Gap Detection in Road Networks
With Linear and Polynomial Models

Allan Garcia, Alex Honda, Nicholas Meves
Faculty Advisor: Todd Wittman
August 8th, 2008
UCLA Department of Mathematics

Abstract

In many road detection algorithms from satellite images, the detected road network contains many gaps and occlusions. These gaps are primarily due to trees, shadows, and other obstructions that block a satellite's view of the road. To create a complete road network, the various gaps that arise must be detected. We propose a model that assigns a probabilistic value to each region that may contain a gap. Taking into account several perceptual factors and the ground truth from aerial images, created to test the validity and accuracy of our model, we have altered our model to be as precise as possible while considering the many cases that arise when analyzing a discontinuity. We hope that with our completed project, it can be used in cooperation with already developed inpainting methods to close existing gaps, and rule out any possible connections with no chance of containing a road network.

Introduction

With the use of satellites, digital aerial images can be extracted and further analyzed using various road detection algorithms to extract road networks. One problem that arises when using aerial images, such as SAR (synthetic aperture radar) images, is the low resolution of the image. Several factors responsible for incomplete maps include vegetation, houses, tall buildings (urban areas), and shadows, which also tend to occlude the underlying details of the image causing gaps in the resulting road network ***(Figure1.1)***.

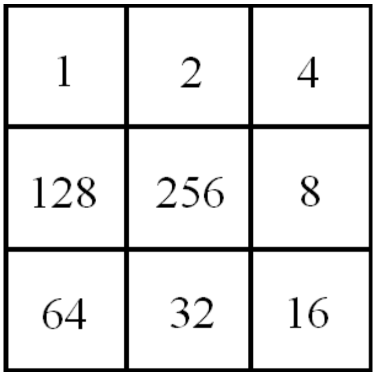
Figure1.1a Here we see the ground truth map where all streets are clearly labeled. [Google Maps 2008]

Figure 1.1b Same image as 1.1a but trees and shadows along left edge cover the street completely from aerial view. [Google Maps 2008]

Road detection in images has been extensively studied in the attempt to extract road networks and fill in their gaps. These approaches to road detection are useful in gaining insight to detecting gaps. Edge and line detection with the Canny operator and Hough transform were common in finding road networks in satellite images. Edge detection, as with the Canny mask, entails finding the image gradient field. Wherever the gradient vector finds local maxima there exists an edge [3]. Another intuitive road model takes an image and parses it into two states: road and background. The detection algorithm parses roads as a logical tree given a starting location and direction [2]. More encompassing models decompose roads into subcategories such as road and intersection. This approach utilizes peak detection on wavelet transforms to determine the corresponding road subcategory [13]. Some models use scale space decomposition to vary the resolution of an image in order to distinguish between salient and non-salient roads. One scale measures continuity and smoothness while another determines road width [7,9]. This approach uses a variation on the deformable contour model known as a ziplock snake [12]. However, while the main purpose of these previous techniques was to close the gaps resulting from undetected roads, where its is generally assumed that the locations of a gap is known, our main focus is to properly identify these gaps and quantitatively describe the probability of existing road.

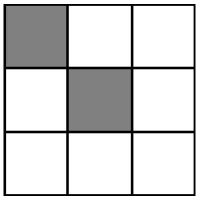
Detecting Road Formations

The first step in road detection is simply identifying the road, a problem that can be made fairly simple by getting a line representation of a road. Before our model can act upon a road map, roads must be one pixel thick, so the algorithm needs a thinned road graph. For our purposes and for extended use of MATLAB, the 3-dimensional matrix of the image can be condensed into a smaller 2-dimensional matrix corresponding to the binary image with non-roads represented by 0; this helps to avoid any complications when applying a convolution mask.

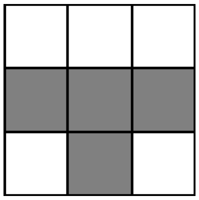


1	2	4
128	256	8
64	32	16

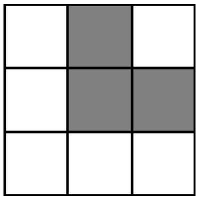
Figure 2.1a Convolution mask with each number raised to a power of 2 to give unique sums.



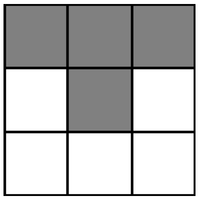
(b) Type: Endpoint
Sum: 257



(c) Type: T-Intersection
Sum: 424



(d) Type: Sharp Turn
Sum: 266



(e) Type: Bump
Sum: 263

Once a binary roadmap exists, a unique 3x3 convolution mask that utilizes powers of two can be convolved with the binary map to identify road formations. A convolution mask is specifically utilized because it can be quickly computed using fast Fourier transforms. The numbers of the 3x3 mask ***(Figure 2.1a)*** cause each unique configuration to have a unique sum. This property allows the convolution value of any pixel to identify the pixels around it without specific examination. With this unique sum, the locations of certain special formations critical to gap detection can be found with a linear search through the convolution matrix for specific convolution values that represent the desired special formation. Once identified, the special points at the center of key formations are saved in a 4xN matrix which contains the following: the row and column indices, the pixel's convolution value, and a value 1-4, used to categorize special point. The convolution map is used to identify four key categories of formations: endpoints ***(Figure 2.1b)***, T-intersections ***(Figure 2.1c)***, sharp turns ***(Figure 2.1d)*** and bumps ***(Figure 2.1e)***.

Once the search through the convolution matrix identifies all special points, steps to find gaps between pairs of these special points can begin.

Eliminating Improbable Connections

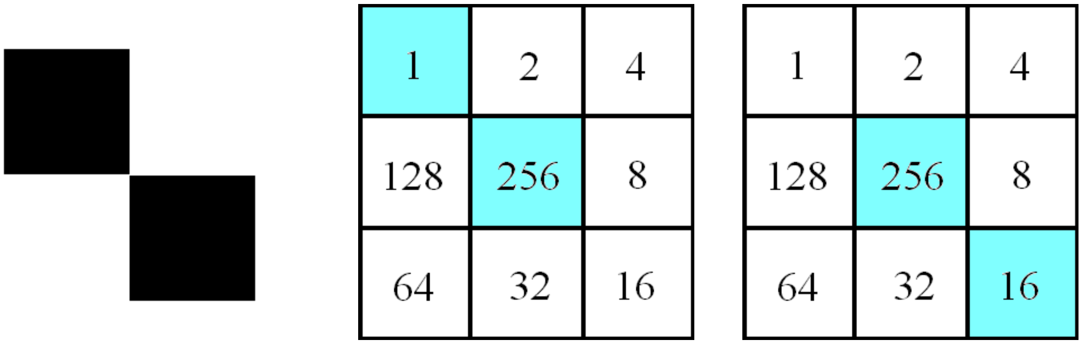
After identifying all the special points, unreasonable pairs of points must be eliminated before performing computationally expensive probability models on them. For N special points, N(N-1)/2 pairs of special points exist. First, using a distance constraint, all pairs of points farther than a certain distance apart, a distance that no gap can possibly exist between, can be eliminated from consideration. Secondly, pairs of special points that have a road between them can also be eliminated; if a satellite detects roads between two special points, then a road directly between the two special points would not be omitted due to occlusions because the other roads were identified ***(Figure 3.1)***.

Figure 3.1 A pair of points that are blocked by other roads

Using a modified version of Bresenham's Line Algorithm, each pixel that would be plotted with the algorithm when creating a line segment between two points is instead inspected for existing roads. Also, since the algorithm could possibly miss detections when it moves to a new height or width at diagonals, the modified algorithm checks the existing road map with double thickness whenever Bresenham's Line Algorithm would have a diagonal. Once the distance and intersection constraints are placed on the possible pairs, only the plausible pairs will then be put into the probability models.

Tracing Roads

Before a probability model can be used on a pair of plausible points, the roads leading up to each of those points must be identified. Both probability models that are discussed later utilized the points on the road leading up to the special points. The MATLAB function *bwtraceboundary* takes in the location of a point and then traces back a continuous line along points connecting to the original point. Expanding on the concepts of *bwtraceboundary,* we created a new line tracing algorithms that efficiently traces along the roads and stores the data in a specific manner more useful to the two probability models. The algorithm utilizes the unique properties of the convolution mask again to efficiently trace back along a road in *O(n)* time until the road stops, makes a sharp turn or branches into multiple roads. Every number in the convolution mask, when added to the number directly across from it, is a multiple of 17.



1	2	4
128	256	8
64	32	16

1	2	4
128	256	8
64	32	16

Figure 4.1 Sample convolution masks of adjacent pixels

Because adjacent roads identify the location of the other road with the two values across from each other ***(Figure 4.1)***, the total sum of pixels in a road includes the following: a multiple of 256 for each center value of the convolution mask, a multiple of 17 for all values across from each other in adjacent pixels, and one last number which identifies the next pixel in the road that was not included in the sum.

$$\sum_{i=1}^{n}\left(x_i - 256\right) \quad \% \;\; 17$$

Figure 4.2 Equation used for the line tracing algorithm.

Using the equation above ***(Figure 4.2)*** on n sequential points in a road, where x_i denotes the convolution value at each point, the next pixel can be found based on the solution. That pixel can then be added to the collection of sequential points in a road and the equation can be used again to find the next.

Probability Model: Linear Method

The first probability model concentrates on modeling straight roads, such as those found in a city grid. The model gives weighted probabilities to three factors: distance, correlation and road momentum. The correlation and road momentum are adjusted with an additional angle check. A probability based on distance is assigned by a decay function. Based on the scale of the road network, the decay rate is adjusted, and the function causes gaps with larger distances to have smaller probabilities.

$$P_{distance} = C_{decay}{}^{distance}$$

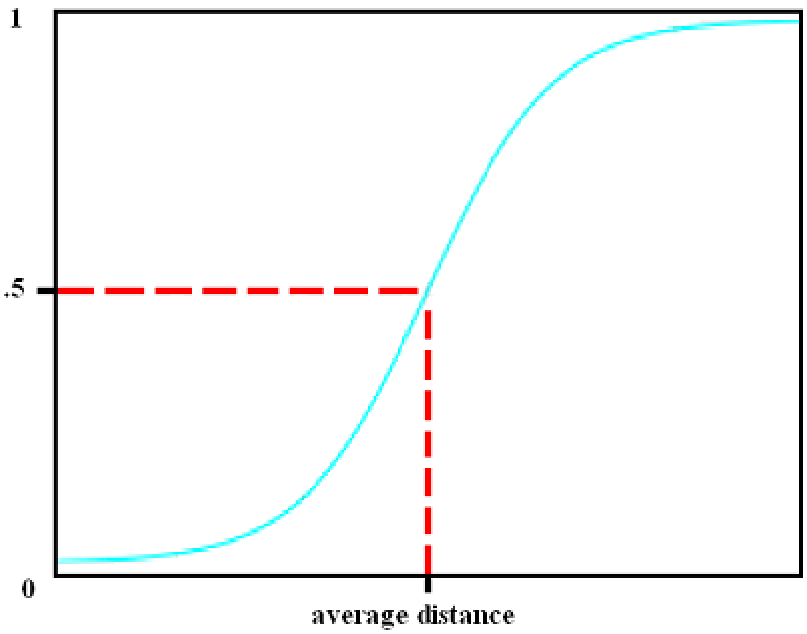
Figure 5.1 Distance probability equation

The model then determines correlation using the roads that were traced back from special points. All the points from both roads leading up to a gap are merged into one data set, and from that merged data set the model computes the correlation coefficient, a numerical value between 0 and 1 displaying how well a set of data correlates to a straight line.

$$P_{correlation} = \frac{cov(x,y)^2}{var(x)var(y)}$$

Figure 5.2 Correlation coefficient equation

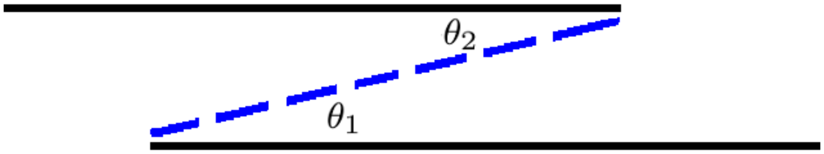
Finally, the model takes into account a road momentum concept, which gives a higher probability to longer than average roads. This road momentum concept stems from the fact that longer, uninterrupted roads are less likely to suddenly stop. The road pairs are assigned a probability based on a logistic curve, translated so the inflection point at .5 corresponds to the average road length.



$$P_{momentum} = \frac{1}{1 + e^{-C_{rate}(distance-average)}}$$

Figure 5.3 Logistic equation and its graph

With correlation and momentum, the probability models must be adjusted to account for the final direction vector of the roads. Some road formations can give high values for correlation and momentum that should low values, such as two parallel, long roads that go past each other in opposite directions. The correlation and momentum are therefore multiplied by an angle ratio, shown below ***(Figure 5.4)***.



$$\theta_{angle-check} = \left(\frac{\theta_1 + \theta_2}{2\pi} \right)^2$$

Figure 5.4 Angle ratio used for the angle check

Once the model determines the distance, correlation and momentum probabilities, the model multiplies the three probabilities by their respective weights and sums them to give a total gap probability. This probability is then multiplied by two final constants, either 1 or 0.8, depending on the type of special points the probability model acts upon, denoted by C_1 and C_2 in ***Figure 5.5*** below. Endpoints are given a weight of 1 while everything else is given a weight of 0.8 because endpoints are more likely to signal a gap than the other special points.

$$P_{total} = C_{distance} P_{distance} + \theta_{angle-check} \left(C_{correlation} P_{correlation} + C_{momentum} P_{momentum} \right)$$

$$P_{final} = C_1 C_2 P_{total}$$

Figure 5.5 Final probability

Probability Model: Polynomial Method

The second probability model uses polynomials to approximate curves. It is weighted by 3 coefficients. The first measures the distance, the second measures how well an estimate matches the data, and the third measures the degree of the approximating polynomial. Other input parameters are used to calibrate the scale of the image, manage clutter resulting from too many matches and to select the maximum degree of the polynomial approximation.

The polynomial model is built off the linear model and has been adapted for detecting larger gaps. Lines can closely approximate very small segments of road networks, but the larger segments are more likely to form curves. To approximate the curve not only do we need to consider more surrounding road data, we need to consider the fact that the road might not be straight. Polynomials were chosen for road modeling because they are the simplest to implement and the MATLAB *polyfit* function allows the ability to specify an arbitrary degree to fit. This makes the polynomial model very flexible to change.

The components of the polynomial probability model are distance, matching, and degree. Distance lowers probability for models connecting far away candidates, matching lowers the probability for models that do not fit the known data well, and degree lowers the probability for higher order polynomials. The distance function was changed to a decaying cosine shape. This allows for larger gaps to retain a higher probability while still decreasing toward zero at the maximum range. The measure of a polynomials ability to match the data is taken to be the L1-norm of the difference between the model and the data in the approximate range where both data and model exist. The last measure of the fitness of a model is the inverse of the degree, which tries to minimize the exponent used.

$$P_{distance} = \left[\frac{1}{2}\cos\left(\frac{(x-1)\pi}{c}\right)+\frac{1}{2}\right] * e^{-\frac{2(x-1)}{3c\pi}}$$

$$P_{final} = C_{distance}P_{distance} + C_{match}P_{match} + C_{degree}P_{degree}$$

Figure 5.6 New distance model and final probability model

One of the downfalls of using *polyfit* is that there is no way to ensure the curve of best fit passes through the endpoints. However, the simplicity and speed of the *polyfit* function make it much more practical to use over a custom best-fit function.

Results

When used on real data, our linear and polynomial models gave promising results. After considering all possible connections, it labels each connection with a color indicating its probability. The linear model, better suited for grid-like maps, was able to make accurate judgments concerning colinearity and distance ***(Figure 6.1)***. However, we sometimes found Bresenham's line algorithm eliminated possible and very probable connections when two terminating collinear roads were at opposite ends of another road ***(Figure 6.2)***. If Bresenham's line algorithm is not used this gives a high probability to this connection. Which is why the intersect test is listed as an option on the GUI for the user to decide. Interestingly, our polynomial model gave similar results when using it in place of the linear model ***(Figure 6.3)***, which further reassures that both models complement each other.

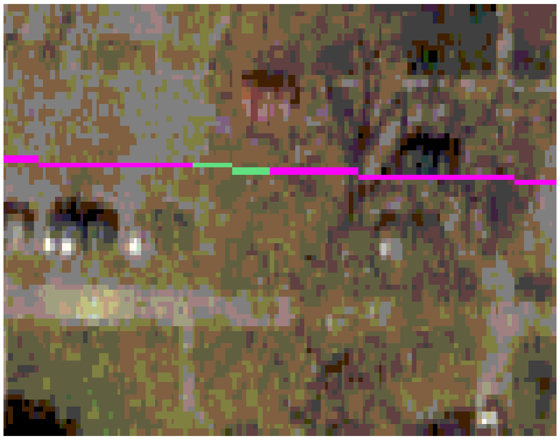
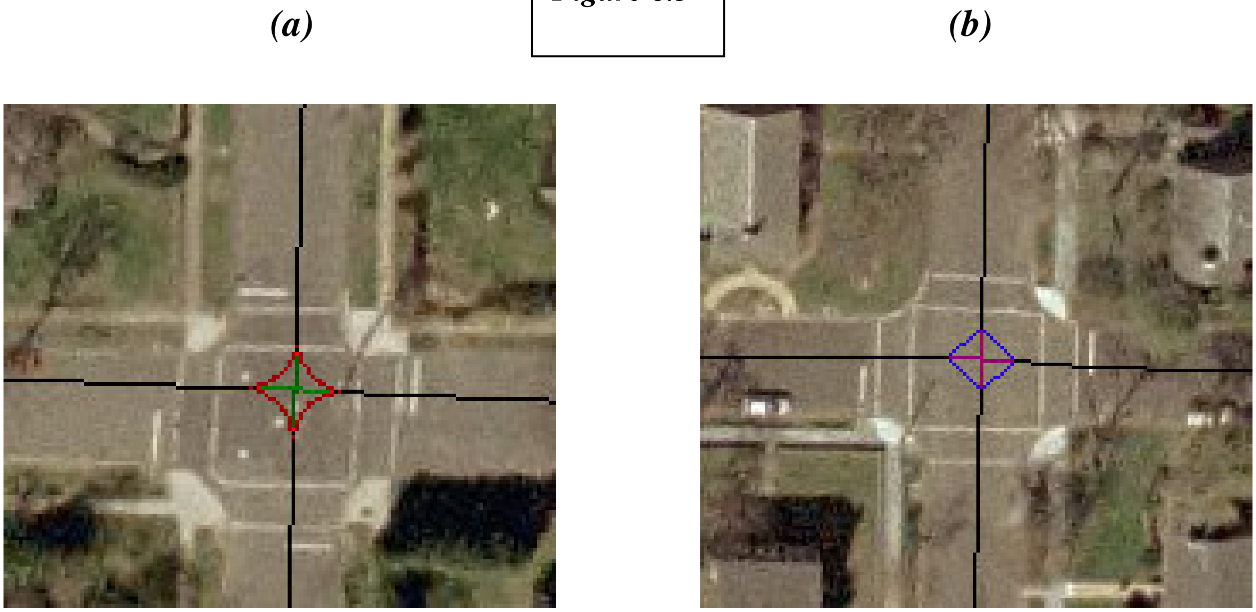


Figure 6.1 The linear method connects two collinear pieces of road. The teal color is the linear match done by the algorithm.

Figure 6.2 The 2 pieces of road separated by the vertical road would not be considered for a possible connection if the Bresenham test were used. When omitted, the linear model detects the gap.

The polynomial model was also effective in finding probable connections, matching roads parametrically with cubic and quadratic functions. Although the user can specify the highest possible degree for a polynomial (highest recommended 6), this allows for more far-fetched connections between gaps. Using original data, the polynomial model identified regions with high probabilistic values, and if it encountered an area with many possible connections, it also gave a probability for each connection ***(Figure 6.4)***. The model does however have its limitations. Whenever the region in consideration is cluttered by many other critical points there are always more possible connections.

Figure 6.3



Here we see results using both, the polynomial method (6.3a) and the linear method (6.3b). They both yield similar results in this intersection.

Sometimes, if the area under examination is too cluttered, the algorithm yields a plethora of results and the best result is not always intelligible since many connections could suffice. Also, a special case of interest is where the region under examination has no critical points to identify in its proximity. If this is the case then no probability will be assigned to this region ***(Figure 6.4)***.

Figure 6.4

Both pictures contain special points not in the vicinity of other special points, causing no gaps to be identified where gaps clearly exist.

Conclusion

Throughout our sample data we were able to detect the standard gaps caused by trees or similar types of occlusions that left small gaps in images. For some regions it was evident by the results from our algorithm and by human recognition that a gap exists. However, sometimes it was difficult to ascertain which connections were the correct connections. Furthermore, if our algorithm yields a variety of possible connections for a region, it is a good indicator that a gap does indeed exist. Our algorithm was also able to identify discontinuities of roads that remained undetected for no apparent reasons, again able to recognize the small "hiccups" in road detection. When encountering different types of context regions, such as urban, suburban, or rural, the coefficient weights played a key role in determining a probable connection for a discontinuity. It's difficult to set a universal set of weights for our coefficients since each map has different arrangements of roads, so our default coefficients in our final projects were empirically derived from testing on different map images. For future work in road detection the Gaussian filter could be used for edge detection. The use of snakes, an active contour model that fuses the properties of ribbon snakes and ziplock snakes, could also be used for gap detection. With our model we hope to set a basis for road detection that will lead to further work and the implementation of these new methods.

Bibliography

[1] Bicego, M., Dalfini, S., Vernazza, G., Murino, P., "Automatic road extraction from aerial images by probabilistic contour tracking." *ICIP03*(III: 585-588).

[2] Geman, D. and Jedynak, B. 1996. "An Active Testing Model for Tracking Roads in Satellite Images." *IEEE Trans. Pattern Anal. Mach. Intell.* 18, 1 (Jan. 1996), 1-14.

[3] Guo Y., Zhengiao B., Yue L., and Yang L., "Genetic Algorithm and Region Growing Based Road Detection in SAR Imaging", in *International Conference on Natural Computation (ICNC 2007)*

[4] Jeon B., Jang J.H., and Hong K.S., "Road Detection in SAR images using Genetic Algorithm", *Dept. of EE POSTECH* (2000) *pp.688*-691.

[5] Lacoste, C., Descombes, X. and Zerubia, J., "Road network extraction in remote sensing by a Markov object process," in *Proc. of IEEE International Conference on Image Processing*, vol. 3, pp. 1017-1020, Barcelona, Spain, September 2003.

[6] Laptev, I., Mayer, H., Lindeberg, T., Eckstein, W., Steger, C., and Baumgartner, A. 2000. "Automatic extraction of roads from aerial images based on scale space and snakes." *Mach. Vision Appl.* 12, 1 (Jul. 2000), 23-31.

[7] Mackaness, William A., and Gordon A. Mackechnie. "Automating the Detection and Simplification of Junctions in Road Networks." GeoInformatica 3 (1999): 185-200.

[8] Mayer, H., Laptev, I., Baumgartner, A. and Steger, C., 1997. "Automatic road extraction based on multiscale modeling, context, and snakes." In: *International Archives of Photogrammetry and Remote Sensing*, Vol. **32** (3-2W32), Haifa, Israel, pp. 106–113.

[9] Merlet, N. and Zerubia, J. 1996. "New Prospects in Line Detection by Dynamic Programming." *IEEE Trans. Pattern Anal. Mach. Intell.* 18, 4 (Apr. 1996), 426-431.

[10] Merlet, N., Zerubia J., "New prospects in line detection for remote sensing images", *ICASSP*, Adelaide, 22 April 1994.

[11] Neuenschwander, W. M., Fua, P., Iverson, L., Székely, G., and Kübler, O. 1997. "Ziplock Snakes." *Int. J. Comput. Vision* 25, 3 (Dec. 1997), 191-201.

[12] Qiaoping Z. and Couloigner I. 2004. "Automatic road change detection and GIS updating from high spatial remotely-sensed imagery." *Geo-Spatial Information Science*. 7, 2 (Jun. 2004), 89-95. *pplication Requirements, Paris* (1999): 185–200.

[13] Rochery, M.[Marie], Jermyn, I.H.[Ian H.], Zerubia, J.B.[Josiane B.], ”Higher-Order Active Contour Energies for Gap Closure.” *JMIV(29)*, No. 1, Septmeber 2007, pp. 1-20.

[14] Rochery, M.[Marie], Jermyn, I.H.[Ian H.], Zerubia, J.B.[Josiane B.], “Gap closure in (road) networks using higher-order active contours.” *ICIP04*(III: 1879-1882).

[15] Rochery, M.[Marie], Jermyn, I.H.[Ian H.], Zerubia, J.B.[Josiane B.], “New Higher-Order Active Contour Energies for Network Extraction.” *ICIP05*(II: 822-825).

[16] Stoica, R., Descombes, X. and Zerubia, J.. “A Gibbs point process for road extraction in remotely sensed images.” *International Journal of Computer Vision*, 57(2): 121-136, 2004.

[17] Zhang, Chunsun, Murai, Shunji, Baltsavias, Emmanuel. “Road network detection by mathematical morphology.” *Proc. of International Workshop on 3D Geospatial Data Production: Meeting Application Requirements, Paris* (1999): 185–200.