

# Automatic Road Detection using Satellite Images

Manohar Yadav<sup>1</sup> and Adil Khan<sup>1</sup>

<sup>1</sup>GIS Cell, Motilal Nehru National Institute of Technology Allahabad, Allahabad, India

**Abstract:** Linear feature extraction has always been a center of attraction for various applications such as urban planning, GIS database update, disaster management, tracking and monitoring and so on. So an accurately extracted image containing roads as the principle component is the need of the hour. With the development of various satellite image processing technologies, the extraction or mapping of linear features such as roads from high resolution satellites images has gained huge momentum in the recent past. This paper presents an automated road detection method using image processing techniques such as connected components analysis and Hough transform. Proposed method is unaffected by noises present in the image. Proposed method was tested on satellite image of developed urban site. Correctness and completeness of 95.6% and 90.5%, respectively were achieved.

**Keywords:** Road detection, Image processing, Connected components, Hough transform

## 1. Introduction

Road network mapping is important for developing geographical information system (GIS) data base related applications for autonomous vehicles, route guidance system, rescue operation and so on. In recent years, many algorithms have been developed to extract linear features such as road from satellite images. Spatiospectral clustering method was introduced [1] for road centerlines identification by analyzing the spatial grouping properties of spectral group using K-Medians clustering. A level set method based on search algorithm was proposed [2]. They determined the evolving curve's boundary pixels to model the road boundaries. A semi-automatic road extraction algorithm using path following approach was introduced [3]. A fully automated road network extraction method from high-resolution satellite multispectral imagery was designed [4]. The road segments were iteratively identified by inspecting the length and width of the object pixels in parallel with the vegetation index. Road was extracted using color of the road or spectrum in case of more spectral bands by calculating the spectral angle with respect to the reference pixel [5]. A neural network based method was proposed for automatic road network extraction using high resolution satellite imagery [6]. Initially road detection was performed using spectral information only and then different texture parameters were computed for each pixel using Gray level co-occurrence matrix from the source image. Extended Kalman filtering and efficient particle filtering based approach was presented for road network extraction [7]. A connected component based technique for automatic extraction of road center line in high resolution satellite imagery was presented [8].

Road extraction from satellite images is affected by shadows of the trees on the road in image, vehicle movement, lane markings, and parking lots. It is also seen that geometrical, topological and radiometric challenges hampers the extraction of roads in the existing literature. Therefore, a new method is proposed for automatic detection of road network from high-resolution satellite image. Proposed method first removes noises in the image and road blockages are also filtered out to minimize possibility of road detection errors.

## 2. Methodology

A new method is proposed for automatic detection of road network using satellite image. Initially input image is pre-processed and converted to binary image then edges are detected. Connected component based segmentation is performed followed by morphological operations to detect dominant edges. Finally Hough transform is implemented to detect road network.

### 2.1. Image Pre-processing and Binary Conversion

It is the first step applied to improve the quality of the image in terms of noise removal. The salt and pepper noise are removed and visibility of edges is enhanced using a Median filter. It helps to achieve better edge detection by smoothing the image, removing salt and pepper noise and enhancing visibility of the edges. The pre-processed RGB color image is converted into binary image using user defined thresholds ( $T_{conversion}$ ). The function inbuilt in Matlab, i.e. `im2bw` is directly used for binary image generation.

### 2.2. Edge Detection and Connected Components Based Segmentation

Edge detection is one of the principle tool used in image processing for feature extraction such as roads, buildings and other manmade objects. Edge detection is used to filter out insignificant information from the image and preserve the structural properties of the objects. Road network in binary image is like connected long edges. So edge detection is applied to preserve edges only and discard all other information. Canny edge detection operator [9] is preferred due to its efficient way to detect edges. This operator is implemented in multi-stages to detect the edges. First, convolution of the image obtained in previous step with the Gaussian filter (see Equation 1) is performed to reduce the impact of noise in the image.

$$G_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i-(m+1))^2 + (j-(m+1))^2}{2\sigma^2}\right); \quad 1 < i, j < (2m+1) \quad (1)$$

Then the intensity gradient of the convolved image is calculated using Sobel filter. The gradient in the X and Y directions are computed using Sobel kernels (see Equation 2) as given below:

$$K_X = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad K_Y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2)$$

Let  $G_X$  and  $G_Y$  are the gradients along X and Y directions, respectively. Then the overall gradient magnitude (see Equation 3) and angle (see Equation 4) are given by:

$$G = \sqrt{G_X^2 + G_Y^2} \quad (3)$$

$$\theta = \arctan\left(\frac{G_Y}{G_X}\right) \quad (4)$$

Non maximum suppression is applied to thin the edges obtained by Sobel filtering that is quite blurred. It performs the suppression of all the gradient values to zero except the local maximal. The local maxima represent pixels location having sharpest change of intensity value. Still few pixels are affected by noise and color variation. Such pixels are filtered out by setting double threshold for gradient and subsequent filtering. The edge pixels having high gradient values are retained while low gradient edge pixels are filtered out by setting two threshold values. The edge pixels are filtered out, which are having gradient value lower than the defined lower threshold value. Similarly the edge pixels having their value higher than the defined higher threshold value are also filtered out while keeping other edge pixels having gradient value lies in between the two thresholds preserved. Further blob analysis is used for edge tracking. It is seen that experimentally the strong edges are

attributed to true edges in the image while the possibility is least to have strong edges due to noise and other spurious elements. But, the possibility is high, where weak edges, i.e. due to noise are also attributed to true edges, therefore they are removed using blob analysis. Blob analysis is executed by looking at weak edge pixels and its connection with the 8 neighborhood pixels. If blob contains at least one strong edge pixel, then associated weak edge point can be identified as one and it should be preserved. It is seen that a weak edge pixel caused from true edges is connected to a strong edge pixel while noise responses are localized and unconnected.

Once the edges are detected the next step is to extract the regions which are not separated by the boundaries. Any set of pixels which are not separated by well-defined edges is called connected, each region of connected pixels is called connected components. These regions of connected pixels i.e. road pixels are extracted using connected component algorithm [10]. Initially each pixel is selected column wise from left to right and then row wise from top to bottom. The object pixel in the binary image is generally represented by 1 while background pixels are represented by 0. The first column of the first row is selected and neighboring pixels of current pixel, i.e. north and west of current pixel (4-pixel connectivity) are selected. If the west of the current pixel is same as the current pixel, then the west pixel is labeled as current pixel otherwise next condition is checked. If the west pixel is different and north pixel is same as current pixel, then north pixel is labeled as current pixel, otherwise again next condition is checked. If both north and west pixels have the same value as the current pixel but are assigned a different label then the minimum of the two labels is assigned to the current pixel otherwise next condition is checked. If both north and west pixels have different labels then a new label is assigned to the current pixel. Next, we create a table and store the equivalence between neighboring labels. Above process is repeated for all selected pixels as current pixel. If element is not background pixel then re-label the pixel with the lowest equivalent label. In this way the connected objects pixels like road pixels are identified by removing the connected objects having pixels less than user defined threshold ( $P_{cc}$ ).

### 2.3. Morphological Operations and Hough Transform

Shape and size of the object pixels are incorporated in the image. Generally the binary image contains internal noise such as shadows by trees, blockages by cars which are present within the object region as well as external noise such as parking lots in the background region. These noises obstruct the accurate road extraction process, hence to remove such noises morphological operations are applied on the image [10]. Therefore, closing operation is used to remove internal noise present within the object region followed by opening operation, which removes the external noise present in the background.

Global edge linking tool, i.e. Hough transform is implemented to detect the linear segments like roads in our case. Hough transform is used to map the spatial domain into parametric space. Basically, a line in the spatial domain (X-Y plane) is mapped into a point in parametric space (m-c plane). For example, if we consider the slope intercept form of a straight line,  $Y = mX + c$ , where  $m$  and  $c$  are slope and intercepts of straight line, respectively. When the normal form of straight line is considered,  $\rho = X \cos \theta + Y \sin \theta$  where  $\rho$  and  $\theta$  are the perpendicular distance from the origin to the straight line and angle made by the perpendicular distance with the X-axis. Then a point in the spatial domain is mapped to a sinusoid in the parametric space ( $\rho - \theta$  plane). Hence the collinear point in the spatial domain is mapped to the set of sinusoids equal to the collinear points and these sinusoids meet at the common point which gives the parametric value, i.e.  $\rho - \theta$  of the straight line in spatial domain. The entire parametric plane is sub divided into accumulator cells and the value in the accumulator cells denotes the number of collinear points in the straight line and the cell location gives the parametric value, i.e. of  $\rho - \theta$  of the particular straight line.

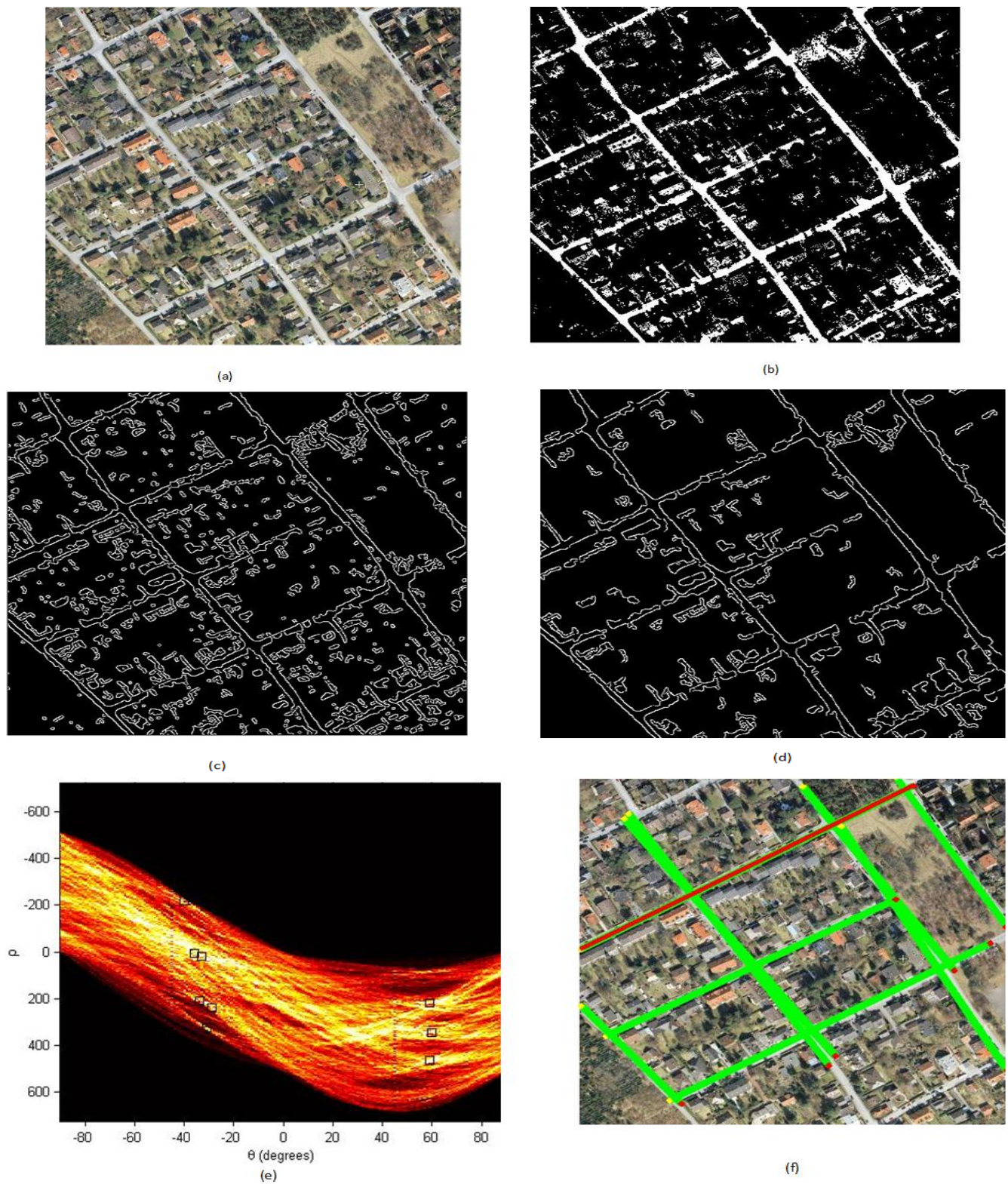


Fig. 1: Two dimensional (2D) view of developed urban region (a) input image, (b) binary image, (c) edge detected image, (d) segmented image, (e) Hough peaks, and (f) output image showing detected road (nature green color).

Finally Hough transform detects road segments based on the user defined threshold ( $L_{road}$ ) on length of Hough lines. It is chosen based on minimum continuous linear road segment in the image.

### 3. Experiment

#### 3.1. Result

Proposed method was tested on the satellite image of developed urban region (see Fig. 1 (a)) using thresholds, i.e.  $T_{conversion}=0.7$ ,  $P_{cc}=50$  pixels, and  $L_{road}=120$  pixels. These thresholds were selected based on several preliminary experiments. Size of input image was 512 pixels by 512 pixels. The result was validated using manually detected road from input image. Length of manually detected road was used as reference road length. Results generated in each stage are shown in Fig. 1.

#### 3.2. Performance Analysis

Detected road network (see Fig. 1 (f)) was validated using the reference road length. Actual detected road length was 281.6 m. Correctly detected road length (TP) was 254.8 m. Incorrectly detected (FP) and missing road length (FN) was 11.7 m and 26.9 m, respectively. Correctness ( $TP/(TP+FP) \times 100$ ) and completeness ( $TP/(TP+FN) \times 100$ ) of 95.6% and 90.5%, respectively were achieved. Proposed method was implemented on Matlab coding platform. Intel (R) Core (TM) i5 @2.50GHz processor and 4 GB RAM enabled PC was used to executed the developed programme. Satellite image of developed urban site ( $512 \times 512$  pixels) was processed in 8 seconds.

### 4. Conclusion and Future Work

A new method for automatic detection of road network from the high resolution satellite imagery is conducted in this paper. First the color image is converted into binary image followed by edge detection. Connected component analysis is done on edge detected image to obtain segmented image contains roads as well as non-road pixels. Internal as well as external noises from the image are removed using morphological operations. Finally Hough transform is applied to extract linear feature, i.e. road network from the imagery. The proposed method was tested on high resolution satellite image of developed urban site. Correctness and completeness of 95.6% and 90.5%, respectively were achieved.

The proposed method fails to detect circular roads using Hough transform. Therefore future work will be detection of curved roads and other complex road structures.

### 5. Acknowledgements

The authors would like to acknowledge the Visualization and Perception Lab of the Department of Computer Science and Engineering at the Indian Institute of Technology at Madras (India) for using the satellite image freely available in its repository.

### 6. References

- [1] P. Agouris, P. Doucette, and A. Stefanidis, "Spatiospectral cluster analysis of elongated regions in aerial imagery," in *Proc. IEEE international conference on image processing*, 2001, pp. 789-792.  
<https://doi.org/10.1109/icip.2001.958612>
- [2] T. Keaton, and J. Brokish, "A level set method for the extraction of roads from Multispectral Imagery," in *Proc. 31st Applied Imagery Pattern Recognition Workshop*, 2002, pp. 141 – 150.  
<https://doi.org/10.1109/aipr.2002.1182268>
- [3] V. Shukla, R. ChandraKanth, and R. Ramachandran, "Semi-automatic road extraction algorithm for high resolution images using path following approach, in *Proc. Third Indian Conference on Computer Vision, Graphics & Image Processing*, 2002, pp. 231-236.
- [4] A. K. Shackelford, "Fully automated road network extraction from high-resolution satellite multispectral imagery," in *Proc. Geoscience and Remote Sensing Symposium*, 2003, pp. 461-463.

<https://doi.org/10.1109/igarss.2003.1293809>

- [5] E. Christophe, and J. Inglada, "Robust Road Extraction for High Resolution Satellite Images," in *Proc. IEEE Conference on Image Processing*, 2007, pp. 437-440.  
<https://doi.org/10.1109/icip.2007.4379859>
- [6] A. Kirthika, and A. Mookambiga, "Automated road network extraction using artificial neural network, in *Proc. IEEE International Conference on Image processing*, 2011, pp. 1061 – 1065.  
<https://doi.org/10.1109/icrtit.2011.5972323>
- [7] J. Subash, "Automatic road extraction from satellite images using extended Kalman filtering and efficient particle filtering," *International Journal of Distributed and Parallel Systems*, Vol. 2, pp. 135-152, November 2011.  
<https://doi.org/10.5121/ijdps.2011.2613>
- [8] C. Sujatha, and D. Selvathi, "Connected component-based technique for automatic extraction of road centerline in high resolution satellite images," *Journal on Image and Video Processing*, Vol. 8, pp. 1-16, December 2015.  
<https://doi.org/10.1186/s13640-015-0062-9>
- [9] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 8, pp. 679-714, November 1986.  
<https://doi.org/10.1109/TPAMI.1986.4767851>
- [10] G. Stockman, and L. G. Shapiro, *Computer Vision*, 1<sup>st</sup> ed. NJ, USA: Prentice Hall, 2001, pp. 69–120.