HÖGSKOLAN
I GÄVLE

FACULTY OF ENGINEERING AND SUSTAINABLE DEVELOPMENT

Department of Industrial Development, IT and Land Management

Building Extraction in 2D Imagery Using Hough Transform

Rucong Zou

Hong Sun

September 2014

Bachelor thesis, Bachelor, 15 Credits

Computer Science

Study Programme for a Degree of Bachelor of Science in Computer Science

Supervisor: Dr. Julia Åhlén

Examiner: Carina Pettersson

Building Extraction in 2D Imagery Using Hough Transform

by

Rucong Zou

Hong Sun

Faculty of Engineering and Sustainable Development

University of Gävle

S-801 76 Gävle, Sweden

Email:

tbs10rzu@student.hig.se
ofk09hsu@student.hig.se

Abstract

The purpose of this paper is to find out whether Hough transform if it is helpful to building extraction or not. This paper is written with the intention to come up with a building extraction algorithm that captures building areas in images as accurately as possible and eliminates background interference information, allowing the extracted contour area to be slightly larger than the building area itself. The core algorithm in this paper is based on the linear feature of the building edge and it removes interference information from the background. Through the test with ZuBuD database in Matlab, we can detect images successfully. So according to this study, the Hough transform works for extracting building in 2D images.

Table of Content

1. Introduction..........1
1.1Aims..........1
1.2 Research Question..........2
2. Theoretical Background..........2
2.1 Image Processing Techniques..........3
2.1.1 Histogram Processing..........3
2.1.2 Histogram Equalization..........4
2.1.3 Binarization..........5
2.1.4 Edge Detection and Line Extraction..........6
2.1.5 Gradient operator..........7
2.1.6 The Sobel Operator..........7
2.2 Hough Transform..........8
2.2.1 Basic Principle of Hough Transform..........9
2.2.2 Straight Extraction..........10
3**. Methodology of Building Extraction..........12**
3.1 Data Collection..........12
3.2 Preprocessing..........13
3.3 Edge Detection and Line Extraction..........14
3.4 Building Identification..........15
3.5 Location of Buildings in Images..........16
4. Result and Discussion..........18
4.1 Result..........18
4.2 Discussion..........21
5. Conclusions and Future perspective..........22
5.1 Conclusions..........22
5.2 Future perspective..........22
Acknowledgements..........24
References..........25
Appendix..........27

1. Introduction

This thesis reviews briefly the study of building image extraction and recognition and the research concerning building image characteristics. It introduces the basic algorithms based on geometric features of typical buildings in image processing. This paper also aims to examine the key steps of the core algorithm stated above and contemplates on how to optimize the image processing in building recognition and extraction.

As an essential part of Digital City system [1], a typical application of image recognition and extraction, image processing in building recognition and extraction has been attracting increased attention. It is used to narrow the search scope of detection, matching and identification for building features, to improve the efficiency of building recognition and 3D reconstruction, and to establish a building texture library in the hope of realizing automatic building texture mapping function. Under this circumstance, it is necessary to design the algorithm that captures information and statistics of the building area effectively.

The image processing has been used widely in many fields. Usually, in some specific projects, the specific item or area of a image is the study target, it occupies a certain area of the whole image and it differs from other surrounding area in the image greatly or precisely that human eyes won't be able to perceive the difference. This thesis we will take only building as subject of interest.

In this thesis we use image extraction to identify and locate buildings in one image. It reduces a large amount of data to be processed in the subsequent image analysis and recognition process while retaining characteristic information of image structure to achieve a better understanding of the information compressed within images, as described in Chen's study, "Remote sensing image recognition and change detection of buildings," [2]. The authors argue that locate buildings precisely is an important part in image engineering because the quality of recognition and extraction determines the effect of the following image processing. Therefore, it is of vital importance to locate buildings accurately and precisely.

1.1Aims

This paper is written with the intention to come up with a building extraction algorithm that captures building areas in images as accurately as possible and eliminates background interference information, allowing the extracted contour area to be slightly larger than the building area itself. The core algorithm in this paper is based on the linear feature of the building edge and it removes interference information from the background. Finally, in the straight line extraction phase, straight line extraction algorithm based on Hough transform is adopted.

1.2 Research Question

Is Hough transform helpful to extract buildings from images ?

2. Theoretical Background

Such as the center of the crowded areas, population, resources, environment and social economy, cities serve as hubs for political, economic and cultural activities of human beings. They are also the congregation of all kinds of information. With the increasingly speedy development of information technology, two-dimensional imaging depiction of cities could no longer satisfy modern people. As a result, Digital City Mode (DCM) has become the hotspot of vivid three-dimensional description of cities [3].

Integrated with such advanced technologies as Geographic Information System (GIS), Remote Sensing (RS), multimedia and virtual simulation, Digital City Mode could automatically capture information about infrastructures and functional mechanisms in the city and monitor their dynamic changes to facilitate related decision making process. Information about building tops can be obtained from aerial and remote sensing images. The two-dimensional GIS could provide entity statistics of buildings while ground shooting images could provide information about profile, outline features and the texture of buildings.

Human creativity and production tools have generated objects that are new to the Nature in the past millions of years. Such objects like buildings, aircrafts and furniture have refined characteristics and obvious geometrical structures. It is thus possible to extract their edge, contour, texture and regional characteristics information in images and analyze the collected information for meaningful usage [4].The extraction of straight line falls into random distribution. Normally, geometry elements are employed to describe buildings.

There are a lot of researchers that study of building image feature extraction. Shackelford used classified images of pixels as input to draw buildings with the context semantic feature parameters and space, which proved to be effective [5]. Hofmann and others used high resolution IKONOS image data combined with airborne laser scanner methods to obtain high resolution digital cadastral map and digital elevation model of urban buildings [6].

Tang Liang, XieWeixin and others extracted edge information of buildings by Canny edge detection operator and identified buildings with a variety of spatial information such as image and object [7]. Sobel Operator is a step of a digital image is mainly used for edge detection and calculation. Technically, it is a discrete first-order differential operator, used to calculate the degree approximate of image brightness function. To use sobel operator at any point of image, it will produce a corresponding gradient vector or a normal vector. Irwin Sobel proposed ("A 3x3 Isotropic Gradient Operator for Image Processing") topic in a doctoral seminar. Yingdong Qu, Chengsong Cui, Sanben Chen and Jinquan Li narrated how fast it is possible to detect sub pixel edge using Sobel Operator method [8]. Because it is detected quickly, while we have a lot of images from ZuBuD, so using this method saves time.

The general process is that making image enhancement processing, then using Sobel Operator for image edge extraction, resulting in a binary edge image, and then using the Hough transform result in an image to feature extraction curve. For example, an image of the street, separate roads and buildings [9]. It can help us discern whether there are buildings or not in these images. Simonton made use of perceptual grouping method to process edge lines and to seek rectangular array of buildings [10]. Kim chose to work out building shapes by matching extracted results with visual perception. The matching linear combination together is likely to be the shape of the buildings [11]. Sun Xian and Wang Hongqi built a network to associate the image segmentation and object recognition, collected target characteristics with classifiers and took tag confidence measures to realize building extraction [12].

TaoChao, Tan Yihua segmented images in the first place to extract candidate area, and then extracted buildings with shadow auxiliary information or sample of the same scenario [13].Gaoput forward three new characteristics and used genetic algorithm to classify [14]. Karantzalos took advantage of empirical knowledge about buildings to segment and extract within the horizontal and variation framework [15]. All these studies take the approach of first segmentation and then extraction. They differ in specific measures and information source.

2.1 Image Processing Techniques

The method of Histogram Equalization and binary Image Preprocessing is very common and efficient algorithms. In this paper, when we building an image preprocessing. According to the characteristics of buildings images, based on image binarization and binary image segmentation algorithm, we can save the image edge features and effectively remove the bottom of the interference information. The following chapter will briefly introduce these two methods.

Usually, images in image systems get degraded in the process of transmission and transformation. Distortion of optical system, relative motion, air flow in shooting or filming will reduce some extent image quality. Therefore, it is necessary to conduct basic disposition such as image demising, image enhancement and image restoration to better analyze images in later procedures. This is called image processing technology in a broad sense.

2.1.1 Histogram Processing

Histogram is used to represent the statistical relationship between each gray level and its occurrence frequency (the number of the gray-scale pixels) in digital image. In rectangular coordinate system, horizontal grayscale is generally represented by X-coordinate and grayscale frequency by Y-coordinate. Assume that the grayscale of digital image is [L - 0, 1], the histogram of the digital image can be formulated by this discrete function:

$$\mathrm{p}(r_k) = n_k / n \quad （1）[23]$$

In this formula, r_k stands for the total pixel number of the 'k'thgrey scale; $p(r_k)$ stands for the frequency of the 'k'thgray scale; n stands for the total number of image pixels. In short, $p(r_k)$ gives the probability estimates of the grayscale of r_k , so the histogram provides the distribution of the gray value of original images, and also provides an overall description of all the grayscale in an image. The Gray scale range, grayscale distribution and the average brightness of the whole image can be described by using the histogram, as shown in figure 1.

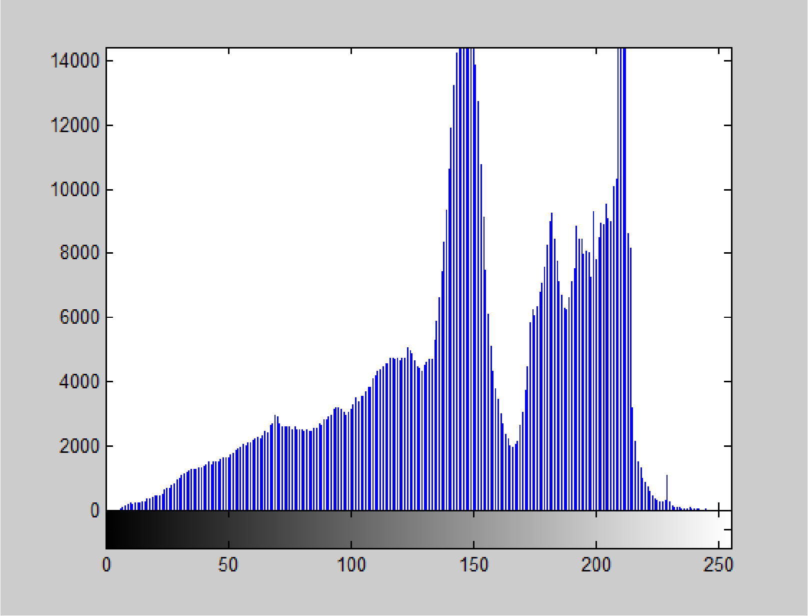
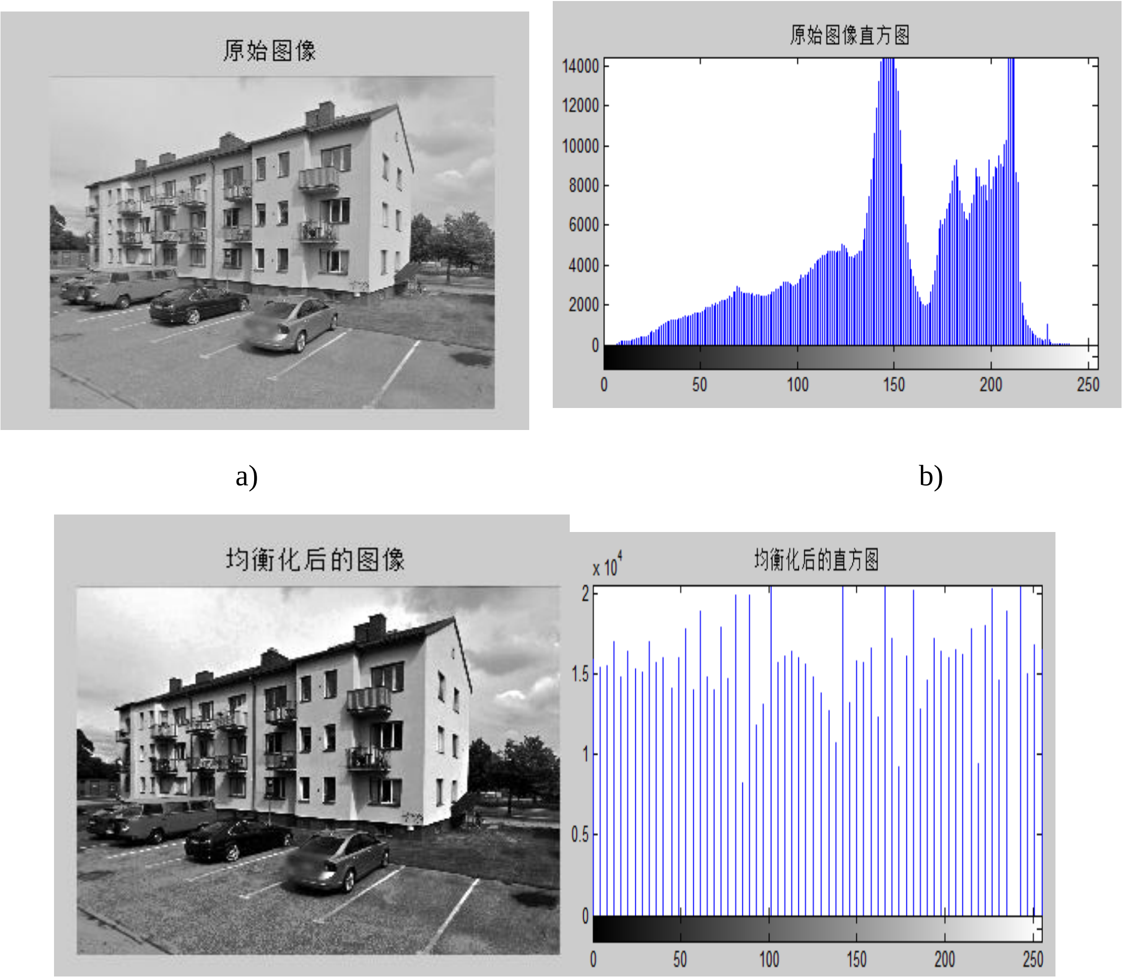


Figure 1 image histogram

2.1.2 Histogram Equalization

The basic idea of equalization of histogram histeq is to change the original image pixel gray value, widen the large number of pixels in the image grayscale, and shrink the small number of pixels of grayscale, the image is evenly distributed in the form of the corresponding histogram transformation, so as to enhance the overall image and contrast and achieve the goal of making the image clear, to get maximum entropy image and the largest amount of information. The example of histeq is given in Figure 2.



a) b)

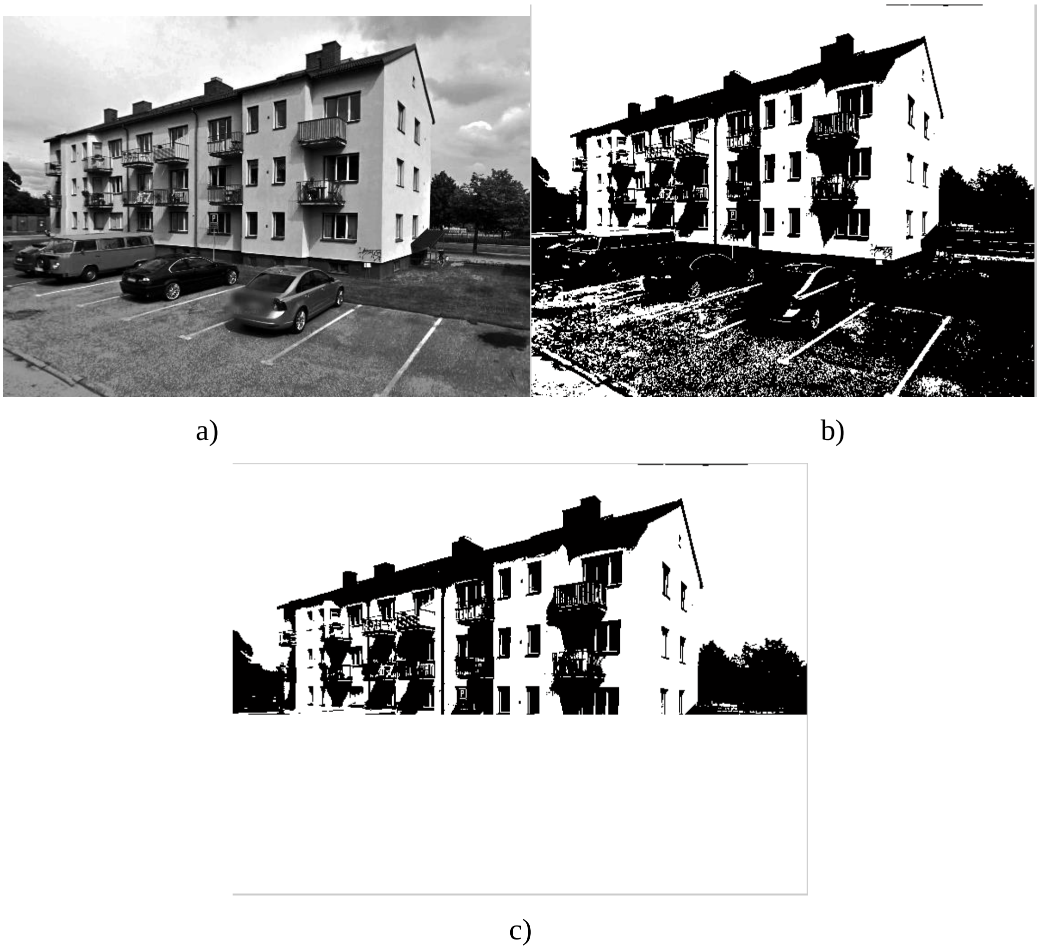
c) d)

Figure 2 a) the original picture, b) the histogram of original, c) the equalization image, d) the histogram of equalization image

2.1.3 Binarization

Digital image binarization is to turn gray image digital signal into figure signal as (0, 1), which is one of the important procedures of preprocessing. By using binarization it is possible to set the gray value with images pixel point to 0 or 255, which makes the whole image showing a clearly black and white effect, making the feature is easy to extract. During the transform, two conditions must be met: 1. Target from its background as obviously as possible; 2. Retain the geometric features of the target as much as possible. There are two different methods of binarization: global thresholding and local thresholding. Global thresholding is comparatively easy. It refers to calculation of the threshold based on the histogram of the whole image [16].

The key part of image binarization is to select a threshold in such ways like whole threshold method, partial threshold and dynamic threshold method according to different image pixel processing and application range of the threshold value. This paper will apply the whole threshold method, which uses a single threshold for image binarization, and this single threshold can be manually set or obtained automatically by the gray histogram. Figure 3(c) threshold shows the comparison of the two pictures before and after binarization.


a) b)

c)

Figure 3 a) Before binarization b) After binarization c) get half after binarization

2.1.4 Edge Detection and Line Extraction

Edge detection of digital image is not only an important basis of image segmentation, target recognition and zone shape extraction, but also an important attribute of image feature extracting during image recognition. In digital image, margin always exists between adjacent domains whose grayscale are different and inconsistent. Therefore, edge detection and line extraction are two important operations during image reading and analysis [23].

Human eyes are sensitive for areas where brightness changes rapidly and for edges where different objects interact. In a certain sense, edges of image contain the majority of image information.

The edges of images refer to areas where the most significant changes take place, changes such as grayscale mutation, color mutation and texture structure mutation and so on. The grayscale section of certain area can generally be regarded as a phase step, where one grayscale jumps to another quite different gray scale with in a very small buffer. Therefore, it can be detected by derivation. There are several frequently-used operators in edge detection like the Roberts operator, the Prewitt operator and the Sobel operator. Differential operations are used in image analysis to enhance grayscale changes. The underlying principle of edge detection is that high differential operation value indicates big grayscale change. These high values can serve as edge strength accordingly. Edge points can be extracted through threshold criterion. Greater differential values than threshold would become edge points.

2.1.5 Gradient operator

Gradient value shows significant changes of image grey value. Different gradient operators correspond to different method of computing image gradient features. Image edge is caused by the discontinuity of grayscale, which can be detected by derivation addressing grayscale changes in certain area of each pixel of images. The first or second order derivative rules are applied to detect edges in marginal areas.

The gradient corresponds to the first derivative and accordingly the gradient operator corresponds to the first derivative operator. For a continuous function f(x,y), its gradient at the point (x,y) can be defined as formula (5):[17].

$$\nabla f = [\frac{\partial f}{\partial x}\frac{\partial f}{\partial y}]^T = [G_x G_y]^T \quad （5）$$

Gradient is a vector whose magnitude and phase can be separately expressed as formula (6) and formula (7) :

$$|\nabla f| = [G_x^2 + G_y^2]^{1/2} \quad （6）$$

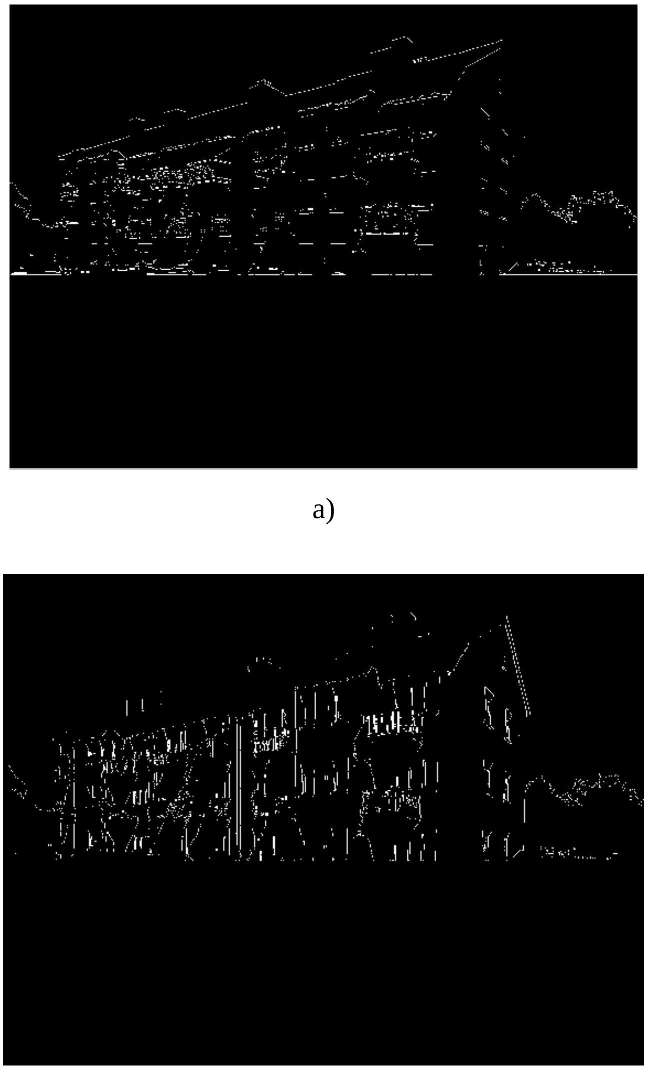
$$\phi(x,y) = \arctan(\frac{G_y}{G_x}) \quad （7）$$

The partial derivative in each formula must calculated the position of every pixel. In practical application, convolution operation of small template is commonly used to approximate, and each has one template. The Roberts operator, the Prewitt operator and the Sobel operator are the most commonly used ones.

2.1.6 The Sobel Operator

The Sobel Operator is a set of direction operator, which detects edges from different directions. This operator strengthens the weight of the central pixels UDLR (up, down, left, right); instead of calculating the difference after average. Its result is a pair of edge images. The Sobel Operator usually performances better when dealing with images which have gradient grayscales and much noise.

The Sobel Operator works the best in convolution operation for edge detection as figure 5[18].



a)

b)

Figure 5 a) Horizontal Edges Detection with Sobel b) Vertical Edge Detection with Sobel

2.2 Hough Transform

Hough Transform is a detecting method that makes use of the overall features of images to detect target contour directly. It is used to connect discrete edge pixels into a closed shape [19]. Hough Transform can get accurate boundary acquisition and export it in the form of a continuous curve. It can also segment and extract known target in very noisy environment with possible sub-pixel accuracy.

2.2.1 Basic Principle of Hough Transform

The basic principle of Hough Transform is the duality of point and line. Collinear points in image space correspond to intersecting lines in parameter space. In the image space of X-Y, Collinear points（x，y）can be defined in Equation (8)

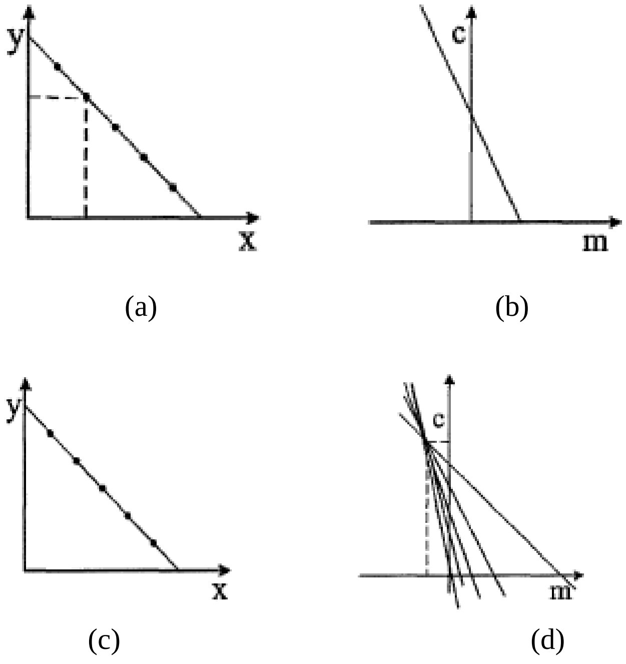
$$y = mx + c \qquad （8）$$

m: linear slope, c: intercept. The above equation can be defined as (9)

$$c = -xm + y \qquad （9）$$

In parameter space of c - m, with x as slope and y as intercept.

It can be seen that (x, y) in image space has its corresponding line in parameter space. A line in image space is determined by (m, c) in parameter space. The two equations are the common constraints of points in both image and parameter spaces according to Hough transform. As shown in figure 7, (a) shows collinear points in image space; (b) shows the bunch of corresponding lines of the collinear points in parameter space. These lines all intersect in one point after Hough transform. The location of this point is definite and it can tell the parameter of lines in image space. Hough transform turns straight line detection in image space into point detection in parameter detection simply by accumulative statistics. As figure 6 Duality of point and line.



(a) (b)

(c) (d)

Figure 6 (a) Single point in image space (b) Straight line in parameter space (c) Collinear points in image space (d) Straight lines in parameter space

2.2.2 Straight Extraction

In the process of concrete calculation, it is needed to discrete M-C in parameter space into two-dimensional accumulative array, which is noted as (m，c) as in Figure7 [20].The scopes of slope and intercept are defined as $[m_{\min}, m_{\max}]$ and $[c_{\min}, c_{\max}]$ respectively. First, it is a must to set full array A zero and set m all possible values in $[m_{\min}, m_{\max}]$ to calculate the corresponding C. Based on the integer values of m and c, it is possible to nail A(m，c)=A(m，c)+I。And determine the values of m and c later by detecting the position of the local peaks in the array A.

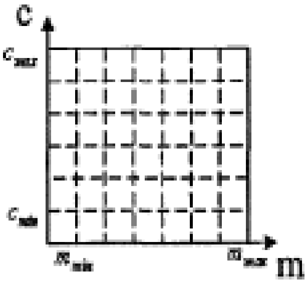


Figure 7 Array in parameter space

If the slope of straight line is infinite (e.g. straight line x = a), equation (10) is not able to handle it. Therefore, straight polar equation brought about by Duda and Hart is adopted in order to correctly identify and detect straight line in any direction and any position.

$$\rho = x\cos\theta + y\sin\theta \quad (10)$$

Hough transform turns figure 5 and 6 to Figure 8 and 9 with sobel edge detection.

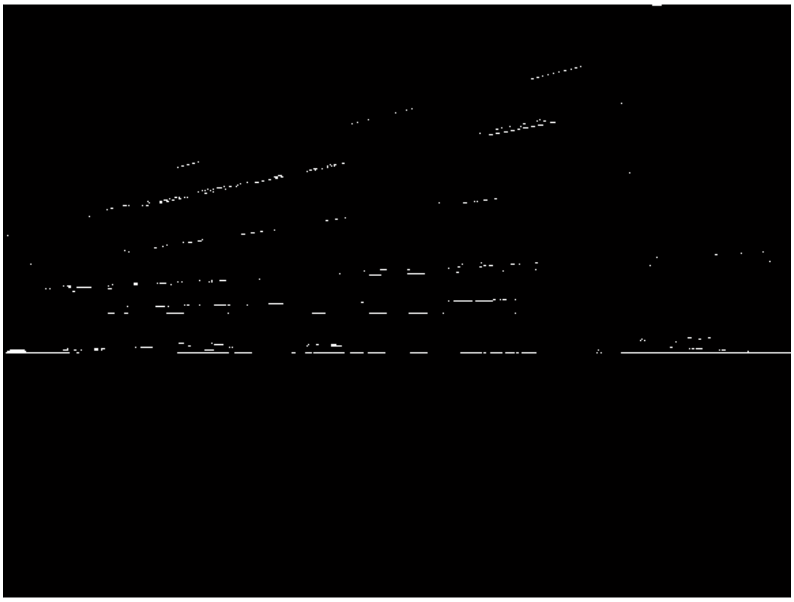


Figure 8 Horizontal line extraction with Hough

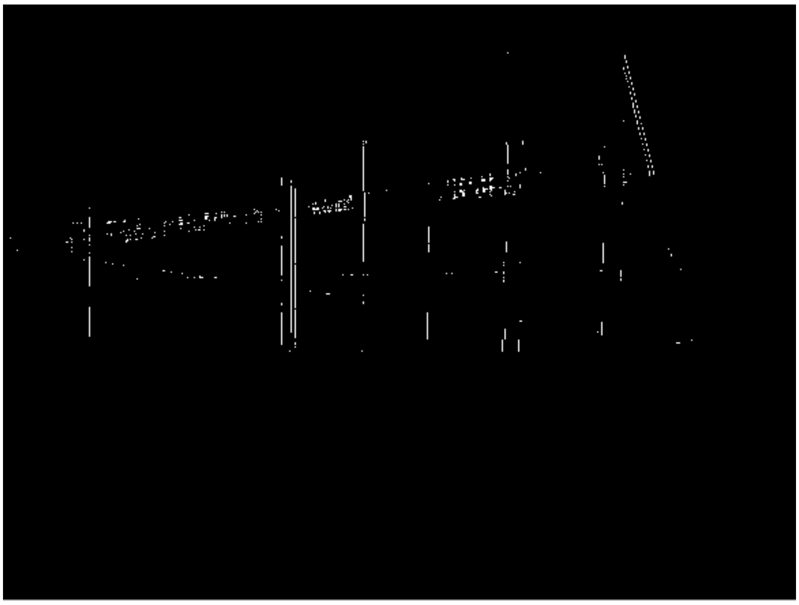


Figure 9 Vertical line extraction with Hough

With Hough transform, edge points in image space are used to calculate the likely trajectory of reference points in the parameter space and figure out the peak after reference points' count in an accumulator. Essentially, Hough transform is more like a voting machine, voting for the discrete points in the parameter space. If a vote value exceeds a certain threshold, it is believed that there are plenty of image points in the straight line determined by the parameters. This method is less affected by noise and linear gaps.

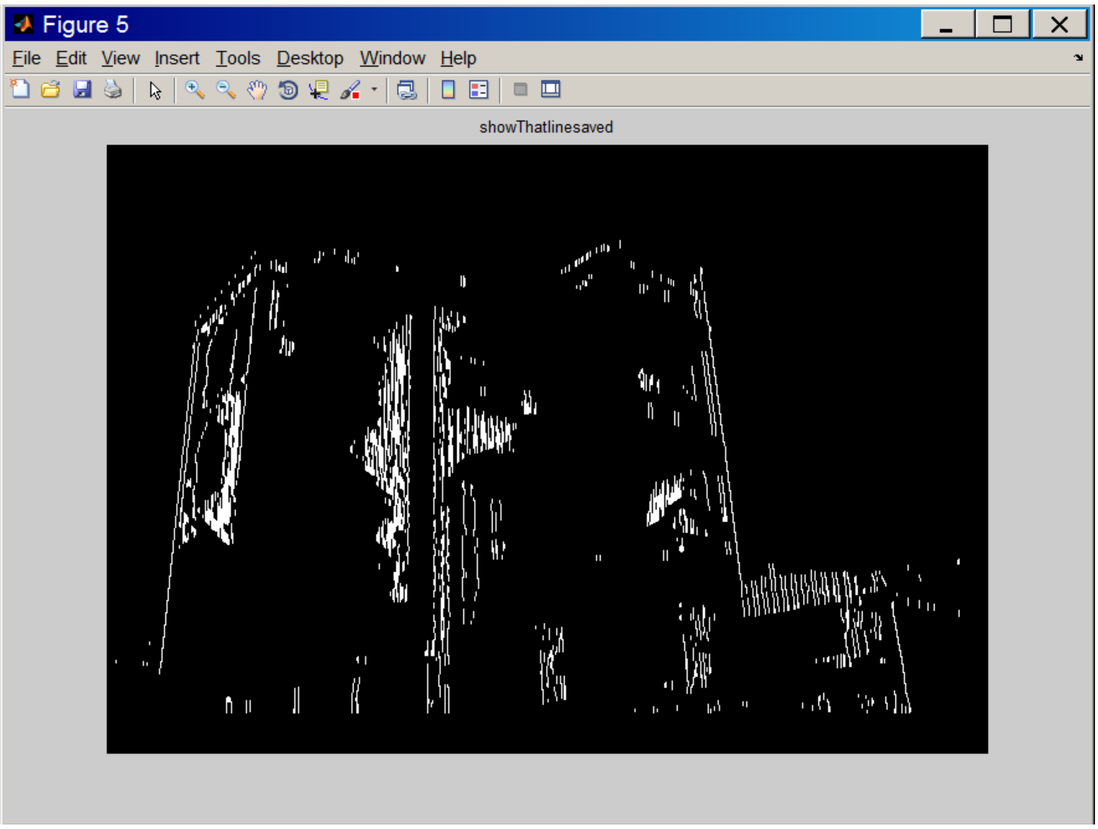


Figure 10 Image after optimization

3. Methodology of Building Extraction

This chapter describes what kind of detector we use to extract buildings information in the process of edge detection, line extraction and building segmentation. A flow-process diagram is shown in figure 11. It is known that different places and regions have their unique architectural styles. European buildings are pointy and detailed in certain area. Buildings in the US usually have the shape of a box. Chinese buildings have flat ceilings and single roofs. Aerial images and remote sensing images capture construction information of building tops while ground shooting takes photos of the sides of buildings. We chose buildings from Europe in the ZuBuD database, these buildings have a lot of windows on their image, thus they have similar line feature characteristic.

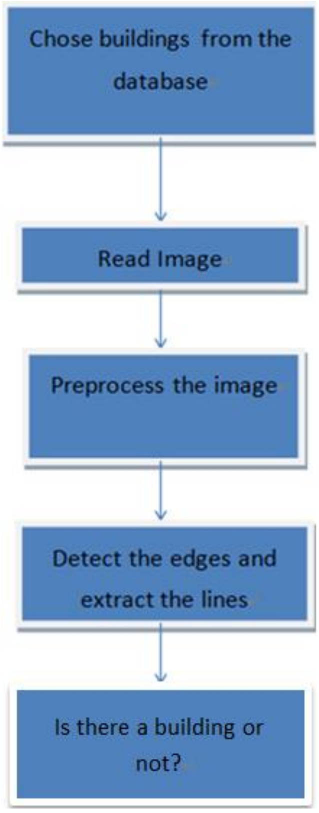

Chose buildings from the database
Read Image
Preprocess the image
Detect the edges and extract the lines
Is there a building or not?

Figure 11 Algorithm of Our Method

3.1 Data Collection

In this chapter we would like to describe how these data can be adopted. Firstly, we discovered ZuBuD database from website [22], we found the whole zip profile with the buildings in PNG format and this helped our research. Then, we chose 100 different buildings from that database in difference angles and then we decided to use this database, as shown Figure 12.

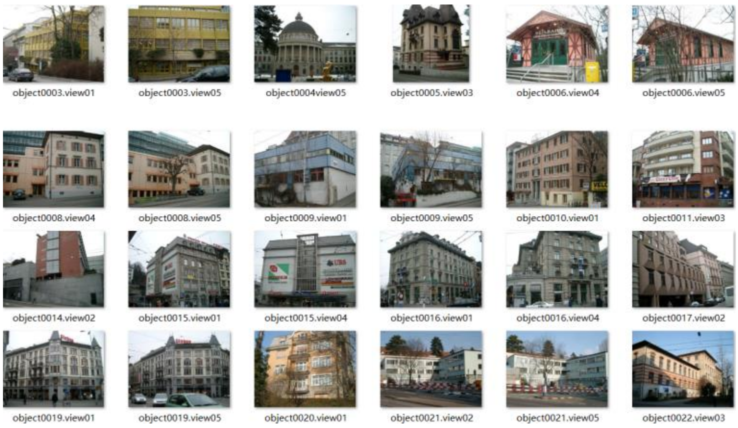


Figure 12 Part of Data base

3.2 Preprocessing

We did two steps in this part. The first step is to use "J=rgb2gray (I)" to get the grayscale, which makes image easy to create a binary image. For the second step, we use binarization (see section 2.1.3) to get the black and white image. Figure 13 shows one original image.

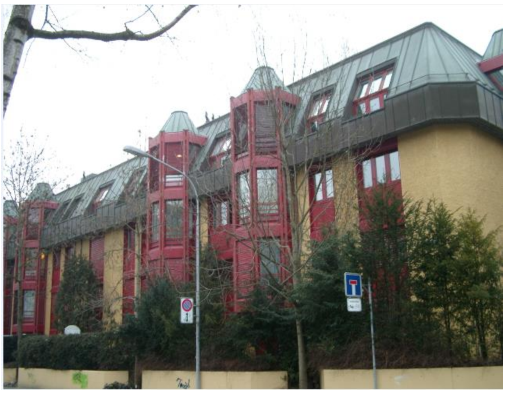


Figure 13 Original image

Binarization benefits the image processing and image analysis. It is helpful in making the graphic become simply, reducing data redundancy and emphasizing the outline of the target to be processed as well.

As a result, usually most recognition systems take binarization as the first step. With our method, we adopt the global thresh holding, as explained in section (2.1.3), for the processing of original building images. Since background information is secondary so we only need to extract the building information. In our system we first change the original image to grayscale image "I"(see the Appendix), then we get the results in figure 14.

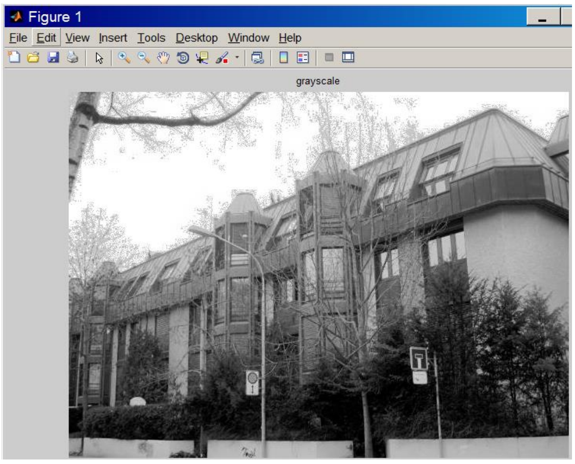


Figure 14 grayscale image

After that we calculate the grayscale image and get the result in "L" (see the Appendix), as shown in figure 15.

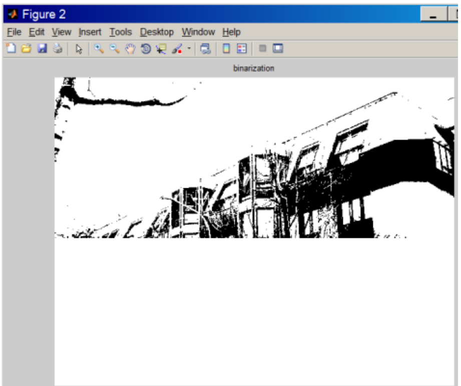


Figure 15 binarization image

As the figure shows, we get the black and white binary image from grayscale image through binarization processing. After this step there are a lot of objects in the image, making it hard to detect where building is. Therefore we have to extract the line and detect edges of buildings in the image.

3.3 Edge Detection and Line Extraction

Building geometry is described by vertical lines in our database. With the intention of remembering and analyzing building images, we detect and extract the lines of images that are already processed. We use Sobel edge detector (as we said in section2) code in our application (see the Appendix) to detect the line as shown in figure 16.

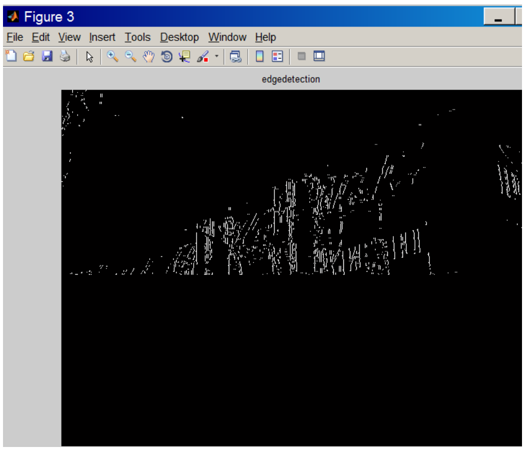


Figure 16 edge detection

The following step is line extraction, in this step we use Hough transform (section 3.2) to extract the lines. There are three steps to complete the transform. First of all is to find out each point that is already processed in edge detection procedure and save the point that we found. This step cannot extract visual figures, because Hough transform calculate the value in the new space. Secondly, we use the other loop (see the Appendix) to create a new space. The last step is to use the point we have saved to figure out the point of intersection, which is the line we want to extract. In this step, we can see optimized line in the image, as shown in Figure 17.

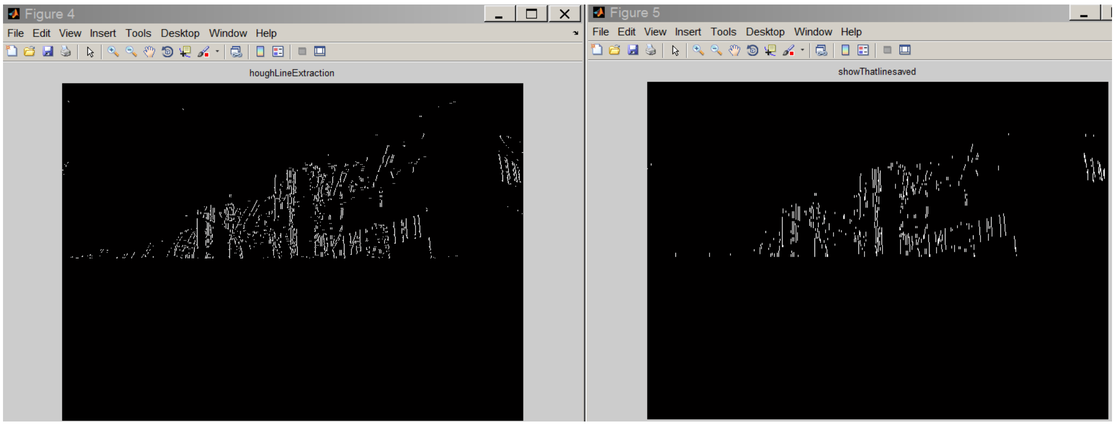


Figure 4
File Edit View Insert Tools Desktop Window Help
houghLineExtraction
Figure 5
File Edit View Insert Tools Desktop Window Help
showThatlinesaved

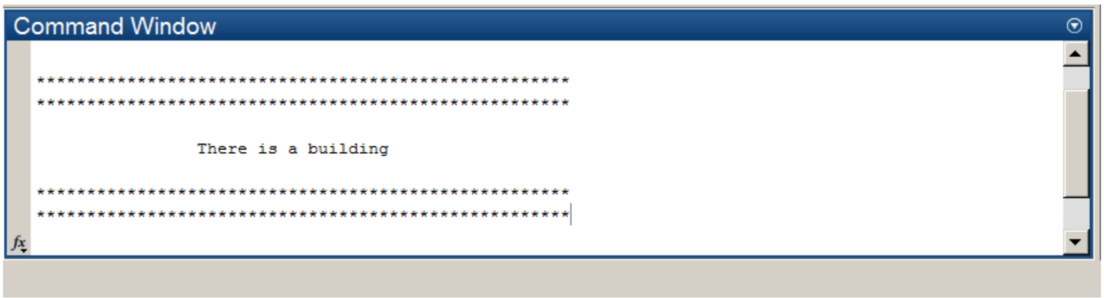
Figure 17 left is before extracted, right is after extracted

3.4 Building Identification

It is assumed that building images are almost upright within a certain shooting range in order to keep the "vertical" edges within the scope of the threshold set in this paper.

It can be seen from Figure10 that line segments after optimization differ in length. Most of these optimized lines belong to the edge line of buildings. Such elements as texture, image noise would get in the way of image preprocessing which cannot reflect the real geometry features. Building texture and complex background information would also interfere with building contour extraction from Hough transform by creating "false line" to affect the effect of straight line detection and to increase the difficulty of subsequent processing. The way is to eliminate partial pixels and to expand partial line segment.

To be specific, it examines the optimization of vertical lines in building detection [21]. But still there are other interfering lines as a result of the complexity of building image background. It is thus necessary to set a threshold value of "Max"(see in Appendix), which could count the longer lines. If the number of long lines in a certain image exceeds Max, it means that there are no buildings in the image and there would appear the note as in Figure 18.



```
*****************************************************
*****************************************************

                There is a building

*****************************************************
*****************************************************
```

Figure 18 the result of test

3.5 Location of Buildings in Images

The boundary of building contour has certain characteristics in ZuBuD database. In this database, the most obvious feature is the line of the buildings, the lines are the main feature of buildings when compared with other objects in this case. We think that linear concentration area is building area. The description of boundary in our algorithm is the lines we have calculated by Sobel operator. The red line area in Figure 19 below is detected area in our system.

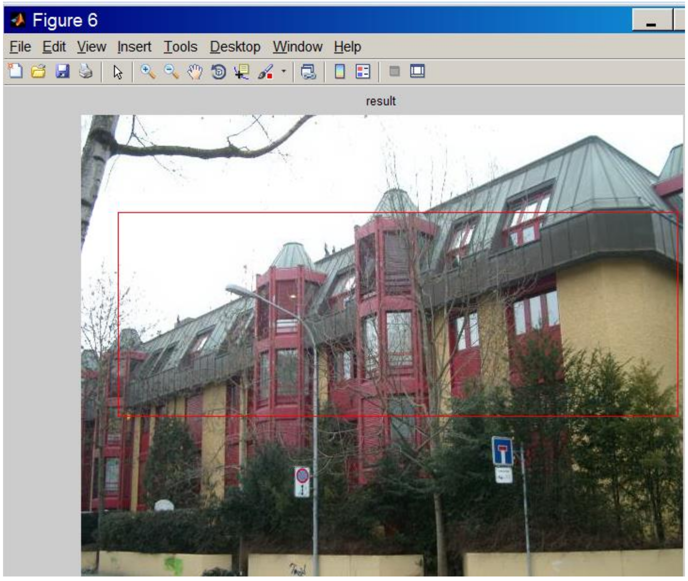
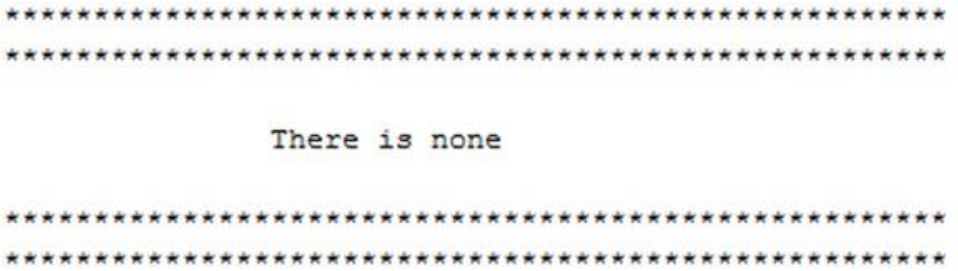


Figure 19 successful delectation

For figure 20 without buildings, matlab simulation system will prompt information as shown in figure 20.

a)



```
*****************************************************
*****************************************************

               There is none

*****************************************************
*****************************************************
```

b)

Figure 20 a) image without building b) the result of test

This shows building identification and location in images and experimental results. We use the threshold "if max>10"to defined where the building is and use "else" to define where there is no building.

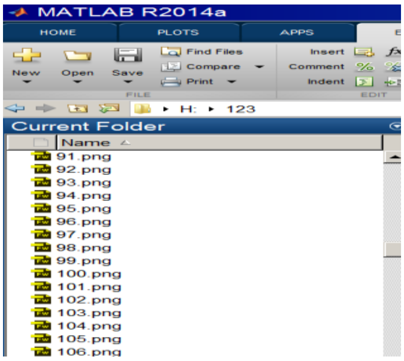
We have tried different values on the variable "max" value in each value scope. If we set the threshold at less than 10, it will present the flower as a building in the above image. If we set the threshold at more than 20, the system will find that the vertical lines are not enough to reach the requirement. Therefore we set the threshold at 20.

4. Result and Discussion

4.1 Result

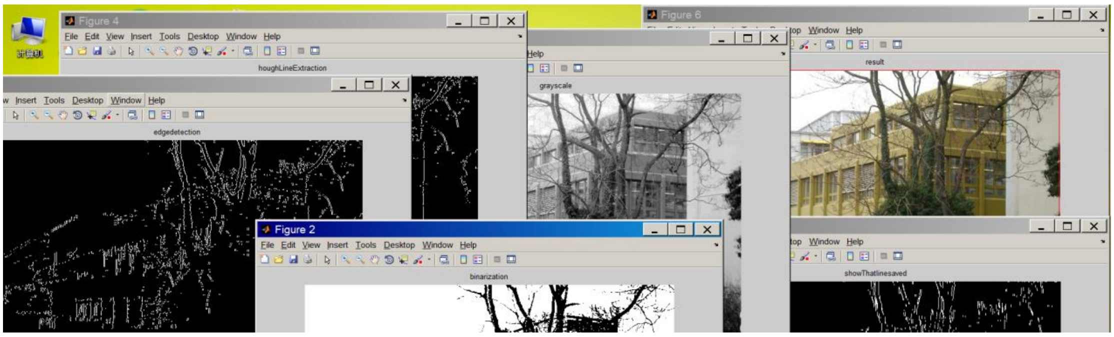
The buildings geometry is portrayed by lines, almost all buildings have the door and lots of windows wherever they come from and whatever regions they belong to. And things like windows or doors have straight lines but other objects like lamp, car, flower and trees have irregular edge. Only regular building edges can be detected by our algorithm.

Our test is focused on the buildings that can be detected by our system. And we have done 100 tests for this application using ZuBuD database. According to statistics, we first experiment with some images from database as shown in figure 21 with our method of building extraction. The rate of success is 96 percent of hundred images we chose from ZuBuD database. Failure occurred in only 4 images.

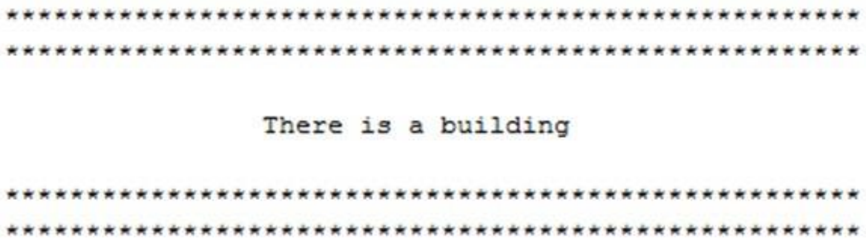


Figue 21 images folder

We type the target name with images to be tested on the top of our method "I=imread ('123.png') ;"(see in Appendix) and then press the run button to get the result we need. We can get 6 figures from each step if it is successful as shown in figure22a, 22c and the display of "there is a building", if it is not successful then we will get 5 different figures as shown in figure 22b, 22d and the display of "there is none".



a)



```
*****************************************************
*****************************************************

                There is a building

*****************************************************
*****************************************************
```

b)

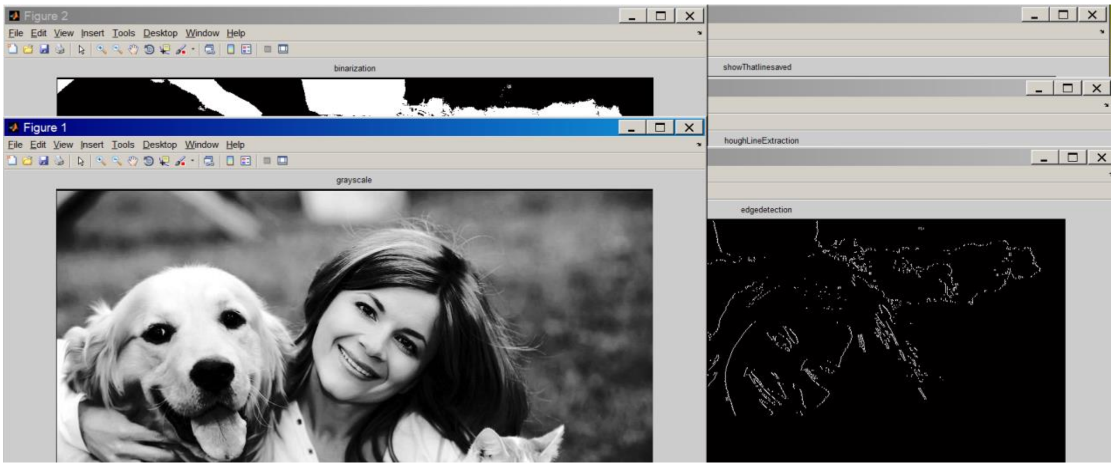
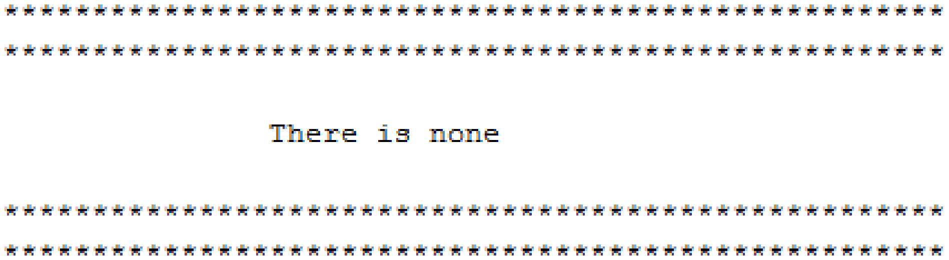


Figure 2
File Edit View Insert Tools Desktop Window Help
binarization
Figure 1
File Edit View Insert Tools Desktop Window Help
grayscale
showThatlinesaved
houghLineExtraction
edgedetection

c)

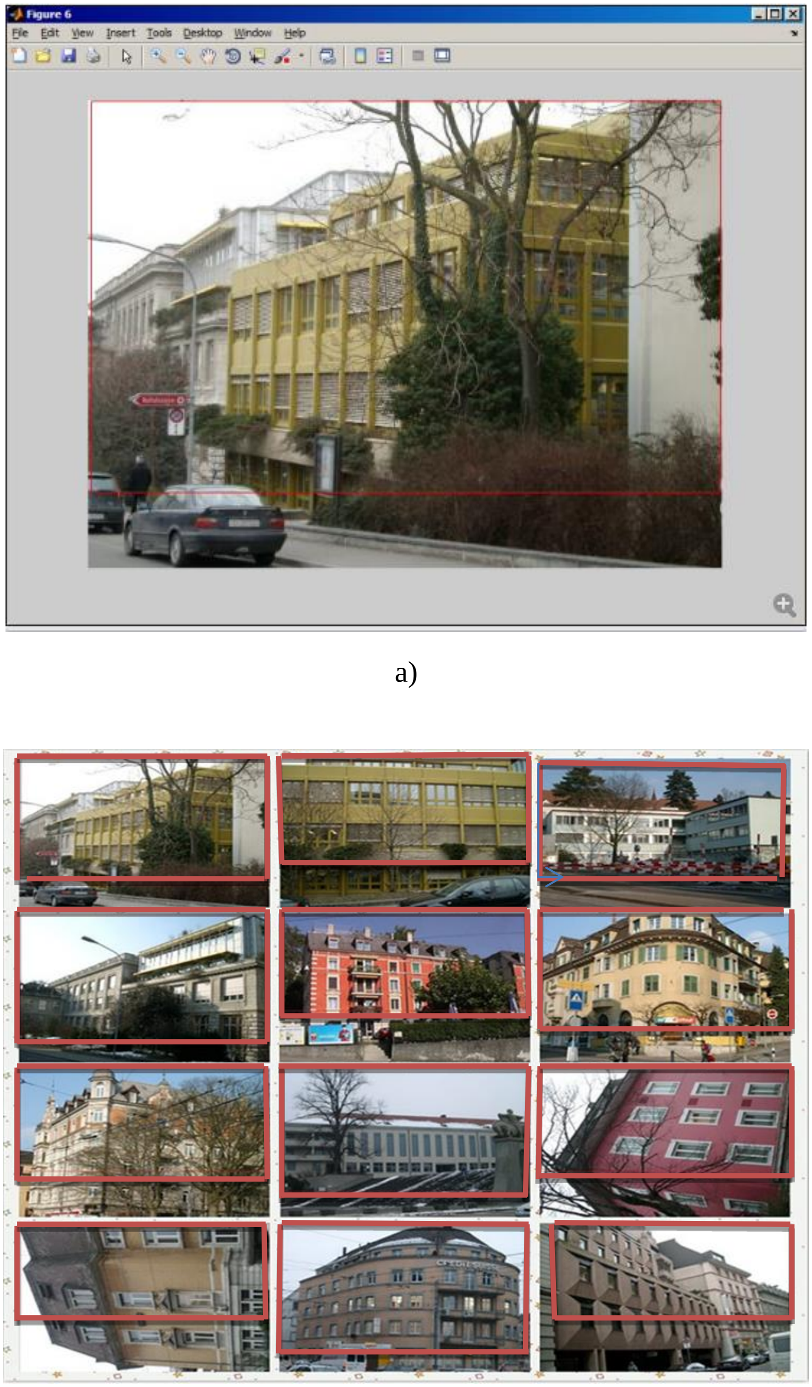


```
*****************************************************
*****************************************************

                There is none

*****************************************************
*****************************************************
```

d)

Figure 22 a) get 6 images after success b) the result above image of test c) get 4 images after failed d) the result above image of test

The following figures with the red line are the results of successful detectation as shown in Figure 23.



a)

b)

Fiugre 23 a) Successful images b) Successful images

In the three images we use, there are no buildings but just flower, dog and car. The three images turned out to be unsuccessful for our algorithm. Two of them get unreasonable results shown as 1 and 4 of Figure 24. In fact, these two images are from other database. Objects in these images possess similar linear characteristics after binarization and straight line extraction as images from ZuBuD database. That explains why the test of these images is a failure.

There is no building in the one shown in number 2, so "There is none". There are just a few shot lines in the result figure, not enough to reach the threshold we set, thus this result touches off the "else" method to gets "There is none". And figure like number 3 does not detect the building clearly. Because liner character of the bottom of building is not obvious enough after binarization and does not reach the threshold we set, resulting in the extraction of the upper half of the building.

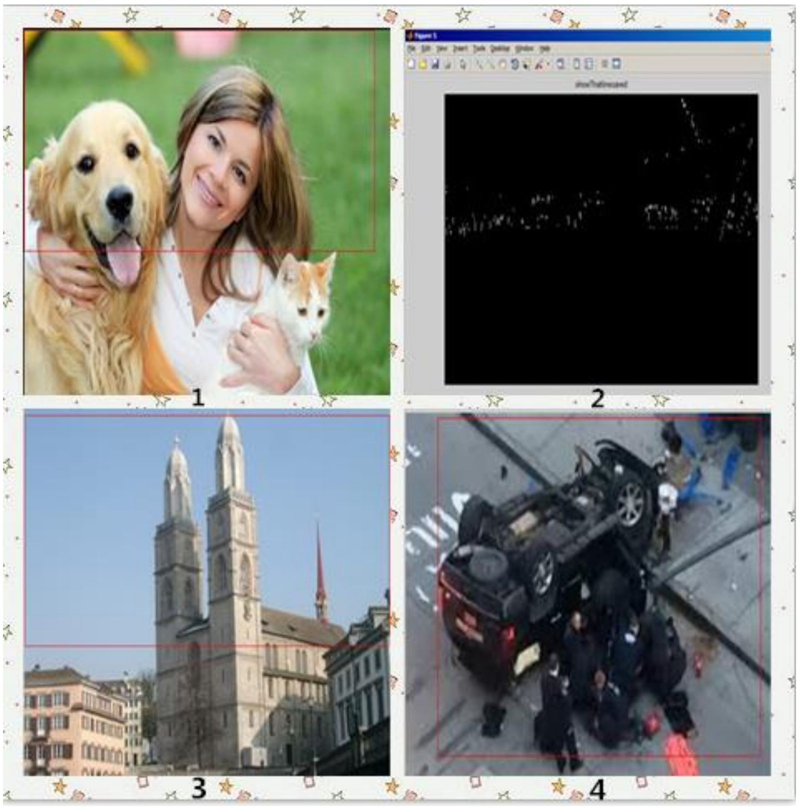


Figure 24 Result unsuccessful

The results are shown in the command window (see figure 18). Most of images in our test shows that have buildings and it prove the techniques we chose work with our algorithm.

4.2 Discussion

The reason for choosing sobel operator is that this operator detects the edge line more easily. Compared with "sobel" or "canny", canny takes more time to get result and it can detect the line more clearly but this will also detect more lines from other objects and this factor will influence the result. Although sobel operator can't detect lines clearly but in this case it can work well.

Hough transform is helpful to extract the buildings, it is commonly used in different fields, especially in feature extraction and recognition system, so we adopt this transform to extract the lines and for building reorganization. Hough transform is widely accepted in modern times. Our application is pre-stage work for GIS. This technique can help people recognize the buildings in satellite pictures. 3D model can be built by this transform, because sometimes people can't purchase laser machine, Users can use this technique to extract building edge lines from difference angles and use the lines extracted to build 3D model and then put some texture on the model.

Our work mainly focuses on line extraction part. The research question is about how to do this since there are often other objects beside the buildings presented in an image. The system may have problems to extract and recognize the building. We are glad about our consequence of recognition and extraction. We test the images from one profile so these images have the same pixels and size, thus we get the recognition rate in a ninety-six percent. Algorithm worked successful because the system detects line feature in ZuBuD data base and the unsuccessful result because we use other data base. Those images of other data base focus on other objects, and those objects have line features as ZuBuD data base. Thus use those data base in our system should be unsuccessful. However, due to the limits of times and ability, we think the following aspects can be further improved in future study.

5. Conclusions and Future perspective

5.1 Conclusions

Based on image edge detection and Hough transform, this article is dedicated to the study of the extraction and detection of building features in 2D Imagery. It reveals the algorithm in image preprocessing phase and adopts image binarization and binary image horizontal segmentation algorithm to extract information of building edges. We only used ZuBuD database to test our algorithm. This method retains building contour information and filters unrelated background information. Line segment algorithm from Hough transform is applied to extract straight lines. And it tries to prove the effectiveness of the method with actual building image detection.

5.2 Future perspective

First, we can try to extract different pixel image as we adopted the images from one database. Secondly, the threshold can be more precise by more tests, because we think that images with different qualities need different thresholds to extract the edges. We chose sobel operator by test, because we think this approach is better than canny operator for this database, the quality of image which we chose is too low, so canny operator will extract the line more clearly, it will extract more lines from other object in the image, sobel is good for ZuBuD database.

We also would like to extend the field of recognition. At present, we just focus on low quality image and we hope to improve it so that it could recognize building image in each quality and extract buildings in images faster.

Acknowledgements

We are greatly indebted to our supervisor, Professor Julia Ahlen for her valuable instructions and suggestions on our thesis as well as her careful reading of the manuscript. She inspired us in many aspects from research method to scientific issues.

Our sincere thanks are also given to our colleges teachers whose lectures turned out to be a treasure for us. Last but not least, we owe much to our friends and classmates for their valuable suggestions and critiques which are of great help and importance in making the thesis a reality.

References

[1] H. Q. Guo, *"Geospatial framework of digital city construction projects 3D city model for building technology,"* Surveying and Mapping, 2013, 4, pp. 86-88.

[2] G. Chen, *"Remote sensing image recognition and change detection of buildings,"* [D]. Nanjing University of Science and Technology, 2005.

[3] L. R. Han, "*Building complex scene image extraction technology research,"*[D]. Ocean University of China, 2010, pp. 220-240.

[4] C. Pan and X. J. Wang, "*Natural background information detected algorithms artificial journal*," Optical Technology, 2008, 34(6): pp. 881-885.

[5] A. K. Shackelford and C. H. Davis, *"A self-supervised approach for fully automated urban land cover classification of high resolution satellite imagery,"* [C] //Proc. 3rd Int. Symp. Remote Sens. Data Fusion Over Urban Areas—Urban. 14-16.

[6] R. Hofmann, *"Extraction of Natural Feature Descriptors on Mobile GPUs,"* [J]. Eingangzum Volltext in OPUS, 2007.

[7] L. Tang, W. X. Xie and J. J. Huang, *"Automatic extraction of high-rise buildings from aerial images,"* [J]. Journal of Computers, 2005, 28(7): pp.1199-1204.

[8] Y. D. Qu, C. S. Cui, S. B. Chen and J. Q. Li, *"A fast sub pixel edge detection method using Sobel-Zernike moments operator,"* [D], Harbin Institute of Technology and Shanghai Jiaotong University, 2003.

[9] C. L. Li, Y. B. Guo and C. Li, *"The High accurate Extraction of Line Features of Object Contour,"* IEEE Transactionon Acta Geodaetica et Carto graphica Sinica, Vol.40, NO.1, pp.67-83, Feb.2011.

[10] E. Simonton, H. Oriot and R. Garello, *"Rectangular Building Extraction from Stereoscopic Airborne Radar Images,"* [J]. IEEE Transactions on Geoscience and Remote Sensing. 2005, 43(10): pp.2386-2395.

[11] Z. W. Kim and R. Nevatia, *"Automatic Description Of Complex Buildings From Multiple images,"* [J]. IEEE Transaction on Computer Vision and Image Understanding. 2004, 96: pp.60-95.

[12] X. Sun and H. Q. Wang, *“Boosting method on the object automatically extraction high-resolution remote sensing image building goals,”* [J]. Electronics and Information Technology, 2009, 31(1): pp.177-181.

[13] C. Tao, Y. H. Tan and H. J. Cai, *“Object-oriented high-resolution remote sensing image classification urban buildings extraction method,”* [J]. Journal of Surveying and Mapping, 2010, 39(1): pp.39-45.

[14] G. Gao, *“An Improved Scheme for Target Discrimination in High-Resolution SAR Images,”* [J]. IEEE Transaction on Geoscience and Remote Sensing, 2011, 49(1): pp.277-294.

[15] K. Karantzalos and N. Paragios, *“Recognition-Driven Two-Dimensional Competing Priors Toward Automatic and Accurate Building Detection,”* [J]. IEEE Transaction on Geoscience and Remote Sensing, 2009, 47(1): pp.133-144.

[16] J. Goodman, *“Global Thresholding and Multiple Pass Parsing,”* Harvard University, 1997.

[17] R. Zhang, *“a color gradient information based image denising algorithms,”*Micro-computer information, 2010, 12, pp. 111-113

[18] Y.L. Xu, J.Y. Zhao and Y. B. Jiao, *“Research Noise image edge detection method,”* [J], Application Research of Computers, 2009, 26(1): pp.387-389.

[19] R. Lang, *“Digital image processing to achieve learning Visual C ++,”* [M], Peking, Hope Electronic Press, 2002.

[20] S. Zhou, *“Target detection and localization algorithm based on image gradient and curvature information constraints research,”*Yanshan University, 2013, 15-16

[21] Y. J. NI and C. R. Wang, *“Of Scientific study of a texture image de-noising method,”*[C]. Chinese Control and Decision Conference Proceedings, 2006, pp.100-101.

[22] http://www.vision.ee.ethz.ch/showroom/zubud/index.en.html

[23]X. Zhao, Y. J. Chen and Y. X. Luan, *“The Edge Detections in Digital Image,”* [D]. China University of Mining & Technology<Beijing>,Vol.11,2006.

Appendix

```
clc;
clear;
I=imread('dog.png');

J=rgb2gray(I);

H=histeq(J);
figure,imshow(H),title('grayscale');
L=im2bw(H);
figure,imshow(L);


[a,b]=size(L);
for i=3:a
    if L(i,35)==0
       for k=i+floor((a-i)/2):a
           for j=1:b
               L(k,j)=1;
           end
        end
    end
end
imshow(L),title('binarization');

img=edge(L,'sobel','vertical',0.3);
figure,imshow(img),title('edgedetection');


[m,n]=size(img);
a=180;
d=round(sqrt(m^2+n^2));
s=zeros(a,2*d);
z=cell(a,2*d);
for i=1:m
    for j=1:n
        if(img(i,j)==1)
            for k=1:a
                p = round(i*cos(pi*k/180)+j*sin(pi*k/180));
                if(p > 0)
                    s(k,d+p)=s(k,d+p)+1;
                    z{k,d+p}=[z{k,d+p},[i,j]'];
                else
                    ap=abs(p)+1;
                    s(k,ap)=s(k,ap)+1;
                    z{k,ap}=[z{k,ap},[i,j]'];
                end
            end
        end
    end
end
for i=1:m
    for j=1:n
        img2(i,j)=0;
    end
end
for i=1:a
    for j=1:d*2
        if(s(i,j)>30)
            lp=z{i,j};
            for k=1:s(i,j)
               img2(lp(1,k),lp(2,k))=1;

            end
        end
    end
end
figure,imshow(img2);title('houghLineExtraction');


for j=1:n
    for i=1:m
        if
img2(i,j)==1&&img2(i+1,j)==1&&(img2(i+3,j)==1||img2(i+4,j)==1||img2(i+5,j)==1)
            img2(i+2,j)=1;
            img2(i+3,j)=1;
        end
    end
end
for j=2:n
    for i=2:m
        if img2(i-1,j)==0&&img2(i,j)==1&&(img2(i+1,j)==0||img2(i+2,j)==0)
```

```
                img2(i,j)=0;
            end
        end
end
for j=2:n
        for i=2:m
            if img2(i-1,j)==0&&img2(i,j)==1&&(img2(i+1,j)==0||img2(i+2,j)==0)
                img2(i,j)=0;
            end
        end
end


figure,imshow(img2),title('showThatlinesaved');
%}
max=0;
for j=1:n
        for i=1:m
            if img2(i,j)==1&&img2(i+1,j)==1&&img2(i+2,j)==1&&img2(i+3,j)==1&&...

img2(i+4,j)==1&&img2(i+5,j)==1&&img2(i+6,j)==1&&img2(i+7,j)==1&&img2(i+8,j)==1
                max=max+1;
                break;
            end
        end
end
if max>20
        disp('                                                     ');
        disp('                                                     ');
        disp('                                                     ');
        disp('*****************************************************');
        disp('*****************************************************');
        disp('                                                     ');
        disp('                 There is a building             ');
        disp('                                                     ');
        disp('*****************************************************');
        disp('*****************************************************');
        disp('                                                     ');
        disp('                                                     ');
        disp('                                                     ');

s=0;
for i=1:m
```

```
        for j=1:n
            if img2(i,j)==1&&img2(i+1,j)==1&&img2(i+2,j)==1&&...
                    img2(i+3,j)==1&&img2(i+4,j)==1&&img2(i+5,j)==1
                s=1+s;
                rol(s)=i;
                col(s)=j;
            end
        end
end

rol2=sort(rol);
col2=sort(col);
r_h=size(rol2,2);
c_h=size(col2,2);

figure,imshow(I),title('result');
hold on;

line([col2(1),col2(c_h)],[rol2(1),rol2(1)],'color','r');
line([col2(1),col2(1)],[rol2(1),rol2(r_h)+70],'color','r');
line([col2(c_h),col2(c_h)],[rol2(1),rol2(r_h)+70],'color','r');
line([col2(1),col2(c_h)],[rol2(c_h)+70,rol2(c_h)+70],'color','r');

else
        disp('                                                        ');
        disp('                                                        ');
        disp('                                                        ');
        disp('********************************************************');
        disp('********************************************************');
        disp('                                                        ');
        disp('                There is none                     ');
        disp('                                                        ');
        disp('********************************************************');
        disp('********************************************************');
        disp('                                                        ');
        disp('                                                        ');
        disp('                                                        ');
end
```