A Fast Offline Building Recognition Application on a Mobile Telephone

N.J.C. Groeneweg, B. de Groot, A.H.R. Halma, B.R. Quiroga,
M. Tromp, and F.C.A. Groen

University of Amsterdam
Informatics Institute
Kruislaan 403
1098 SJ Amsterdam
{njgroene, bgroot, ahalma, bquiroga, mtromp, groen}@science.uva.nl

Abstract. Today most mobile telephones come equipped with a camera. This gives rise to interesting new possibilities for applications of computer vision, such as building recognition software running locally on the mobile phone. Algorithms for building recognition need to be robust under noise, occlusion, varying lighting conditions and different points of view. We present such an algorithm using local invariant regions which allows for mobile building recognition despite the limited processing power and storage capacity of mobile phones. This algorithm was shown to obtain state of the art performance on the Zürich Building Database (91% accuracy). An implementation on a mobile phone (Sony Ericsson K700i) is presented that obtains good performance (80% accuracy) on a dataset using real-world query images taken under varying, suboptimal conditions. Our algorithm runs in the order of several seconds while requiring only around 10 KB of memory to represent a single building within the local database.

1 Introduction

In today's society mobile technology plays an important role. With the widespread use of next generation mobile phones a large percentage of the population carries a potent processing unit, which nowadays is accompanied by a digital camera. This setting has presented interesting opportunities for the development of new applications of artificial intelligence, in particular of computer vision. One interesting idea in this direction is an application for a mobile device for the recognition of buildings in an urban environment using only a still camera image as input. Such an application could be used within the tourist industry to provide people with interesting information on photographed buildings, such as opening hours and historical background information, without the need for any additional hardware other than a camera equipped mobile phone.

Although such applications have recently been described within the literature [2] these systems all rely on a client-server architecture in which a user takes a photograph with his mobile device, sends the image to the server which carries

J. Blanc-Talon et al. (Eds.): ACIVS 2006, LNCS 4179, pp. 1122–1132, 2006.

the computational workload and subsequently returns information to the user. Such an architecture has some serious drawbacks. The communication overhead is costly in time and is expensive for the user since mobile network providers usually charge data transfer across their network. Secondly, such an application requires coverage by a mobile network, limiting the application to use within urban environments where coverage is high. A more interesting version of such an application would run locally on the mobile phone without requiring any communication with an external server, assuming the application is acquired on beforehand. However, developing a computer vision application that works in acceptable time on an limited device such as a mobile phone requires a careful selection of techniques, as the mobile platform imposes constraints on both computational power and storage capacity. The application suggested here tackles these practical issues and performs fast offline building recognition in a robust manner.

The aforementioned constraints strongly influence the choice of algorithms and restricts one to a selection of relatively cheap techniques which require as little storage as possible. This implies that great care should be taken to ensure that the representation of buildings in the local database (on the mobile phone) is as compact as possible without affecting the performance of the classification algorithm used. Ideally, one would train a learning algorithm offline and provide the mobile phone with a compact representation of the original database which contains only those image features which are required to correctly classify the buildings in the database. The mobile device would subsequently perform feature extraction/construction on the captured image and feed the result to a compactly represented decision rule, thusly eliminating the need of communication with an external server. On the other hand the application requires an approach that can handle the varying circumstances under which query images are taken. A user might take a picture of a building in which he is interested during any time of the day, under different weather conditions, from any viewpoint. The approach chosen will therefore be required to work under different lighting conditions, be able to handle low resolution images and preferably also be invariant under affine transformations. A solution that meets both the performance criteria and these practical constraints will be presented.

An outline of our approach, which uses local invariant regions, will be discussed, along with a simple baseline classifier providing a reference point for 'naive expectation' performance. The performance of these techniques was evaluated on a personal computer on a standard industrial dataset; the Zurich Building Database (ZuBuD) [10], the results of which will be discussed. On the basis of these results it will be argued that the ZuBuD, although widely used within the literature, is perhaps too easy to serve as a performance reference for our envisaged application. A custom database which does not suffer from this drawback will therefore be introduced for the evaluation of our method on the mobile phone. It will be shown that our method can compete with more computationally elaborate approaches from the literature in terms of classification performance, whilst still running in acceptable time on the low capacity processor of a mobile telephone and requiring only a minimal amount of storage.

2 Local Invariant Regions

In order to identify a building, the intrinsic properties of the object, such as shape, color and texture, have to be compared with those of known buildings. The main difficulty is to get rid of the extrinsic properties, such as scale, viewpoint and illumination conditions.

There are two classes of approaches to object recognition, based on either global or local features. The global approach characterizes the object by its image as a whole. A buildings color distribution is such a global feature. Because many buildings have similar colors it is very unlikely that only color information is sufficient for the task at hand. More sophisticated ways of characterizing objects globally have several problems. In general they are not robust to occlusion, which is unavoidable with objects the size of buildings, not invariant to the viewpoint and they might require a prior segmentation of the image, which is a hard task in the case of buildings.

The other approach characterizes an object by a representations of a number of local features. Usually the features found in an image are compared with the features seen in the database of known objects, where some sort of voting scheme determines which known object fits best. This method is frequently used successfully in literature [2,3,4,6,8,9,11,13]. The advantages of this approach are that it is robust to occlusion and clutter. Local features are more easily described invariantly of scale, viewpoint, and other extrinsic properties.

In the case of building recognition specifically local features prove successful [11,4,2,3,8,5,6]. Recently a more efficient version of SIFT [4] has been proposed, called i-SIFT, for the recognition of buildings on a mobile telephone using a client server architecture [2]. i-SIFT reduces the runtime of the SIFT algorithm by selecting only informative features and reducing the size of the representation of them. Both SIFT and i-SIFT are very robust approaches and i-SIFT in particular has been shown to yield good performance for building recognition[2]. Unfortunately both approaches are less suited for local execution on a mobile phone, since they require numerous Gaussian convolutions. These proved to be a serious execution time bottleneck on the low-end processor available in the average mobile device. Since we explicitly want to avoid the need for client-server communication during the classification process, a method that is more suitable for execution on the mobile phone is required.

Here we will present a novel object recognition method, based on local invariant features, that is optimized for mobile building identification. Our algorithm follows the basic scheme of finding interest points, representing them invariantly and using a voting scheme based on a distance measure between feature representations to determine the best match. The method makes uses of the characteristics of the problem limit the required resources to a minimum, while still performing very well compared to computationally more expensive approaches.

The problem can be described in more detail as follows: Given a low resolution picture of a building taken with a mobile phone, decide which known building is most similar regardless of viewpoint, scale and illumination conditions. From

each known building a set of pictures from various viewpoints is available to create the application. The classification should be done locally on the mobile phone itself within acceptable time. The resolution of the query image is 160×120, since this is the only available resolution Java software was able to capture on the mobile phone we used. Although this severely reduces the amount of information available for classification, it also reduces the amount of computation and memory needed.

There are some assumptions we can make to make that make problem easier. First of all we can assume that query pictures are always taken upright, which is fair since buildings newer appear rotated. Secondly, pictures are taken approximately under the same vertical angle as the training images. Furthermore, the method exploits the fact that buildings often exhibit many repetitions and planar surfaces.

2.1 Feature Detection

Recognition of objects based on local features requires a method to select interest points that are repeated in different images of the same object independently. The more points that are repeated throughout different images of the object the better, because only regions that are found in both training and query images facilitate correct classification. Regions that only occur in either training or query images can only lead to false associations. Unfortunately it is inevitable that a method detects false interest points, that are not repeated. Extrema in the intensity image have proved to work well as interest points [13] and are very cheap to detect using an optimized non-maximum suppression algorithm. Intensity extrema that also occur versions of the image that are blurred with a Gaussian kernel are more reliable, because they are repeated more often [11]. Gaussian blurring is a time consuming operation on the mobile phone we used (several seconds for a single blur), however when a convolution kernel is used in which all coefficients are powers of 2, bit shift operations can be used to create a fast implementation for this specific kernel.

Other types of interest points that can be found in the literature were less suitable, because they were expensive to detect. Extrema in the difference-of-Gaussian scale space [4], for instance require many expensive convolution operations. Corner points are also more expensive, but also have the drawback that they tend to appear very frequently at places where the object in not planar, which makes it more difficult to characterize those local features invariantly.

2.2 Feature Representation

Given a local intensity extremum, the small image region around this interest point is used as the basis of the local feature that is used for recognition. The shape and size of these regions are determined in such a way that they adapt itself to the viewpoint of the object. The border points of the region correspond

roughly to sharp intensity changes while radiating from the local intensity extremum, as described in [13]. Unlike what is done frequently, we do not fit an ellipse to these points. In small regions where the object is planar, the (perspective)projection of this region is approximately affine. Because we have assumed that the images are not rotated and the training images are taken under the same vertical angle, vertical lines in the region also appear as vertical lines in the training images. The remaining differences can be compensated with a vertical shear and both vertical and horizontal rescaling. A parallelogram of which the left and right sides are held vertically that is fitted to the border points of the region, would therefore capture the transformations that effect the appearance of the region. To make the feature more distinctive we double the size of the fitted parallelogram. This approach is in particular well suited for man-made structures.

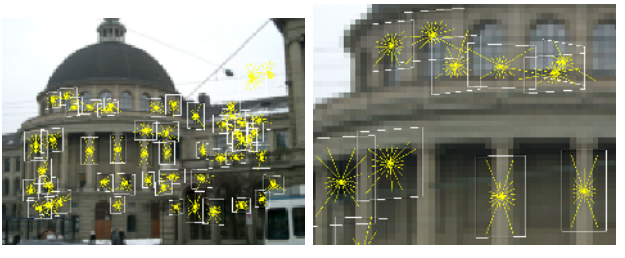

Fig. 1. Examples of found local invariant regions for a building taken from ZuBuD

Unlike others who use 'Generalized Color Moments' introduced by Mindru [7] we represent the regions in a more direct way by transforming the image contents in each region to a fixed size square of size 10×10. The RGB color values of the pixels in this square characterize the region. To make the representation invariant to illumination intensity we divide each value by the sum of the intensities of all pixels in the region. The chosen representation is not rotationally invariant, because this is not needed for our application, making the representation more descriptive than Generalized Color Moments.

In order to reduce the amount of information needed to represent each region, we use Principal Component Analysis to compress the data. Once we have collected all regions from all buildings from the database and represent the pixel data in a normalized region as a vector, we can use PCA to determine a more efficient basis for this feature space. By selecting the first n principal components we can reduce the space required represent a regions significantly and also reduce the amount of computation required for region comparison. We have used the first 30 components, leading to a representation with a space reduction to around 9 % while retaining 96 % of the original variance of the feature space. Theoretically a drop in performance could be expected in case discriminating features got lost due to this less accurate representation, but this does not appear to be the case in practice.

2.3 Reduction of the Number of Regions

Because we have several images of each building, features which are not repeated throughout these images are likely to be noise or at least not characteristic. Discarding such features therefore not only leads to less storage requirements and faster classifications it also gives a higher accuracy. Repeated regions are not only found because there are more views of each building, but also because many buildings have many repetitions in themselves (i.e. a row of identical windows). It would be a waste of space to store all copies of such repeating regions, so only one prototype is stored. This is achieved by clustering the features found for all images of a building. Singleton clusters are removed, because such features are not characteristic or noise. Of all other clusters, only the centroids are kept as prototypes of the cluster members. We used linkage clustering with an empirically determined threshold based on the maximal distance between the instances.

This novel addition reduces the storage needed to around 20 % of the original size, and makes our approach especially suitable for implementation on mobile phones. The clustering step also boosts performance, since it automatically filters out noise that could otherwise reduce classification accuracy.

2.4 Image Classification

When a new image is captured it will undergo the same process of creating a list normalized pixel data of fixed size regions. The data is then projected on the principal component basis. Now the new image can be compared to the database. Classification of the image is performed using a weighted majority voting scheme, wherein each region found within the query image votes for the building belonging to its nearest neighbor in the principal component space.[1]. The weight of the vote equals $1/(d + \epsilon)$, where d is the Euclidean distance to the nearest neighbor and ϵ a small constant number to prevent the vote to have infinite weight. Any monotonically decreasing function would work, however experiments have shown that this function gave the highest performance. The query building will be classified as the building with the highest total vote.

3 Baseline Comparison

In order to obtain a basic 'naive expectation' measure to indicate the relative performance of our approach, a simple classifier based on normalized RGB (*rgb*) histograms was implemented [12].

For this classifier a histogram is built for the r and g channel of every building in the database with 100 bins for each channel.[2] The histogram is then normalized and stored in the database.

[1] Determining the nearest neighbor can be done in $\mathcal{O}(\log n)$[1].

[2] Note that the b channel contains no extra information since $r + g + b = 1$ and can therefore be dropped.

To classify a new image a histogram is constructed of the query image according to the same parameters as the histograms in the database. The χ^2 distance between the histogram of the query image and every histogram in the database is calculated, after which the query image is classified as being the building for which the χ^2 distance is the smallest.

4 Experiments and Results

To test the viability of our local invariant region approach it has first been implemented in MatLab, where it was tested on the 'Zürich Building Database', ZuBuD[10]. The obtained results will be discussed below. The performance of our approach will be compared to that of other approaches from the literature.

4.1 The ZuBuD Database

The ZuBuD consists of pictures of 201 different buildings taken in the city of Zürich, Switzerland. For every building there are five different views within the database, each of which are 640×480 pixels in size. Differences between the views include the angle at which the picture is taken, relatively small scaling effects and occlusions. Examples of images from this dataset can be found in figure 2. The ZuBuD comes with a standardized query set, consisting of 115 images of buildings occurring in the database. These query images have a resolution of 320×240 and are taken with a different camera under different conditions.

Results on ZuBuD. Since on the mobile telephone we only make use of query images with a low 160×120 resolution, we downscaled the ZuBuD images to this resolution as well. We used these images to test our method. For a fair comparison, we also used these downscaled images for the color histogram approach. The results obtained can be found in Table 1, along with performance of other methods found within the literature. Several things can be noted about these results. First of all the local invariant regions classifier we propose performs well on the ZuBuD. The performance of 91 % is identical to that of the i-SIFT algorithm [2] and of the same order of magnitude as that of most other algorithms mentioned in the literature, despite the fact that it uses very little resources, because it has to run within acceptable time on a mobile phone. Secondly it is interesting to note that the rgb histogram based classifier shows a surprisingly high performance on ZuBuD. So high in fact, that it outperforms most methods found in the literature, including our own. Note that this is achieved with only information present in the downscaled images. Intuitively, you suspect that a method based on global color distributions is very sensitive to illumination conditions, scale, occlusions and differences between cameras. Using normalized RGB, the method is invariant to illumination intensity, but the representation is still influenced by the illumination color, which is determined by, amongst others, the weather and

Fig. 2. Examples of different views of a single building from the Zürich Building Database

time of the day. Scale influences the building's representation because the ratio between background colors and building colors is different for a different scale. Occlusions add noise to the representation. When there are different buildings with similar colors, when they are made from the same materials for instance, the discriminative power of color histograms can be expected to be too small to cope with these factors, so the method would fail easily. The high performance of the color distribution approach on ZuBuD can be explained from the fact that the ZuBuD query images are very similar to the reference images in terms of weather, viewing direction and scale. It also helps that many of the buildings in the database are painted in nice pastel colors, which increases the discriminative power of color.

Our own method incorporates shape and texture in its representation of a building, but also relies on color. Illumination intensity is compensated by normalizing the total intensity in a region, but the method is not insensitive to other photometric conditions. In light of the above observations on ZuBuD, we created a custom building database ourselves, which is more realistic for our application. This allowed us to evaluate our method under harder conditions and verify our intuition that global color distributions are not discriminative enough in general. The database also allowed us to test our method on the mobile telephone without traveling to Zürich.

4.2 Custom 'Roeterseiland' Database

The 'Roeterseiland' database consists of images of 7 buildings of the Roeterseiland complex of the University of Amsterdam. From each building between 4 to 11 photos are included, depending on the amount of visible sizes of the building and the diversity between the sides. The images have a resolution of 160×120 pixels and are resized from the originals are shot with a 5.0 megapixel camera. The set of query images consists of 45 images, which are taken independently by someone else using the built-in cameras of a Sony Ericsson K700i

Table 1. Performance of the local invariant regions approach on ZuBud, together with results found in the literature

method	performance (% correct)
HPAT indexing [11]	86 %
SIFT[4]	86 %
I-SIFT[2]	91 %
Local Invariant Regions	**91 %**
Baseline matching [3]	92 %
Sublinear indexing [8]	93 %
***rgb* histograms**	**94 %**
Random subwindows [5]	96 %
LAF [6]	100 %

Fig. 3. Examples of images from the 'Roeterseiland' database (left) and query set (right), showing some differences in viewpoints

and a Nokia 6630 mobile telephones. The query images show the same buildings from different angles, at different scales and with various kinds of weather. In several images the buildings are partially occluded by people, cars or trees. A few examples of the images in the 'Roeterseiland' database are shown in Fig. 3.

Results on the 'Roeterseiland' Database on the Mobile Telephone. The results obtained on the custom 'Roeterseiland' database can be found in Table 2. We begin to note that on this dataset *rgb* does not perform very well, which is more along the lines of expectation than its performance on ZuBuD, reflecting the more realistic quality of the query sets used on our database. Our local invariant region approach still performs quite well, showing 80 % accuracy on the database. This indicates that the performance of our method can not just be ascribed to the lack of difference between training and query images from the ZuBuD, but proves to be reasonably robust. The representation of buildings in our method is invariant to illumination intensity, but not to other photometric properties. The results show that the used features have enough discriminative power to overcome this shortcoming. It might help to use training images taken under identical conditions, to avoid a bias for buildings captured with the same weather for instance.

Table 2. Performance on the 'Roeterseiland' database

method	performance (% correct)
Local invariant regions	80 %
rgb histogram	24 %

The algorithm takes less than 5 seconds to classify a building on a Sony Ericsson K700i, and requires only 63 KB bytes of storage for the database (less than 10 KB for a single building). These statistics show that our approach can be efficiently implemented on a mobile phone and that an application performing building recognition locally on the mobile device is indeed feasible in practice.

The number of buildings in the 'Roeterseiland' database is small, which influences the performance measured positively. The ZuBuD shows that the method's accuracy is scales to a large number of buildings. The execution time of the method consists largely of the constant time it takes to extract the image's features. The comparison of each feature with the database is logarithmic in the number of stored regions, when implemented efficiently [1]. In terms of time performance the method can be considered scalable too. Furthermore, in many cases the number of candidate buildings might not be very large. This is the case when the approximate location of the mobile phone is known, using information about the mobile phone network cell for instance.

5 Conclusions

A new local invariant region algorithm for building classification was proposed that could combine robustness with fast performance on a mobile phone while requiring only limited storage. The algorithm was evaluated on a personal computer on the ZuBuD and was shown to be capable of obtaining state of the art results on this database (91 % accuracy).

In order to obtain a baseline performance score as reference material, a very simple classifier using only *rgb* histograms was also implemented. It was shown that this very simple method can outperform most of the known methods from the literature on ZuBuD, indicating that the query images of the ZuBud database are too easy to differentiate between simple methods and robust advanced methods. We therefore created a small database of our own, consisting of seven buildings and several training and query images for each building. When creating this database we used a high quality camera for the creation of the database and two low-quality built-in mobile phone cameras to create the query images from different viewpoints, at different scales, under different illumination conditions and with realistic small occlusions.

Our algorithm was implemented on a mobile phone (Sony Ericsson K700i) and tested on our custom database along with the baseline *rgb* histogram classifier. For the naive color distribution approach performance performed very poorly on this database (24 % accuracy), whereas the local invariant region algorithm

kept performing well (80 % accuracy). This indicates that the custom database is more challenging and more suitable for obtaining an estimate of the level of performance that can be expected in practice.

References

1. Bentley, J. L., Weide, and Yao, A.: Optimal expected time algorithms for closest point problem. ACM Transactions on Mathematical Software, Vol. 6, No. 4, 1980, pp. 563-580.
2. Fritz, G., Seifert, C. and Paletta,L.: A Mobile Vision System for Urban Detection with Informative Local Descriptors. ICVS '06 **4** (2006) p. 30
3. Goedeme, T., Tuytelaars, T., van Gool, L.: Fast Wide Baseline Matching for Visual Navigation. CVPR'04 **1** (2004)
4. Lowe, G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 2004.
5. Marée, R., Guerts, P.,Piater,J. et al.: Decision Trees and Random Subwindows for Object Recognition. ICML workshop on Machine Learning Techniques for Processing Multimedia Content MLMM '05
6. Matas, J., Obdržálek, S.: Object Recognition methods Based on Transformation Covariant Features, XII. European Signal Processing Conference 2004.
7. Mindru, F., Moons, T. and van Gool, L.: Recognizing color patterns irrespective of viewpoint and illumination, IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 368-373, 1999.
8. Obdrzálek, S. and Matas, J.: Sub-linear indexing for large scale object recognition. Proceedings of the British Machine Vision Conference volume **1**, pages 1-10, 2005
9. Obdrzálek, S. and Matas, J.: Image Retrieval Using Local Compact DCT-based Representation. DAGM'03, 25th Pattern Recognition Symposium September 10-12, 2004, Magdeburg, Germany. Proceedings of the British Machine Vision Conference volume **1**, pages 1-10, 2005
10. Shao, T. S. H. and Gool, L. V.: Zubud-zurich buildings database for image based recognition. Technique report No. 260, Swiss Federal Institute of Technology, 2003.
11. Shao, H., Svoboda, T., Tuytelaars, T. and van Gool, L.: indexing for fast object/scene recognition based on local appearance. Image and Video Retrieval, Second International Conference,CIVR 2003, page 71-80
12. Swain, M. and Ballard D.: Color Indexing, International Journal of Computer Vision, Vol. 7, pp. 11-32, 1991
13. Tuytelaars, T. and van Gool, L. J.: Wide baseline stereo based on local affinely invariant regions. British Machine Vision Conference, 2000.
14. Zhang, W. and Kosecka, J.: Localization based on Building Recognition. Workshop on Applications for Visually Impaired, IEEE Conference, CVPR, 2005