Knowledge Based Single Building Extraction and Recognition

JULIA ÄHLEN
University of Gävle
Department of Industrial Development, IT and Land Management
Kungsbäcksvägen 47 Gävle, 801 76 Gävle
SWEDEN
jae@hig.se

STEFAN SEIPEL
University of Gävle
Department of Industrial Development, IT and Land Management
Kungsbäcksvägen 47 Gävle, 801 76 Gävle
SWEDEN
ssl@hig.se

Abstract: Building facade extraction is the primary step in the recognition process in outdoor scenes. It is also a challenging task since each building can be viewed from different angles or under different lighting conditions. In outdoor imagery, regions, such as sky, trees, pavement cause interference for a successful building facade recognition. In this paper we propose a knowledge based approach to automatically segment out the whole facade or major parts of the facade from outdoor scene. The found building regions are then subjected to recognition process. The system is composed of two modules: segmentation of building facades region module and facade recognition module. In the facade segmentation module, color processing and objects position coordinates are used. In the facade recognition module, Chamfer metrics are applied. In real time recognition scenario, the image with a building is first analyzed in order to extract the facade region, which is then compared to a database with feature descriptors in order to find a match. The results show that the recognition rate is dependent on a precision of building extraction part, which in turn, depends on a homogeneity of colors of facades.

Key–Words: Building, extraction, recognition, Chamfer metrics

1 Introduction

Modern mobile phones have developed into computational devices equipped with high-quality color displays, high-resolution digital cameras, and real-time 3D graphics. The information can be transmitted over data connections and GPS provides for locations of the phone users. Mobile devices enable many types of services such as navigation aid, weather reports, or a tool for restaurant guide. For most of these services the geographical location is an essential part, but that is not enough. For example, a person can be interested in finding information on the object that the device pointing at. Recognition of the target will allow for augmentation in the viewfinder either with graphics or services. In this paper we focus on buildings as a target for recognition. Building recognition can be used in various kinds of applications, including surveillance [1], 3-D city reconstruction, real-time mobile device navigation [2], and robot localization [3].

A number of building recognition systems have been proposed in recent years. However, most of them are based on a complex feature extraction process. Buildings are hard to define since no obvious descriptors can be defined. A human observer easily recognizes the differences between a building and a box with drawers. For a computer vision those two objects have similar qualities, rectangular shape, smaller rectangular shapes inside and homogeneity in colors, at least in most of the cases. That makes it a very challenging task to define a set of specific feature descriptors for a building. Generally, most of the existing building recognition systems adopt a complex feature extraction process to represent an image. For example, both global features and shape [4], texture [5], and local features such as SIFT and SURF [6] are integrated to obtain satisfactory performance. Using more features may bring better results [7], however, it also means the feature representation requires more computational cost and is not easy to implement. In light of this, we investigated whether there is a simple way for feature extraction in the building recognition task. A common approach to segment an object from images is to use a prototype shape, and search for it
in the image. This leads to the task of shape matching, which has numerous applications, such as object localization, image retrieval, model registration, and tracking. One way to represent a shape is by a set of feature points, for example edges. In order to match two shapes, point correspondence on the two shapes have to be established.

Generally, there is always some knowledge about the building that is coded in a set of feature descriptors. We suggest to use knowledge about the surroundings of the building to successfully segment it out. In this paper, we propose a straightforward building recognition model, where the building of interest is extracted from the rest of the image based on global image characteristics and then compared to a database of horizontal gradient images using Chamfer metrics. The fact that only buildings or building parts are present after the first step will significantly improve matching rate since no other interfering objects are compared.

This paper is organized as follows. In Section 1.1, we review related work on building recognition and shape extraction. In Section 2 we present the newly proposed model for building recognition (BRM) in detail. In Section 3, we evaluate the performance of BRM. Section 4 concludes the paper and provides some discussion.

1.1 Related Work

Existing building recognition systems can be roughly divided into two categories: clustering-based methods and feature representation-based algorithms. Clustering-based methods aim to discover the relationships among different image structures by grouping them into different clusters. Zhang and Kosecká [8] proposed a building recognition system based on vanishing point detection and localized color histograms. Detected line segments are grouped into dominant vanishing directions and vanishing points are estimated by the expectation maximization (EM) algorithm. After that, image pixels satisfying some certain constraints will be divided into three groups, namely left, right, and vertical, and localized color histograms will only be computed on these pixels. Because of the fast indexing step using localized color histograms, this method achieved some improvement in efficiency and has attracted the most attention. Feature representation-based algorithms focus on the process of feature extraction in building recognition.

Hutchings and Mayol [9] designed a building recognition system for mobile devices to serve as a tourist guide in the world space. Given a query image, its local features are extracted and described by the Harris corner detector [10] and the SIFT descriptor, respectively. In the matching process, a scale is selected for each query image according to its GPS position. This results in the reduction of search space and the computational cost. However, the system fails in dealing with very large viewpoint changes.

Some models for building recognition are simply using local orientation for feature definition [11]. The described model is very simple, however, it offers a modular, computationally efficient, and effective alternative to other building recognition techniques.

Proposed decades ago, Chamfer matching remains to be the preferred method when speed and accuracy are considered. Chamfer matching was first proposed by Barrow et al [12] and improved versions have been used for object recognition and contour alignment. The basic idea is that given two sets of points whereas \( U = u_i \) and \( V = v_j \) are template and query image edge maps respectively. Chamfer distance between each point \( u_i \in U \) and its closest edge in \( V \):

\[
d_{ch}(U, V) = \frac{1}{n} \sum_{u_i \in U} \min_{v_j \in V} |u_i - v_j| \quad (1)
\]

The template image \( U = u_i \) is superimposed on the distance image \( V = v_j \). An average of the pixel values that the template hits is the measure of correspondence between the edges, called the edge distance. A perfect fit between the two edges will result in edge distance zero, as each template point will then hit an edge pixel. The actual matching consists of minimizing the edge distance. There are many variants of matching measure averages, e.g. arithmetic, root mean square and median.

When using a single template, chamfer matching cannot handle large shape variations. The chamfer distance is not invariant in regard to translation, rotation or scale. Furthermore, the number of templates needed increases with object complexity. Each of these cases has to be handled by matching with different templates. In scenes with cluttered building facades the chamfer cost function will typically have several local minima. In order to make a decision about the object location, orientation and scale, it may be necessary to use a subsequent verification stage [13]. Scalable Vocabulary Tree (SVT) algorithms are tested.
in [14] and a very good performance is presented, however partial occlusion of the building causes the distribution of features to change, thus affecting the entropy based scoring metric, as well as the SVM training. This fact as well as the requirement of a much larger data set for the improved performance, will make this approach not suitable for our study.

2 Building Extraction and Recognition

This section describe data and present the algorithm for building extraction and recognition.

2.1 Data Description

Images used to test and develop our method are taken from Zurich Building Image Database (ZuBuD), which is acquired and prepared by the Department of Information Technology and Electrical Engineering - Computer Vision Laboratory in Zurich, Switzerland. Example of various buildings from ZuBuD used for tests are given in Figure 1.

![Figure 1: ZuBuD images](image1)

All 1005 images of ZuBuD database were captured by digital cameras of resolution 640x480 without flash. This database contains, for each building, five images were acquired at random arbitrary view points.

In addition there are other building images taken by the authors in the city of Gavle mostly at Campus of University of Gavle. Those images are taken at various resolutions at different angles without flash. These images serve to copy the situation when the user is targeting a mobile devise at comfortable distance from the building. In many cases the buildings are occluded by trees or cars. Examples of these images are in Figure 2

![Figure 2: City of Gavle images](image2)

2.2 Algorithm

The approach suggested in this work emerges from assumption that images of buildings contain quite large areas of sky and in many cases large areas of street pavement or other street coverage. Using this assumption the building extraction module processes in the following sequence:

1. Calculate a location of all pixels where blue color is predominant. In RGB images each pixel consists of three values R, G and B. We will keep pixels where blue value is dominant in the comparison to red and green. For that purpose we set a threshold for the blue value and then we scan the image to find locations where the blue value is larger than both the defined threshold and values of red and blue. The image $f = (f_R, f_G, f_B)$ is color filtered according to the following rule:

   $$f = \begin{cases} 
   f & \text{if } f_B \geq T \text{ and } f_B \geq f_R, f_G \\
   0 & \text{if } f_B \leq T \text{ and } f_B \leq f_R, f_G 
   \end{cases}$$

   where $T$ is predefined threshold for the blue channel.

   This step results in an image with only sky and other blue components segmented out. The rest is set to black, see Figure 3.

2. As a next step we calculate labeled regions of all blue objects in the image and perform comparison of color values of the biggest blue region and the remaining labeled objects. We assume that the largest blue region will correspond to a sky, which is most logical in any outdoor scene. We put a constraint on how much color values may differ from the biggest blue region. Means of red, green and blue channels of the biggest blue area are compared to means of the rest of blue regions. Based on the difference value the pixel is set
Figure 3: Original Image and Image with only blue regions.

either to 0 or left unchanged. The result of this operation is shown in Figure 4

Figure 4: Image with only sky regions.

3. In the next step we create a complement image of the previous result in order to get all non-sky regions. After that we label all objects in the image and perform calculations of properties on those objects. These include calculation of area of the region and its bounding box coordinates. The calculated areas are sorted in the descending order so that the biggest labeled area candidate will be evaluated first.

We argue that since we have removed sky pixels from the background, the biggest areas that remain in the image are the building itself and street coverage. There is a reason to assume that street will be situated below the building and thus we test if the biggest labeled area’s bounding box is not below the predefined location coordinates. We define the constraints on location coordinates as following: In an image $f$ of size $(x, y)$ the upper left bounding box coordinate may not have value below $(x/2, y)$. We assume by this constraint that the target building will have upper limits above the mid point coordinate of the image.

If the bounding box constraint is not violated, we state that the object belongs to the building and we keep it as a reference, otherwise we proceed to the next biggest labeled area and do the comparison again. Street coverage and other objects below the building level will be omitted by this step.

4. Since only color values are needed in the next evaluation step, we transform the remaining non-sky image to HSV domain and calculate mean values of the H channel of the biggest building area that we have got in the previous calculation. Then we perform a comparison of Hue mean with the rest of the objects. If the color mean value differences are within a predefined threshold, than we keep those values unchanged. In other cases we set the value to 0. In most tested images this calculation results in an image where only building or parts of the building are extracted, see Figure 5

Figure 5: Image with the extracted building parts.

In the recognition module we perform identification of the building based on the assumption that vertical edges provides enough information about the distances within windows and between windows in the extracted building. As conventional Chamfer matching the method we propose use low level features such as vertical edge map of an image, which is then distance transformed. The approach in our application is to extract only vertical edges since we use window positions and distances within windows of buildings as feature descriptor. Some skewness and rotation will not influence the matching outcome because the distance in window frames will be the same. Vi match edge points with the same gradient direction only. The outcome of this operation is a Chamfer distance value calculated using threshold to avoid outliers and in some cases missing edges:
\[
d_{Ch}(U,V) = \frac{1}{n} \sum_{u_i \in U} \max(\min|u_i - v_j|, \tau) \quad (3)
\]

where \(\tau\) is quite high in our case, since we deal with imperfect building extraction where features that occlude the building will be extracted as well. \(\tau\) is 20\% of the maximum value of the chamfer distance map. The result of this operation is a value of sum of the averages between the template edge map and the extracted building edges. If the chamfer distance value is in between 0.01 and 0.1 we have a match. We deliberately avoid a perfect match value of 0 since it is unlikely to occur because it is difficult to take an image from the exact location and under the exact lighting conditions as a template image. In Figure 6 we see a vertical edge map of the extracted building, edge map of the reference image and an edge distance map of the reference image. Edge map of the extracted building and edge distance map of the reference are inputs to Chamfer transform.

![Figure 6: Edge map of the extracted building, edge map of the reference and edge distance map of the reference.](image)

<table>
<thead>
<tr>
<th>Tested Images</th>
<th>Max Chamfer Value</th>
<th>Min Chamfer Value</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZuBuD</td>
<td>1.5</td>
<td>0.02</td>
<td>0.773</td>
</tr>
<tr>
<td>Gavle</td>
<td>0.105</td>
<td>0.001</td>
<td>0.566</td>
</tr>
</tbody>
</table>

Table 1: Chamfer values after test on ZuBuD database and images of Gavle.

When comparing a large database with images that contain a reference image but at a different angle, the value of Chamfer distance is higher than in the case of building that is similar to a reference. It was not our goal to evaluate run times for the different sets of images, however, we should mention that the ZuBuD database was tested by importing 10 images at a time in the algorithm testing environment. We suggest an interval of matching based on the table above. If we set it to \(0.001 < ch < 1.4\) we get a reasonable amount of matches from both sets of images. The higher value of interval the less matches we face. Thus we choose 1.4 instead of 1.5 as in the table. Using values aimed for ZuBuD database on a set of images taken in Gavle produced 40\% of false matching. The recognition rate depends on the amount of building parts extracted from the image, meaning that the more accurate the extraction of the whole building the more precise the matching. In cases where building colors are in the same tone as the sky or in cases where the sky does not have any bluish tones the recognition rate is down to 0 since we cannot extract building parts. Testing a large database to the 25 templates we could observe that the recognition rate is 90\% for the images where the sky regions contain any blue tone and house coloring is similar to the template. In cases where the image is occluded by trees in most of the building facade we could still have a satisfactory recognition rate since higher parts of the building without trees would serve as an extracted region and thus could have a matching value to a reference in the defined interval.

3 Results

We use color and local position as descriptors to enable the extraction of a building in outdoor scenes. For building matching we suggest a BRM algorithm that uses modified Chamfer distance transform where inputs are an edge map of the extracted building and an edge distance map of the references. We prepared a reference database with 25 templates of edge distance maps of the buildings from both images sets. These templates were inputs for the comparison with the extracted image. By testing the algorithm on images from ZuBuD database and also on 20 images of various buildings taken in the city of Gavle (Sweden) we can present the following results, see Table 1.
4 Conclusion and Discussion

In this paper, we present a novel building recognition model, which contains two stages, the extraction of the building parts and matching part. The matching is done by extraction of local features, such as a vertical edge map and calculation of Chamfer distance on the template and the extracted part. Based on experimental results we can conclude that the algorithm is rather stable in cases with outdoor scenes with similar lightning conditions and building color gamut. Running the algorithm on two different sets of images with different resolution and content of the surroundings we can state that some adjustment to the threshold should be done manually or automatically in order to apply the correct Chamfer distance value as an indication of recognition. The model is easy to implement and requires no manual extraction of the building prior to the recognition task.

Even though the tested algorithm is very effective for building recognition in the described databases, it is important to point out that this model is not meant to be used generally, since there are still drawbacks connected to the nature of Chamfer distance described earlier. It might be regarded as an alternative solution for many vision-based applications. Furthermore, it is clear that incorporating GPS coordinates for the location of the building will lead to a possibility of recognition based on a database with descriptions of buildings or services in that area.

The algorithm part for automatic building extraction provides a satisfactory ground for recognition that follows. For successful recognition of a building is essential to eliminate parts that do not belong to the object of interest. The extraction process is fully automatic in our application and thus we significantly decrease cases with false matching due to similarities of the query image and templates. Another point that should be discussed is a constraint that should be met in order to segment a large enough part of the target. Since color hue is a part of the extraction algorithm, the surroundings of a building should have a considerable difference in color representation, otherwise there is a risk for outliers and false recognition. Incorporating orientation of the local features into the extraction part would allow for more exact target extraction, however, the authors argue that in the current stage, when working with two building databases, the results are within the expected level.

Acknowledgements: The research was supported by the University of Gävle(Sweden).

References:
