From Structure-from-Motion Point Clouds to Fast Location Recognition

Arnold Irschara[1,2], Christopher Zach[2], Jan-Michael Frahm[2], Horst Bischof[1]
[1]Graz University of Technology
{irschara, bischof}@icg.tugraz.at

[2]University of North Carolina at Chapel Hill
{cmzach, jmf}@cs.unc.edu

Abstract

Efficient view registration with respect to a given 3D reconstruction has many applications like inside-out tracking in indoor and outdoor environments, and geo-locating images from large photo collections. We present a fast location recognition technique based on structure from motion point clouds. Vocabulary tree-based indexing of features directly returns relevant fragments of 3D models instead of documents from the images database. Additionally, we propose a compressed 3D scene representation which improves recognition rates while simultaneously reducing the computation time and the memory consumption. The design of our method is based on algorithms that efficiently utilize modern graphics processing units to deliver real-time performance for view registration. We demonstrate the approach by matching hand-held outdoor videos to known 3D urban models, and by registering images from online photo collections to the corresponding landmarks.

1. Introduction

Image-based localization is an active and highly relevant research topic, e.g. self localization using cell phone cameras is an interesting and important future application for touristic site identification. Tracking solutions like GPS can satisfy the demand to some degree in outdoor environments, but do not work in areas with an occluded sky, downtown areas and indoor environments. These areas can only be addressed with image-based solutions given that drift-free inertial system solutions are economically not feasible.

The demand for image based location recognition has not been satisfied despite the tremendous progress in image based recognition [15]. Our proposed approach to this problem leverages the recent progress of 3D scene reconstruction from images/videos [16, 22, 11] to allow a superior recognition system. Both research areas have independently made enormous progress in the last decade. It is now possible to efficiently build 3D models from large image collections or videos. Our proposed approach employs the fact that the obtained 3D models allow to impose stronger geometric constraints on possible scene views than traditional image based methods. These geometric constraints are mostly orthogonal to the image based constraints and deliver the pose for the query image directly. Accordingly we can also utilize the significant progress in image based recognition that occurred over the past decade leading to near real-time image retrieval from huge databases containing millions of images. Our proposed approach combines these two disciplines and uses their state-for-the art techniques to advance location recognition. Unlike previous methods, we propose to compute a representative set of 3D point fragments that cover a 3D scene from arbitrary view points and utilize a vocabulary tree data structure for fast feature indexing. A subsequent matching approach and geometric verification directly delivers the pose of the query image, as shown for instance in Figure 1.

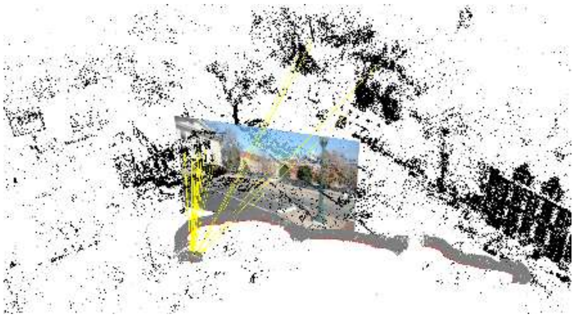


Figure 1. Registration of video frames with respect to a sparse 3D scene, reconstructed by structure from motion techniques.

There is significant literature on image based location recognition [19, 25, 26]. Schindler *et al.* [20] present a method for large-scale location recognition based on geo-tagged video streams and specific trained vocabulary trees. Self-localization in indoor and smaller-scale environments using image or video data is also addressed by the visual

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

SLAM (simultaneous localization and mapping) literature. Ethan and Drummond [4] propose a vocabulary tree-based approach for real-time loop closing, using a reduced SIFT-like descriptor containing 16 components and thus a smaller vocabulary. Combining bag-of-features approaches with geometric verification to improve the precision of object recognition is also proposed in [24].

Related work in the augmented reality context includes [6] and [18]. In particular, our approach is similar to that of Gordon and Lowe [6], also utilizing structure from motion point clouds for pose estimation. However, the size of the employed 3D models in their approach is about two orders of magnitude smaller, thus a compact 3D representation as proposed in our approach is not required.

The work by Simon *et al.* [21] shares algorithmic similarities with the scene compression proposed in this work. Their method aims on computing a minimal canonical subset of views that best represents a collection of given images. A greedy method is employed due to the intractability of finding the true optimal solution. Our compressed 3D scene representation presented in Section 2.3 encounters a similar underlying combinatorial problem approximately solved by a greedy procedure.

2. 3D Scene Representation

This section describes the compact representation of a 3D model (or a set of models) that we use to register new query images. Naturally, the underlying 3D models are created from images using multiple-view vision methods. A set of images registered to the 3D model is always required in order to retrieve the necessary image features and associated descriptors for the 3D points of the model. Since we employ point features in the query image, only a sparse point cloud needs to be maintained for our purpose and we can omit the costly dense geometry generation.

2.1. Model Reconstruction

Since the main focus of this work is not the reconstruction aspect, we only briefly describe the steps relevant for the subsequent processing. In particular we rely on previous work described in [8]. The approach taken in this work uses calibrated cameras in order to largely avoid degenerate configurations and robustify the reconstruction process. Further, the resulting sparse models exhibit many more 3D points than models generated e.g. from uncontrolled image collections [22, 11], which turns out to be necessary for higher registration rates.

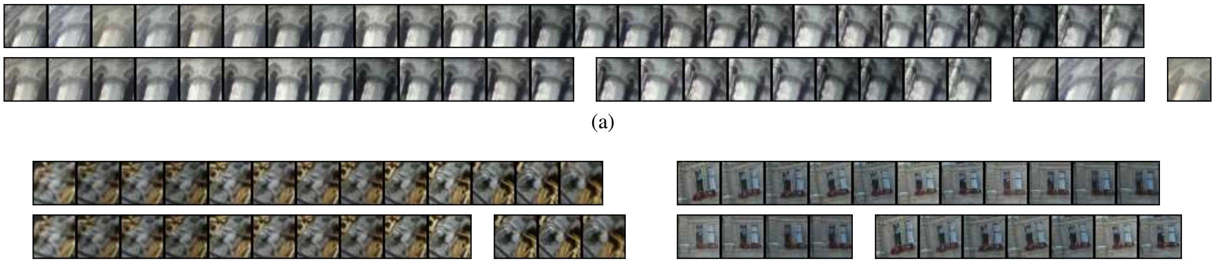
Our method utilizes the very effective SIFT keypoint detector and descriptor [12] as the primary tool to represent point features (but is of course not limited to this choice). Consequently, any 3D point in the resulting sparse model has a set of associated image features with a variety of view dependent descriptors. The list of descriptors can be very long for highly stable 3D points (i.e. points visible and matchable in many source images). Typically, the descriptor list for such points shows high redundancy, and the descriptor set can be compressed without loss in registration performance. Thus we apply mean-shift clustering [3] to quantize SIFT descriptors belonging to each 3D point, thereby lowering the memory footprint of the 3D representation and speeding up matching time. Mean-shift clustering enables to set a global threshold h (bandwidth) on the maximally allowed inter-cluster dissimilarity $2h$. Hence, if two feature descriptors have a distance d before mean-shift clustering, then the distance of the cluster centers is at most $d + 2h$. Figure 2 shows image patches of respective SIFT descriptor and the grouping after mean-shift clustering. The reduction in memory consumption is significant, e.g. 1.500.000 SIFT descriptors (≈730 MB) are compressed to less than 600.000 descriptors (300MB).

Beside the list of corresponding feature descriptors, every reconstructed 3D point has an associated scale induced by the keypoint detector. Under a fronto-parallel surface assumption, the scale found in the image can be extrapolated to a 3D scale. This 3D scale value is subsequently used to estimate the size of a 3D feature in synthetic views and thereby affecting the patch's visibility. Under the fronto-parallel surface assumption each descriptor also carries a directional component pointing towards the camera in which the descriptor was extracted.

2.2. Synthetic Views

As described in the previous section, the reconstructed model is represented as a 3D point cloud with associated scale values and feature descriptors. In addition, the set of images used to build the model with known orientation is available. This information allows registration of new views sufficiently close to the original ones, but in order to be able to compute the poses for images taken far from the originally provided set of views we propose the creation of "synthetic" views located at additional positions not covered by the original images.

Our application is targeted towards localization in urban environments. Hence we can restrict the placement of synthetic cameras to the "eye-level" plane induced by the original views to simplify the problem. Generally, our approach is not limited to terrestrial camera positions. A more powerful descriptor like the VIP features [23] might prove beneficial for registering images captured from significantly different viewpoints (e.g. aerial views). It is sufficient to place these synthetic cameras uniformely on this plane and we do not consider any optimal placement strategy (like proposed in [2]). Under the assumption of dominant horizontal viewing directions, we use 12 for the camera rotation. This corresponds to a 30° rotation between the cameras. The

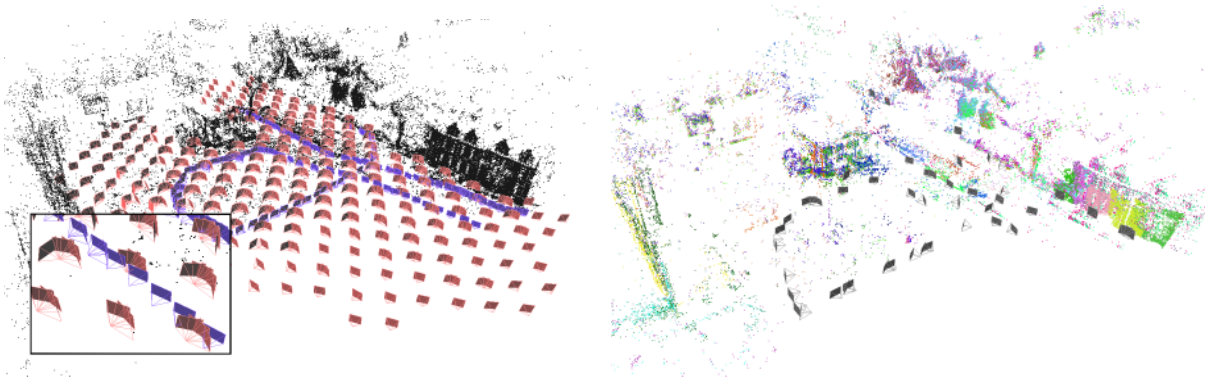


(a)

(b)

(c)

Figure 2. (a)-(c) The first row of each figure shows image patches belonging to the same triangulated 3D point (track). The patches are associated to regions where SIFT keys are extracted in each input image. The second row depicts the grouping result after mean-shift clustering (bandwidth $h = 0.22$). For track (a) 26 SIFT descriptors are reduced to 4 clusters (26/4), in (b) 13/2 and (c) 11/2.



(a)

(b)

Figure 3. (a) A structure from motion point cloud and the raw views/documents (blue camera glyphs for real images and red ones for the full set of synthetic views). (b) Compressed view/document set with the color coding indicating the associated 3D points.

30° are approximately the off image plane rotation that the SIFT descriptor is robust against [13]. In-plane camera rotations are largely handled by the rotational invariance of the SIFT descriptor. The intrinsic parameters for the synthetic views are empirically set to a field of view α and $m \times n$ pixel resolution. Not all generated synthetic views are really useful. Given the 3D position and the respective scale of each triangulated point in the sparse model, one can estimate the projected feature size in the synthetic images and therefore infer the visibility of each 3D point given the set of visible features. More precisely, a 3D point is potentially visible in a synthetic view, if the following criteria are met: (i) the projected feature must be in front of the camera and lie within the viewing frustum; (ii) the scale of the projected 3D feature must be larger or equal to one pixel in terms of the respective DoG scale space extrema to ensure detectability; and (iii) the one of the associated descriptors is extracted from an original image with a sufficiently similar viewing direction due to the limited repeatability of the SIFT descriptor under perspective distortion [13]. For the viewing angle criterion we set the threshold in the viewing angle difference to 30°, which again corresponds to the stability region of the SIFT descriptor. This criterion acts as a "face culling" test by removing 3D points oriented away from the synthetic camera.

There is a one-to-one correspondence between synthetically generated views and the 3D points visible therein. The set of 3D points (potentially) visible in a particular synthetic view represents the document later retrieved through in the vocabulary tree search and in the subsequent 2D-3D point correspondence estimation. Analogously, the 3D points triangulated in the original images form "3D documents" with respect to the original views. Figure 3(a) illustrates this concept by displaying the original views utilized for structure and motion computation (blue) and the additional synthetic views generated by uniform sampling (red). In general the created synthetic views will have a high degree of redundancy especially given the fact that the original views additionally sample the scene. In the next section we will discuss a technique to perform a compression of these views into a representative subset of views covering the scene.

2.3. Compressed Scene Representation

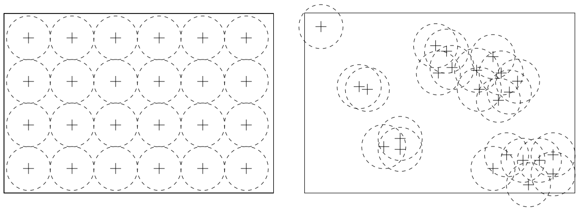
The aim of our compression procedure is to build a compact as well as efficient 3D document database. A reduced set of documents has two major advantages over utilizing the full set of real and synthetic views: the signal-to-noise ratio for vocabulary tree queries (see Section 3.1) is increased, since it is expected that a reduced document set is more discriminative for their respective scene content. Further, the smaller database size has a positive impact on the run-time efficiency in general. Hence, we take a different approach than [20], where visual words voting for a particular document in the vocabulary tree also support documents associated with spatially close views.

The overall goal of our proposed compression strategy is to keep a minimal number of documents while still ensuring a high probability for successful registration of new images. Thus, the key question in evaluating a document summarization is, whether a particular set of documents is sufficient to determine the pose of admissible images. In order to reduce the computational complexity of determining a representative document set, we only consider views which are subsets of real and synthetic views. Thus, we do not create new 3D documents during the compression process. In the following we state these objectives more precisely.

Let V be an admissible view. The sparse 3D model projects into this view as a set of putatively visible 2D point features with associated descriptors. Under the assumptions for the image resolution (see Section 2.2), only a fraction of 3D points is estimated to be visible due to the corresponding scale of the features. 3D points with a too small scale in their projection will be discarded besides the features that are not within the field of view of the camera. We assume that a view V can be successfully registered by a set of 3D points $\mathcal{P}$, if a certain number of 3D points from $\mathcal{P}$ is visible in V and has a good spatial distribution in the image. Consider Figure 4: while the number of features is equal in (a) and (b), the uniform spatial distribution of point features in (a) can be regarded as more reliable than the one shown in (b). Hence, we weight the raw number of features (or correspondences) by an estimate for the covered image fraction yielding an effective feature/correspondences count. This weighting is utilized for determining the effective number of correspondences for view registration (Section 3), too.

For the document reduction procedure we require 150 effective 3D points from $\mathcal{P}$ to be visible in V (according to the above-mentioned assumptions on feature repeatability).

For given sets of 3D documents and views a binary matrix can be constructed, which has an entry equal to one, if the respective document covers the particular view, and zero otherwise. Since in our setting the 3D documents correspond to combined (real and synthetic) views, this matrix is square. In order to have every view covered by at least one document, a document covers its corresponding view by default. This situation can arise if a particular real image has only a few extracted features and thus only a few triangulated 3D points are visible at all. The objective is now to determine a subset of the documents, such that every view is still covered by at least one 3D document. This is an instance of the set cover problem, one of the earliest problems known to be NP-complete [10]. We use a straightforward greedy approach [9] to determine a reduced but representative subset of documents with low time complexity. Algorithm 1 illustrates the greedy algorithm for a given binary view cover matrix A. The actual implementation employs a sparse, set-based representation for A.



(a) (b)

Figure 4. (a) Coverage of uniformly and (a) non-uniform distributed image measurements.

Algorithm 1: Greedy Set Cover

```
Input: Binary matrix $A \in \{0,1\}^{m \times n}$
Output: $\mathcal{S} \subseteq \{1, \ldots, M\}$
$\mathcal{S} \leftarrow \emptyset$
while $A \neq \mathbf{0}$ do
    $i^* \leftarrow \arg\max_i \sum_j A_{i,j}$
    $\mathcal{S} \leftarrow \mathcal{S} \bigcup \{i^*\}$
    $A_{i,:} \leftarrow \max(0, A_{i,:} - A_{i^*,j})$ for all $i$
end
```

Algorithm 1 delivers a representative subset of views needed to cover the 3D scene. This subset can now be deployed for an efficient recognition of the scene context. The next section will describe our search method.

3. View Registration

Geometric registration of an incoming query image $\mathcal{Q}$ to the existing 3D database involves finding potentially matching relevant documents, for which we employ a vocabulary tree with a subsequent geometric verification. This verification step simultaneously validates the putative matches and determines the pose of the query image with respect to the 3D model. If maximal run-time performance is targeted, 3D document retrieval needs to be very precise in order to avoid costly geometric verification of irrelevant documents. Thus, we designed a novel scoring function to rank documents according to the raw votes obtained by the vocabulary tree, and we utilize the computational power of modern graphics processing units to accelerate several highly data-parallel steps in the view registration procedure.

3.1. Vocabulary Tree And Document Scoring

A critical step in the overall approach is to determine relevant documents that are tested for geometric plausibility later on. We employ a vocabulary tree approach [15] to obtain potential matches between query image features and the keypoint descriptors associated with the 3D documents in an efficient manner. The utilized tree is a complete tree with $D = 3$ levels and $K = 50$ children for every internal node. The leaves of the tree correspond to quantized feature descriptors (visual words) obtained by a hierarchical K-means clustering procedure. The tree structure allows the efficient determination of the approximately closest visual word by $K \cdot D$ descriptor comparisons.

In our approach we select a different scoring function than the one proposed in [15] based on the following reasoning. Without loss of generality, we assume that the number $|\mathcal{Q}|$ of extracted features in the query image is less or equal the number of features in the considered document $|\mathcal{D}|$. If $|\mathcal{Q}| > |\mathcal{D}|$ the roles of the query image and the documents can be exchanged by their intrinsic symmetry. Then, if $f_i^{\mathcal{Q}}$ denotes a feature descriptor in the query image $\mathcal{Q}$, and $f_j^{\mathcal{D}}$ denotes the corresponding feature in a matching document $\mathcal{D}$, i.e. $sim(f_i^{\mathcal{Q}}, f_j^{\mathcal{D}}) \geq \theta$, then there is a (relatively high) probability, that the corresponding visual words $w(f_i^{\mathcal{Q}})$ and $w(f_j^{\mathcal{D}})$ are the same. Thus, it is expected that both features fall into the the same leaf node with probability

$$P\left(w(f_i^{\mathcal{Q}}) = w(f_j^{\mathcal{D}}) | \mathcal{Q} \equiv \mathcal{D}\right), \tag{1}$$

where we denote the existence of a true geometric relation between query image $\mathcal{Q}$ and a 3D document $\mathcal{D}$ by $\mathcal{Q} \equiv \mathcal{D}$. This probability depends on the actual features, but we assume it has a universal value p_1. On the contrary, the visual word $w(f_i^{\mathcal{Q}})$ votes for an unrelated document $\bar{\mathcal{D}}$ by pure coincidence (e.g. due to the lower dimensional discretization of the descriptor space), thus we have to estimate the probability

$$P\left(w(f_i^{\mathcal{Q}}) = w(f_j^{\bar{\mathcal{D}}}) | \mathcal{Q} \not\equiv \bar{\mathcal{D}}\right). \tag{2}$$

In this case, we cannot simply assume a universal value for this probability, since it largely depends on the fraction of leaf nodes the incorrect document $\bar{\mathcal{D}}$ is participating in. Under the assumption that features vote for unrelated documents uniformly, the above probability can be estimated as

$$\bar{p}(\bar{\mathcal{D}}) := \frac{\#\bar{\mathcal{D}}}{\#leaves} = \frac{\#\bar{\mathcal{D}}}{K^D}, \tag{3}$$

where $\#\bar{\mathcal{D}}$ denotes the number of leaves in which document $\bar{\mathcal{D}}$ is appearing in the respective inverted files. Further, under the simplifying assumption of feature independence, the chance of having k votes for a relevant document is given by a binomial distribution

$$k \sim B(|\mathcal{Q}|, p_1) \quad \text{if } \mathcal{Q} \equiv \mathcal{D}, \tag{4}$$

and the probability of k votes for an irrelevant document $\bar{\mathcal{D}}$ is

$$k \sim B(|\mathcal{Q}|, \bar{p}(\bar{\mathcal{D}})) \quad \text{if } \mathcal{Q} \not\equiv \bar{\mathcal{D}}. \tag{5}$$

In order to obtain a suitable score for each document given an observed number of raw votes, we determine the posterior probability by Bayes' rule, i.e.

$$\frac{P(\#votes = k | \mathcal{Q} \equiv \mathcal{D})}{P(\#votes = k | \mathcal{Q} \not\equiv \mathcal{D})}. \tag{6}$$

Of course, in practice the log-likelihood ratio is utilized. Thus, the score of documents e.g. occupying all leaves is zero or negative. Note that we did not include incorrect votes for a relevant document $\mathcal{D}$ in the consideration above. A simple approximation to include false positive votes for correct documents is to increase the probability $P\left(w(f_i^{\mathcal{Q}}) = w(f_j^{\mathcal{D}}) | \mathcal{Q} \equiv \mathcal{D}\right)$ from p_1 to $p_1 + \bar{p}(\mathcal{D})$. With this definition of the scoring function a natural tradeoff between positive evidence for documents based on visual words and document distinctiveness is achieved.

Determining the visual words for the features extracted from the query image requires traversal of the vocabulary tree and a number of comparisons for the query feature with the node descriptors. Since the features from the query image are handled independently, the tree traversal can be performed in parallel for each feature. Hence, we employ a CUDA-based approach executed on the GPU for faster determination of the respective visual words. The speed-up induced by the GPU (about 15 - 20 on a GeForce GTX280 vs. Intel Pentium D 3.2Ghz) approach allows to incorporate more descriptor comparisons, i.e. a deeper tree with a smaller branching factor can be replaced by a shallower tree with a significantly higher number of branches. As pointed out in [20], a broader tree yields to a more uniform, hence representative sampling of the high-dimensional descriptor space.

3.2. Feature Matching and Pose Verification

After the score of 3D documents with respect to a new query image is determined, the geometric relationship between the top-ranked documents and the query image needs to be established. First, the extracted features in the query image are exhaustively compared with the descriptors associated with the 3D points in the tested document. Since exhaustive SIFT descriptor matching essentially requires a matrix multiplication between large but dense matrices, the run-time performance can be increased by employing modern GPUs as well. Our approach to feature matching consists of a call to dense matrix multiplication in the CUBLAS library with subsequent instructions to apply the distance ratio test and to report the established correspondences.

If enough putative feature matches are obtained, the actual pose for the query image needs to be determined (if such pose exists at all with respect to the currently considered document). We distinguish between two scenarios: if real-time performance is targeted, we assume that the intrinsic parameters of the camera are (approximately) known, hence we can rely on fast RANSAC methods (e.g. [17]) to determine the absolute pose from three point correspondences [5, 7]. If the camera intrinsics (mainly the focal length) are not known, a 4-point perspective pose approach [1] simultaneously estimating the pose and the focal length was recently presented. The major drawback of that method is the rather low run-time performance to generate the hypotheses, that makes it not suitable for a RANSAC procedure. Hence, we discretize a reasonable range for the focal length and apply the standard 3-point algorithm for all potential focal length values. The pose/focal length pair with the highest number of inliers is reported.

4. Results

We evaluate the performance of our proposed view registration approach on two potential applications. In our first experiment we perform inside out tracking. Essentially we match hand-held outdoor videos to a database of known urban 3D models. Here we target at high frame rates, calibrated camera settings are assumed. In the second experiment we take images from online photo collections and compute the according camera poses with respect to 3D models of city landmarks.

4.1. Tracking by Recognition

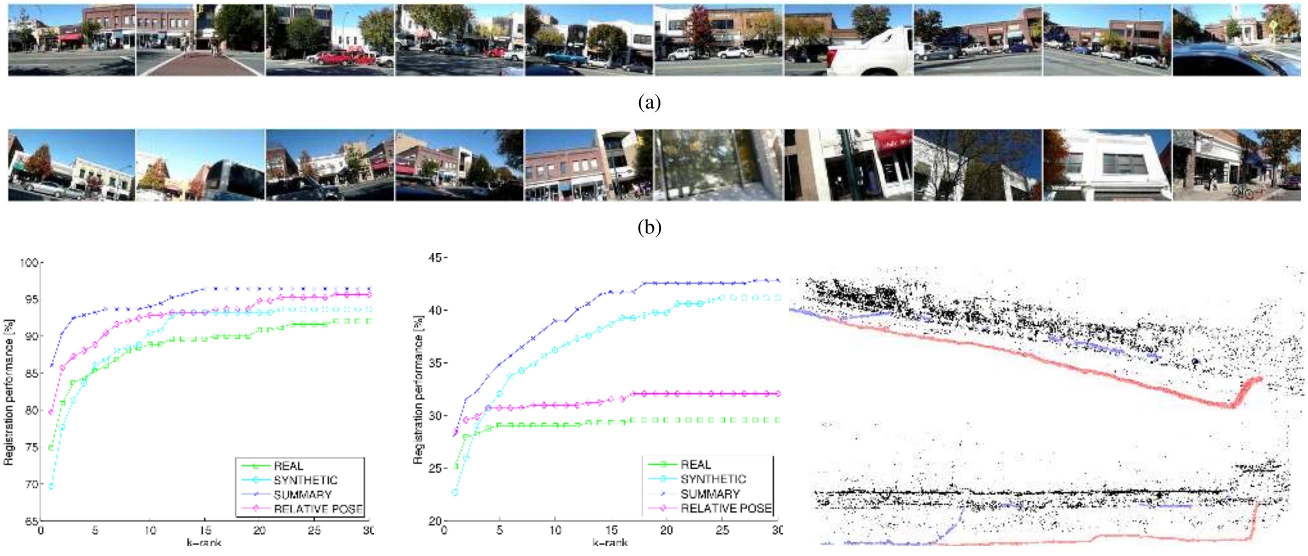
For our first experiment we have reconstructed seven landmarks of a single city from still images taken with a standard consumer digital camera. The camera is precalibrated, images are of resolution 3072×2304 and taken at wide angle (65.4° FOV). In addition, we acquired several video sequences of resolution 848×480 pixel from the same locations as the landmark reconstructions. This data is later used for evaluation. For 3D model reconstruction 1054 images are processed and 400.000 points triangulated from $1.500.000$ SIFT descriptors. After applying mean-shift clustering, the number of descriptors reduces on average to 40% of the original size. These value varies between the seven 3D models, with respect to the scene complexity and the number of redundant views used in the reconstruction process. For each 3D model we estimated an average ground plane and evenly placed synthetic views with a distance of equivalently $2m$ in between. At each grid position we inserted 12 synthetic views with field-of-view $\alpha = 65°$ and resolution 1024×1024 pixel (to model portrait and landscape mode images simultaneously). The heading between cameras is 30°, therefore a full panoramic view at the given position is covered. Since 3D structure is only expected above the ground plane, the cameras are tilted 10° towards the positive horizon. The full set of synthetic and real views contains 11700 documents, which are subsequently reduced to 50% by our compression procedure.

Operation	time $[ms]$
SiftGPU 848×480	33
Vocabulary Tree Traversal K=50 D=3	4
Inverted File Scoring	15
Matching 1600×2500 SIFT key's	$10 \times k$
RANSAC 3-point (up to 500 samples)	$15 \times k$

Table 1. Average timings of our system on a Intel Pentium D 3.2Ghz and a GeForce GTX 280. k is the number of top-ranked documents geometric verification is applied on.

We evaluate the view registration performance by measuring the percentage of video frames for which a valid pose is found after considering the k-th top ranked 3D document from the vocabulary tree scoring. A pose returned by the RANSAC procedure is only considered as reliable, if ten effective correspondences are found. The effective number of inliers is determined in terms of coverage times the raw number of inliers. This is a more robust measure than the standard raw inlier count, since also the spatial distribution of points is taken into account. Of course, the effective inlier number does not reflect a ground truth, but at least in our experiments we did not find false positives among the set of registered frames. Critical thresholds in terms of timings are the maximal number of RANSAC iterations N_{max} and the number of extracted features $|\mathcal{Q}|$ in the input image and 3D document $|\mathcal{D}|$. We set $N_{max} = 500$ (corresponding to a maximal outlier fraction of $\epsilon \approx 0.8$ at a 95% confidence level), $|\mathcal{Q}| = 1600$ and $|\mathcal{D}| = 2500$, which results in execution times of $25ms$ to test a single 3D document on average. By using the publicly available SiftGPU[1] software and only testing the first-ranked 3D document from the vocabulary tree scoring, view registration can be done in real time. Average timings are listed in Table 1. Figures 5(c) and 5(d) show registration performance for two hand-held video sequences (V1,V2) with respect to different 3D document strategies. Our evaluation includes also a comparison to a pure image based method, with the Five-Point algorithm[14] used for pose verification (relevant parameters are adjusted to get comparable timings to the 3-point method). Note, V1 was taken at nearby positions as the views from model reconstruction, therefore higher recognition rates are achieved than for the more challenging sequence V2, that follows a different path approaching the facades. For both cases, the reduced document set based on synthetic and real views gives the best registration performance. Another 3D model and the registered frames of a hand-held captured video are depicted in Figure 6.

[1] `http://cs.unc.edu/~ccwu/siftgpu`



(a)

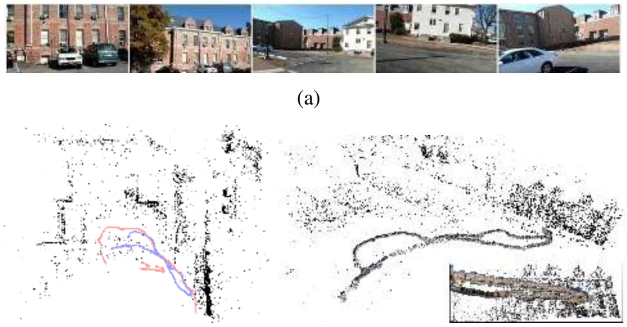
(b)

(c)

(d)

(e)

Figure 5. (a),(b) Some sample frames of two video streams V1, V2 acquired with a hand-held camera. V1 was taken close to original camera position of real views (images from model reconstruction), whereas V2 follows a different path.(c) and (d) show registration performance measured in terms of percentage of registered views after considering the k-top ranked images from the vocabulary tree scoring for V1 and V2, respectively. Each graph shows: *REAL*, set of 3D documents formed by views from model reconstruction; *SYNTHETIC*, synthetic views; *SUMMARY*, reduced set of 3D documents computed by scene compression; *RELATIVE POSE*, image based retrieval with Five-Point relative pose verification. (e) Side and top view showing registered views from video stream V1(red) and V2(blue), respectively.



(a)

(b)

(c)

Figure 6. (a) Some frames of video sequence V3. (b) Locations of successfully registered views from sequence V3 (blue) and V4 (red) to the respective 3D model. From sequence V3, 1878 out of 2760 frames are registered after testing the first ranked 3D document, 2367/3000 from sequence V4.(c) Side view of registered poses from sequence V3 by considering the first ten top ranked 3D documents (every tenth frame shown here).

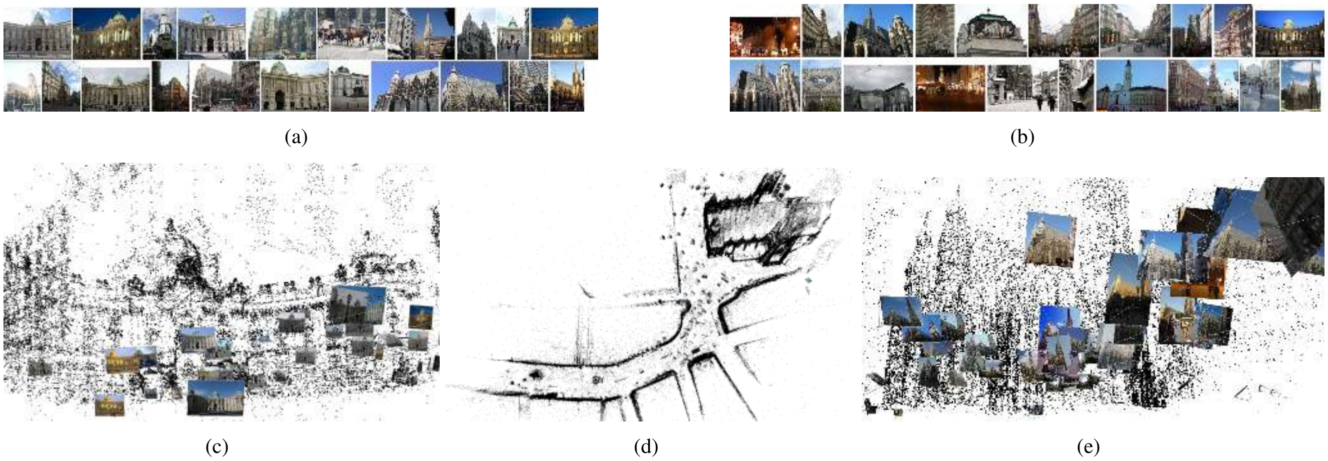
4.2. Community Photo Collections

In our second experiment we apply our view registration technique to images from the web. Again, calibrated cameras were used and three landmarks of Vienna were reconstructed from 117, 128 and 622 images, respectively. For these particular landmarks we gathered a set of images from the Panoramio website geographically associated with these places of interest. We select a relevant subset of 266 images, that have a potential visual overlap with the reconstructed scenes. To determine the camera poses we use the calibrated 3-point method and exhaustively test ten focal lengths with respect to a field-of-view range $[30°..90°]$. By using our approach we are able to efficiently register 165 images from total 266 by considering up to ten top ranked 3D documents. Qualitative registration results are shown in Figure 7.

5. Conclusion

We introduced a novel method for image based real-time scene recognition. The main contributions of the proposed method are (i) the introduction of synthetic views to allow better registration of images taken from novel viewpoints, (ii) an effective document compression procedure for provided real imagery and the synthetic ones in order to reduce the database size, and (iii) a novel scoring function to rank the documents returned by vocabulary tree queries. Video-based inside-out tracking for large outdoor environments can be achieved with real-time performance. The algorithm was tested on a variety of data and showed superior results compared to existing methods.

Acknowledgment: This work was supported by the Austrian Science Fund (FWF) under the doctoral program Confluence of Vision and Graphics W1209.



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

Figure 7. (a) Examples of successfully registered views, and (b) some images that could not be registered (after testing up to 10 top ranked 3D documents) in the database. (c)-(d) Camera poses of registered images to sparse landmark reconstructions of Vienna.

References

[1] M. Bujnak, Z. Kukelova, and T. Pajdla. A general solution to the P4P problem for camera with unknown focal length. In *Proc. CVPR*, 2008.

[2] S. Y. Chen and Y. F. Li. Automatic sensor placement for model-based robot vision. *IEEE Trans. Systems, Man and Cybernetics*, 34(1):393–408, Feb. 2004.

[3] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell*, 24(5), 2002.

[4] E. D. Eade and T. W. Drummond. Unified loop closing and recovery for real time monocular SLAM. In *Proc. BMVC*, 2008.

[5] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Communication Association and Computing Machine*, 24(6):381–395, 1981.

[6] I. Gordon and D. G. Lowe. What and where: 3D object recognition with accurate pose. In *CLOR06*, pages 67–82, 2006.

[7] R. M. Haralick, C. Lee, K. Ottenberg, and M. Nölle. Analysis and solutions of the three point perspective pose estimation problem. In *Proc. CVPR*, pages 592–598, 1991.

[8] A. Irschara, C. Zach, and H. Bischof. Towards wiki-based dense city modeling. In *Workshop on Virtual Representations and Modeling of Large-scale environments (VRML)*, 2007.

[9] D. S. Johnson. Approximation algorithms for combinatorial problems. *J. of Comput. System Sci.*, 9:256–278, 1974.

[10] R. M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, pages 85–103, 1972.

[11] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *Proc. ECCV*, 2008.

[12] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[13] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 65:43–72, 2005.

[14] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–770, 2004.

[15] D. Nistér and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proc. CVPR*, pages 2161–2168, 2006.

[16] M. Pollefeys et al. Detailed real-time urban 3d reconstruction from video. *IJCV*, 78(2-3), 2008.

[17] R. Raguram, J.-M. Frahm, and M. Pollefeys. A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus. In *Proc. ECCV*, 2008.

[18] G. Reitmayr and T. Drummond. Initialisation for visual tracking in urban environments. In *Proc. ISMAR 2007*, pages 161–160, Nov. 13–16 2007.

[19] D. Robertson and R. Cipolla. An image-based system for urban navigation. In *Proc. BMVC*, 2004.

[20] G. Schindler, M. Brown, and R. Szelisk. City-scale location recognition. In *Proc. CVPR*, 2007.

[21] I. Simon, N. Snavely, and S. M. Seitz. Scene summarization for online image collections. In *Proc. ICCV*, pages 1–8, 2007.

[22] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. In *Proceedings of SIGGRAPH 2006*, pages 835–846, 2006.

[23] C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys. 3D Model Matching with Viewpoint Invariant Patches (VIPs). In *Proc. CVPR*, 2008.

[24] J. X. Xiao, J. N. Chen, D. Y. Yeung, and L. Quan. Structuring visual words in 3D for arbitrary-view object localization. In *Proc. ECCV*, 2008.

[25] W. Zhang and J. Kosecka. Image based localization in urban environments. In *Proc. 3DPVT*, pages 33–40, 2006.

[26] Z. W. Zhu, T. Oskiper, S. Samarasekera, R. Kumar, and H. S. Sawhney. Real-time global localization with a pre-built visual landmark database. In *Proc. CVPR*, 2008.