Building Detection from Terrestrial Images

Teerapat Chaloeivoot and Suebskul Phiphobmongkol
Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand
Email: Teerapat.Ch@Student.chula.ac.th, suebskul.p@chula.ac.th

***Abstract*—This paper presents a method to detect buildings in terrestrial images. High resolution terrestrial images are normally taken from land survey vehicles. These images and other surveyed data along roads are needed by many agencies that require new data as time passes by. Land use in rural area is an example that needs information about buildings and can benefit from terrestrial images. The proposed method was aimed to detect buildings in terrestrial images to benefit the above needs. The method consists of two stages. The first stage removes unwanted objects, performs image segmentation, and finds regions of interest. Image processing techniques such as greenness extraction, sky detection, color segmentation, color detection, shape detection are used. The second stage performs building detection. It includes the possible building parts detection, projection profiles finding, and the building determination. The method can identify a partial building if the whole building is not shown in an image. The proposed method was tested on 936 images (332 images with buildings and 604 images without buildings). The images were from Google Street View. The accuracy was determined by human inspection. The method gave promising results with an average accuracy of 82.5%. Positive faults were 4.7% average.**

***Index Terms*—building detection, terrestrial image, image processing, image segmentation**

I. Introduction

Land use may include living, grazing, agriculture, industries. Many agencies need information about land use for many purposes, such as, taxation, community growth measurement, new dweller area finding. Building identification is a feature that helps specifying living or industrial land use. Building identification from aerial survey photos is a widely used method. However, in Thailand, this method is not used frequently due to the high cost of survey. An option is to use human surveyors to survey the area. This option requires many surveyors, and takes a long time. Another option is to use a survey vehicle that can take high-resolution photos with GPS coordinates and look for buildings later on computer. This method is considered more efficient and save time. At present, there are agencies that operate such vehicles. If a computer can detect buildings in the surveyed images there would be a new source of land use information for the relevant agencies.

The proposed method was aimed to find buildings or building parts in real environment images. Most buildings have specific elements in common, such as, roof, gables, doors, windows, and pillars. Difficulties include the detection of an incomplete structures or shapes of the buildings, building obstructed by other objects. Algorithms based on color detection, Line Hough transform, automatic seeded region growing were used to find potential features that can help specify a building. In the building determination process, it considers detected building parts such as gables, long horizontal lines, parallel vertical lines, rectangles, unnatural colors by using rules described later in the paper.

II. Related Work

Most researches of building detection from terrestrial images detected fa çade, windows, and plane. Zhizhong, Sisi, and Ben (2005) [1] introduced an automatic detection of range variance of facades by projective difference of corresponding points on a common projective plane. Therefore, the algorithm could separate building fa çade from cross road. Haider *et al*. (2007) [2] presented windows detection in the urban environment based on Adaboost to optimize cascaded classifier for detection. Martin and Wolfgang (2009) [3] applied Adaboost classification framework to classify feature of building parts. This research focused on fa çade, roof, windows and windowpane. Viraj, Rohan and Hong (2009) [4] described a method to identify windows from a building frontal fa çade by projection profiles and snake algorithm. Vincent and Caroline (2011) [5] presented a method for gable roof detection from terrestrial images. Milos and Thomas (2012) [6] used image-processing to detect windows in facades at several different orientations and scales.

An approach that detect buildings from aerial images used LIDAR data and image processing techniques [7]. Others used image-processing such as Antonis and Hichem (2008) [8], Xiuyun and Yan (2008) [9], Qiongchen and Zhiguo (2009) [10], Masoud and Parvaneh (2009) [11] and Melissa and Parvaneh (2012) [12].

III. Proposed Method

The proposed method has two stages. The first stage removes unwanted objects, performs image segmentation using automatic seeded region growing and performs region analysis. The second stage determines whether the image contains a building or not. It uses the result from

Manuscript received July 17, 2015; revised October 29, 2015.

doi: 10.18178/joig.4.1.46-50

building-part extraction and projection profiles. The system overview of the methodology is shown in Fig. 1.

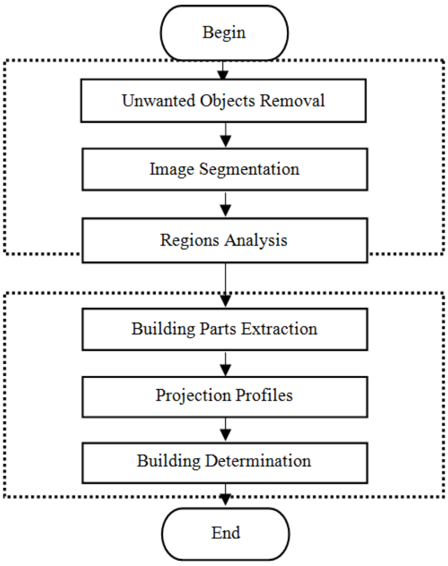

Figure 1. System overview

A. *Unwanted Object Removal*

This process was intended to remove the objects that are not part of a building, e.g. trees, sky. The result will then be appropriate for image segmentation. Two steps are described as follows:

Tree removal: This step was used to remove trees by detecting green color of trees by using HSV color space [13], [14]. The transformation of RGB to HSV performed using the following equation (1) to (6), where R, G, and B are in the range [0, 1], H is [0°, 360°], and S and V are [0, 100]:

$$\max = \max(R,G,B) \tag{1}$$

$$\min = \min(R,G,B) \tag{2}$$

$$Chroma = \max - \min \tag{3}$$

Equation (1) to (3) calculated Chroma value. Then, compute HSV value in (4), (5) and (6).

$$H = \begin{cases} 0°, & if\ Chroma = 0 \\ 60°\ x\ \dfrac{G-B}{Chroma} mod\ 6, & if\ max = R \\ 60°\ x\ \dfrac{B-R}{Chroma} + 2, & if\ max = G \\ 60°\ x\ \dfrac{R-G}{Chroma} + 4, & if\ max = R \end{cases} \tag{4}$$

$$S = \begin{cases} 0\ , & max = 0 \\ \dfrac{Chroma}{max}, & max \neq 0 \end{cases} \tag{5}$$

$$V = \max \tag{6}$$

The proper range of Hue to detect green color of tree was 40-120. Saturation was 19-100. Value was 0-66.

Sky removal: In landscape images, sky is mostly brighter than objects on the ground. Automatic Threshold method [15], [16] was used to separate the sky in the images. From the experiment, the proper range of threshold values was 130-150. The sky removal method can be performed using (7) where T is threshold value, src(x,y) is horizontal and vertical addresses of source image and dst(x,y) is horizontal and vertical addresses of destination image.

$$dst(x,y) = \begin{cases} 0, if\ y > Image\ height \div 2 \\ 0, if\ \ src(x,y) > T \\ src(x,y), otherwise \end{cases} \tag{7}$$

B. *Image Segmentation*

This process segmented each area of an image according to colors. The algorithm was automatic seeded region growing as used by Shih and Cheng (2005) [17]. This algorithm was developed from Seeded Region Growing which was presented earlier by Adams and Bischof (1994) [18]. YC_bC_r color space was used in Shih and Cheng method. The RGB to YC_bC_r conversion performed using (8), where R, G, and B are in the range [0, 1], Y is [16, 235], and C_b and C_r are [16, 240]:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 65.481 & 128.553 & 24.996 \\ -39.797 & -74.203 & 112 \\ 112 & -39.786 & -14.214 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \times \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \tag{8}$$

The process began with initial seeds. The seed selection determined 2 conditions. First condition is a similarity seed pixel checking. It was considering a 3×3 neighborhood. It started with calculates a standard deviation in (9):

$$\sigma_x = \sqrt{\frac{1}{9}\sum_{i=1}^{9}(x_i - \bar{x})^2} \tag{9}$$

where x are Y, C_b, C_r and $\bar{x}$ is mean value of 3x3 neighborhood. The similarity of pixel to its neighbor is defined as:

$$H = 1 - \left(\frac{\sigma_Y + \sigma_{C_b} + \sigma_{C_r}}{\sigma_{max}}\right) \tag{10}$$

where σ_{max} is maximum standard deviation in the image. A similarity value (10) must be higher than a threshold value. From our experiment, a proper similarity threshold value is 0.80.

Second condition is a relative Euclidean distance to its neighbor checking. The formula is as follow:

$$d_i = \frac{\sqrt{(Y-Y_i)^2 + (C_b - C_{b_i})^2 + (C_r - C_{r_i})^2}}{\sqrt{Y^2 + C_b^2 + C_r^2}} \tag{11}$$

where i=1, 2, 3, …, 8. The relative Euclidean distance in (11) is used to find maximum distance in (12):

$$d_{max} = max_{i=1}^{8}(d_i) \tag{12}$$

The maximum relative Euclidean distance to its neighbor must be less than a threshold value. A proper threshold value from our experiment is 0.08

A pixel that met two conditions above is classified as seed pixel. Next, the growing method started with labeling seeded pixel. Then, unclassified pixels were calculated a relative Euclidean distance to its adjacent regions in (13):

$$d_i = \frac{\sqrt{(Y_i - \bar{Y})^2 + (C_{b_i} - \bar{C}_b)^2 + (C_{r_i} - \bar{C}_r)^2}}{\sqrt{Y_i^2 + C_{b_i}^2 + C_{r_i}^2}} \tag{13}$$

where ($\bar{Y}$, $\overline{C_b}$, $\overline{C_r}$) are average value of Y, C_b, C_r information in that region. The pixel merged into adjacent minimum relative Euclidean distance region. This step performed until no unclassified pixel left. Next, the region similarity checking performed using (14):

$$d(R_i, R_j) = \frac{\sqrt{(\bar{Y}_i - \bar{Y}_j)^2 + (\bar{C}_{b_i} - \bar{C}_{b_j})^2 + (\bar{C}_{r_i} - \bar{C}_{r_j})^2}}{\min\left(\sqrt{\bar{Y}_i^2 + \bar{C}_{b_i}^2 + \bar{C}_{r_i}^2},\ \sqrt{\bar{Y}_j^2 + \bar{C}_{b_j}^2 + \bar{C}_{r_j}^2}\right)} \quad (14)$$

The region merged into adjacent minimum relative Euclidean distance region. The minimum relative Euclidean distance must be less than threshold value. We used 0.04 as threshold value. Final step is merged small region size. Our threshold value described below:

$$\text{Threshold} = \frac{\textit{Image height} \times \textit{Image width}}{150} \quad (15)$$

where 150 was obtained from experiment. The region that size smaller than (15) merged into adjacent minimal distance region. The threshold of minimal distance Euclidean distance of adjacent region from our experiment must be higher than 0.02.

C. Regions Analysis

This process culled out regions that are unlikely to contain buildings. It involves three steps: noise removal, left-over tree removal, and low-region removal.

Noise removal: The tiny regions would be removed because they were too small for the detection and can be burden or cause error in determination process. In our experiment, if the region size was less than 500 pixels, it will be eliminated.

Left-Over tree removal: From trees removing step, it cannot remove all green color because of several reasons such as light, various color of leaves. Removal occurred if the number of white pixels is greater than 30% (from the experiment). This method will wipe out the left-over trees regions.

Low-Region removal: Since the source images were taken from a survey vehicle, the bottom parts were mostly road, grass, or wall. Eliminating such region helps reduce error and burden to the process.

After all, unwanted objects in original image (Fig. 2a) are mostly eliminated. The result presents in Fig. 2b.

(a) Original image (b) Pre-Processing results

Figure 2. Pre-Processing results before building parts extraction process

D. Building Parts Extraction

Color detection: The colors: red, blue, brown, yellow and white are widely used as building paints which are clearly distinct from nature. Each color will be evaluated by size and position. Red and blue detected only the top half of images for checking rooftop. Brown, yellow and white are detected in the middle of images for checking body of building. The proper HSV color ranges for building detection are shown in Table I.

TABLE I. THE PROPER HSV COLOR RANGES FOR BUILDING DETECTION

Color	H	S	V
Red	0-34, 340-360	35-100	33-86
Blue	150-200	35-94	19-78
Brown	216-244	1-44	66-80
Yellow	16-50	10-17	11-35
White	30-60	19-78	39-58

Line detection: Structure of most buildings contains rectangles. In the real environment, it is hard to find perfect shapes because of obstacles such as trees. This step uses the benefits of Line Hough Transform [19] to help finding the straight edges of buildings. The angle finding formula (16) is as follows:

$$\text{Angle (in degree)} = -\left(\tan^{-1}\left(\frac{y2-y1}{x2-x1}\right) \times \left(\frac{180}{\pi}\right) \right) \quad (16)$$

where $y1$, $y2$, $x1$, $x2$ are pixel coordinates at the tips of a line.

Line detection can be categorized into four cases below:

Horizontal line: Detection of long horizontal line(s). The line length must be at least 100 pixels.

Parallel vertical lines: Detection of a pair of parallel vertical lines whose centers are almost of the same vertical level. Each line is far from the each other more than a threshold value. Our test used a threshold value at least 100 pixels.

Rectangle detection [20]: Detection of quadrilateral shape with four corners. This step aims to extract windows and doors. The size of quadrilateral shape must be larger than a threshold value. Our threshold value is 300 pixels.

Gable detection: Detection of two lines with opposite angle in the same horizontal area and their top points are near each other (less than a threshold value). From our experiment, threshold value used ±7 pixels from top point of line.

a) Superimposing building parts (b) Projection profiles

Figure 3. Superimposing and projection profiles

E. Group All Building Parts

After performing building parts extraction process, this step superimposes all results (Fig. 3a) and use projection profile to find the height and width of each group (Fig 3b).

F. Building Determination

To consider these elements, decision-making rules are needed. The rules use size and aspect ratio values as described below. Though these values were chosen empirically from 20 trial images at the beginning of our experiment, the values were applied to all tested images. Further investigation for more appropriate values should be done for real use.

Size: The group elements size must be at least 100 pixels × 100 pixels.

Aspect ratio: The width to height ratio of each group must be between 4:3 to7:5.

IV. Experiments

A. Data Test Sets

The location that the test images were taken was in the rural area of Thailand. The buildings had various structures and many of them were blocked out by obstacles, e.g. trees. These images were from Google Street View survey vehicles. The resolution was 640 x 600 pixels. Many kinds of building textures such as wood and cement were found. Some buildings had no facade, some were covered by trees.

B. Unwanted Object Removal and Image Segmentation

Sky, vegetation, noise, and left-over trees were removed after automatic seeded region growing method. Building parts would be left in the resulting images. The results are shown in Table II.

TABLE II. Object Removal and Image Segmentation Results

Method	Accuracy
Remove sky	94%
Remove vegetation	97%
Regions Analysis	100%

Errors were caused by dark cloud, trees with green colors which were out of the defined range. The problems can mostly be solved by the left-over tree removal.

C. Experiments on Images with Buildings

Testing was performed on 100 images of no obstacles in front of buildings and 100 images of building with obstacles. The results are shown in Table III.

TABLE III. Results on Images with Building (with/without Obstacles)

Descriptions	Accuracy
Building without obstacles	92%
Building with obstacles in front	83%

The results showed satisfied accuracy in both cases. The errors were caused by the edge joint of source images (Fig. 4a) and by no-wall building (Fig. 4b), which made buildings undetected in projection profiles.

(a) Image with edge joint (b) Building without wall

Figure 4. Error cases

D. Experiments on Mixed Data Set

Experiment was done on 936 images (332 images with buildings and 604 images without buildings) in the rural area environment. The results are shown in Table IV.

TABLE IV. Results from Mixed Dataset

Descriptions	Accuracy
Images with buildings	82.5%
Images without building	95.3%

Some false positives occurred due to trees, burned agricultural area, and large water surface. Some trees were not totally removed by the prior process and created faults. The reason was the confusing Hough line detection in many angles. The burned agricultural area was detected as large brown color and water reflection detected as large white color in color detection. Thus, the situation led to mistaking the area as buildings. The results are shown in Table V.

TABLE V. False Positive Cases

Reasons	Number of images
Trees that were not removed in early process	21
Burned agricultural area	5
Water reflection	2

False negatives also occurred mostly due to objects between the cameras and the buildings. Far-away buildings were also too small to be detected. Too few building parts detected can also cause false negatives. The false negatives are shown in Table VI.

TABLE VI. False Negative Cases

Reasons	Number of images
Small Building (far away)	11
Obstacle objects (trees)	39
Few building parts were detected	8

V. Conclusion

This proposed method can detect buildings in terrestrial images that contain whole buildings or part of them. The test images were taken in the rural area of

Thailand by Google. Building detection was tested on 936 images. An average accuracy of 82.5% was achieved with average positive faults of 4.7%. Errors were from objects, such as trees that were over the buildings in images. Too small buildings and unclear building parts also caused errors. Other causes were from out-of-range tree colors, water reflection, and burned agriculture. The program, sometimes, did not detect some walls and may detect more than one building. The difficulties were from the handling of objects over buildings and the analysis of existed buildings from the detected components. However, the result showed that the proposed method could be for real use and is satisfied.

REFERENCES

[1] Z. Kang, S. Zlatanova, and B. Gorte, "Automatic detection of range variance of facades from vehicle based image sequence," in *Proc. 5th International Symposium on Mobile Mapping Technology*, Italy, 2007.

[2] H. Ali, C. Seifert, N. Jindal, L. Paletta, and G. Paar, "Window detection in facades," in *Proc. 14th International Conf. on Image Analysis and Processing*, 2007, pp. 837-842.

[3] M. Drauschke and W. Förstner, "Selecting appropriate features for detecting buildings and building parts," in *Proc. 21st Congress of the International Society for Photogrammetry and Remote Sensing*, Beijing, China, 2008, vol. 16, pp. 447-452.

[4] V. Kulkarni, R. Nagesh, and H. Wu, "Window detection in frontal façades," Project work at CS294-69 Image Manipulation and Computational Photography, University of Barkley, 2011.

[5] V. Brandou and C. Baillard, "Gable roof detection in terrestrial images," in *Proc. Photogrammetric Image Analysis*, 2011, pp. 97-108.

[6] M. Miljanovic, T. Eiter, and U. Egly, "Detection of windows in facades using image processing algorithms," *IJCSE*, vol. 3, 2012.

[7] J. Xiao, "Automatic building outlining from multi-view oblique images," *Proc. ISPRS Annu. of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, no. 3, pp. 323-328, 2012.

[8] A. Katartzis and H. Sahli, "A stochastic framework for the identification of building rooftops using a single remote sensing image," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 46, no. 1, pp. 259-271, 2008.

[9] X. Wu and Y. Li, "Integrated method of building edge line feature extraction from aerial imagery and digital surface model," in *Proc. 18th International Conf. on Geoinformatics*, 2010, pp. 1-5.

[10] Q. Wang, Z. Jiang, J. Yang, D. Zhao, and Z. Shi, "A hierarchical connection graph algorithm for gable-roof detection in aerial image," *Geoscience and Remote Sensing Letters*, vol. 8, no. 1, pp. 177-181, 2001.

[11] M. S. Nosrati and P. Saeedi, "A novel approach for polygonal rooftop detection in satellite/aerial imageries," in *Proc. 16th IEEE International Conf. on Image Processing*, 2009, pp. 1709-1712.

[12] M. Cote and P. Saeedi, "Automatic rooftop extraction in nadir aerial imagery of suburban regions using corners and variational level set evolution," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 51, no. 1, pp. 313-328, 2013.

[13] M. K. Agoston, *Computer Graphics and Geometric Modeling: Implementation and Algorithms*, London: Springer, 2005, pp. 300-306.

[14] R. G. Kuehni, *Color Space and Its Divisions: Color Order from Antiquity to the Present*, UK: Wiley, 2003.

[15] R. C. Gonzalez and R. E. Woods, *Thresholding in Digital Image Processing*, Pearson Education, 2002, pp. 595-611.

[16] M. Sezgin, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 146-168, 2004.

[17] F. Y. Shih and S. Cheng, "Automatic seeded region growing for color image segmentation," *Image and Vision Computing*, vol. 23, no. 10, pp. 877-886, September 2005.

[18] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 641-647, 1994.

[19] N. Aggarwal and W. C. Karl, "Line detection in images through regularized Hough transform," *IEEE Trans. on Image Processing*, vol. 15, no. 3, pp. 582-591, 2006.

[20] C. R. Jung and R. Schramm, "Rectangle detection based on a windowed Hough transform," in *Proc. 17th Brazilian Symposium on Computer Graphics and Image Processing*, 2004, pp. 113-120.

Teerapat Chaloeivoot received his B.Eng. degree in computer engineering from Department of Engineering, Faculty of Engineering, King Mongkut's University of Technology Thonburi, Thailand in 2011. He is currently pursuing master degree in Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand. His current research interests are image processing, vision and machine learning.

Suebskul Phiphopmongkol received his B.Eng. degree in electrical engineering in 1980 and M.Sci. degree in computer science in 1982 from Chulalongkorn University, Bangkok, Thailand. He received his Ph.D. in computer science from Auburn University in 1991. His current research interests are digital image processing, intelligent transportation systems, artificial intelligence, computer vision, real-time simulation and computer hardware.