RAPID IMAGE RETRIEVAL FOR MOBILE LOCATION RECOGNITION

G. Schroth, A. Al-Nuaimi, R. Huitl, F. Schweiger, E. Steinbach

Institute for Media Technology, Technische Universität München, Munich
{schroth, anas.alnuaimi, huitl, florian.schweiger, eckehard.steinbach}@tum.de

ABSTRACT

Recognizing the location and orientation of a mobile device from captured images is a promising application of image retrieval algorithms. Matching the query images to an existing georeferenced database like Google Street View enables mobile search for location related media, products, and services. Due to the rapidly changing field of view of the mobile device caused by constantly changing user attention, very low retrieval times are essential. These can be significantly reduced by performing the feature quantization on the handheld and transferring compressed Bag-of-Feature vectors to the server. To cope with the limited processing capabilities of handhelds, the quantization of high dimensional feature descriptors has to be performed at very low complexity. To this end, we introduce in this paper the novel Multiple Hypothesis Vocabulary Tree (MHVT) as a step towards real-time mobile location recognition. The MHVT increases the probability of assigning matching feature descriptors to the same visual word by introducing an overlapping buffer around the separating hyperplanes to allow for a soft quantization and an adaptive clustering approach. Further, a novel framework is introduced that allows us to integrate the probability of correct quantization in the distance calculation using an inverted file scheme. Our experiments demonstrate that our approach achieves query times reduced by up to a factor of 10 when compared to the state-of-the-art.

Index Terms— Image Retrieval, Location Recognition, Bag-of-Features, Mobile Media Search

1. INTRODUCTION

Information about the location, orientation, and context of a mobile device is of central importance for future multimedia applications and location-based services (LBS). While GPS can provide sufficient location accuracy, its applicability is limited to outdoor scenarios with few obstacles. Unfortunately, most interesting LBS could be provided in densely populated environments, which include urban canyons and indoor scenarios, where the accuracy and availability of GPS is often limited. Utilizing recorded images as a visual fingerprint of the environment and matching them to an existing georeferenced database like Google Street View [1] allows us to derive the pose in a very natural way (see Fig. 1). This task, known as Content Based Image Retrieval (CBIR), has been an area of intensive research for the last few decades [2, 3, 4, 5, 6]. In feature based retrieval approaches, the similarity of images is typically determined by a score based on the count of matching feature descriptors. To avoid a query time and memory requirements which scale linearly with the number of database images, Sivic and Zisserman [2] addressed these challenges by adopting text retrieval approaches, quantizing descriptors into visual words. In their work, an image is no longer represented by descriptors but by a visual word frequency histogram, the so called Bag-of-Features (BoF) vector. Computing the distances between BoF vectors via inverted files determines the similarity of an image pair in an efficient manner. Objects recorded at different size, pose, and background can be distinctively described by BoF vectors with the aid of a robust yet fine quantization of high dimensional descriptors into visual words. The application to mobile location recognition complicates these requirements. Typically, only sparse reference data can be assumed. For instance, Google Street View panoramas are available online at a distance of 12.6 m on average. The three exemplary panoramas in Fig. 2 illustrate the problem of wide baselines, different lighting conditions, and dynamic objects. However, most importantly, very low retrieval times are essential due to the rapidly changing field of view of the handheld caused by the constantly changing user attention. Real-time location recognition is an essential prerequisite for most LBS and especially for SLAM algorithms [7]. The retrieval times are governed by the feature extraction on the mobile device, the retrieval of database images on the server, and in particular the delay caused by the transmission of the features from the handheld to the server. To reduce this transmission delay, we follow the approach proposed by Chen et al. [8], where, instead of features, compressed BoF vectors are transferred. This allows for a more than 5x rate reduction compared to compressed features [9] and thus a significant reduction of the overall query time. However, this approach requires to perform the quantization of high dimensional descriptors into visual words on the mobile device at very low complexity to cope with the limited processing power.

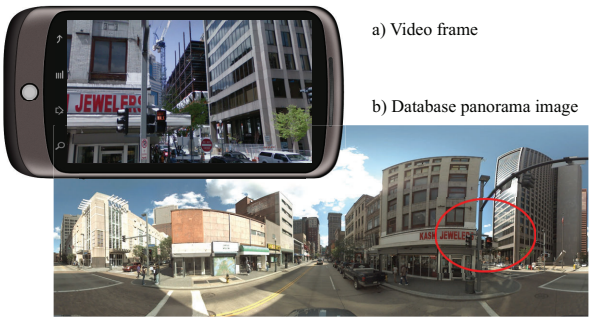


Fig. 1. Google Street View panorama matched to a low resolution video recording (downtown Pittsburgh) using the Multiple Hypothesis Vocabulary Tree. The red ellipse indicates the parts common to both query frame and retrieved panorama.

To achieve this goal, we introduce in this paper the Multiple Hypothesis Vocabulary Tree (MHVT), which allows for the robust quantization of 1000 feature descriptors on a Nexus One with a 1 GHz CPU within 12 ms. Retrieval performance comparable to significantly more complex approaches is accomplished by applying an overlapping buffer at each quantization step, an adaptive clustering approach, and by integrating the probability of correct quantization into the BoF distance calculation.

Fig. 2. Sample images from the Google Street View dataset of Pittsburgh; panoramas are on average 12.6 m apart from each other.

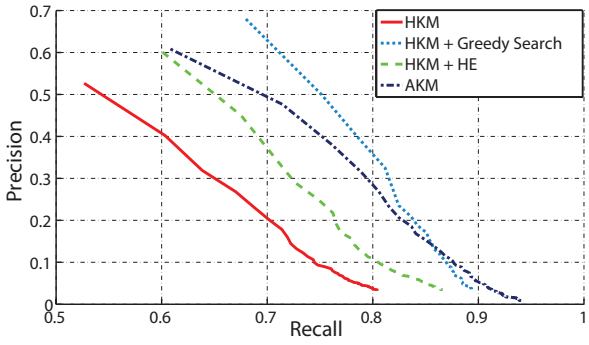


Fig. 3. Comparison of state-of-the-art quantization and indexing structures based on a Google Street View database. Panoramas within a radius of 10 m around the query location are considered relevant.

2. RELATED WORK

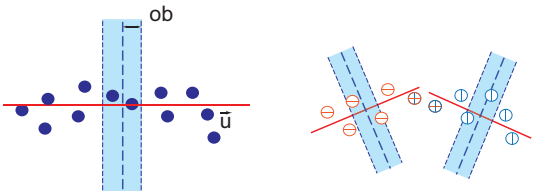
The quantization and thus the robust assignment of descriptors to visual words under varying conditions, e.g., perspective and illumination changes, is a challenging task and essential for the accuracy of image retrieval. In this chapter we provide a comparison of state-of-the-art approaches and evaluate them on a dataset typical for mobile location recognition. Nistér and Stewénius [3] quantize the descriptors with a k-means tree (HKM), which recursively subdivides the features into k clusters until a certain tree depth L is reached. This allows for increasing the number of visual words and thus the distinctiveness while significantly reducing the query time. Further, not only the leaves but also the inner nodes can serve as visual words. This so-called hierarchical scoring allows us to improve the tradeoff between distinctiveness (fine quantization) and correct assignment to a visual word (rough quantization), as inner nodes serve as umbrella terms for the subjacent words. As described for instance in [3, 4], an increase of the branching factor k results in an enhancement of the retrieval performance, which ultimately leads to non-hierarchical k-means quantization at the cost of significantly increased query time. Philbin et al. [5] propose approximate k-means (AKM) clustering to reduce the computational complexity of learning such a flat vocabulary, which minimizes total distortion. Further, they assign a descriptor not only to the closest visual word but also to words in its vicinity in order to increase the chance that matching query and database descriptors are assigned to the same visual word (soft assignment). Schindler et al. [4] propose a so-called Greedy Search within the k-means tree, which allows us to adaptively improve the probability of quantizing a query descriptor to the closest visual word at the cost of increased query time. Jégou et al. [6] follow a different approach by increasing the distinctiveness of features quantized to the same visual word with the aid of Hamming Embedding.

To provide a systematic comparison of the state-of-the-art, we query a georeferenced database extracted from Google Street View of an area of about 4 km^2, which consists of 5000 panoramas at a distance of 12.6 m, each composed of 12 rectified images. As vague prior knowledge on the location, e.g., derived from Cell-IDs, can be always assumed, the size of the database can be limited. Query images are recorded at a resolution of 800x480 pixels and placed halfway between the panoramas. Approximately 400 SURF [10] descriptors per query image are extracted. Fig. 3 shows the precision recall graphs of the individual approaches. *Precision* is the percentage of retrieved images (locations) that are relevant to the query. *Recall* is the percentage of all the relevant images (locations) in the database which are retrieved. Relevance is defined in location recognition by a given radius around the query location. Hence, for a maximum recall of 1, all panoramas within a radius of 10 m around the query location have to be retrieved in Fig. 3. As the graphs show the average over all query images, recall and precision can take values between 0 and 1. This very challenging scenario allows us to effectively evaluate the properties of the individual approaches. As shown in Fig. 3, the basic HKM quantization with 1 M leaves, branching factor $k = 10$ and tree depth $L = 6$ seems to be clearly inferior to the other approaches. However, only $kL = 60$ L_2 distance computations are required per query, rendering the approach extremely fast. Applying a Greedy Search [4] at query time significantly boosts the performance while requiring 510 distance computations in this configuration. Hamming Embedding [6] (HE) requires only about one third of the computations, however, at the cost of increased memory requirements. The AKM [5] approach is set to perform 192 distance computations as part of the backtracking in 8 randomized kd-trees to query a flat vocabulary of 1 M visual words. Soft assignment is applied to the 4 closest visual words. The results of the AKM can be improved by additional backtracking, e.g., at 768 distance computations a 3% increase in recall can be observed, however we try to minimize the query time.

Hence, among the state-of-the-art algorithms, HKM, which requires about 25 ms per query image on a 2.4 GHz desktop CPU and can be adaptively improved with the aid of greedy search, is most suitable for rapid quantization of descriptors. While this would be sufficient on a regular PC, the limited processing power of mobile devices calls for even faster approaches. To this end, we introduce the novel Multiple Hypothesis Vocabulary Tree (MVHT) as a step towards real-time mobile location recognition.

3. MULTIPLE HYPOTHESIS VOCABULARY TREE

Instead of applying a large branching factor k, intended to improve quantization in k-means trees [3, 4], we limit the structure to a binary tree to minimize the query time. Thus, we organize the data hierarchically by separating the space iteratively with hyperplanes. At each node, a vector $\tilde{\mathbf{u}}$, heading in the direction of maximum variance, and the median of all data points projected onto $\tilde{\mathbf{u}}$ are determined. Thus, $\tilde{\mathbf{u}}/\|\tilde{\mathbf{u}}\|$ is the normal vector of the hyperplane that separates the node at the median, resulting in two mutually exclusive child nodes. It is essential for the performance of the algorithm that the normal vector is aligned with the direction of maximum variance. While close to optimal results can be obtained by spanning the vector between the two mutually farthest point clusters, the optimal splitting axis can also be determined by the eigenvector corresponding to



(a) Quantization step 1 (b) Quantization step 2

Fig. 4. Quantization with overlapping buffers using previously proposed suboptimal pivot selection (solid line)

the largest eigenvalue of the node's covariance matrix. However, in this case the time required to build the tree is increased.

The ratio between the query vector comparisons for a binary tree relative to a tree with branching factor k, at corresponding number of leaves, is given by Eq. 1, where L is the depth of the k-ary tree.

$$\frac{Comp_{binary}}{Comp_{k-ary}} = \frac{\log_2\left(k^L\right)}{k \cdot L} = \frac{\log_2\left(k\right)}{k} \tag{1}$$

While for HKM every node has to be represented by a high-dimensional centroid, no separation axes have to be stored for the leaf nodes of a binary tree. In Eq. 2 the ratio between the number of separation axes S_C and the node count N_C is given, proving that the binary tree requires at most an equal amount of memory.

$$\frac{S_C}{N_C} = \frac{k^L - 1}{(k^{L+1} - 1)/(k - 1)} \tag{2}$$

A query descriptor is quantized by proceeding down the tree performing high-dimensional dot products with $\tilde{\mathbf{u}}$ to evaluate on which side of the hyperplane it is located. Descriptors close to the splitting boundaries have a high probability of matching to a descriptor in the neighboring node and would require backtracking to be found. Hence, adapted from [11], an *overlapping buffer* around the boundary with width $ob = \tau \cdot \|\tilde{\mathbf{u}}\|$ is introduced (see Figure 4). All database descriptors projected inside the buffer are assigned to both child nodes. Hence, descriptors that cannot be clearly distinguished by the current decision boundary are not separated from each other at the current node. The differentiation is delayed to the child nodes where the probability of lying far from the separating hyperplane is larger. As the nodes are no longer mutually exclusive, additional quantization steps can be required, which are relative to the parameter τ. However, in practice this is only a minor fraction of the overall number of quantization steps and hardly adds to the query time.

The size of the tree is mainly determined by the aspired quantization level, which can be defined by the number of features assigned to a visual word. Hence, the splitting process described above proceeds recursively until the number of descriptors is less than the defined maximum, the *naive count*. However, large databases result in differently sized descriptor clusters depending on the frequency of the corresponding image textures (e.g. windows in an urban environment). Thus, using a fixed *naive count* to stop the quantization is suboptimal. As self-contained feature clusters can be assumed to have similar variance in most directions, the percentage of features inside the overlapping buffer is very high. This allows us to evaluate the separability of a node and to stop the quantization once a certain percentage ρ of features fall into the buffer. Thus, overfitting of descriptor clusters can be effectively avoided, resulting in smaller trees with significantly increased performance.

The assignment of features to an overlapping buffer is comparable to the soft assignment strategies described in [5]. However, instead of assigning all descriptors to multiple leaf nodes whose centroids are closer than a certain threshold, we allow database descriptors to follow only hypothetical paths that a query descriptor could traverse. Thus, the probability of finding the matching descriptor in the leaf the query has been assigned to, is significantly increased while limiting the loss in distinctiveness of the nodes. We do not assign multiple hypothetical paths to the query descriptor since this would clearly increase the query time. A hierarchical scoring as applied in [3], which can also be interpreted as a weighted multiple assignment of descriptors to leaf nodes, should be no longer necessary. Instead of less distinctive umbrella terms, multiple hypotheses of feature descriptors that correspond to likely word spellings are considered. With the multiple hypothesis approach, the scoring energy can be concentrated on the leaf nodes and extensive time consuming random memory accesses are avoided.

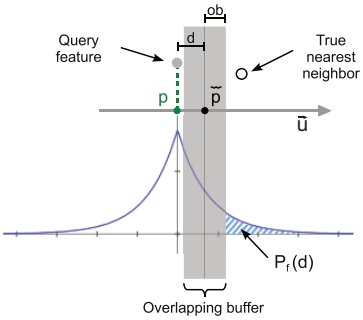


Fig. 5. Two dimensional illustration of the separation axis $\tilde{\mathbf{u}}$, the overlapping buffer, and the determination of the *false quantization probability* $P_f(d)$ as a function of the distance d between the query feature and the separating hyperplane.

3.1. Weighted Scoring

As described above, the probability of assigning a descriptor to the correct child node depends on the distance d between the feature and the separating hyperplane, and the size of the overlapping buffer ob (see Figure 5). With the aid of this buffer, quantization effects are reduced. Further, we would like to account for the probability of assigning matching query and database descriptors to the same leaf node. We determined descriptor differences between matching SURF features to be Laplacian in each dimension, via empirical evaluation. The mean of this distribution is zero, as it is the difference of two identically distributed random variables. Thus, the probability P_f that a matching feature is "incorrectly" quantized to the neighboring child node can be determined by the cumulative 64-dim. Laplacian distribution function. As illustrated in Figure 5, this probability corresponds to the integral over the shaded area beyond the overlapping buffer (ob). Thus it depends on the distance d between the query feature and the separating hyperplane.

We assume that the distribution of the entries of the difference vector between a query and a matching reference descriptor are independently distributed following a Laplacian distribution. Thus, we only have to consider the one dimensional cumulative distribution to determine the probability P_f of finding a match on the other side of the buffer (Eq. 3). Here, σ is relative to the variance of the matching descriptor differences D.

$$P_f\left(d\right) = \frac{1}{2} e^{-\frac{|d+ob|}{\sigma}}; \qquad \sigma = \sqrt{\frac{\mathrm{var}(D)}{2}} \tag{3}$$

The probability of assigning matching descriptors to the same visual word corresponds to the probability of quantizing matching features to the same node $(1 - P_f)$ in all quantization steps m (Eq. 4).

$$\alpha_i = \prod\nolimits_m \left(1 - P_{f\,m}\right) \tag{4}$$

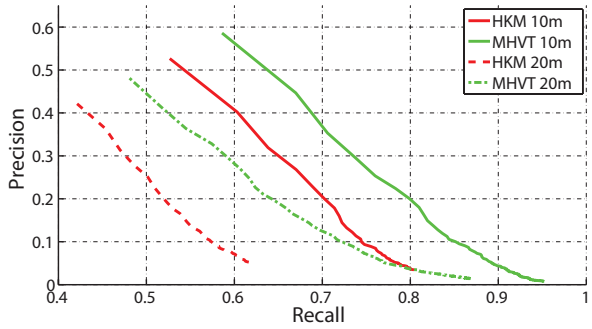


Fig. 6. Comparison of MHVT and HKM at two vicinity levels.

This probability α_i can be utilized to weight the individual comparisons between query ($\mathbf{q}$) and reference ($\mathbf{d}$) BoF vector entries in Eq. 5, which correspond to the visual words. This allows us to reduce the influence of unconfident visual word quantizations. In this equation, the comparison between a query and a single database image is shown, with the sum iterating over all dimensions of the BoF vector.

$$\sum_i \alpha_i \left|q_i - d_i\right|^P = \sum_{i|d_i=0} \alpha_i \left|q_i\right|^P + \sum_{i|q_i=0} \alpha_i \left|d_i\right|^P + \sum_{i|q_i\neq 0 \wedge d_i \neq 0} \alpha_i \left|q_i - d_i\right|^P \quad (5)$$

This equation has to be reformulated to allow for a distance calculation via inverted files. Only those terms of the BoF vector comparison can be computed efficiently where both $q_i \neq 0$ and $d_i \neq 0$. The BoF vector entries where either $q_i = 0$ or $d_i = 0$ are inaccessible as they are not stored in the inverted file system. Since BoF vectors are normalized to unit length, we can express the sums over inaccessible entries by their complements (see Eq. 6).

$$\sum_{i|d_i=0} \alpha_i \left|q_i\right|^P = \sum_i \alpha_i \left|q_i\right|^P - \sum_{i|d_i\neq 0} \alpha_i \left|q_i\right|^P \quad (6)$$

Since no weights α_i exist for $q_i = 0$ in the second term of Eq. 5, we set them to a constant value c and perform the same substitution as in Eq. 6 resulting in Eq. 7.

$$\sum_i \alpha_i \left|q_i - d_i\right|^P = \sum_i \alpha_i \left|q_i\right|^P - \sum_{i|d_i\neq 0} \alpha_i \left|q_i\right|^P + \\ + \sum_i c|d_i|^P - \sum_{i|q_i\neq 0} c|d_i|^P + \sum_{i|q_i\neq 0 \wedge d_i \neq 0} \alpha_i \left|q_i - d_i\right|^P \quad (7)$$

Hence, only terms with $q_i \neq 0$ and $d_i \neq 0$, as well as the weighted norms of the query and the database image, remain in Eq. 7, which can be computed efficiently within the inverted file approach.

Combining the multiple hypotheses vocabulary with the weighting based on the plausibility of the query descriptor quantization allows us to cope with the continuous feature descriptor space while hardly increasing the query time. In practice, less than 21 high-dimensional vector comparisons have to be performed per query descriptor on average. As time consuming hierarchical scoring techniques [3] can be avoided, an overall query time of about 2.5 ms per 1000 descriptors on a 2.6 GHz single-core CPU using the same database as in [3] can be achieved at superior retrieval performance. In Fig. 6, we compare the HKM approach at 1 M leaves with the MHVT at *naive count* $= 200$ and $\tau = 0.06$ using the same database as in Fig. 3 comprising 24 M descriptors. Locations within 10 m and 20 m, are to be retrieved, respectively. Larger values of τ increase retrieval performance further at the cost of higher memory requirements of the inverted file system, which is comparable to the AKM approach with soft-binning.

4. CONCLUSION

In this paper, we consider BoF based image retrieval with respect to its applicability to mobile location recognition as part of the federal-funded project NAVVIS. To achieve the required low query times, transmission delay is minimized by performing the feature quantization on the mobile device and sending compressed BoF vectors. To cope with the limited processing capabilities of mobile devices, we introduce the Multiple Hypothesis Vocabulary Tree, which allows us to perform the feature quantization at very low complexity. A further increase of retrieval performance is accomplished by integrating the probability of correct quantization in the distance calculation. By achieving an at least 10 fold speed up with respect to the state-of-the-art, resulting in 12 ms for 1000 descriptors on a Nexus One with a 1 GHz CPU, mobile vision based real-time localization becomes feasible. In combination with the feature extraction as proposed in [12], which takes 27 ms per frame on a mobile device, extraction and quantization of 500 features can be performed at 30 fps.

5. ACKNOWLEDGEMENTS

This research project has been supported in part by the space agency of the German Aerospace Center with funds from the Federal Ministry of Economics and Technology on the basis of a resolution of the German Bundestag under the reference 50NA1107.

6. REFERENCES

[1] "Google Street View," http://maps.google.com/streetview.

[2] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *ICCV*, Nice, October 2003.

[3] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *CVPR*, New York, June 2006.

[4] G. Schindler, M. Brown, and R. Szeliski, "City-scale location recognition," in *CVPR*, Minneapolis, June 2007.

[5] 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*, Anchorage, June 2008.

[6] H. Jégou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *Int. Journal of Comp. Vision*, vol. 87, no. 3, pp. 316–336, February 2010.

[7] A. Angeli, D. Filliat, S. Doncieux, and J.A. Meyer, "A fast and incremental method for loop-closure detection using bags of visual words," *IEEE Trans. on Robotics, Special Issue on Visual SLAM*, vol. 24, pp. 1027–1037, 2008.

[8] D.M. Chen, S.S. Tsai, V. Chandrasekhar, G. Takacs, J. Singh, and B. Girod, "Tree histogram coding for mobile image matching," in *IEEE Data Compression Conference*, Snowbird, March 2009.

[9] M. Makar, C. Chang, D. Chen, and S. Tsai, "Compression of Image Patches for Local Feature Extraction," in *ICASSP*, Taipeh, April 2009.

[10] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Lecture Notes in Comp. Science*, pp. 404–417, May 2006.

[11] T. Liu, A.W. Moore, A. Gray, and K. Yang, "An investigation of practical approximate nearest neighbor algorithms," in *Neural Information Processing Systems*, Vancouver, May 2004.

[12] G. Takacs, V. Chandrasekhar, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod, "Unified Real-Time Tracking and Recognition with Rotation-Invariant Fast Features," in *CVPR*, San Francisco, June 2010.