Hierarchical Building Recognition

Wei Zhang, Jana Košecká

Department of Computer Science, George Mason University, USA

Abstract

In urban areas, buildings are often used as landmarks for localization. Reliable and efficient recognition of buildings is crucial for enabling this functionality. Motivated by the applications which would enhance visual localization and navigation capabilities we propose in this paper a hierarchical approach for building recognition. In the first recognition stage the model views are indexed by localized color histograms computed from dominant orientation structures in the image. This novel representation enables quick retrieval of a small number of candidate buildings from the database. In the second stage the recognition results are refined by matching previously proposed SIFT descriptors associated with local image regions. For this stage we propose a method for selecting discriminative SIFT features and a simple probabilistic model for integration of the evidence from individual matches based on the match quality. This enables us to eliminate the sensitive choice of threshold for match selection as well as the sensitivity to the number of features characterizing different models. The proposed approach is validated by extensive experiments, with images taken in different weather conditions, seasons and with different cameras. We report superior recognition results on a publicly available database-ZuBuD and one additional database of buildings we collected.

Key words: building recognition, man-made structures retrieval, localization, visual landmark recognition

1 Introduction

In this paper, we study the problem of building recognition. As an instance of the recognition problem, this domain is interesting since the class of buildings posses many similarities, while at the same time calls for techniques which are capable of fine discrimination between different instances of the class. The problem is of importance in the context of navigation for robots or visually impaired, where the buildings or similar man-made structures are often used as landmarks for localization in urban areas. In order to tackle the so called kidnapped robot (person) problem, the landmarks (e.g. buildings) have to be reliably recognized from various viewpoints, given the database of previously recorded views. Once the most likely landmark has been found, the pose of the camera between the query view and reference model view can be computed. From the perspective of navigation applications the natural concern is efficiency and scalability.

1.1 Related work

One of the central issues pertinent to the recognition problem is the choice of suitable representation of the class and its scalability to a large number of exemplars.

There is a large amount of literature on general object recognition. The existing methods exploit geometric and/or appearance information, consider part based models or more holistic representations. In the context of the presented work we will review some related works which obtain the desired representations from image appearance, computing either global and local image features.

Global approaches typically consider the entire image as a point in the high-dimensional space and model the changes in the appearance as a function of viewpoint using subspace methods [17]. Given the subspace representation the pose of the camera can be obtained by spline interpolation method, exploiting the continuity of the mapping between the object appearance and continuously changing viewpoint. Alternative global representations proposed in the past include responses to banks of filters [31], multidimensional receptive field histograms [22] and color histograms [29]. These representations do not encode spatial information inherent in the image. Although quite robust to changes in viewpoint and/or scale, they are often not very discriminative. Partial means of encoding the spatial information can be obtained by computing the global descriptors over different image regions separately [30]. [18] Alternative representations in terms of local features have become very effective in the context of different object/category recognition problems. In this case the descriptors are computed only over local image regions, whose location is first determined using various saliency measures. These representations perform favorably in the presence of large amount of clutter and changes in viewpoint. The representatives of local image descriptors include scale invariant features and their associated descriptors, which are robust with respect to affine transformations [12,24,32,13]. The SIFT features proposed by [11] achieved best performance in the matching context based on comparison tests reported by Mikolajczyk and Schmid [14]. Further improvements were proposed by Ke and Sukthankar [6] developed an alternative descriptor by applying PCA to image patches detected by SIFT. Recently, Mikolajczyk and Schmid conducted more comprehensive comparison [15], which showed that SIFT based descriptors outperformed others.

Given a particular representation computed from model view(s) one can proceed with recognition. There is a large body of related work on techniques employed for recognition. Simplest methods for recognition using global descriptors typically use the nearest neighbor classification since each model image is represented by a single descriptor. In case of recognition based on local features very effective and simple method for object recognition is based on voting scheme. In voting however, the contribution (vote) of each match is sensitive to the choice of threshold and matches are considered independent. To alleviate these problems several structured probabilistic models for recognition have been proposed in the past, which account properly for the quality of the match [16], spatial relationships between features as well as global coherence of the hypothesis [19,23]. They further differ in the means of integrating the individual matches to form the final hypothesis of object being present in the image. In our work we will point out that in the context of our problem the previously proposed models are not suitable and propose an alternative discriminative probabilistic recognition scheme which favorably accounts for both the number and quality of the matches.

In the previous sections we reviewed the several object recognition recognition approaches, which most closely motivate the proposed work. Below we review few approaches for tackling the problem of recognition, localization and/or image retrieval of buildings from large image databases. In [27] authors worked on locating buildings in a given image. In the context of CBIR application, consistent line cluster was proposed [9] for classifying images as building/no building images. More challenging problem of detection of man-made structures in cluttered scenes has been addressed in [8], where the authors modelled the image spatial dependencies using Markov Random Fields. In [20] authors proposed matching descriptors associated with interest regions extracted from rectified views, obtained using vanishing point information. Authors in [26] proposed to extract invariant regions and used a set of color moments to represent them, recognition was based on the number of matched regions. In [5] the recognition was achieved by matching descriptors associated with line segments, epipolar constraint was imposed to reject the false matches. The methods which rely solely on local feature based matching are often quite slow as pointed out in work of [20]. In [4] the authors proposed to detect buildings using SIFT descriptors combined with the discriminative feature selection mechanism which reduced the overall complexity of the representation. Alternatively when dealing with large databases, it is desirable to use a global descriptor, to preselect a small number of candidates before carrying out recognition based on local features. For this stage color histogram seems to be a good choice, because of its simplicity and robustness to changes in object's scale, orientation and to some extent viewpoint. In [33] the authors described an approach to recognizing location from mobile devices using image-based Web search utilizing a hybrid color histogram and keyword search technique. However global color histograms are not very discriminative since images with similar color distributions but different content are often present. Furthermore in case of buildings, images contain background (e.g. trees, sky), which can change dramatically with the change of viewpoint. Without further labeling of the object (building) and background area, the global histogram would be affected by background pixels. The localized color histograms introduced in this paper favorably circumvents the two difficulties and demonstrates robustness to moderate light changes.

1.2 Outline

In this paper, we propose to tackle the building recognition problem by a two stage hierarchical scheme. The first stage is comprised of an efficient indexing 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, which is based on the matching of scale invariant keypoints. A simple probabilistic model for recognition is proposed to integrate the evidence from individual matches. We assume that there is only one building to be recognized in each image. The two main contributions of the presented work are the novel localized color histogram representation which enables efficient and accurate indexing scheme and a discriminative probabilistic model for recognition based on local features, which correctly accounts for both the quality and the number of obtained matches. Our approach is tested with the ZuBuD database [25] and an additional database of buildings which we collected. For comparison we show better results than previously reported work [26] [5].

The rest of the paper is organized as follows: In Section 2 we describe the localized color histogram representation and first recognition stage based on this representation. In Section 3 we introduce and justify the probabilistic model for recognition based on local features and in Section 4 we report the experimental results for both recognition stages. Section 5 contains additional discussion and concludes the paper.

2 Localized color histograms

Man made structures like buildings contain geometric regularities, such as parallel and orthogonal lines and planar structures. Parallel lines in the world intersect in the image plane at vanishing points. We propose to compute the color distribution only based on pixels whose gradient direction complies with main vanishing directions, which are likely to come from buildings. Discriminating power is gained by weakly encoding the spatial information, achieved by handling the histograms of the different principle directions separately. In [28], the authors have suggested to improve discrimination power of plain color indexing technique by encoding of spatial information, by dividing the image into 5 partially overlapping regions. Our method is advantageous because it does not rely on fixed regions, thus it's more robust to translation and scale change. The term ”localized color histogram” reflects two facts: pixels contributing to the histogram are typically localized in the building area and are further divided into several groups, based on the vanishing directions they belong to. The contribution of each pixel group is then captured by one of the histograms. The whole process will be described 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 the simultaneous grouping of lines into dominant vanishing directions and estimation of vanishing points using expectation maximization algorithm (EM). During each iteration, the posterior probabilities $p(\mathbf{v}_k \mid \mathbf{l}_i)$ are computed given the currently available vanishing points estimates. Then in the maximization step, the vanishing points are estimated by minimizing negative log likelihood. This yields a linear least-squares estimation problem

$$J(\mathbf{v}_k) \; = \; \min_{\mathbf{v}_k} \sum_i w_{ik} (\mathbf{l}_i^T \mathbf{v}_k)^2 \; = \; \min_{\mathbf{v}_k} \|(WA\mathbf{v}_k)\|^2 \tag{1}$$

where $\mathbf{v}_k$ is a vanishing point associated with k-th direction, $W \in \Re^{n\times n}$ is a diagonal matrix of weights and rows of $A \in \Re^{3\times n}$ are the detected line segments. In our experiments, the EM algorithm typically converges in 3-5 iterations, due to effective initialization stage based on peaks in the orientation histogram. The EM algorithm is initialized by detecting the straight lines in the images and computing their orientation histogram. The number of peaks in the orientation histogram typically serves as good initial guess for number of present vanishing directions. During the EM iteration, small groups will be merged or removed. The line segments which do not align with principal directions are classified as outliers and discarded. For buildings which lack dominant orientations, the vanishing point estimation process is terminated due to the lack of line support. In such cases, the first recognition stage is bypassed and the matching based on local descriptors is carried out. We have not encountered this situation throughout our experiments.

2.2 Pixels membership assignment

The EM algorithm 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. We will refer to these directions 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. Such labels remain the same for a wide range of out-of-plane and in-plane rotation.

Once the vanishing directions are computed each image pixel with its gradient (obtained through convolution of the image with Sobel edge detector) magnitude above

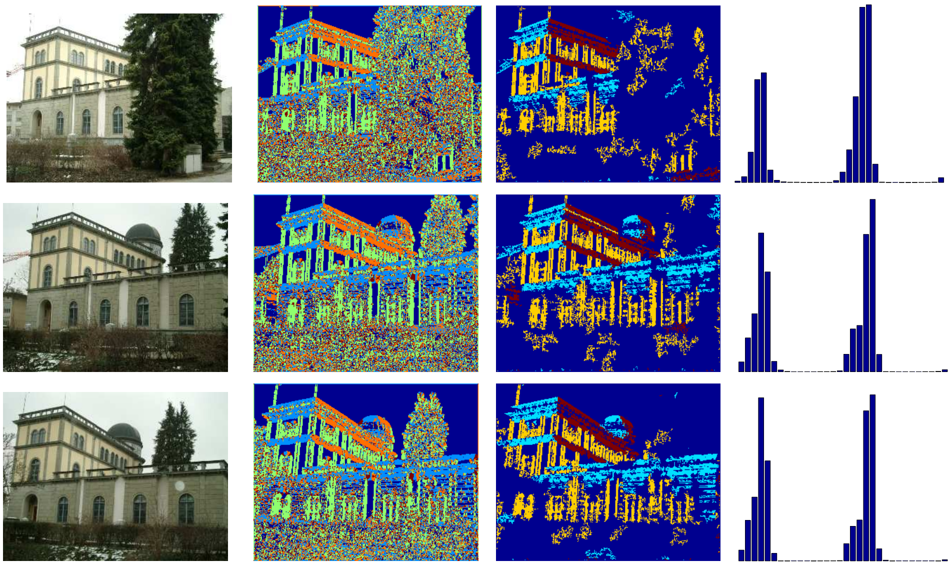

Fig. 1. Three views of the same building. First column: original images. Second column: pixel membership assigned using geometric constraints. Third column: pixel membership assigned after connected component analysis. Background pixels are coded with deep blue, while red, light blue and yellow color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

some small threshold (2 in our experiments)is classified as belonging to one of the groups (left, vertical and right) if the difference between its gradient direction 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 [3] 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 the second column of Figure 1 shows, the pixels belonging to the background clutter (trees and grassland) still remain, because their gradient directions may be aligned with one of the principal direction. Note that those pixels are located in the area where gradient directions change frequently, so their neighboring pixels are unlikely to belong to the same group, *i.e.*, the same group of pixels are unlikely to be connected in the area of background clutter. Hence most of the remaining clutter can be eliminated in the following way: for each group of pixels, find connected components based on those pixels only, then remove connected components whose sizes are less than some threshold. The threshold is set to be half of the image width in our experiment, so it's adaptive to the change of image size. The final group membership assignments are shown in the third column of Figure 1, where bushes and trees have been eliminated. Note that the color coded membership of foreground pixels remains stable across different views. This enables us to achieve a representation which is robust with respect to change of viewpoints. We will next demonstrate that a highly discriminative descriptor can be obtained by extracting color information augmented by the membership information.

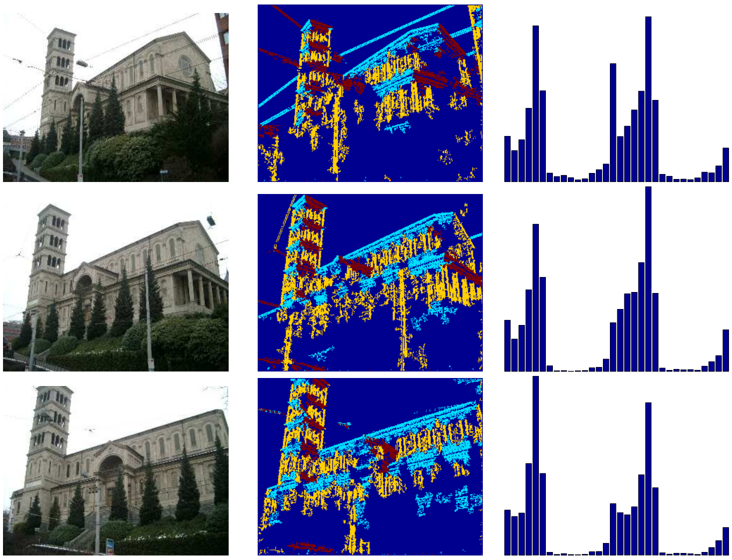

Fig. 2. Three views of another building with more background clutter and viewpoint change. First column: original images. Middle column: pixel membership assignments. Last column: indexing vector for each image. Note that the color of this building is similar to the building in Figure 1, but their indexing vectors are quite different.

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 RGB space or HSV space, we adopt the 1D chromaticity (hue) representation proposed in [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

$$H = arctan(C_b, C_r)/\pi \qquad -1 \leq H \leq \tag{2}$$

The hue histogram of each group of pixels is computed and quantized into 16 bins. very fast. 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: the hue histogram representation is robust to illumination change and the indexing vector is more

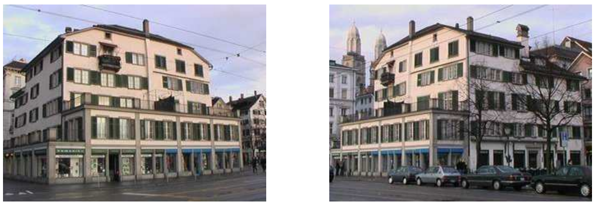

Fig. 3. Two views of the same building. Pixels which belong to the right principal group in the left image belong the left group in the right image.

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.

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

$$d(h^1, h^2) = d(h^1_x, h^2_x) + d(h^1_y, h^2_y) + d(h^1_z, h^2_z). \tag{3}$$

There is one subtle issue: because of the viewpoint change, pixels which belong to the left group in one view may be in the right group in another image, and vice versa. For instance, in Figure 3 pixels in the front facade belong to different groups. Consequently, if the distance is computed by the above formula, the result will be sensitive to this viewpoint change. We resolve this problem by combining the histograms of left and right groups into 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 last columns 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. If going one step further and combining the three histograms into one, the discrimination capability would greatly deteriorate, as shown in Table 1.

To compare a test image to different models we use the χ^2 distance between two histograms. Given the indexing vector of a test image h_t and model view h_p, their χ^2 distance is defined as:

$$\chi^2(h_t, h_p) = \sum_{k=1}^{32} \frac{(h_t(k) - h_p(k))^2}{h_t(k) + h_p(k)} \tag{4}$$

[1] As surveyed by T. Huang in [21], vectors are typically on the order of 10^2.

where k is the index of the histogram bins. 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, 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:

$$Am = \frac{\chi_1^2}{\frac{1}{n-1}(\sum_i \chi_i^2)} \tag{5}$$

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. The number of results N_r to be considered by the second stage is calculated as $N_r = \lceil N_m \times Am^2 \rceil$, here N_m is the maximum size of the list, 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 the 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 performance obtained in this first recognition stage in Section 4.1.

The two histograms which compose indexing vector are meaningful by themselves, they represent the color distribution of vertical and horizontal pixels respectively. Therefore, the first stage can be further accelerated by separating them, e.g. the histogram of vertical group can be used as an indexing vector to choose a subset of models, then histogram of horizontal group can be used to refine the subset. We carried out experiments with this sequential indexing method. The first step is based only on the vertical histogram, N_r is set to 201 (the number of models in our experiments), the indexing hit rate [2] is 99% with average size of subset 38.83 (20% of original models). Then the horizontal histogram is used to further reduce the subset with N_r setting to be 20. The resulting hit rate is 94.5%, which is almost the same as using them together (95% as in Table 1). This sequential indexing scheme has clear advantage for extremely large databases .

3 Local feature based refinement

The purpose of the second recognition stage is to refine the results obtained in the first stage and identify the correct model. In this stage we exploit the SIFT keypoints and their associated descriptors introduced by [12]. The SIFT features represent salient image locations, which are stable across variations in scale. Candidate

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

locations are obtained by searching for local extrema in the pyramid $D(x, y, \sigma)$ obtained by taking a difference of two neighboring images in the scale space:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned} \tag{6}$$

The image scale space $L(x, y, \sigma)$ is first build by convolving the image with Gaussian kernel with varying σ, such that at particular σ, $L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$. This difference operation approximates the convolution of an image with Laplacian of Gaussian $D(x, y, \sigma) \approx \sigma^2 \nabla^2 G * I(x, y)$, with additional scale normalization [10]. Each region is endowed with a 128 dimensional descriptor f, which captures the gradient orientation information of the region, is rotationally invariant and has been shown to be robust with respect to large variations in viewpoint and scale. For each model image, the keypoints are extracted off line and saved in the database along with the color indexing vectors. After extracting keypoints from a test image, its descriptors are matched to those of the models selected in the first recognition stage. Figure 6 shows examples of detected SIFT keypoints. In the original matching scheme described in [12] a pair of keypoints is considered a match if the distance ratio between the closest match and the second closest one is below some threshold τ_r:

$$\frac{d^2(f, f_{1st})}{d^2(f, f_{2nd})} < \tau_r^2 \tag{7}$$

where $f \in \Re^n$ is the descriptor to be matched and f_{1st} and f_{2nd} are the closest and the second closest descriptors from the model database, with $d(.,.)$ denoting the Euclidean distance between two descriptors. The threshold $\tau_r = 0.8$ suggested in [12] was found effective for the general object recognition. This ratio threshold is effective because correct discriminative keypoints often have the closest neighbor significantly closer than the closest incorrect match. In the context of buildings, which contain many repetitive structures (e.g. windows), the above criterion will reject many possible matches, since often multiple neighbors may have very close distances. We chose to add another criterion, which considers two keypoints as matched, when the cosine of the angle between their descriptors f and g is above some threshold τ_c:

$$\cos(f, g) = \frac{f^T g}{\|f\|_2 \|g\|_2} < \tau_c \tag{8}$$

where $\|f\|_2$ represents the L2-norm of a vector. In case multiple features pass τ_c (this happens because of repetitive structures), only the one with the highest cosine value is kept. Although the matches obtained by this criterion may not be the true correspondences [3], they indicate likely existence of correct matches. Using this additional criterion the overall number of correct matches will increase as Figure 4 shows, which benefits the subsequent recognition.

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

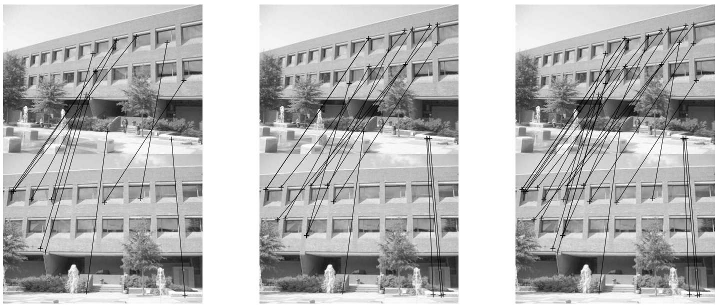

Fig. 4. Matches obtained using: distance ratio (left, 11 matches); cosine measure (middle, 15 matches); combining both measures (right, 24 matches).

In many instances using a simple voting scheme along with the augmented matching criteria described above is sufficient to resolve the ambiguities and pick the correct model from the first stage. This is demonstrated in Figure 5 showing an example of SIFT based matching. The test image is shown 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. In our experiments with ZuBuD database, voting scheme always selects the correct model from the candidates list chosen in the first stage. The reason is that most candidates are rejected by the first stage. As shown in Section 4.2, using SIFT based matching and voting directly without the first stage, the recognition rate decreases.

Even though the voting scheme seems good enough for ZuBuD database, it does have some problems. Note that the basic point matching process depends on a threshold. If the threshold between two descriptors is too high, few matches will be found. On the other hand, if it is set too low, false matches will be introduced. Moreover, once passing the threshold, every match will be treated equally although the matches with higher cosine value are more likely to be correct. To address these two problems with standard voting, next we introduce a simple probabilistic model which takes into account both the number of matches and the quality of the attained matches. This enables us to set relatively loose thresholds and weigh the matches according to their quality. The advantages of the probabilistic model compared to the voting approach are demonstrated in Section 4.3 with an additional database.

3.1 Probabilistic recognition

Several probabilistic formulations have been considered in the past in the context of object recognition. The existing models [23,2,19] differ in their complexity and the number of aspects they account for. In our case the repetitive structures often cause features to be matched to different locations, consequently spatial relationships between corresponding features wouldn't be maintained. Therefore spatial relationships are not taken into account in our model, making it simpler and more

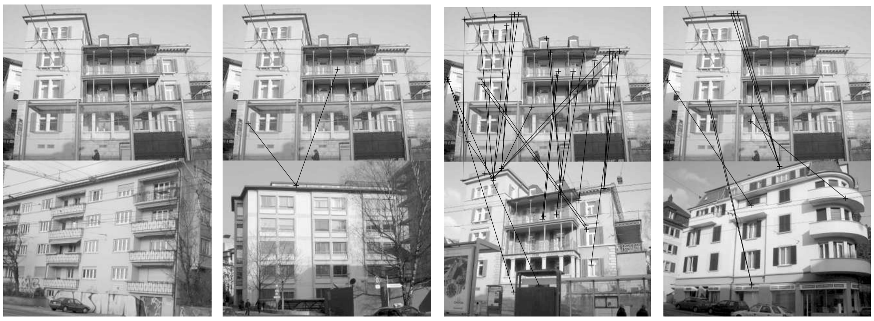

Fig. 5. The correct model has much more matches, although it was listed as a third candidate by the coarse recognition stage (Figure 10). The top row shows the test image, the bottom row shows the top 4 candidates returned by the first stage.

efficient. Even though it seems less powerful not to consider spatial relationships, our experiments show that our probabilistic model outperformed voting scheme without introducing substantial amount of computation.

Note that we need to reconcile the two criteria based on distance ratio and cosine measure in order to attain a single quality measure. For keypoints pairs that get matched by exceeding the distance ratio threshold, their similarity scores can be relatively low, which does not reflect the fact that they are potentially high quality matches. This is reconciled by learning the match quality for these matches. More details about this stage can be found in [34].

In the probabilistic setting the recognition problem can be formulated as the problem of computing posterior probability of the building model given a query image Q, $P(B = j|Q)$. The probability that the j^{th} building appears in a query image Q depends only on the set of matches $\{m_i\} = \{\mathbf{C}_i(Q, B_j)\}$ and can be written as $P(B = j|\{m_i\})$. Since the spatial relationships between individual matches are not considered, the indexes of the keypoints are omitted. Instead of computing $P(B|\{m_i\})$ directly, we consider the probability that Q doesn't contain the j^{th} building which can be expressed as $P(B \neq j|\{m_i\}) = 1 - P(B = j|\{m_i\})$. Here we assume that only one building appears in each query image. In the following section we also omit the model index j to improve the clarity, and use $\bar{B}$ to represent $B \neq j$. Assuming that all matched pairs are independent we obtain:

$$\begin{aligned} P(\bar{B}|\{m_i\}) = \frac{P(\{m_i\}|\bar{B})P(\bar{B})}{P(\{m_i\})} &= \frac{\prod_{i=1}^{n} P(m_i|\bar{B})P(\bar{B})}{\prod_{i=1}^{n} P(m_i)} \\ &= \frac{\prod_{i=1}^{n}(1 - P(m_i|B))P(\bar{B})}{\prod_{i=1}^{n}(P(m_i|B) + P(m_i|\bar{B}))} \end{aligned} \qquad (9)$$

where n is the number of matches and $P(\bar{B})$ is the prior which is assumed to be uniform for all models. The confidence measure obtained from the first stage of recognition can be used in the place of prior. Denoting $\alpha_i = \frac{(1-P(m_i|B))}{(P(m_i|B)+P(m_i|\bar{B}))}$ we have:

$$P(B|Q) = 1 - \prod_{i=1}^{n} \alpha_i P(\bar{B}). \tag{10}$$

The term α_i naturally depends on the quality of the i^{th} match pair and is also related to the number of features representing each model. For every detected 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 $(1-\epsilon)^N$. The larger the value of N, the smaller the probability. Since the number of detected keypoints for different models ranges from hundreds to thousands, models with more keypoints are likely to get more matches (votes). To alleviate the bias caused by different number of keypoints, we propose a simple feature selection scheme which enables us to choose approximately the same number of keypoints for each model. In Section 3.2 we will show how to accomplish this goal.

The rationale of the proposed probabilistic formulation is the following: Each correspondence in $\mathbf{C}_i(Q, B_j)$ serves as the evidence contributing to the probability that a query image belongs to a particular model and $P(B \neq j|m_i)$ represents the probability that this contribution to the final classification is wrong. $P(B \neq j|\{m_i\})$ represents the probability that all the classifications are wrong (i.e. none of the keypoints in Q belongs to B_j). Thus $P(B = j|\{m_i\})$ represents the probability that at least one keypoint is classified correctly, that is to say, at least one keypoint belongs to B_j. Note that we are trying to classify all the keypoints as belonging to one building, if one keypoint belongs to a building, so do the others. Therefore, $P(B = j|\{m_i\})$ represents the probability that B_j appears in the query image Q.

3.2 Model Feature Selection

In order to choose the number of keypoints (features) to be approximately the same in all models, we need to consider the quality of each feature. This quality of a feature can be measured by how it's repeatable across views and how well it characterizes particular model. 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 for one model than those present only in views of that single model. The two quality measures can be characterized by the probability $P(f_i^j|B_j)$, where f_i^j is the i^{th} keypoint descriptor of the j^{th} model. This probability represents how likely f_i^j comes from the model B_j and is approximated by:

$$P(f_i^j|B_j) = \frac{\sum_k w_k^j}{\sum_l \sum_k w_k^l} \tag{11}$$

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

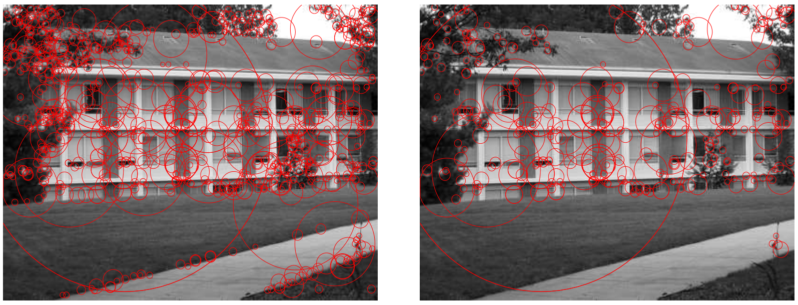

Fig. 6. The original set of 684 keypoints is shown in the left and the selected set of 326 keypoints is shown in the right. The circle center represents location of a keypoint and its scale is represented by the size of circle.

f_k^l and f_i^j as:

$$w_k^l \propto exp\left(\frac{||f_k^l - f_i^j||^2}{2\sigma^2}\right) \tag{12}$$

where σ is set to be $\varepsilon/3$. If a feature is more repeatable for one model, it's neighborhood contains more features from same model, thus the numerator in Equation 11 will be larger and the probability will be higher. When a feature is more characteristic for one model, it's neighborhood will contain less features from other models, thus the normalization factor (the denominator in Equation 11) will be smaller and probability will also be higher.

For each model image, we keep only those features with $P(f_i^j|B_i)$ 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 does not degrade the recognition. This is due to the fact that the removed features are less repeatable thus unlikely to appear in the test image (i.e. their chance of getting matched is low) and they are likely to appear in multiple models, consequently interfering the recognition. As shown in Figure 6, the selected features are mostly located on buildings. Because model images typically capture buildings from different viewpoints, background pixels appearing in one view may not appear in other views and hence are not selected.

3.3 Learning the parameter α

With the influence of the number of model features removed, the relationship between α_i and the match quality d_i (cosine measure) computed for two descriptors can be represented by a function: $F : d_i \rightarrow \alpha_i$. We learn this function in a supervised experiment setting. Given one test image and a number of model images (20 in our experiments, with correct model included), by fixing a particular threshold τ_c, we can obtain a large set of matched pairs between the test image and those

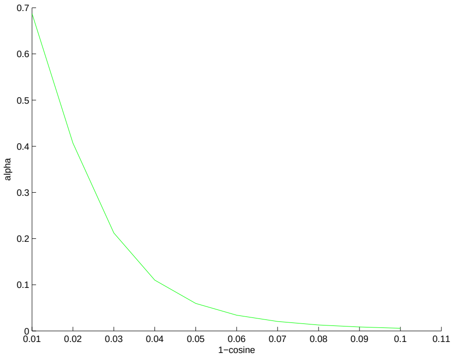

Fig. 7. The relationship between α and $1 - d$.

model images. Suppose the total number of matches for a particular threshold is M and among them we can identify a set of N correct matches. The ratio N/M reflects the probability of true correspondences for that given τ_c. For more accurate estimation, 8 test images were used to obtain the average $\bar{M}$ and $\bar{N}$. And $\frac{\bar{M}-\bar{N}}{\bar{M}}$ is the α^{τ} parameter we want for this τ_c. Repeating this for a number of different τ_c's, we can approximate the mapping in Figure 7. We adopt the following robust function to approximate F.

$$\alpha_i = F(d_i) = \frac{2s^3}{\pi(s^2 + (1 - d_i)^2)^2} \tag{13}$$

This function closely approximates the above mapping when s is 0.3.

4 Experiments

The experiments we report in this section were carried out using mainly the ZuBud database which is described in detail in [25]. 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.

4.1 Validation of the proposed indexing vector

To demonstrate the benefits of using the localized histogram we compared it with few alternatives: a) using pixels from the detected straight lines only to form one color histogram; detected, unfortunately too 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

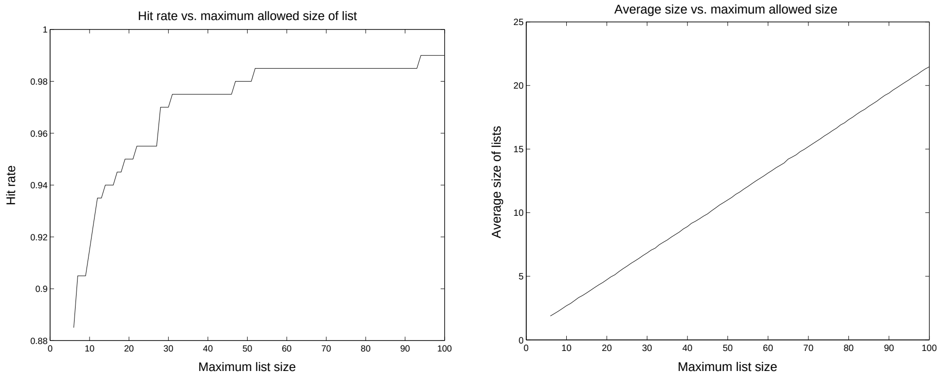

Fig. 8. Hit rate and average list size vary with regard to the maximum allowed list size.

in Table 1. The first three columns of the table list the hit rate of top 1, top 5 and the entire list, the last column shows the average size of lists for all the test images.

Table 1
Comparison with other methods.

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

Table 1 shows the clear advantage of the indexing vector. We can also see the benefit of using variable top k list. While the top 5 list shows 93% hit rate, an average size of 5.1 list provides 95% hit rate.

We also tried to evaluate how the indexing vector performs with regard to N_m, the maximum allowed list size. The results are shown in Figure 8. The average size of lists increases almost linearly with the increasing of N_m, while the hit rate raises quickly at first, then increase very slowly. Based on this curve, we set N_m to be 20 for all our experiments. There are two reasons: the curve is concave downward around 20; the corresponding average size of list is around 5 (top 5 list is often considered in other experiments).

4.2 Experiment with ZuBuD query database

To compare with the existing work, we use the query image database of ZuBuD [25]. With the first stage only, we got 90.4% correct recognition and 96.5% of them have the correct model in top 5 list. Some results are shown in Figures 9 and 10. The remaining 4 images are rather difficult to recognize. Three of them come from one building, they failed because of a significant lighting and viewpoint change between the query and the model views. The 4-th failure is due to a dramatic viewpoint change, which is difficult to recognize even for humans. Figure 11 shows the two buildings. We can see the 32 dimensional indexing vector has a very good discrimination capability. The first recognition stage alone shows better recognition rate than the results reported by [26] (86%). Because of multiple model images in this setting, the ambiguity measure we obtained in this experiment is very small. For 64 query images only 1 candidate is selected. For the actual lists returned, 9 candidates are listed at most, and the average size is 2.2087. All the correct models in the selected candidates lists are identified by the second stage, see Figure 5 for an example. Thus the combined two stage recognition brings 96.5% recognition rate, 4.5% better than the best reported work [5]. Since the second stage only needs to compare few models, the speed problem is greatly alleviated.

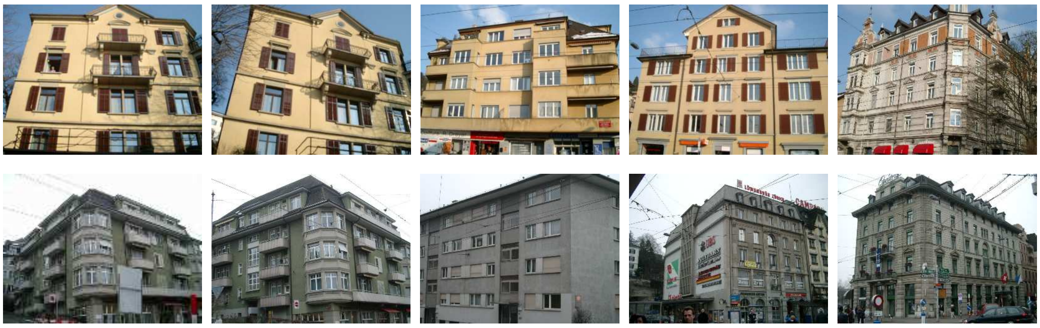

Fig. 9. Correctly recognized test images by the first stage. The query image and the top four results are listed from left to right.

To understand better the advantage of proposed indexing vector, we carried out another experiment with ZuBuD database, which uses SIFT based matching directly without the first stage. The voting scheme is used as a comparison baseline with our original experiment. The result was 90.4% recognition rate and with 94.8% hit rate in the top 5 list. We can see that the first stage alone provides better hit rate than the second stage alone, although their recognition rates are the same (note the correctly recognized buildings are different). The second stage is necessary because it uses complementary information which can further improve the recognition.

4.3 Validation of the probabilistic model

For the experiment with ZuBuD database, since the first stage screens out most of the candidates, the correct model has significantly more matches than other chosen candidates. Thus simple voting scheme is enough to produce good results and the benefit of the probabilistic model is not so apparent. We carried out additional experiment based on a database of 68 buildings with no color information available. Figure 12 shows some examples of this database. The first view of each building is used as the model and the other two views are used as test images. Table 2 summarizes our results. For 6 images which are misclassified by the voting method, the probabilistic recognition gives correct result with the probability score more distinct than the number of matches (see Figure 12). Figure 13 shows one example.

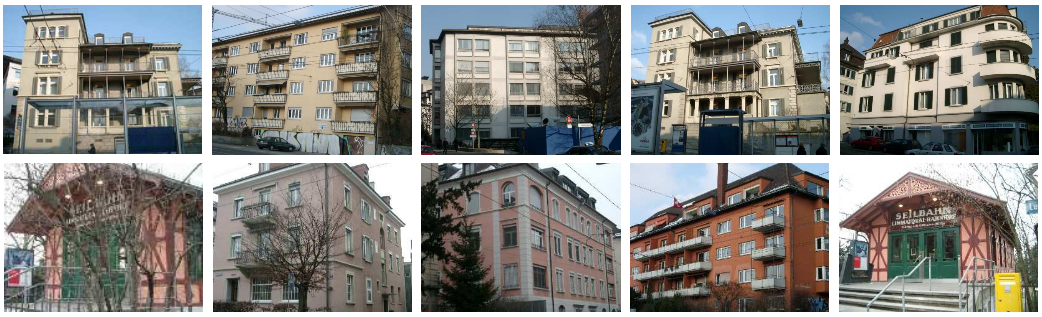

Fig. 10. Incorrect recognition with the correct model in the top 4 list.

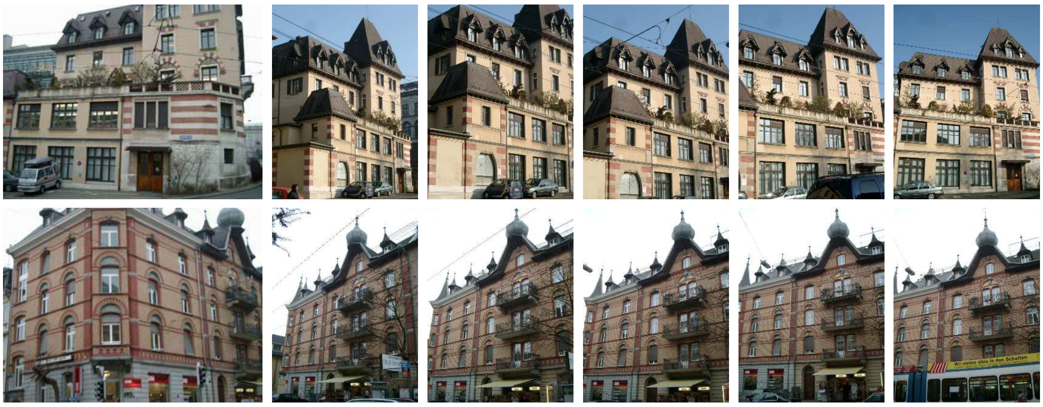

Fig. 11. The two buildings which were failed to be recognized. The query images are in left, with their corresponding five model views right. Top: One query view of the building which has 3 query views in the database, all of the three query views got wrong result. Bottom: the building which causes $4th$ failure. We can see those queries are rather difficult to recognize.

Table 2
Recognition performance of SIFT based matching.

testing method	view 2	view 3
voting	95.5%	88.2%
probabilistic	98.5%	91.2%

Fig. 12. Examples of additional building database.

This clearly demonstrates the benefits of using the probabilistic model for integration of the local descriptor matches.

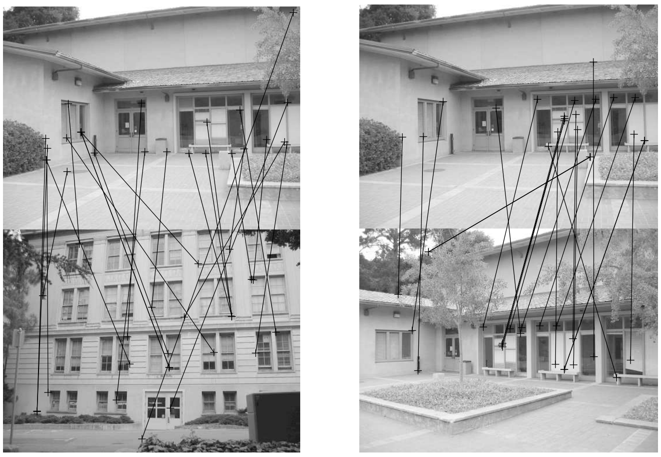

Fig. 13. Wrong model gets 25 matches (left), while the correct model (right) gets 22 matches. U sing voting only would produce incorrect result. The probability of the wrong model however is only 0.03, much less than the correct one which is 0.12.

From the efficiency standpoint our current implementation in MATLAB and C++ takes about 2s for processing a single query image on a 1.5GHz notebook computer with the database of 201 building models. In case a planar motion can be assumed, the processing time can be further improved. Additional speed up can be achieved using fast approximate nearest neighbor matching in both stages. The first phase of the proposed recognition scheme is also amenable for implementation using currently available camera cell-phone image processing capabilities.

5 Conclusion and future work

We have described a hierarchical scheme for building recognition which can be applied for urban navigation and image retrieval. Exploiting the assumption about the presence of man-made structures, in the first stage, an indexing vector which consists of localized color histograms is used to select a small subset of models. Our experiments show that the localized color histogram has rather good discrimination capability which is comparable to the local feature based techniques, without the need of finding correspondences. Extraction of the indexing vector is also very efficient. In the second stage we used local feature based matching to identify the true model from the subset. A simple and efficient probabilistic model is proposed for integration of the matching results. The number of matches and the match quality are both taken into account in the model, making the approach insensitive to the matching threshold. The proposed recognition scheme scales well to large databases due to the compact sized indexing vector.

In the future, we will consider preprocessing of the image with a suitable color normalization strategy, so that more dramatic illumination can be tolerated. In the absence of color information, solely the second local descriptor stage can be applied, with the selection of the discriminative features as well as probabilistic integration of the matches. Once the correct model is identified further geometric constraints can be used for computation of the relative pose of the query image with respect to the model view. We have demonstrated this in our previous work [35]. Providing that the global information about the location of model image is available (as provided for example by GPS), the proposed method could provide a complete solution to the vision based localization problem.

References

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

[2] M. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *Proceedings of ECCV*, 1998.

[3] 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.

[4] G. Fritz, C. Seifert, M. Kumar, and L. Paletta. Building detection from mobile imagery using informative SIFT descriptors. In *SCIA*, pages 629–638, 2005.

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

[6] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2004.

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

[8] S. Kumar and M. Hebert. Man-made structure detection in natural images using a causal multiscale random field. In *CVPR 2003*, pages 119–126, 2003.

[9] Y. Li and L. G. Shapiro. Consistent line clusters for building recognition in CBIR. In *ICPR 2002*, pages 952–956, August 2002.

[10] T. Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2), 1994.

[11] D. Lowe. Object recognition from local scale invariant features. In *International Conference on Computer Vision*, pages 1150–1157, 1999.

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

[13] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the 13th British Machine Vision Conference,*, pages 384–393, 2002.

[14] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *CVPR*, 2003.

[15] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 2005.

[16] R. Mohr, S. Picard, and C. Schmid. Bayesian decision versus voting for image retrieval. In *Proceedings of Computer Analysis of Images and Patterns*, 1997.

[17] S. Nayar, S. Nene, and H. Murase. Subspace methods for robot vision. *IEEE Transactions on Robotics and Automation*, 6(5):750–758, 1995.

[18] G. Pass and R. Zabih. Comparing images using joint histograms. *Multimedia Systems*, 7:234 – 240, 1999.

[19] A. Pope and D. Lowe. Probabilistic models of appearance for 3D object recognition. *International Journal of Computer Vision*, 40(2):149–167, 2000.

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

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

[22] B. Schiele and J. L. Crowley. Object recognition using multidimensional receptive field histograms. *International Journal of Computer Vision*, 2000.

[23] C. Schmid. A structured probabilistic model for recognition. In *Proceedings of CVPR, Kauai, Hawai*, pages 485–490, 1999.

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

[25] H. Shao, T. Svoboda, and L. Van Gool. ZUBUD-Zurich buildings database for image based recognition. *Technical report No. 260, Swiss Federal Institute of Technology*, 2003.

[26] H. Shao, T. Svoboda, T. Tuytelaars, and L. Van 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.

[27] A. Stassopoulou. Building detection using bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 83(5):705–740, 2000.

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

[29] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991.

[30] 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.

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

[32] T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *IJCV*, 59, 2004.

[33] T. Yeh, K. Tollmar, and T. Darrell. Searching the web with mobile images for location recognition. In *CVPR*, 2004.

[34] W. Zhang and J. Kosecka. Experiments in building recognition. In *Technical Report GMU-CS-TR-2004-3, George Mason University*, 2004.

[35] W. Zhang and J. Kosecka. Localization based on building recognition. In *Workshop on Computer Vision Applications for the Visually Impaired, CVPR*, 2005.