our mobile visual searching system [4] is divided into two parts: the mobile client and the server backend. The user takes photos of interest using the client software, which then sends the image to the server for recognition. From the user point of view the system architecture is a quite ordinary web service accessible with any kind of relatively modern mobile phone equipped with a camera and an internet connection. In some applications, the local descriptors are extracted directly on the phone and sent to the server for matching [19], which is only meaningful when the size of the descriptors are significantly smaller than the original images. However, the size of standard descriptors (e.g. SIFT [10] or SURF [1]) with normal parameter settings is often about three times the size of the original images, unless some algorithm can be applied to select the useful descriptors, which means even more computational burden on the phone. Therefore, in our current application, we resize the query images on the client to 640×480 pixels (about 25–50 kB in size) and send the resized images to the server. The scale of the query image is an important parameter for both matching accuracy and speed, and even a query image smaller than this is often sufficient for recognition. In this work, we extract and use SURF descriptors for the matching.

3.1 Sub-Linear Indexing

A practical method for image recognition and matching must support sub-linear indexing, i.e. it has to match the query image to the database images with complexity that does not grow linearly with the size of the database. With methods that explicitly compare the query to each item in the database, the response time will at some point be unacceptable. This is crucial especially for methods that describe the images with non-global descriptions, such as using sets of local features for each image. Standard methods for sub-linear indexing include hashing [8] and tree-based approaches [11].

The classical kd-tree algorithm [7] splits the data from the median in the dimension which has the largest variance of data among the dimensions, but fails to provide any speed-up with high-dimensional spaces. Therefore, a common approach is to use some approximate algorithm, such as Best-Bin-First [10], multiple randomized kd-trees [16], or hierarchical k-means [12].

Recently, Silpa-Anan and Hartley proposed an approximate version of kd-tree which uses multiple randomized kd-trees [16]. A randomized kd-tree selects the dimension to split the data randomly from the first M dimensions with the greatest variance in the data, and in their method N_t such trees are constructed. When searching, a single priority queue is maintained for the N_t trees so that the search can be ordered by distance to each bin boundary. The degree of approximation is determined by examining a fixed number of leaf nodes, at which point the search is terminated and the best candidates returned. In the following, we refer to the multiple randomized kd-trees as a *randomized kd-forest*.

Neither the kd-tree nor the randomized kd-tree can be modified after construction, i.e. new branches or nodes cannot be added or deleted from the tree without rebuilding. The time to build the tree is also relatively long when the dataset is large. If new data is continuously added and old data is removed from the database, it is infeasible to constantly keep rebuilding the tree. Therefore, in order to handle the constant changes in a dynamic database, we use multiple forests of randomized kd-trees. When a new batch of descriptors is added to the database, these descriptors form a separate randomized kd-forest. Similarly, when a certain batch of data is removed, we can just remove the corresponding forest. This multiple forests approach facilitates also parallel processing, which can further speed up the searching, increase accuracy, and enable query-time restrictions to the database (cf. Section 3.3).

In our current project, the image database consists of a set of recently published issues of a certain magazine or magazines from a publisher. When a new magazine issue is published, it is added to the database with each page as a separate image, a new randomized kd-forest is built for the magazine issue, and the forest is added to the database index. Similarly, the outdated magazine issues are removed from the database by removing the corresponding randomized kd-forests from the index.

3.2 Descriptor Pruning

A common method for limiting the number of descriptors extracted from images is to reduce the resolution of the images. This can also be combined with restricting the number of image-wise descriptors based on some magnitude criterion. E.g. the SURF feature adopts a fast multi-scale Hessian keypoint detector for the extraction of the keypoints, and the number of descriptors can be restricted using a threshold for the Hessian.

In the setup of this paper, the magazine pages we are considering are quite different from common images, as there are large portions of text on many pages, which is common cause of wrong matches, and we cannot reduce the resolution of the images too much as the system has to be able to recognize also small details from the pages. As a result, with default parameter settings and a sufficient resolution, each magazine page can generate over 10 000 descriptors.

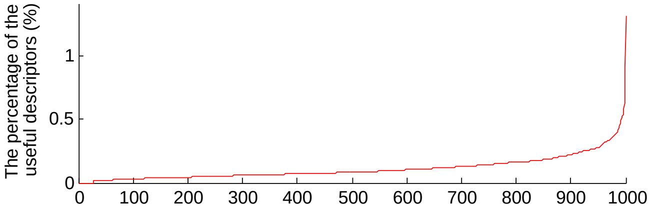

Fig. 1. The percentage of survived nearest neighbors in the clusters.

Since each page generates a large number of descriptors, many of them ineffective, it is advantageous to study if and how much we can reduce the number of descriptors without compromising the search accuracy. Two straightforward approaches for accomplishing this are to increase the Hessian threshold and to randomly sample the keypoints. A third approach proposed in this paper is to classify the keypoints based on their estimated probability of matching. For this, we use a clustering-based classification method.

As the training data set for the clustering, we selected one magazine outside of the testing database, extracted SURF descriptors from each magazine page, and build a randomized kd-forest for all the descriptors. The full set of descriptors, 650 000 in total, was then clustered using k-means with 1000 clusters. We collected a total of 100 images taken with a mobile phone as query images, used the recognition engine to find the matching magazine pages, and recorded all the query descriptors that were matched correctly. We then assigned each matching descriptor from the data set to its cluster, recorded the number of matches for each cluster, and sorted the clusters according to their total sums of matches. The relationship between the clusters in sorted order and the percentage of matched descriptors is depicted in Fig. 1.

From Fig. 1, we can observe that the last 20% of the clusters contain more than half of the matched keypoints in the data set, which suggests that it could be feasible to remove a large portion of the clusters and associated descriptors with marginal effect to the matching performance. We can utilize these sorted clusters to prune descriptors from other data sets as well, by removing the descriptors associated to clusters below a certain cluster threshold.

3.3 Matching with Multiple Indices

The recognition of query images received from the mobile clients is implemented using a two-stage algorithm described in this section. Assume we are matching a query image q to N_f randomized kd-forests. The first stage begins after d_q descriptors have been extracted from q. The N_n nearest neighbors of each query descriptor are returned from each randomized kd-forest. We thus obtain a total of $N_f N_n d_q$ descriptors, each associated with a certain database item. We calculate for each item the total number of its descriptors that belong to this set. Finally, the N_c best-scoring magazine pages are selected as candidates for the second stage.

On the second stage, for the N_c candidates, we do a full pair-wise matching of the d_q query descriptors as in [10] to find the overall best matching pages. At this stage, we only accept nearest neighbors whose distance is less than τ of the distance of the second nearest neighbor. The approximate nearest neighbor algorithm typically results in a substantial number of wrong pair-wise correspondences. In the studied application domain, especially the body text on the magazines produces incorrect matches. Therefore, to exclude the wrong matches from further analysis, we estimate a homography between the point correspondences for the N_c candidates using RANSAC [6] and remove the outliers.

4 Experiments

In this section, we describe our experiments with two databases: a collection of nine issues from three different magazines and the publicly available ZuBuD Zurich Buildings database [15]. The latter is used to validate and compare our method to other published results, as the Magazines database is not public. We use the OpenCV implementation of randomized kd-trees from the Fast Library for Approximate Nearest Neighbors (FLANN) [11], which uses a fixed $M = 5$ and constructs a set of N_t randomized kd-trees to be searched in parallel. We use the parameter values $N_t = 4$, $N_n = 1$, $N_c = 5$, and $\tau = 0.6$ in these experiments.

4.1 Magazines Database

In the Magazines data set, a total of $N_f = 9$ issues are included from three different magazines, each containing about 80–130 pages. The size of each page image is 771×1024 pixels, and a total of 6.5 million descriptors are extracted. The three descriptor pruning approaches are applied before building the randomized kd-forests as described in Section 3.2.

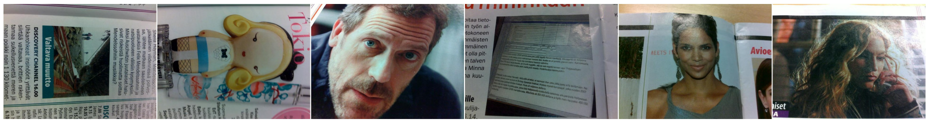

Fig. 2. A random sample of the query images in the Magazines data set.

For testing the recognition accuracy, we took a total of 300 query images from three issues, each from a different magazine. The images were taken by a Nokia E71 phone camera and resized to 640×480 pixels. The images were taken of such content that could be potentially interesting to the readers of the magazines and mostly contain only a small portion of whole page. Some of the query images are illustrated in Fig. 2. In the server, the query images are first resized with a scale of 0.5, that is to 320×240 pixels, as we have observed in our initial experiments that size to work well both in accuracy and speed.

In the extraction of the SURF descriptors, the Hessian threshold is initially set to the default value of 500 of the OpenCV implementation, and the three proposed descriptor pruning methods are then applied. As the name implies, the *random method* samples the descriptors randomly from each page before forming the kd-forests. In these experiments, the random descriptor sample varies from 10% to 90% with 10% intervals. In the *Hessian pruning method*, we sort the descriptors from each page according to their Hessian values, use page-wise thresholds between 800 and 8000 descriptors, and remove the surplus descriptors. With the *clustering-based descriptor classification method*, we prune the descriptors mapped to 40%–90% of the clusters with the lowest fractions of matching descriptors as shown in Fig. 1. The results of these experiments are shown in Fig. 3, from where we can observe that the matching accuracy of the clustering-based classification method is somewhat higher than the other two methods, especially when the size of the kd-forests is only a small fraction of the whole database. In particular, the matching accuracy remains over 0.9 with only 18% of the whole database remaining. The average matching time is about 500 ms.

The SURF descriptor is well known for multiscale matching due to the generation of descriptors using multi-scale scanning. However, the size of the query image determines the number of descriptors and in practice has a notable effect on the recognition accuracy. In the above experiments, the query image was scaled to 0.5 of the original size, which seems to work well overall, but also results in some failed recognitions. Thus, in the following experiments we use a multi-scale approach, where the matching is initially done with the scale of 0.5 and if no match was found, again with scales of 0,67, 0.8, and 1.0. The same experiments for the random, Hessian and clustering-based methods are performed with the multiple-scale approach, and the matching accuracy is shown in Fig. 4.

Comparing the two figures, the multiple-scale matching attains slightly better results than the single scale of 0.5, with the accuracy remaining near 0.95 with only 18% of the database used. Furthermore, since most of the query images are matched correctly with the initial scale, only a small portion of the images are processed with multiple scales, so the multiple-scale approach decreases the average response time only slightly while increasing the system accuracy.

4.2 ZuBuD Database

We have also experimented with a publicly available image database to compare our method and our recognition results with results published in previous works. The ZuBuD database contains color images of 201 buildings in the city of Zurich. There are a total of 1005 images as each building is represented by five shots, taken from different viewpoints and in different lighting conditions. In addition, there are 115 query images included, each having a correct answer among the 201 buildings in the database. The database is relatively easy, as many works report high average accuracies, and it is small enough so that exhaustive pairwise image matching can be used.

Table 1 shows recognition results from two sources, [20, 2], which both use exhaustive matching and report accuracies of over 0.95. With this database, we

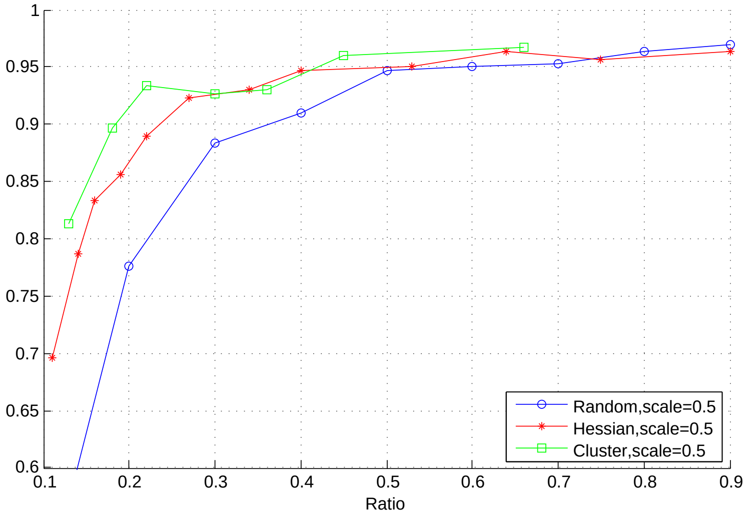

Fig. 3. A comparison for the recognition accuracies of the three descriptor pruning methods, using a fixed scale of $s = 0.5$.

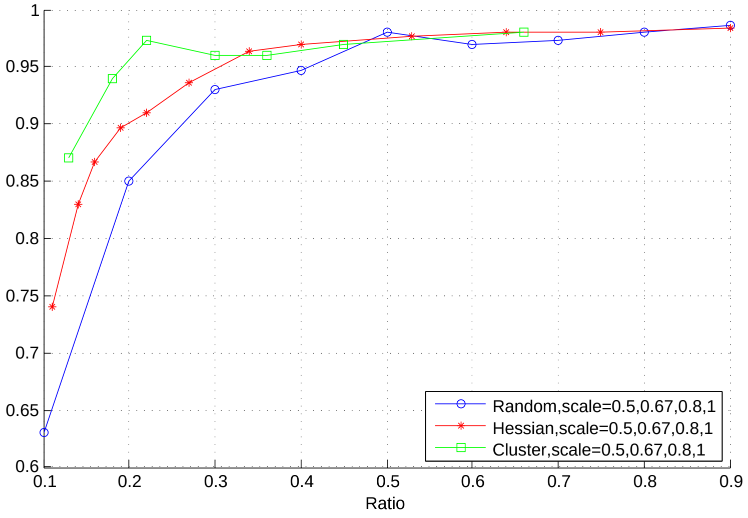

Fig. 4. A comparison for the recognition accuracies of the three descriptor pruning methods, using multiple scales $s \in \{0.5, 0.67, 0.8, 1.0\}$.

used $N_f = 10$, Hessian pruning with different thresholds, and multiple scales of the query images. The results are shown in Table 1, which shows that our method is able to reach similar performance with sub-linear matching. The average matching time is about 300 ms. It can be noted that with 600 keypoints, as in [2], we get only three failed recognitions, and that these failures are due to a missing homography. If in this case we accept the image that is most often selected for the second stage as the recognition result, we get accuracy of 1.0.

Table 1. Results with the ZuBuD database; from the literature (left) and our results (right).

Method	# of keypoints	Accuracy
[2] (SIFT)	600	0.956
[2] (CHoG)	600	0.974
[20]	unknown	0.965

Method	# of keypoints	Accuracy
Hessian	600	0.974
Hessian	400	0.948
Hessian	300	0.921
No homog.	600	1.0

5 Conclusions

In the project described in this paper, we maintain a dynamic image database with magazines from a publishing company. As the database is constantly updated by adding new issues and removing old ones, the separate randomized kd-forest for each magazine fulfills the requirements for the high flexibility, and facilities the use of multiple threads to speed up the matching. Due to the demand for high matching accuracy with query images that are often of poor quality, we use direct matching of descriptors instead of visual words. The magazine pages usually contains a large number of descriptors due to high resolution and large amounts of text. In order to reduce the size of the kd-forests while preserving the matching accuracy, a clustering-based descriptor classification method is applied to the descriptor database. By keeping the descriptors only from the selected clusters, the matching accuracy can reach 0.9 with only 15% of the descriptors. Therefore, the proposed method significantly reduces the memory consumption while retaining a high matching accuracy.

References

1. Bay, H., Tuytelaars, T., Gool, L.V.: SURF: Speeded up robust features. In: Proc. ECCV 2006 (May 2006)
2. Chandrasekhar, V., Chen, D.M., Lin, A.L., Takacs, G., Tsai, S.S., Cheung, N.M., Reznik, Y., Grzeszczuk, R., Girod, B.: Comparison of local feature descriptors for mobile visual search. In: IEEE International Conference on Image Processing (ICIP). Hong Kong (September 2010)
3. Chen, D., Tsai, S., Vedantham, R., Grzeszczuk, R., Girod, B.: Streaming mobile augmented reality on mobile phones. In: Proceedings of International Symposium on Mixed and Augmented Reality (ISMAR 2009). Orlando, Florida (October 2009)

4. Chen, X., Koskela, M., Hyväkkä, J.: Image based information access for mobile phones. In: Proceedings of 8th International Workshop on Content-Based Multimedia Indexing. Grenoble, France (June 2010)
5. El Choubassi, M., Nestares, O., Wu, Y., Kozintsev, I., Haussecker, H.: An augmented reality tourist guide on your mobile devices. In: Proceedings of 16th International Multimedia Modeling Conference (MMM 2010). pp. 588–602. Chongqing, China (January 2010)
6. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381–395 (1981)
7. Friedman, J.H., Bentley, J.L., Finkel, R.A.: An algorithm for finding best matches in logaritmic expected time. ACM Transactions on Mathematical Software 3(3), 209–226 (1977)
8. Gionis, A., Indyk, P., Motwani, R.: Similarity search in high dimensions via hashing. In: Proceedings of 25th International Conference on Very Large Data Bases (VLDB'99). pp. 518–529. Edinburgh, Scotland, UK (September 1999)
9. Kähäri, M., Murphy, D.: MARA – Sensor based augmented reality system for mobile imaging. In: Proceedings of International Symposium on Mixed and Augmented Reality (ISMAR 2006). Santa Barbara, CA (October 2006)
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (November 2004)
11. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. In: Proceedings of International Conference on Computer Vision Theory and Applications (VISAPP'09). Lisboa, Portugal (February 2009)
12. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: Proceedings of IEEE CVPR 2006. vol. 2, pp. 2161–2168 (2006)
13. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (June 2007)
14. Philbin, J., Isard, M., Sivic, J., Zisserman, A.: Descriptor learning for efficient retrieval. In: Proceedings of the 11th European Conference on Computer Vision (ECCV 2010). vol. 6313, pp. 677–691 (2010)
15. Shao, H., Svoboda, T., van Gool, L.: ZuBuD - Zurich buildings database for image based recognition. Tech. Rep. 260, ETH Zurich (April 2006)
16. Silpa-Anan, C., Hartley, R.: Optimised KD-trees for fast image descriptor matching. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2008)
17. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proc. of ICCV'03. vol. 2, pp. 1470–1477 (Oct 2003)
18. Takacs, G., Chandrasekhar, V., Gelfand, N., Xiong, Y., Chen, W.C., Bismpigiannis, T., Grzeszczuk, R., Pulli, K., Girod, B.: Outdoors augmented reality on mobile phone using loxel-based visual feature organization. In: Proceeding of the 1st ACM international conference on Multimedia information retrieval (MIR '08). pp. 427–434. Vancouver, British Columbia, Canada (2008)
19. Tsai, S.S., Chen, D., Chandrasekhar, V., Takacs, G., Cheung, N.M., Vedantham, R., Grzeszczuk, R., Girod, B.: Mobile product recognition. In: ACM Multimedia (ACM MM). Florence, Italy (October 2010)
20. Zhang, W., Kosecka, J.: Hierarchical building recognition. Image and Vision Computing 25(5), 704–716 (May 2007)