Localization Based on Building Recognition

Wei Zhang and Jana Košecká
Department of Computer Science, George Mason University
4400 University Drive, MS 4A5, Fairfax, VA 22030
{wzhang2, kosecka}@cs.gmu.edu

Abstract

Navigational capabilities of people in urban areas are to a large extent determined by their knowledge of current location. In addition to location information available by means of global positioning sensors, images can provide additional and often complementary information about relative position and/or viewpoint of the person with respect to some known landmarks. In order to enable such functionality, landmarks (e.g. buildings) have to be reliably and efficiently recognized.

In this paper we describe a hierarchical approach for recognition of buildings. At the first stage, we use a novel and efficient representation named localized color histograms. This representation enables efficient retrieval of a small number of candidate matches from the database. At the second stage, the recognition is refined by matching descriptors associated with local image regions. Once the correct building is identified, the relative pose with respect to the building is recovered. The proposed approach is validated by extensive experiments, with images taken in different weather conditions, seasons and with different cameras.

1 Introduction

With the wide dissemination of digital cameras and various navigational aids and sensors, images acquired by the cameras can provide additional means for determining the position of a person in the urban area. This can be achieved in a two stage process, by first acquiring a database of buildings and/or locations of a particular area from different viewpoints, followed by recognition of a new query view by matching it to the closest model in the database. From the application standpoint this problem is of interest for above mentioned navigation task. This problem is also interesting as an instance of object recognition problem. The class of buildings posses many similarities, while at the same time calls for the techniques which are capable of fine discrimination between different instances of the class. From the perspective of applications the natural concern is efficiency and scalability, which will be addressed in this paper.

1.1 Related work

The localization problem as considered in this application comprises two phases: location recognition and relative pose recovery of the camera with respect to the query view. In the presented work, we focus on the recognition aspect and point the reader to some standard techniques for pose recovery. The problem of location and building recognition has been addressed by several authors in the past, mostly considering outdoors scenes. In [9] authors used vanishing direction for alignment of a building view in the query image to the canonical view in the database and proposed matching using descriptors associated with interest regions, followed by the relative pose recovery between the views from planar homographies. Authors in [6] proposed to extract invariant regions and used a set of color moment invariants to represent them. Recognition was based on the number of matched regions. In [3] the recognition part was achieved by matching line segments and their associated descriptors. False matches were rejected by imposing epipolar geometry constraint. An alternative approach was proposed in work of [14] on context-based place recognition. The representation of individual locations was in this case obtained by integrating responses of the bank of filters over coarse spatial regions and fitting a Gaussian mixture model to the responses. This method enabled coarse classification of locations and also exploited spatial relationships between locations captured by Hidden Markov Model. The proposed location model did not allow for actual pose recovery of the camera with respect to the scene.

One of the central issues pertinent to the recognition problem is the choice of suitable representation of the class and its scalability to large number of exemplars. In the context of object recognition, both global and local image descriptors have been considered. Commonly used global descriptors, which provide some invariance to occlusions and

clutter proposed in the past include responses to banks of filters [15] and multi-dimensional histograms [4]. In [12], the authors suggested to improve the discriminant power of plain color indexing technique with encoding of spatial information, by dividing the image into 5 partially overlapping regions. Local feature based techniques have recently become very effective in the context of different object recognition problems. They perform favorably in the presence of large amount of clutter and changes in viewpoint. The representatives of local image descriptors include scale invariant features [8, 11] and their associated descriptors, which are invariant with respect to rotation or affine transformations. From the perspective of the application the efficiency of the approach has to be considered. The methods which employ solely geometric and local feature based matching techniques are often quite slow as pointed out in work of [9]. Therefore, when dealing with large databases, it's desirable to have some simple indexing vector for all models, so that unlikely models can be eliminated in advance.

1.2 Paper overview

In this paper, we propose to tackle the building recognition problem by a two stage hierarchical approach. The first stage is comprised of an efficient coarse classification scheme based on localized color histograms computed over dominant orientation structures in the image. A small number of *best* candidate models is chosen for the second recognition stage comprised of matching scale invariant keypoints and their associated descriptors. Once the most likely model view is found, we recover the relative pose of the camera with respect to the query view. The main contribution of this paper is the localized color histogram descriptor used in the fast indexing scheme. We demonstrate high discrimination capability of the proposed descriptor in extensive experiments using the ZuBuD database [5] which contains 201 buildings of Zürich with 5 views each. These images are taken with two cameras, under different weather conditions, seasons and with deliberate occlusion.

2 Localized color histograms

In order to exploit the efficiency and compactness of the histogram based representations and at the same time gain the advantages of discrimination capability and robustness of local feature descriptors, we propose a representation of buildings which trades these characteristics favorably. The representation is motivated by the observation, that buildings contain constrained geometric structure, such as parallel and orthogonal lines and planar structures. Parallel lines in the world intersect in the image plane at vanishing points. In case of urban environments the dominant line directions are typically aligned with three orthogonal axes of the world coordinate frame.

We propose to compute the color distribution only based on pixels whose orientation complies with main vanishing directions, which are more likely to come from buildings. Consequently, our histograms representation is robust to background change and occlusion, which cause big trouble for standard global histogram techniques. Discriminating power is gained by weakly encoding the spatial information. This is achieved by treating the histograms of the different dominant orientations separately. We coined the representation "localized color histogram" for two reasons: pixels contributing to the histogram are localized in building area; those pixels are divided into several groups and each group has its associated histogram. The whole process will be described to more detail in the following section.

2.1 Dominant vanishing directions

The detection of vanishing directions in the image, which are due to the presence of dominant man-made structures is based on our earlier work where we proposed an efficient vanishing point detection scheme [7]. The detection of line segments is followed by simultaneous grouping of lines into dominant vanishing directions and estimation of vanishing points using expectation maximization algorithm (EM). The EM algorithm typically converges in several iterations, due to effective initialization stage based on peaks in orientation histogram. In our experiments, the number of EM iterations is set to be 10, but we often observe good convergence after less than 5 iterations. For buildings which lack dominant orientations, the vanishing point estimation process is terminated due to the lack of straight line support. In such cases, the first recognition stage would be bypassed and matching based on local descriptors is carried out. We have not encountered this situation throughout our experiments.

2.2 Pixels membership assignment

In the above step, we have obtained the principal orientations of detected line segments. The EM process typically returns two or three vanishing points, which correspond to principal directions $\mathbf{v}_x$,$\mathbf{v}_y$ and $\mathbf{v}_z$ in the world coordinate frame. These directions can be labelled and referred to as left ($\mathbf{v}_x$), right ($\mathbf{v}_y$) and vertical $\mathbf{v}_z$, based on coordinates of their corresponding vanishing points with respect to the center of an image. The label remains the same for a wide range of out-of-plane and in-plane rotation.

Once the vanishing directions are computed each image pixel with gradient magnitude above some threshold is classified as belonging into one of the groups (left, vertical and right) if the difference between its gradient direc-

tion and the principal direction $\mathbf{v}_x$,$\mathbf{v}_y$ and $\mathbf{v}_z$ is less than some threshold τ_o; $\tau_o = 30^o$ in our experiments. Otherwise the pixel is classified as an outlier and removed. Coughlan and Yuille [2] have demonstrated that small objects like bike and robot can be detected using such an outlier model. While sky like background will be removed as middle row of Figure 1 shows, the pixels belonging to background clutter (e.g. trees and grassland) still remain. Note that those pixels are located in area where gradient direction changes frequently, so their neighboring pixels are unlikely to belong to the same group. Hence most of the remaining clutter can be eliminated by doing connected component analysis for each group of pixels separately and removing small connected components.

The final group membership assignments are shown in the third row of Figure 1, where bushes and trees have been eliminated. Note that the color coded membership of foreground pixels remains stable across different views, which enables us to achieve representation which is robust with respect to change of viewpoints. We will next demonstrate that highly discriminative descriptor can be obtained by extracting color information guided by the membership information.

2.3 Indexing vector formation

Color information of (only) pixels which belong to principal directions is considered in the next step. Unlike the traditional color indexing technique where pixel color is represented in 3D RGB space or 2D hue-saturation space, we adopt the 1D hue representation [1]. The RGB is first transformed to (Y, C_b, C_r) defined as

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.2125 & 0.7154 & 0.0721 \\ -0.1150 & -0.3850 & 0.5000 \\ 0.5000 & -0.45400 & -0.0460 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}.$$

The hue value is then calculated by

$$H = arctan(C_b, C_r)/\pi \quad -1 \leq H \leq 1 \qquad (1)$$

The hue histogram of each group of pixels is computed and quantized into 16 bins. In order to avoid the boundary effects, which cause the histogram to change abruptly when some values shift smoothly from one bin to another, we use linear interpolation to assign weights to adjacent histogram bins according to the distance between the value and the bin's central value. Finally, the three histogram vectors h_x, h_y and h_z are concatenated into one indexing vector h to represent each image. The benefit of using only the hue information is two fold: hue histogram representation is robust to illumination change; the indexing vector is more compact compared to other indexing vectors[1]. As our experiments show, the hue histogram is quite discriminative. This is partly due to the object (building) pixels being grouped according to their direction, thus the spatial information is weakly encoded in the indexing vector.

[1]The descriptors surveyed by T. S. Huang in [10] are typically on the order of 10^2.

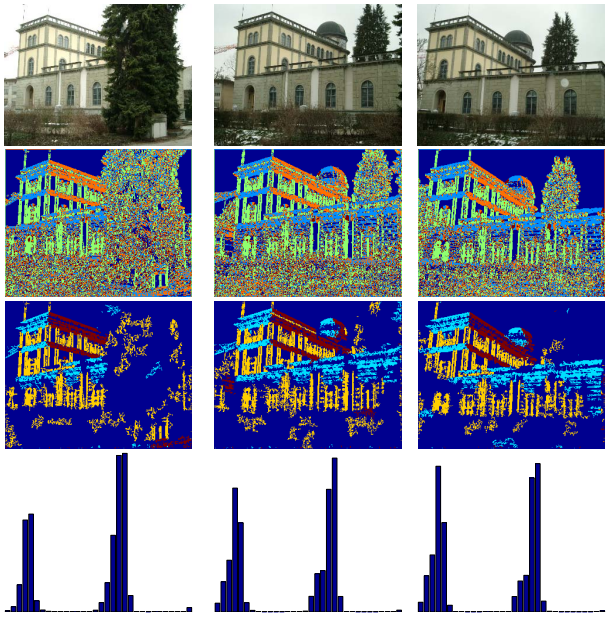


Figure 1. Three views of the same building. Top row: original images. Second row: pixel membership assigned using geometric constraints. Third row: pixel membership assigned after connected component analysis. Background pixels are coded with deep blue, while red, light blue and yellow color represent three group of pixels, respectively. Bottom row: indexing vector for each image

.

2.4 Building retrieval

Given a $16 \times 3 = 48$ dimensional indexing vector representing each image, building retrieval can proceed by comparing histogram vector of the test image and model images. The distance between two indexing vectors is the sum of three individual histogram distances

$$d(h^1, h^2) = d(h^1_x, h^2_x) + d(h^1_y, h^2_y) + d(h^1_z, h^2_z). \qquad (2)$$

There is one subtle issue: because of the viewpoint change, pixels which belong to left group in one view may be in the right group in another image, and vice versa. For instance pixels in the front facade belong to different groups for two images in Figure 3. Consequently, if the distance is computed by the above formula, the result will be sensitive to this viewpoint change.

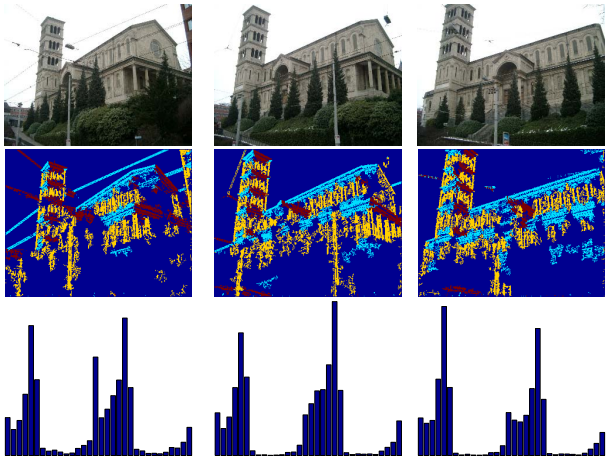


Figure 2. Three views of another building with more background clutter and view point change. Top row: original images. Middle row: pixel membership assigned in the end. Background pixels are coded with deep blue, while red, light blue and yellow color represent three group of pixels, respectively. Bottom row: indexing vector for each image. Note that the color of this building is similar to the building in Figure 1, but their indexing vectors are quite different.

We resolve this problem by combining the histograms of left and right groups, i.e. we consider those two groups of pixels as one large group and represent its color distribution using one histogram. The $16 \times 2 = 32$ dimensional indexing vector still shows high discriminative power in our experiments. The bottom rows of Figures 1 and 2 show the actual indexing vectors. As a byproduct we obtain a shorter indexing vector, which is good for both storage and comparison. Going one step further and combining the three histograms into one, greatly deteriorates the discrimination capability, as shown in Table 1.

To compare a test image to different models, different distance measures can be used. We tried $L1$, $L2$ and χ^2 distance. Though χ^2 distance is not a metric (triangle inequality doesn't hold), we obtained best results using it. Given the indexing vector of a test image h_t and model view h_p, their χ^2 distance is defined as:

$$\chi^2(h_t, h_p) = \sum_k \frac{(h_t(k) - h_p(k))^2}{h_t(k) + h_p(k)} \tag{3}$$

where k is number of histogram bins ($k = 32$ in our case). The small size of the descriptor makes the comparison very fast, which is especially beneficial when dealing with very large databases. As the output of the first recognition stage,

Figure 3. Two views of the same building. Because of the viewpoint change, pixels which belong to the left principal group in the left image will be in the right group in the right image.

we choose a subset of models, which will be further considered in the second stage. The cardinality of the subset will depend on how ambiguous the recognition is. The ambiguity is quantified as:

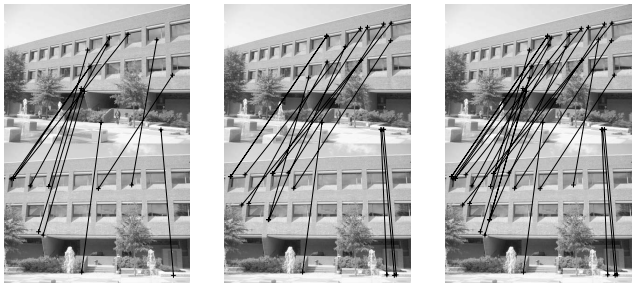
$$Am = \frac{\chi_1^2}{\frac{1}{n-1}(\sum_i \chi_i^2)} \tag{4}$$

where $i = 2, 3, ..., n$, and χ_i^2 is the i_{th} closest distance of the result. We set n to be 5 in our experiments. The ambiguity measure will be very low when the test image is easy to classify and is close to 1 when it's hard to identify. Then the number of results to be listed N_r can be calculated by $N_r = \lceil N_m * Am^2 \rceil$, here N_m is maximum size of the list we allowed, which we set to be 20. Thus when the closest candidate is very distinctive, the first recognition stage may return only single candidate and obviate the second recognition stage. If more candidates are closely matched, we provide a list of candidates with correct model included. The typical size of the list is around 3. When each object model has multiple views in the database, the smallest χ^2 distance among those views is used to compute the size of the list. We report the recognition rates obtained in this first recognition stage in Section 5.

3 Local feature based refinement

The purpose of the second recognition stage is to further refine the results and identify the correct model. In this stage we exploit the SIFT keypoints and their associated descriptors introduced by [8]. For each model image, the keypoints are extracted off line and saved in the database along with the color indexing vectors. After extracting features from a test image, its descriptors are matched to those of the models selected in the first recognition stage.

In the original matching scheme suggested in [8] a pair of keypoints is considered a match if the distance ratio between the closest match and second closest one is below



a) b) c)

Figure 4. Matches obtained using a) distance ratio (11 matches); b) cosine measure (15 matches); c) and result using both measures (24 matches).

some threshold τ_r. In the context of buildings, which contain many repetitive structures, the criterion will reject many possible matches, because up to k nearest neighbors may have very close distances. One option for tackling this issue would be to perform some clustering in the space of descriptors to capture this repeatability as suggested in the context of texture analysis [13]. We instead choose to add another criterion, which considers two features as matched, when the cosine of the angle between their descriptors is above some threshold τ_c. The cosine measure between two feature f_a and f_b is:

$$\cos(\angle f_a, f_b) = \frac{f_a^T f_b}{\|f_a\|_2 \|f_b\|_2}$$

In case multiple features pass τ_c (this happens because of repetitive structure), only one with highest cosine value is kept. Although the matches obtained by this criterion may not be true correspondences[2], they indicate the likely presents of correct matches (which are likely to pass τ_c, but not with highest cosine value). Hence the overall number of correct matches will increase favorably as Figure 4 show, which will benefit the later voting scheme.

Denoting the number of matches between each candidate model image I_j and the test image Q by $\{\mathcal{C}(Q, I_j)\}, j = 1, 2, \ldots, N_r$. the most likely model can be determined using simple voting strategy. In such case the best model is the one with the largest number of successfully matched points:

$$C = \max_j \{\mathcal{C}(Q, I_j)\} \text{ for } j = 1, 2, \ldots, N_r$$

Figures 6 and 5 show two examples where SIFT based matching helps to identify the correct model. Each figure has the test images in the top row. The top four candidates returned by the first recognition stage are listed from left to right in the bottom row. Note that the correct models have many more successful matches than other candidates.

[2]The two points are in correspondence, when they are projections of the same point in 3D world.

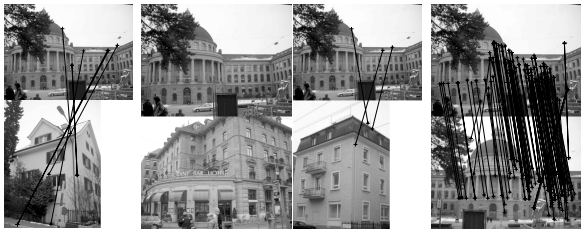


Figure 5. The other three models listed as in top 3 in coarse recognition stage (Figure 8), have much less matches than the correct model.

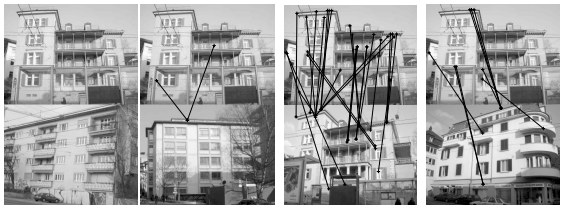


Figure 6. Another example that appearance based technique helps identify correct model. The correct model has much more matches, although it was listed as a third candidate by the coarse recognition stage (Figure 8).

3.1 Model Feature Selection

The number of detected keypoints is often quite different for different images, ranging from hundreds to thousands. For every candidate keypoint, even though it's not a correct match, it still has a small probability ϵ of getting matched. Consequently, for a model image with N keypoints, the probability that none of its keypoints match a keypoint from a test image will be $1 - (1 - \epsilon)^N$. Therefore, models with more keypoints are likely to get more matches. To eliminate this bias, we choose approximately same number of keypoints for each model.

In order to choose the number of features to be approximately the same in all models, we need to consider a quality of each feature. This quality can be measured by its repeatability and distinctiveness. A feature which appears in multiple views of the same model is more repeatable and likely to appear in a new view. On the other hand, a feature which appears in multiple models is less characteristic than those present only in views of single model. The repeatability and

distinctness of each feature f_i^j (i_{th} feature of j_{th} model) can be characterized by the probability $P(f_i^j|m_j)$. This probability represents how likely f_i^j comes from model m_j, which can be obtained using Parzen window approach:

$$P(f_i^j|m_j) \propto \sum_k w_k^j \tag{5}$$

where w_k^l is the contribution of each feature f_k^l located inside a local neighborhood ε, $||f_k^l - f_i^j|| < \varepsilon$. The contribution depends on the distance between f_k^l and f_i^j.

$$w_k^l \propto exp\left(\frac{||f_k^l - f_i^j||^2}{2\sigma^2}\right) \tag{6}$$

where σ is set to be $\varepsilon/3$.

The probability would be higher when the feature is more repeatable and characteristic, and low otherwise. For each model, we keep only those features with $P(f_i^j|m_i)$ higher than certain threshold τ_p; $\tau_p = 0.03$ in our experiments. If the number of features is still large, we keep the top 500 discriminative features. On the average, this procedure discarded around 50% of features from the original feature set reducing the storage requirement and matching computation. Based on our experiments, the feature selection step didn't deteriorate recognition.

4 Pose recovery

Once the correct building has been identified, one can know that he is close to some known landmark. Further information about relative position to that landmark needs to be obtained for navigation purpose. We can either determine the relative pose of the camera with respect to the building or the pose of the camera with respect to the model view. This can be achieved by a number of standard techniques, exploiting single on two view geometric relationships between the views. In [9] authors recover the pose using planar homographies between the model and the test view. We instead recover the pose of the camera, by utilizing rectangular structure based technique proposed earlier [16]. Using robust extraction of dominant rectangular structure, the camera pose can be recovered with regard to the dominant structure, from the homography between the test image and the world structure.

5 Experimental results

The experiments we report on in this section were carried out using the ZuBud database which is described in detail in [5]. The database is comprised of 201 buildings. 5 images per building were acquired with large variation of viewpoints, in different seasons, weather and illumination conditions and by two different cameras. Purposely some occlusions by trees and other objects were included in some images. Some of the images in ZuBuD database were taken with camera rotated 90^o, so we pre-rotated them before experiments, because the 90^o rotation will change the group label.

5.1 Building recognition

To demonstrate the benefits of using localized histogram we compared it with few alternatives: a) form one color histogram using pixels on the detected straight lines only; because those pixels are likely to belong to foreground. b) using all the pixels from the three groups to form one color histogram. The first views of the 201 buildings are chosen as models, the second views are chosen as test images. The results are summarized in Table 1. The first three columns of the table list the hit rate[3] of the top k list, the last column shows the average size of lists for all the test images.

	1^{st}	top 5	list	average size
Line pixels	65.5%	83.5%	88%	5.5
One histogram	69%	89%	92%	5.0
Our approach	83.5%	93%	95%	5.1

Table 1. Summary of the first experiment

As shown in Table 1 with one view per building, we obtain 83% recognition rate, which clearly outperforms the alternatives. We also tried color index based on whole image, the result is even worse than the alternatives. Table 1 also shows the benefit of using variable top k list. While the top 5 list obtains 93% hit rate, an average size of 5.1 list provides 95% hit rate.

We conducted the second experiment using the query image database of ZuBuD. Same as [3] and [6], all 5 views are used as reference images for each building. Out of 115 query images, we got 104 (90.4%) correct recognition result, and 111 (96.5%) of them have correct model in top 5 list. Some results are listed in Figures 7 and 8. The remaining 4 images come from two buildings, as shown in Figure 9, they are rather difficult to recognize. Three of them come from one building, they failed because of significant lighting and viewpoint change between the query and the model views. The 4-th failure is due to dramatic viewpoint change, which is difficult to recognize even for human. We can see that the 32 dimensional indexing vector has very good discriminating capability. The first stage recognition alone shows better recognition rate than the results reported

[3] Hit rate is defined as $\frac{N_c}{N_t}$, where N_c is number of lists which include correct models, N_t is total number of lists.

Figure 7. Example of correct recognized test images by the first stage. The query image and top four results are listed from left to right. Some images are resized for display purpose.

Figure 8. Not correct recognitions but has correct models in list.

by [6] and is comparable to the line matching technique described in [3]. The ambiguity measure we obtained in this experiment is typically small. For 64 query images, only 1 candidate is selected. The maximum list size is 9, and the average size is 2.2087. Therefore, the SIFT based stage only needs to choose from less than 3 models on average.

It is worth mentioning another experiment which uses SIFT based matching directly without first stage; the recognition rate is 90.4% and with 94.8% in top5 list. We can see that the recognition based on the first stage alone is slightly better than the second stage alone. Though their recognition rates are same (note that the correctly recognized buildings are different), the hit rates of the two top5 lists are different. The second stage is necessary because it uses complimentary information which can further improve recognition. Since the second stage only needs to compare few models, the speed problem is greatly alleviated. The combined two stage recognition brings 96.5% recognition rate.

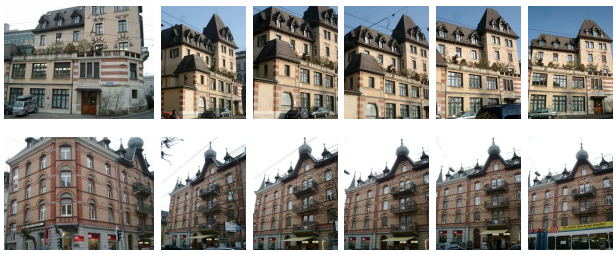


Figure 9. The two buildings which failed. The query images are in left, with their corresponding five model views right. Top: One query view of the building which cause 3 failure. Bottom: the $4th$ failure.

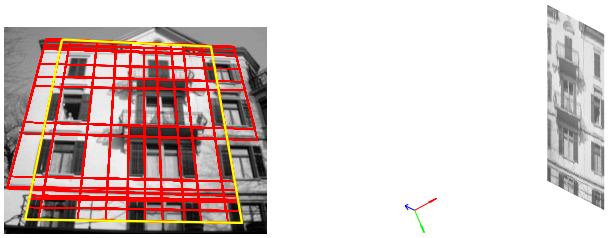


Figure 10. Rectangular structure based pose recovery. Top: the detected structures. The yellow lines delimitate the largest structure. Bottom: Camera pose with regard to the largest structure.

5.2 Pose recovery

Figure 10 shows the pose recovery result for the first test image in Figure 7. The camera pose is shown with related to largest (dominant) structure, the camera coordinate system is depicted by the three arrows. Both the camera orientation and relative distance to the building are obtained.

5.3 Implementation Issues

In the current implementation, we use general purpose hardware and do not use any additional information about approximate position of the user. The overall system we envision as a navigational aid, is similar to the system proposed in [9]. In their setting the query views are taken by a cell phone camera and relayed to the server where the matching and final recognition is done. From the efficiency standpoint, both the building indexing vector and the hierarchical matching scheme is superior to the previously proposed methods [9, 3] and applicable to more general man-made structures. Our approach does not require in the

matching stage any dominant planar structures or repeatable line segments and their associated descriptors. Our current implementation mainly uses MATLAB (with two functions written in C++), where whole processing of a test image takes less than 2 seconds on a $1.5GHz$ notebook computer. If planar motion can be assumed, the processing time can be improved further, since the vanishing directions are known a-priori. The first phase of the proposed matching stage (computation of the indexing vector) is also amenable for implementation using currently available camera cell-phone image processing capabilities such as Nokia 3659 TM.

6 Conclusion and future work

In this paper, we proposed a hierarchical scheme for building recognition which can be used for urban navigation. Localized color histogram is used in the first recognition stage. Our experiments show that it has rather good discrimination capability which is comparable to the local feature based techniques, without the need of finding correspondences. When multiple views of models are available, the selected candidates are more accurate, often correct results are obtained in the first stage. Due to its compact size representation, the methods scales well to large databases. Extraction of the representation vector is also very efficient. In the second stage we used local feature based matching. Candidates selected by the first stage are identified, which further improves recognition. The bias toward model with more features is resolved by a feature selection process.

We are currently investigating the robustness of the color descriptor with respect to large change in illumination and poor image quality obtained by cell-phone camera or PDA. We also plan to implement the remaining parts in C++ to further reduce the processing time.

References

[1] H. Aoki, B. Schiele, and A. Pentland. Recognizing personal location from video, 1998.

[2] J. M. Coughlan and A. L. Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *ICCV (2)*, pages 941–947, 1999.

[3] T. Goedeme and T. Tuytelaars. Fast wide baseline matching for visual navigation. In *CVPR'04*, pages 24 – 29, 2004.

[4] B. S. H. Aoki and A. Pentland. Recognizing places using image sequences. In *Conference on Perceptual User Interfaces*, San Francisco, November 1998.

[5] T. S. H. Shao and L. V. Gool. Zubud-zurich buildings database for image based recognition. *Technique report No. 260, Swiss Federal Institute of Technology*, 2003.

[6] T. T. H. Shao, T. Svoboda and L. V. Gool. Hpat indexing for fast object/scene recognition based on local appearance. In *computer lecture notes on Image and video retrieval*, pages 71–80, July 2003.

[7] J. Kosecka and W. Zhang. Video compass. In *Proceedings of European Conference on Computer Vision*, pages 657 – 673, 2002.

[8] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.

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

[10] Y. Rui and T. Huang. Image retrieval: Current techniques promising directions and open issues. *Journal of Visual Communnication and Image Represent*, 10:39–62, 1999.

[11] C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *Pattern Analysis and Machine Intelligence*, 19:530–535, 1997.

[12] M. Stricker and A. Dimai. Spectral covariance and fuzzy regions for image indexing. *Machine Vision and Applications*, 10:66–73, 1997.

[13] C. S. S. Lazebnik and J. Ponce. Affine-invariant local descriptors and neighborhood statistics for texture recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 649–655, 2003.

[14] A. Torralba, K. Murphy, W. Freeman, and M. Rubin. Context-based vision system for place and object recognition. In *International Conference on Computer Vision*, 2003.

[15] A. Torralba and P. Sinha. Recognizing indoor scenes. *MIT AI Memo*, 2001.

[16] W. Zhang and J. Kosecka. Extraction, matching and pose recovery based on dominant rectangular structures. In *High Level Knowledge in Vision Workshop, ICCV 2003. Nice, France*, 2003.