Urban Detection Based on Straight Line

JIN HE
HEXJ03058504
hejin.udem@gmail.com

Abstract

In this project, I classified urban or rural areas based on straight lines that extracted from high resolution satellite image for the purpose of land development. We could easily distinguish urban and rural regions according to several features of line structures, the common use features as following: 1) average line length 2) entropy of line length 3) average line contrast 4) entropy of line contrast. The reason we choose these 4 features to describe a region is based on the fact that wilderness and rural areas produce more random line length and lower line contrast, these features allow us to get a good distinction between urban and rural area. Next, I made use of PCA method in dimension reduction procedure in order to decrease computation complexity. And then I utilized Bayesian classifier to detect urban or rural areas. The final results is very good with sensibility 100% and specificity 83.3%.

keyword: urban detection, straight line, PCA , Bayesian classifier, EM parameters estimation.

1 Introduction

As we all known, land development is playing an important role in many fields: region planing, disaster prevention, national defense etc., the identification and tracking of land development across large scale areas and also cost lot of time, nowadays, high resolution satellite images offer a new way in solving these problems. In this project I built an automatic image analysis system to classify land use.

My classification system works based on the statistics of straight lines structure of that image. For more details I will talk in later section. My goal in this project is to classify image into regions of little or no development (wilderness or rural) and developed regions (urban or residential). I applied PCA dimension reduction, EM parameters estimation and Bayesian classifier to label each image after extracting straight lines. And I obtained good results (sensibility 100% and specificity 83.3%.).

In related work, Lin and Nevatia[1] and Nevatia and Price[2] detected buildings and other structures in aerial images. Roessner et al.[3] addressed detecting urban areas from airborne hyper-spectral images. Although there has been extensive work on land use classification, quite few structural approaches to this problem. My approach, being totally based on straight lines, offers a very good solution.

2 Straight Line Extraction

2.1 Line Support Region

In contrast to classic edge detectors, the Burns et al. [4] method defines a line segment as an image region, the line support region, namely a straight region whose points share the similar gradient orientation. The whole algorithm in three steps:

1. Partition the image into line support regions by grouping connected pixels that share the similar gradient angle up to a certain tolerance.

2. Approximate each line support region into the best line segment.

3. Validate or not each line segment based on the information in the line support region.

The step 1 is described in this section. Steps 2 and 3 will be described in the following two sections.

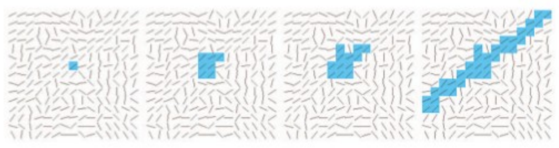

Fig. 1 Growing process of a region. The gradient orientation field is represented by dashes. Marked pixels are the ones forming the region. From left to right: first, second, and third iterations, and final result.

Fig.1 illustrates the procedure of growing a line support region. Each region starts with one pixel and the initial region angle is set to be level line angle (gradient angle) at that pixel. Then, the pixels adjacent to the region are tested; the ones with gradient orientation equal or similar to the region angle is added to this region. At each iteration, the region angle is updated until no new adjacent pixel is added to that region, the region angle is defined by

$$\arctan\left(\frac{\sum_i \sin(ang_i)}{\sum_i \cos(ang_i)}\right)$$

The pseudocode Algorithm 1. (RegionGrow) gives more details. Seed pixels with larger gradient magnitude are tested first as they are more likely to belong to straight edges. When a pixel is added to a region, it is marked "used" and never visited again.

```
Algorithm 1. REGIONGROW
  input:  An image I; a starting pixel $(x, y)$; an angle
          tolerance $\tau$; an image $Status$ where pixels used
          by other regions are marked.
  output: A list $Region$ of pixels.
 1  $Region \leftarrow (x, y)$;
 2  $\theta_{region} \leftarrow$ LevelLineAngle$(x, y)$;
 3  $S_x \leftarrow \cos(\theta_{region})$;
 4  $S_y \leftarrow \sin(\theta_{region})$;
 5  foreach $pixel\ P\ in\ Region$ do
 6      foreach $\bar{P}$ neighbor of $P$ and Status$(\bar{P}) \neq Used$ do
 7          if $Diff(LevelLineAngle(\bar{P}),\ \theta_{region}) < \tau$ then
 8              Add $\bar{P}$ to $Region$;
 9              Status$(\bar{P}) \leftarrow$ Used;
10              $S_x \leftarrow S_x + \cos(\text{LevelLineAngle}(\bar{P}))$;
11              $S_y \leftarrow S_y + \sin(\text{LevelLineAngle}(\bar{P}))$;
12              $\theta_{region} \leftarrow \arctan(S_y/S_x)$;
13          end
14      end
15  end
```

The fact that connected pixels with common level line orientation (gradient angle) always correspond with straight edges is a surprising discovery due to Burns et al. [4]

2.2 Rectangle Approximation of Regions

A valid line support region must be associated with a line segment. A line segment is a rectangle actually, determined by its two endpoints and its width. Its approximation as shown in Fig. 2 includes all of these parameters.

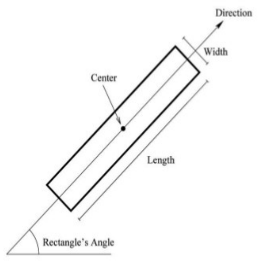

Fig. 2. Line segments are characterized by a rectangle determined by its center point, angle, length, and width.

But how we get these parameters? First, we regard the region of pixels as a solid object and gradient magnitude of each pixel is used as the "mass" of that point. Then the center of mass is used to selected as the center of rectangle and the angle of first inertia axis is selected as the rectangle orientation. Finally, the length and the width are set to the smallest values that make the rectangle to cover the full line support region.

the center of the rectangle (c_x, c_y) is set to

$$c_x = \frac{\sum_{j \in Region} G(j) \cdot x(j)}{\sum_{j \in Region} G(j)}$$

$$c_y = \frac{\sum_{j \in Region} G(j) \cdot y(j)}{\sum_{j \in Region} G(j)}$$

where $G(j)$ is the gradient magnitude of pixel j , j runs over the pixels in the region, the main rectangle's angle is set to the angle of the eigenvector associated with the smallest eigenvalue of the matrix.

$$M = \begin{pmatrix} m^{xx} & m^{xy} \\ m^{xy} & m^{yy} \end{pmatrix}$$

with

$$m^{xx} = \frac{\sum_{j \in Region} G(j) \cdot (x(j) - c_x)^2}{\sum_{j \in Region} G(j)}$$

$$m^{yy} = \frac{\sum_{j \in Region} G(j) \cdot (y(j) - c_y)^2}{\sum_{j \in Region} G(j)}$$

$$m^{xy} = \frac{\sum_{j \in Region} G(j) \cdot (y(j) - c_y)(x(j) - c_x)}{\sum_{j \in Region} G(j)}$$

Fig. 3. shows an example of the result.

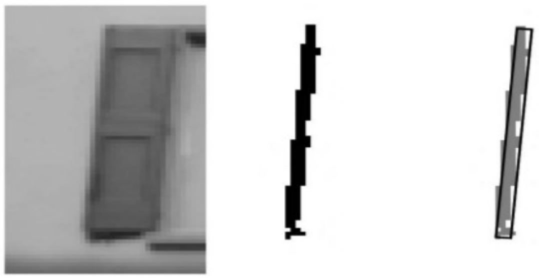

Fig. 3. Example of a rectangular approximation of a line-support region. Left: Image. Middle: One of the line-support regions. Right: Rectangular approximation superposed to the line-support region.

2.3 Line Segment Validation

Each line segment is subject to a validation procedure. The pixels in the rectangle whose level line angle corresponds to the angle of the rectangle up to a certain tolerance τ are called aligned points. The total number of pixels in the rectangle n , and its number of aligned points k , are counted and used to validate or not the rectangle as a detected line segment.

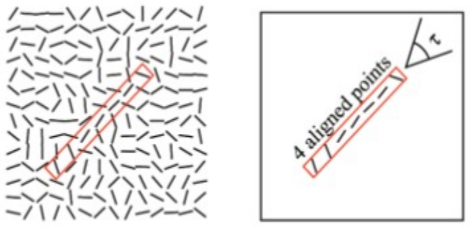

Fig. 4 Left: One line segment shown over the level line orientation field Right: The number of aligned points up to an angular tolerance τ is counted for each line segment. The line segment shown has four aligned points among seven.

The validation step is based on the contrario approach and the Helmholtz principle proposed by Desolneux, Moisan, and Morel.[5] The Helmholtz principle states that no detection should be produced on an image of noise. In the Desolneux et al.[5] a contrario approach, detection is treated as a simplified hypothesis testing problem. Indeed, in the classic decision framework, two probabilistic models are required: one for the background and one for the objects . In a contrario method, the objects are directly detected. But the background model H_0 (white noise model) usually produces non-meaningful objects that we don't need. As Desolneux et al.[5] showed, a background model is just one in which all gradient angles are independent and uniformly distributed. They showed that this is the case for a Gaussian white noise image.

According to the opinion of Desolneux et al.[5], a valid line segment should not be presented by the background model. In other words, structured events are defined being rare in a contrario model.

In the case of line segments, the number of aligned points is the key. We consider event that line segment in a contrario model has as many or more aligned points, as in the observed line segment. Given an image i , and a rectangle r , we note $k(r,i)$ is the number of aligned points and $n(r)$ is the total number of pixels in r , then, expected number of events which as good as observed one is

$$N_{test} \cdot P_{H_0}[k(r,I) \geq k(r,i)]$$

where the number of tests N_{test} is total number of possible rectangles being considered, P_{H_0} is the probability on this contrario model H_0 , and I is a random image following H_0 . The H_0 stochastic model fix the distribution of the number of aligned points $k(r,I)$,which only depends on the distribution of the level line field associated with I . Thus H_0 is a noise model for image gradient orientation rather than a noise model for the image.

The a contrario model H_0 used for line segment detection is therefore defined as a stochastic model of level line field satisfying the following properties:

1) $\{LLA(j)\}_{j \in Pixels}$ is composed of independent random variables.

2) $LLA(j)$ is uniformly distributed over $[0, 2\pi]$.

where $LLA(j)$ is the level line angle at pixel j . Under hypothesis H_0 , the probability that a pixel on a contrario model is an aligned point is $p = \tau / \pi$ and, as a consequence of the independence of the random variables $LLA(j)$, $k(r,I)$ follows a binomial distribution. Thus the probability term as:

$$P_{H_0}[k(r,I) \geq k(r,i)]$$

is given by

$$P_{H_0}[k(r,I) \geq k(r,i)] = B(n(r), k(r,i), p)$$

where $B(n,k,p)$ is the tail of the binomial distribution

$$B(n,k,p) = \sum_{j=k}^{n} \binom{n}{j} p^j (1-p)^{n-j}$$

The number of tests N_{Test} corresponds to the total number of rectangles that could show an alignment at a fixed precision. Notice that the rectangles are oriented, meaning that the order of their starting and ending points is not arbitrary: it encodes which side of the line segment is darker. Thus, a rectangle from point A to point B is a different test from point B to point A. The exhaustive choice is to take all the rectangles starting and ending at image pixels. In a $N \times M$ image this gives $NM \times NM$ different rectangles. Also $\sqrt{NM}$ different width values are considered for each one.

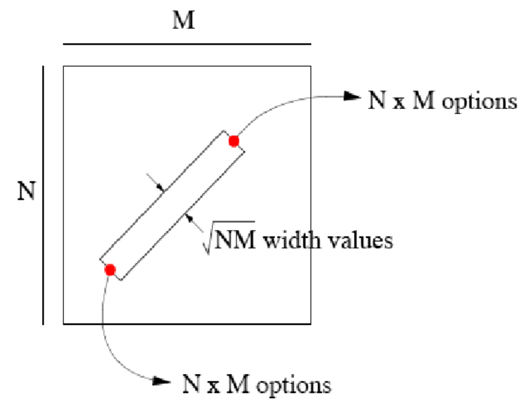

Fig.5. Possible rectangle detected, its starting point has $N \times M$ options , ending point has $N \times M$ options, and its width has $\sqrt{NM}$ options. The total number of possible rectangle in a picture of $N \times M$ pixels is $NM^{5/2}$

The number of rectangles considered is the $(NM)^{5/2}$ The precision p is initially set to the value τ / π but other values are also tested to cover the relevant range of values; γ is the number of different p values. Each rectangle is tested with each p values. The final number of test is $(NM)^{5/2} \cdot \gamma$ Finally, we could get the Number of False Alarms (NFA) :

$$NFA(r,i) = (NM)^{5/2} \cdot \gamma \cdot B(n(r), k(r,i), p)$$

This corresponds to the expected number of rectangles which have a sufficient number of aligned points to be as rare as r under H_0 . When the NFA is large, this means that such an event is expected on a contrario model. On the other hand, when the NFA value is small, the event is a meaningful one. A threshold ε is set to the small enough value, we select the valid line segment which meet the inequality: $NFA(r,i) \le \varepsilon$

2.4 The Complete LSD Algorithm

Algorithm 2. shows a pseudocode for the complete algorithm. Computing input image gradient and gives three outputs: the level line angles, the gradient magnitude, and an ordered list of pixels. There are three parameters involved in LSD: ρ , τ , and ε . The parameter ρ is a gradient threshold: Points with gradient magnitude smaller than ρ are discarded. This parameter will be discussed later. To construct the list, the pixels of image are sorted according to their gradient magnitude; the list starts with the pixel that has the largest gradient magnitude and ends with the smallest one, because the straight edges are more likely to be formed by the pixels with larger gradient value. Parameters τ and ε are angle tolerance and NFA's threshold respectively, explained in the previous section. Algorithm 2. as following shows:

```
Algorithm 2. LSD: LINE SEGMENT DETECTOR
    input:  An image I, parameters $\rho, \tau$ and $\varepsilon$.
    output: A list out of rectangles.
 1  (LLAngles, GradMod, OrderedListPixels) ← Grad($I, \rho$);
 2  Status(allpixels) ← NotUsed;
 3  foreach pixel P in OrderedListPixels do
 4      if Status(P) = NotUsed then
 5          region ← RegionGrow($P, \tau$, Status);
 6          rect ← RectApprox(region);
 7          nfa ← NFA(rect);
 8          nfa ← ImproveRect(rect);
 9          if nfa < $\varepsilon$ then
10              Add rect to out;
11              Status(region) ← Used;
12          else
13              Status(region) ← NotIni;
14          end
15      end
16  end
```

Line 5. RegionGrow, is used to obtain a line-support region. Line 6. RectApprox, gives a rectangle approximation of the region. Line 7. NFA, used to validate or not each line-segment, and these methods described in Section 2.1, 2.2 and 2.3

In this project, the best rectangular approximation of a line support region is the one that gives the smaller NFA value. The step of ImproveRect tries several perturbations to the initial approximation in order to get a better approximation. With a finer precision some points may stop from being aligned, so $k(r,i)$ may decrease; the balance between a smaller p and a smaller $k(r,i)$ can increase or decrease the NFA value depending on the case. And then the best one is kept ($NFA(r,i) \le \varepsilon$).

2.5 Internal Parameters

There are three parameters involved in LSD: ρ, τ, and ε. ρ is a threshold on the gradient magnitude: Pixels with small gradient are not considered. Desolneux et al.[5] showed that gray level quantification produces errors in the gradient orientation angle. This error is negligible when gradient magnitude is large, but can be significant for a small gradient magnitude. Assuming a quantization noise n and an ideal image i we observe:

$$\tilde{i}=i+n\,,\ \ \nabla\tilde{i}=\nabla i+\nabla n$$

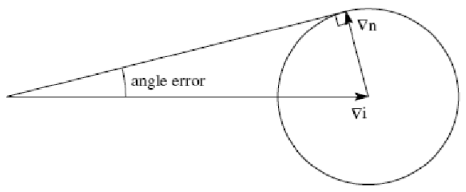

Fig.6. Angle error is generated by adding noise

we have

$$|angle\ error|\leq \arcsin\left(\frac{q}{|\nabla i|}\right)$$

where q is a bound on $|\nabla n|$. The criterion used is to reject pixels where the angle error is larger than angle tolerance τ used in the RegionGrow algorithm. That is we impose $|angle\ error|\leq\tau$ and we get $\rho=\frac{q}{\sin(\tau)}$ .In the usual case , the pixel values are quantized to integer values in $\{0,1,2\ldots\ldots255\}$. Thus the maximal possible error in the gradient is 2 (the worse case is when adjacent pixels get errors of +1 and -1). In addition, the pixels with small gradient magnitude usually correspond to flat zones or slow gradients. Thus I set $q=5$ here.

There are three options for tolerance τ : 11.25, 22.5, 45. The first case is too restrictive and the region is too small: with 45 degree regions often expand to far from the edge. 22.5 is a good compromise.

2.6 Examples

The following examples (600×600 pixels) try to give an idea of result that obtained with line support region method. I got these original high resolution satellite image from google map. The first two images are rural areas, and last two images come from urban region.

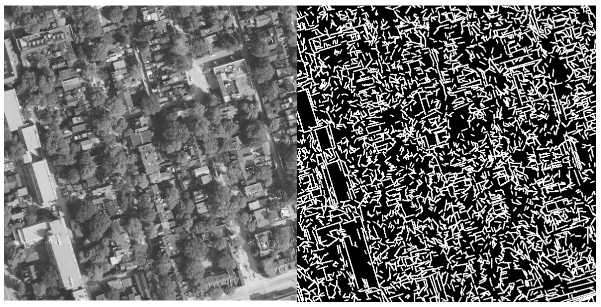

Fig.7. rural region example 1

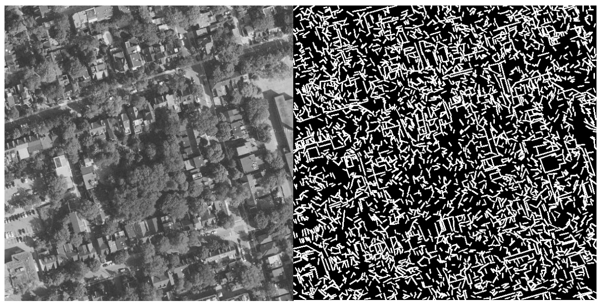

Fig.8. rural region example 2

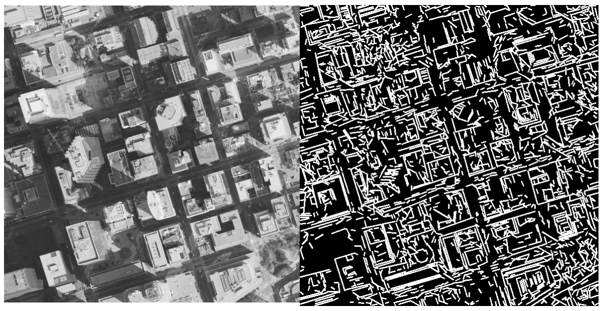

Fig.9. urban region example 1

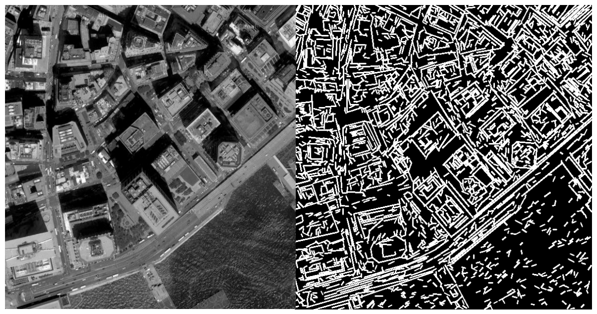

Fig.10. urban region example 2

as showed pictures above, there are lot of details detected because of the high resolution of image like shadow or stuff on the roof, but we still could simply draw a conclusion that the line structure are well organized in urban area, but randomly distributed in rural region. Therefore 4 features are selected in order to measure an image.

- average line length, $\bar{\mu}_l$;
- entropy of line length, E_l ;
- average line contrast, $\bar{\mu}_c$;
- entropy of line contrast, E_c .

3 Statistics Feature Extraction

3.1 Length

The dominant shape of the buildings and street segments in urban regions is rectangular (or compositions of rectangles). In wilderness (or rural) regions, due to the absence of human activity, we expect shorter and more randomly distributed straight lines. This observation generates two statistical features. The first is the sample mean of the length of straight lines, $\bar{\mu}_l$, in the given image.

The second feature measures the entropy in the distribution of line lengths, E_l , in the given image. According to our hypothesis, not only will image features be shorter in undeveloped areas, they should also be more randomly distributed . Therefore, we capture the randomness in the entropy. First forming a histogram of 37 bins (with binwidth = 4 pixels), The maximum bin value represents the longest line segment . The binwidth is picked to obtain a fairly smooth histogram.

Next the histogram is normalized to approximate a probability mass function. Let the normalized vote for bin i be $h_l(i)$ for $i=1,\ldots,37$. The entropy of line length is then obtained from equation:

$$E=-\sum_{i=1}^{N}[h(i)\log_2(h(i))]$$

3.2 Contrast

Contrast provides another indicator of the level of development in a given area. In urban regions for example, we encounter greater variation in the color of buildings, sharp dark light transitions between roads and shiny roofs, and the juxtaposition of buildings and vegetation. Such high contrast variations are less likely in wilderness (or rural) regions. This observation let us to define two features based on contrast.

We define the contrast for each line support region as follows. Let pixels in a given line support region form a set LS ; $(x,y)\in LS$ represents a pixel in that line support region. The contrast value for the line support region, CLSR is the maximum directional derivative magnitude over the region . We also obtain the average directional derivative magnitude, $\bar{\mu}_c$, from CLSR values .

$$CLSR=\max_{(x,y)\in LS}\left[\max_{d\in(x,y)}|G_d(x,y)|\right]$$

Our second feature is the line contrast entropy E_c . To compute it, we quantize the contrast range to 31 bins (with binwidth = 95), These values are obtained as values obtained in length histograms. We then normalize the histogram as usual. Let the normalized vote for bin i and $h_c(i)$ for $i=1,\ldots,31$. Then, the contrast entropy is obtained from same equation as length entropy.

Lastly, when we got a set of 4 features: $\bar{\mu}_l$, E_l , $\bar{\mu}_c$, E_c , we have to consider computation complexity, in order to increase computation efficiency, I reduce the dimension of data with PCA (Principal Component Analysis) method from 4D to 2D. The PCA is a classic approach in resolving dimension reduction problem as it is simple and easily implemented, some basic functions are provided by OpenCV Library.

4 Bayesian Classifier

4.1 Parameters Estimation

Extracted features contain two classes in general : urban area or rural area, and the distribution of the data respect to two mixtured Gaussian, but we don't know labels, mean vectors and matrix covariances for each class. These parameters can be estimated by means of EM algorithm with K-means initialization. And then we can get 4 parameters: μ_0 , μ_1 , Σ_0 , Σ_1 for two Gaussian distribution.

4.2 Bayesian Decision Theory

In this project, I use minimum-error-rate classification to decide possible urban or rural. We define a decision function according to Bayesian theory :

$$p(w_i|x)=\frac{p(w_i)\times p(x|w_i)}{p(x)} \qquad i=0 \quad , \quad 1$$

if $p(w_0|x)>p(w_1|x)$ x belongs to class 0 otherwise it is in class 1.

5 Experiments Results and Analysis

The whole system was built by C/C++ with OpenCV under Linux environment. All high contrast satellite images are from google map with size of 600×600 pixels.

	Training	Test
Urban Area	16	8
Rural Area	26	12

Table 1. Experiment data

To evaluate our system, including sensibility, specificity, the definitions of these criteria as following :

$$sensibility=\frac{TP}{TP+FN}$$

$$specificity=\frac{TN}{TN+FP}$$

Experiment performance:

	Urban Area	Rural Area
Urban Area	TP: 8	FP: 2
Rural Area	FN: 0	TN: 10

Table 2 . Experiment results

My system gives a very good result with sensibility 100% and specificity 83.3%.

6 Conclusion and Discussion

My objective is to learn what information about land development could be extracted efficiently from the photometric structure in the image. LSD has better shape description of the object comparing to other land development detection methods, straight line based classification approach gives us more robust and precise result.

Among the several features that could be used in this project, length and contrast are the most promising for classification, because these features do not depend heavily on the city model, they could be applied to most cities around the world.

Although we get very good result by means of analyzing photometric and geometric characteristics of straight-line segments , it is not easy to detect a region where the pattern is not as distinct as that in urban or rural area in this project, such as suburban residential regions which are in the middle of urban and rural areas, some new features perhaps can help us if we think about the periodicity of straight lines, because residential regions usually have stronger periodicity feature in line segment arrangement than urban or rural areas.

This project indicate that the straight lines of image structure, as captured by the spatial organization, does provide an effective indicator of land development activity. Next I intend to extend these methods to multi-spectral images, combining them with spectral information, to study the impact on classification rates.

References

[1] C. Lin and R. Nevatia, "*Building detection and description from a single intensity image*," Comput. Vis. Image Understand., vol. 72, no. 2, pp. 101–121, Nov. 1998.

[2] R. Nevatia and P. K. E., "*Locating structures in aerial images*," IEEE Trans. Pattern Anal. Machine Intell., vol. 4, pp. 476–484, Sept. 1982 .

[3] S. Roessner, K. Segl, U. Heiden, and K. H., "*Automated differentiation of urban surfaces based on airborne hyperspectral imagery*," IEEE Trans. Geosci. Remote Sensing, vol. 39, pp. 1525–1532, July 2001.

[4] J. B. Bums, A. R. Hanson, and E. M. Riseman, "*Extracting straight lines*," IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-8, pp. 425–455, July 1986.

[5] A. Desolneux, L. Moisan, and J.M. Morel, "*Meaningful Alignments*," Int'l J. Computer Vision, vol. 40, no. 1, pp. 7-23, 2000.