5/7/96

Obstacle Detection Algorithms for Rotorcraft Navigation

NASA Grant NAG 2-1410

The Final Report Submitted To
NASA Ames Research Center
Grants Officer, N241-1
Moffett Field, California 94035-1000
NASA Technical Contact: Albert Ahumada

by

Rangachar Kasturi and Octavia I. Camps
Principal Investigators
Department of Computer Science and Engineering and Electrical Engineering
The Pennsylvania State University
University Park, PA 16802
Tel: (814) 863-4254 and (814) 863-1267
E-mail: kasturi,camps@cse.psu.edu

Students:
Ying Huang
Anand Narasimhamurthy
Nitin Pande

Abstract

In this research we addressed the problem of obstacle detection for low altitude rotorcraft flight. In particular, the problem of detecting thin wires in the presence of image clutter and noise was studied. Wires present a serious hazard to rotorcrafts. Since they are very thin, their detection early enough so that the pilot has enough time to take evasive action is difficult, as their images can be less than one or two pixels wide. After reviewing the line detection literature, an algorithm for sub-pixel edge detection proposed by Steger was identified as having good potential to solve the considered task. The algorithm was tested using a set of images synthetically generated by combining real outdoor images with computer generated wire images. The performance of the algorithm was evaluated both, at the pixel and the wire levels. It was observed that the algorithm performs well, provided that the wires are not too thin (or distant) and that some post processing is performed to remove false alarms due to clutter.

Contents

1 Introduction 1

2 Needs and Requirements 2

3 Wire Detection using Computer Vision 3

3.1 Line Detection Algorithms 3

3.2 Steger's Unbiased Detector of Curvilinear Structures 6

3.3 Post Processing 8

4 Data Modeling and Simulation 8

4.1 Illumination Model 9

4.2 Coordinate Systems and Mapping Matrices 10

4.3 Geometric Model 13

4.4 Image Generation of Wire Structures 15

4.5 Noise Model 16

4.6 Background Image 16

5 Performance Evaluation of the Detection Algorithms 17

5.1 Definitions and Notation 17

5.2 Pixel Level Performance Indices 18

5.3 Wire Level Performance Indices 18

6 Experimental Results 19

7 Summary and Conclusions 22

References 23

List of Figures

1 Thin wire with clouds in the background, and noise due to camera jitter. 1

2 Normal representation of the line 4

3 (a) Smoothed parabolic line profile. (b) Convolution with the first derivative of a Gaussian. (c) Convolution with the second derivative of a Gaussian. 7

4 Illumination on a surface. 9

5 World and airborne coordinate systems. 11

6 Camera and Airborne Coordinate Systems. 13

7 (a) Cables of hanging bridges have a parabolic profile. (b) Cables from power lines have a catenary profile. 14

8 Mapping between wire structure element and an image pixel. 15

9 Three cables ($\phi = 18mm., 21.5mm., 45mm$) at 25 seconds to collision for a helicopter moving at (a) 100 km/h and (b) 400 km/h; (c) and (d) edges detected in (a) and (b) using Steger's algorithm; (e) and (f) edges after Hough transform of (c) and (d). 20

10 Pixel level indices (PDR, PFA, PRI) vs distance. 21

11 Wire level performance evaluation. 21

12 Wire level performance evaluation (fragments detected). 22

1 Introduction

Continued advances in the fields of image processing and computer vision have raised interest in their suitability in aiding pilots to early detection of possible obstacles in their flight trajectories. For example, as part of the High Speed Research (HSR) Vision program at the NASA Ames Research Center, we have recently studied, designed, and tested a computer vision system capable of real-time obstacle detection during mid-flight of an aircraft [5, 11]. Before that, obstacle detection on runways during take-offs and landings was also studied [10].

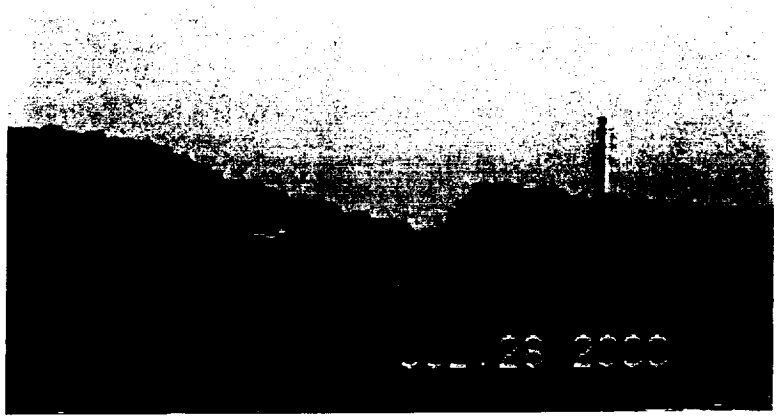


Figure 1: Thin wire with clouds in the background, and noise due to camera jitter.

Some of our previous results can be naturally extended to obstacle detection for low-altitude flight of helicopters. However, the fact that the aircraft is close to the ground for most of the time, places more severe requirements to the algorithms to be used. For example in the image shown in figure 1, the system must deal with multiple ground-based obstacles such as wires, trees, etc. in the presence of severe camera jitter and ever present cluttered background due to the ground. Of these obstacles, the most difficult to detect are wires since they are very thin and their image from the rotorcraft can be

less than a pixel wide at the time to collision.

In this report we describe a preliminary study on the use of a line detection algorithm to detect wire obstacles in the path of rotorcrafts flying at low altitudes. An algorithm proposed by Steger [15] and a Hough transform to eliminate false alarms were identified as good candidates for this task since they are capable of line detection with sub-pixel accuracy. The algorithms are described in section 3. Due to the fact that real data was not available to test the algorithms, a set of testing data was generated by combining natural background images with computer generated wires, corrupted with synthetic noise. The procedure used to generate the data is described in section 4. In order to evaluate the performance of the algorithm a set of experiments were conducted for different sizes of wires at various distances. The experimental protocol used in these experiments is described in section 5 and the obtained results are summarized in section 6. Finally, the conclusions and directions for future research are discussed in section 7.

2 Needs and Requirements

NASA's need for enhanced capabilities in obstacle detection using image processing requires robust, reliable and fast techniques. Low-altitude rotorcraft navigation must often avoid ground-based obstacles such as electric wires, antennas, poles, trees and buildings. Electric wires are very thin objects and hence their images can have sub-pixel thickness. On the other hand, trees and buildings typically occupy several pixels. Furthermore, low-altitude flight implies, in general, severe background clutter due to the ground. Thus, the obstacle detection techniques should provide a high probability of *timely* detection while maintaining a low probability of false alarm in noisy, cluttered images of obstacles exhibiting a wide range of sizes and complexities. Moreover, these techniques should work well under the controlled conditions found in a laboratory and with data closely matching the hypothesis used in

the design process, but it must be insensitive – i.e. must be *robust* – to data uncertainty due to various sources, including sensor noise, camera jitter, weather conditions, and cluttered backgrounds.

3 Wire Detection using Computer Vision

Electric wires between poles hang forming catenary curves and wires holding hanging bridges hang forming parabola curves. However, for detection purposes these curves, can be approximated as piece-wise linear. Thus, for this study we have confined ourselves to the problem of line detection in cluttered images. Furthermore, since wires are thin and their images from a far enough distance are typically less than a pixel wide, we paid special attention to algorithms that could provide sub-pixel accuracy.

3.1 Line Detection Algorithms

Detection of curvilinear and piece-wise linear structures in gray scale images has a wide range of applications including medical imaging, remote sensing, photogrammetry and line drawing understanding and has been the focus of much attention in computer vision research. Next, we present a brief overview of line detection techniques. For more details see for example [9].

- **Edge detection based approaches**

  Lines can be detected by locating "edges" – i.e. pixels where the image gray levels undergo large variations. Thus, most edge-based line detection techniques rely on operators approximating the image gradient. Examples of this approach are the Roberts, Prewitt, and Sobel operators where the image gradient components are approximated as weighted averages of gray level differences in the pixel neighborhood, computed using a pair of small masks. Edges are then found by thresholding the magnitude of the image gradient. This procedure typically results in "thick edges" that must be thinned and gaps that must be closed using a cleaning procedure.

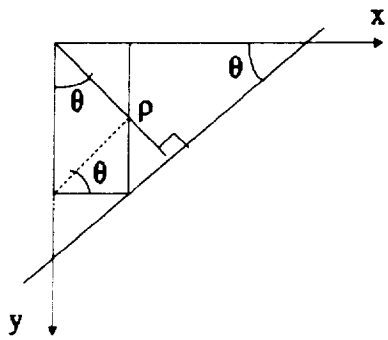


Figure 2: Normal representation of the line

Alternatively, edges can be located by finding the zero-crossings of the image Laplacian. Since the second derivative operator is very sensitive to noise, the image Laplacian is usually applied in conjunction with a noise filtering such as a Gaussian filter.

Probably, the edge detector most commonly used today is the Canny edge operator which uses first derivative of Gaussian filters to closely approximate the operator with optimal signal-to-noise ratio and edge localization.

- **Hough transform based approaches**

Lines and curves can be found by linking adjacent edges into contours. The Hough transform was introduced to detect complex patterns and quickly adapted to detect lines and curves. The main idea of the Hough transform is to map the pattern detection problem into the easier problem of detecting a peak in the space defined by a set of parameters describing the pattern being sought. Consider for example, a line expressed using its normal representation (see figure 2):

$$\rho = x \cos\theta + y \sin\theta$$

where ρ represents the distance between the image origin and the line and θ is the line orientation. Each edge pixel (x_p, y_p) constrains the set of possible pairs of parameters (ρ, θ) of lines containing

the edge by the sinusoidal expression:

$$\rho = x_p \cos\theta + y_p \sin\theta$$

Collinear edges share an unique pair (ρ, θ) which must satisfy all the constraints and thus corresponds to the point in parameter space where all the associated constraints intersect. The longer the line, the more edges sharing the same pair of parameters and the larger the number of constraints intersecting at one point in parameter space. Thus, lines can be found by discretizing the parameter space, associating to each cell a counter of the number of constraints passing through the cell, and finding peaks among the counter values.

- **Curve fitting based approaches**

  Splines are widely used to represent curves. Although splines can be made by joining any kind of function end to end, the most commonly used splines use piecewise cubic polynomials. Cubic polynomials provide enough degrees of freedom to determine edge location and orientation. Algorithms for edge detection using B-splines are described in [2] and [7].

- **Detection of thin lines**

  Thin lines can be detected by modeling them as objects with parallel edges [4, 6] and using a pair of edge detector filters to find the left and right edges of the line or by using differential geometry properties to find ridges and ravines on the image surface $z(x, y)$ [13, 12, 3, 1, 8]. Recently, Steger [15] proposed a detection algorithm based on differential geometry capable of detecting lines with sub pixel accuracy. He applied the algorithm to detect of roads from satellite images and to detect very thin lines in MR and angiogram medical images. Steger's algorithm was capable of detecting lines at different scales, even in the presence of severe clutter. Furthermore, the algorithm retrieved the precise line locations (defined as their median axis) and the line widths

with sub pixel accuracy. Due to the quality of these results, and the similarity between the complexity of the images used in [15] and the ones we are interested for this study, it was decided to evaluate the feasibility of using this algorithm for wire detection.

3.2 Steger's Unbiased Detector of Curvilinear Structures

Next, the main ideas of the detection algorithm proposed by Steger are summarized for the 1D case. A more complete description, including its generalization to 2D, can be found in [15, 14, 16, 17].

The algorithm is based on the concept that a line can be thought of as a one-dimensional manifold in $\mathcal{R}^2$ with a well defined width w. Similarly, curvilinear structures in 2D can be modeled as curves $s(t)$ that exhibit a 1D line profile in the direction perpendicular to the line – i.e. perpendicular to $s'(t)$.

Then, an ideal one-dimensional line profile can be modeled as a symmetrical bar-shaped profile given by

$$f_b(x) = \begin{cases} h, & \mid x \mid \leq w \\ 0, & \mid x \mid > w \end{cases} \tag{1}$$

where w is the width of the line and h is the contrast, or as a more general asymmetrical bar-shaped profiled given by

$$f_a(x) = \begin{cases} 0, & x < -w \\ 1, & \mid x \mid \leq w \\ a, & \mid x \mid > w \end{cases} \tag{2}$$

where $a \in [0, 1]$. A more gradual drop between the line and the background can be modeled by a parabolic profile:

$$f_p(x) = \begin{cases} h(1 - (x/w)^2), & \mid x \mid \leq w \\ 0, & \mid x \mid > w \end{cases} \tag{3}$$

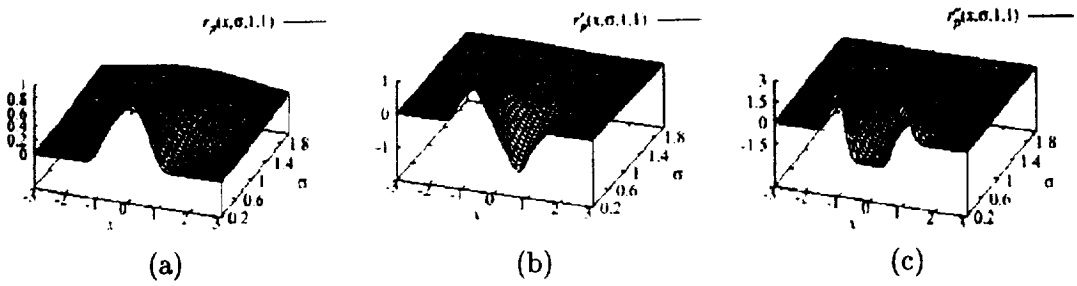


Figure 3: (a) Smoothed parabolic line profile. (b) Convolution with the first derivative of a Gaussian. (c) Convolution with the second derivative of a Gaussian.

A line with a profile given by (3) can be found in an ideal noiseless image $z(x)$ by determining the points where $z'(x) = 0$. Salient lines can be identified by imposing that the magnitude of the second derivative $z''(x)$ at the point where $z'(x) = 0$ should be sufficiently large. In the presence of noise, as discussed in the previous section, the derivatives of the image should be estimated by convolving the image with the derivatives of a Gaussian smoothing kernel with standard deviation σ. The space-scale behavior of the smoothed parabolic profile and its convolution with the first and second derivative of Gaussian filters is shown in figure 3. It can be seen that it is possible to detect the *precise* location of the line *for all* σ. However, if the line follows a profile like (1) or (2), it can be shown that the magnitude of the convolution with the second derivative of the Gaussian has a clear maximum at the true image location *only* if

$$\sigma \geq \frac{w}{\sqrt{3}}$$

The width of the line can be estimated by looking at the places where the magnitude of the output of the first derivative of the Gaussian is maximum. It can be shown that if the line profile is symmetrical and the width is small the width will be estimated too large. Furthermore, in the case of parabolic profiles, the width will be estimated too large for a range of widths. In either case, the mapping between the estimated and the true width can be inverted and the true width can be determined with high accuracy. However, if the profile is not symmetrical, the estimated line location is biased towards

the weak side of the line:

$$l = -\frac{\sigma^2}{2w} \ln(1 - a)$$

but if a is known, the bias can be corrected.

3.3 Post Processing

After lines are detected using the above algorithm, noise and false alarms due to image clutter are reduced by rejecting short lines. This is accomplished by thresholding a Hough transform of the image obtained using the line pixels. The threshold used was fixed to

$$Th = \text{mean} + 0.5(\text{max} - \text{min}) \tag{4}$$

where mean, max, and min are the mean, maximum and minimum counter values in the parameter space, respectively.

4 Data Modeling and Simulation

In order to characterize the performance of the detection algorithm using statistical tests with a given accuracy, we must have large populations of sample representative images. Unfortunately, at the time of this study real testing data was unavailable. Therefore, realistic testing data was generated by combining real (background) images with synthetically generated wires that were corrupted using noise models. The procedure used to generate these images is described next.

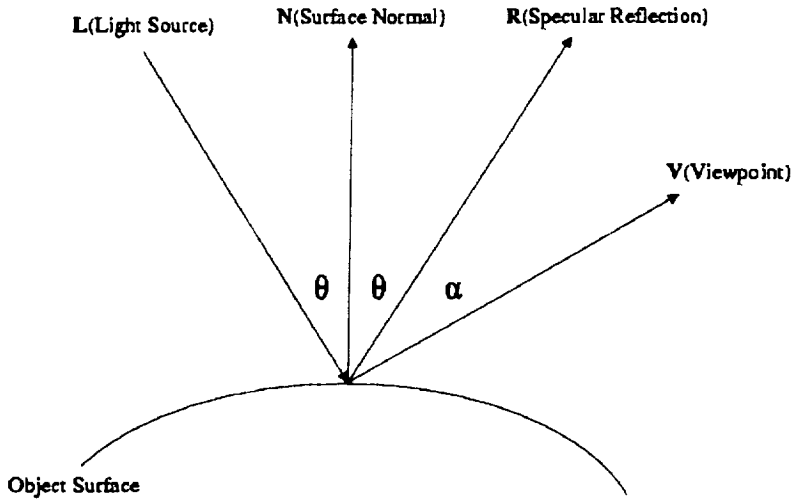


Figure 4: Illumination on a surface.

4.1 Illumination Model

The scene illumination was assumed to have two distinct light sources: ambient light (i.e. diffused light from the landscape, sky and clouds) and a distant point light source (i.e. the Sun/Moon) as shown in figure 4. Furthermore, it was assumed that the ambient light impinged equally on all surfaces of the wire, from all directions. Thus, the reflected light from the wire to the image plane is given by

$$I = I_a K_a$$

where I_a is the intensity of the ambient light and K_a is the ambient reflection coefficient. For the point light source, the Phong illumination model was used. Thus, the reflected light from the wire to the

image plane due to a point light source can be modeled as

$$I = f_{att} I_p [K_p cos(\theta) + w(\theta) cos^n(\alpha)]$$

where I_p is the light intensity of the point light source, f_{att} is the light attenuation factor[1], K_p is the diffusion-reflection coefficient of the wire surface w.r.t. the specific light spectrum of the light source, $w(\theta)$ is the material specular-reflection coefficient.

The typical thickness of power lines ranges between 5mm and 45mm. The typical cruising speed of a helicopter is between 100MPH and 400MPH. Thus, the distance from the camera on board the helicopter to the lines to be detected is such that the image width of the wire is typically less than 1 or 2 pixels. Therefore, the contribution from the specular-reflection component is insignificant and can be set to 0. In addition, the distance from the power lines to the point light source (the Sun) is very large and f_{att} can be set to 1.0. Thus, combining the ambient light model and the point light model, we have

$$I = I_a K_a + I_p K_p cos(\theta)$$

4.2 Coordinate Systems and Mapping Matrices

We will employ three coordinate systems: World System, Airborne System, and Camera System. The *world coordinate system* is a coordinate system fixed with the ground. Wire structures are stationary relative to this coordinate system. The *airborne coordinate system* moves with the helicopter and it has 6 degrees of freedom relative to the world coordinate system: 3 degrees of freedom for its origin (X_a, Y_a, Z_a) and 3 degrees of freedom for its orientation (α, β, γ), where α is called the angle of attack,

[1]For example $f_{att} = min(\frac{1}{C_1 + C_2 r + C_3 r^2}, 1)$ or $f_{att} = \frac{1}{r^2}$, where r is the distance of the object from the light source.

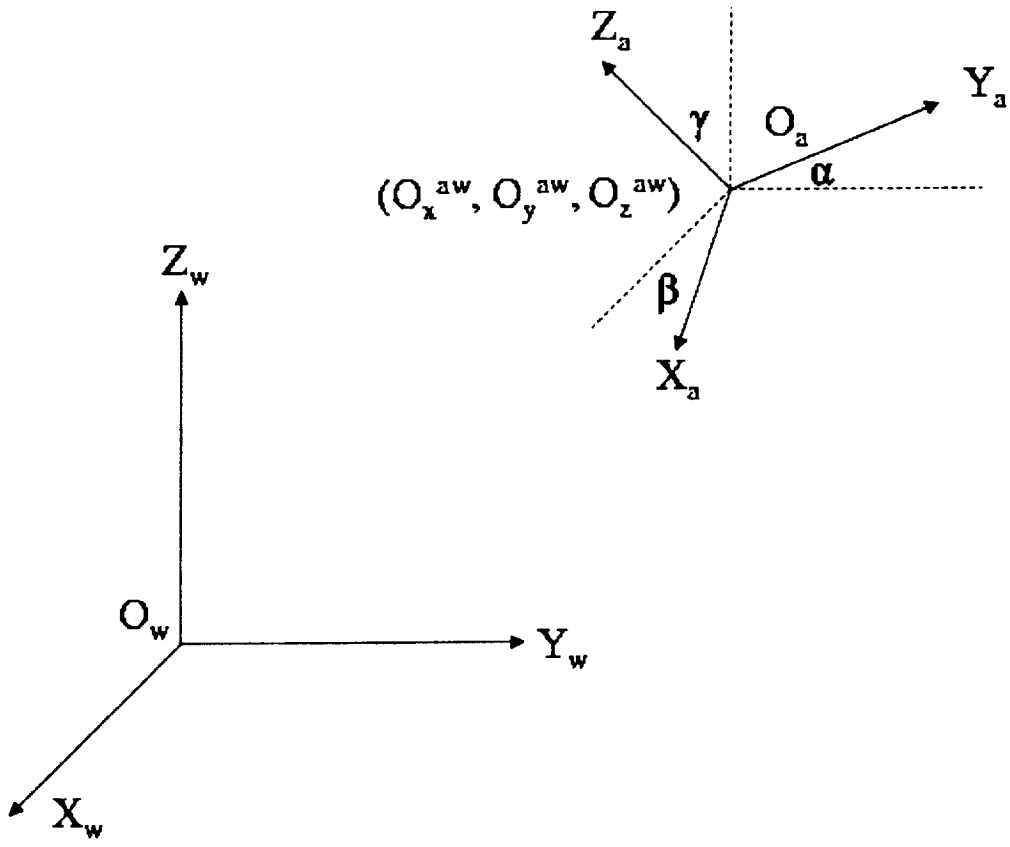


Figure 5: World and airborne coordinate systems.

β the yaw angle, and γ the roll angle (see figure 5). The transformation (mapping) matrix between the world and the airborne coordinate systems can be derived as:

$$\begin{bmatrix} X_a \\ Y_a \\ Z_a \end{bmatrix} = M_3 M_2 M_1 \left(\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} - \begin{bmatrix} O_x^{aw} \\ O_y^{aw} \\ O_z^{aw} \end{bmatrix} \right) \tag{5}$$

where,

$$M_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \quad M_2 = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad M_3 = \begin{bmatrix} \cos\gamma & 0 & -\sin\gamma \\ 0 & 1 & 0 \\ \sin\gamma & 0 & \cos\gamma \end{bmatrix} \tag{6}$$

The *camera coordinate system* is a 2D system fixed in the camera image plane. The pixel coordinates of the images captured by the camera are expressed in this coordinate system. The origin of the camera coordinate system is chosen to be the same as that of the airborne coordinate system (see figure 6). The mapping from the airborne system to the camera system is called the perspective transformation or imaging transformation. This transformation projects 3D points onto a 2D plane. Unlike the regular coordinate transformation, the perspective transformation is a non-linear transformation. Let f be the focal length of the camera. Then, the mapping between the camera and the airborne coordinate systems is given by

$$\frac{X_c}{f} = -\frac{X_a}{Y_a - f}$$

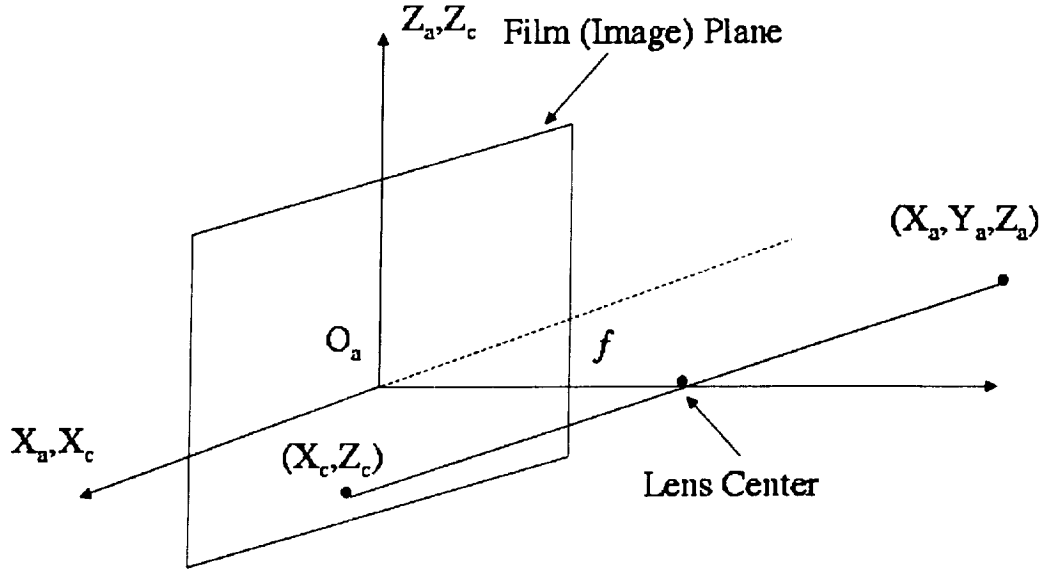


Figure 6: Camera and Airborne Coordinate Systems.

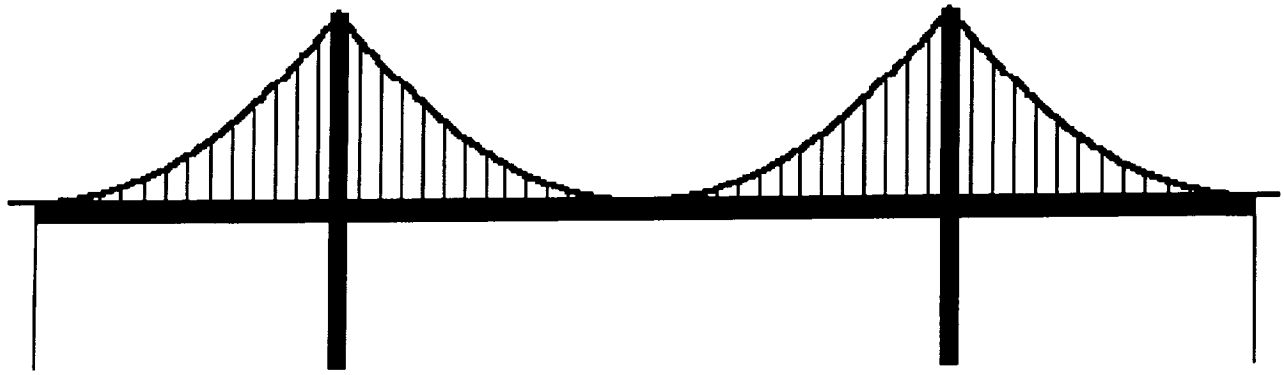
$$\frac{Z_c}{f} = -\frac{Z_a}{Y_a - f} \tag{7}$$

4.3 Geometric Model

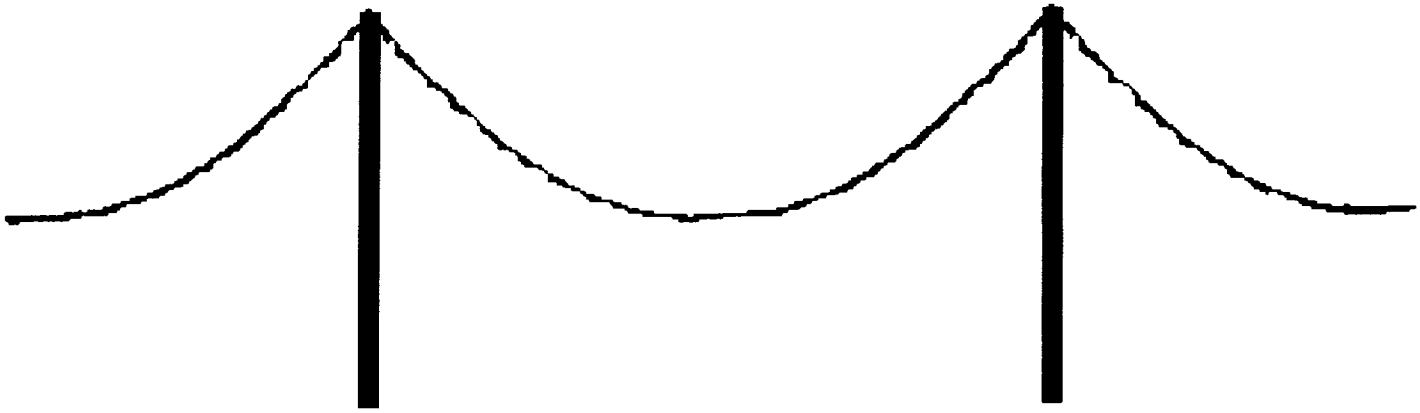
There are two kinds of curve structures which may be of importance to the present application: hanging bridges and power lines (see figure 7). Let μ be the linear weight density of the bridge deck, and T be the cable tension force. Then, with an appropriate choice of the coordinate system, the equation of the cables for the hanging bridge is that of a parabola,

$$y = \frac{\mu}{2T}x^2 \tag{8}$$

Similarly, let μ be the linear weight density of a power line, and T be the power line tension force. With an appropriate choice of the coordinate system, the equation of the power lines is the catenary



(a)



(b)

Figure 7: (a) Cables of hanging bridges have a parabolic profile. (b) Cables from power lines have a catenary profile.

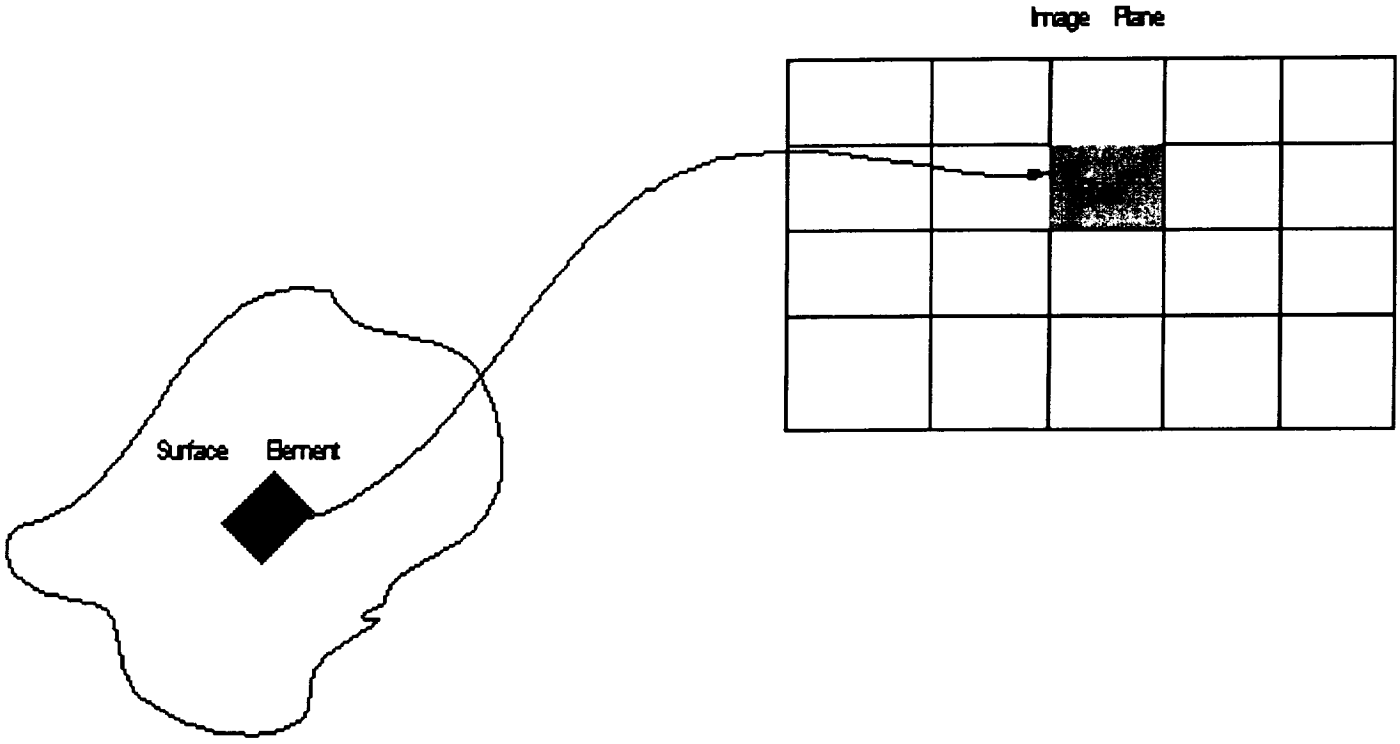


Figure 8: Mapping between wire structure element and an image pixel.

curve:

$$y = \frac{T}{\mu} cosh(\frac{\mu}{T}x) - \frac{T}{\mu} \tag{9}$$

4.4 Image Generation of Wire Structures

Knowing the illumination and geometric models, we can use the mapping between the world and the camera coordinate systems to obtain synthetic images of wire structures. First, the surface of a wire structure is divided into finite elements. The size of each element is chosen so that its mapped area on the image plane is less than 1 pixel. Then, the reflected light from this element is mapped to the pixel within which the center of the element is mapped into (see figure 8). Finally, the pixel gray level value is computed as

$$I_{new} = I_{old} + A_e/A_{pixel} * (I_R - I_{old}) \tag{10}$$

where I_{old} and I_{new} are the image value of this pixel before and after this surface element is mapped, respectively, I_R is the light intensity of the reflected light from this surface element, A_e is the mapped area of this pixel into the image plane[2], and A_{pixel} is the area of a pixel, which is a fixed size for a given digital camera. The image of a power line structure is generated when all surface elements have been mapped to the image plane.

4.5 Noise Model

Noise may be added to the generated image. The noise appears as breaks in the image of the wire structure. The location of the breaks is assumed to have a uniform distribution. The number of breaks follows a Poisson distribution, which has the following probability density function (pdf):

$$f_{poisson} = \frac{\mu^n}{n!} \exp^{-\mu} \tag{11}$$

where μ is the mean of the distribution. The size of the breaks (i.e. number of pixels for a break) is assumed to follow a Rayleigh distribution, whose pdf is

$$f_{rayleigh} = \frac{x}{\sigma^2} \exp^{-\frac{x^2}{2\sigma^2}}, \quad x = 0, \ldots, +\inf \tag{12}$$

with mean $\frac{2}{\pi}\sigma$.

4.6 Background Image

For images captured from a low-flying helicopter, the background mainly consists of two things: clouds and landscape. Thus, to simulate realistic images, the background of the images were obtained by capturing real images using a digital camera. Then, computer generated images of power line structures

[2]Note that the mapped area may not be rectangular any more, even if the original element is.

were superimposed onto these real images.

5 Performance Evaluation of the Detection Algorithms

A performance evaluation protocol for the detection algorithms was designed based on the one described in [18]. The protocol measures to what extent the algorithm detects wires present in the image and whether the algorithm falsely detects wires in the background. These measurements are performed at both the pixel and the wire level.

5.1 Definitions and Notation

Before describing the protocol, the following terms must be defined:

Ground truth image (I_g). Original data, which is the basis of comparison with the detection result. Ground truth images are synthetically generated and consist of one or more dark wires on a white background. Each wire has a different pixel value which is used as an id of the wire.

Number of true wire pixels (P_g) . The number of ground truth image pixels that belong to a wire.

Detected image (I_d) Binary image, resulting after the scene (noisy) image has undergone the defined strategy for detecting edges (Steger's algorithm, followed by a Hough transform).

Number of detected wire pixels (P_d) . The number of detected image pixels that were labeled as belonging to a wire.

Overlap. If there is an edge pixel at position (x, y) in I_g and there exist an edge pixel at the same position (x, y) or any of its eight neighbors in I_d, then an overlap is said to occur between the two edge pixels. Let $P_{g \times d}$ be the number of overlap pixels between I_g and I_d.

5.2 Pixel Level Performance Indices

Next we define a set of indices to measure the performance of the algorithm at the *pixel level.* These indices provide information about the number of wire pixels correctly and incorrectly labeled.

True Positives or Pixel Detection Rate (PDR). The *pixel detection rate* (PDR) is the rate of positive responses in the presence of instances of the sought feature:

$$PDR = \frac{P_{g \times d}}{P_g} \tag{13}$$

Pixel False Positives or False Alarms (PFA). The *pixel false alarm* (PFA) is the rate of positive responses in the absence of the sought feature:

$$PFA = 1 - \frac{P_{g \times d}}{P_d} \tag{14}$$

Pixel Recovery Index (PRI). The *pixel recovery index* (PRI) combines the PDR and the PFA in a single index:

$$PRI = \alpha PDR + (1 - \alpha)PFA, \quad 0 \leq \alpha \leq 1 \tag{15}$$

where α weights the relative importance of true positives and false alarms. (In our study, $\alpha = 0.5$.)

5.3 Wire Level Performance Indices

The pixel level performance criteria defined above do not provide a measure of how many individual wires or which fragments of each wire were detected. For this purpose, the following *wire level indices* are defined:

Wire Detection Rate (WDR). A wire is said to be detected if a number greater than a threshold

(in our case 50%) of its pixels are detected. The *wire detection rate* (WDR) is the ratio of the total number of wires detected to the total number of wires in the ground truth image.

$$WDR = \frac{\text{Number of wires in } I_d}{\text{Number of wires in } I_g} \tag{16}$$

Detection Fragmentation Rate (DFR) A measure of the fragment of each wire detected. The *detection fragmentation rate* (DFR) is defined as

$$DFR = \frac{\text{Number of pixels detected in a wire}}{\text{Number of pixels in the wire}} \tag{17}$$

6 Experimental Results

Synthetic images were generated following the procedure described in section 4. Each image had three wires of different diameters (18 mm., 21.5 mm. and 45 mm.) viewed from different distances ranging between 560 m. to 2,800 m. Figure 9(a) and (b) show images where the time to collision is 25 seconds for helicopter speeds of 100 km/h (694.4 m), and 400 km/h (2777.78m), respectively. Edges were detected in the synthetic images by using an implementation of Steger's algorithm provided by the author. Examples of the results are shown in figure 9(c) and (d).

The three pixel level indices for the different cable distances are shown in figure 10. As expected, the performance degrades as the distance increases. Due to time constraints, the results illustrated here were obtained by applying only Steger's algorithm, without post processing. While the false alarms are relatively high, as it is seen in figure 9 (e) and (f) post processing does eliminate most of the false alarms. The wire detection rate and the detection fragment rate are shown in figures 11 and 12, respectively. These plots show that most of the misdetection errors are due to the thinnest of the wires, indicating a limitation on the diameter of the wires that can be safely detected.

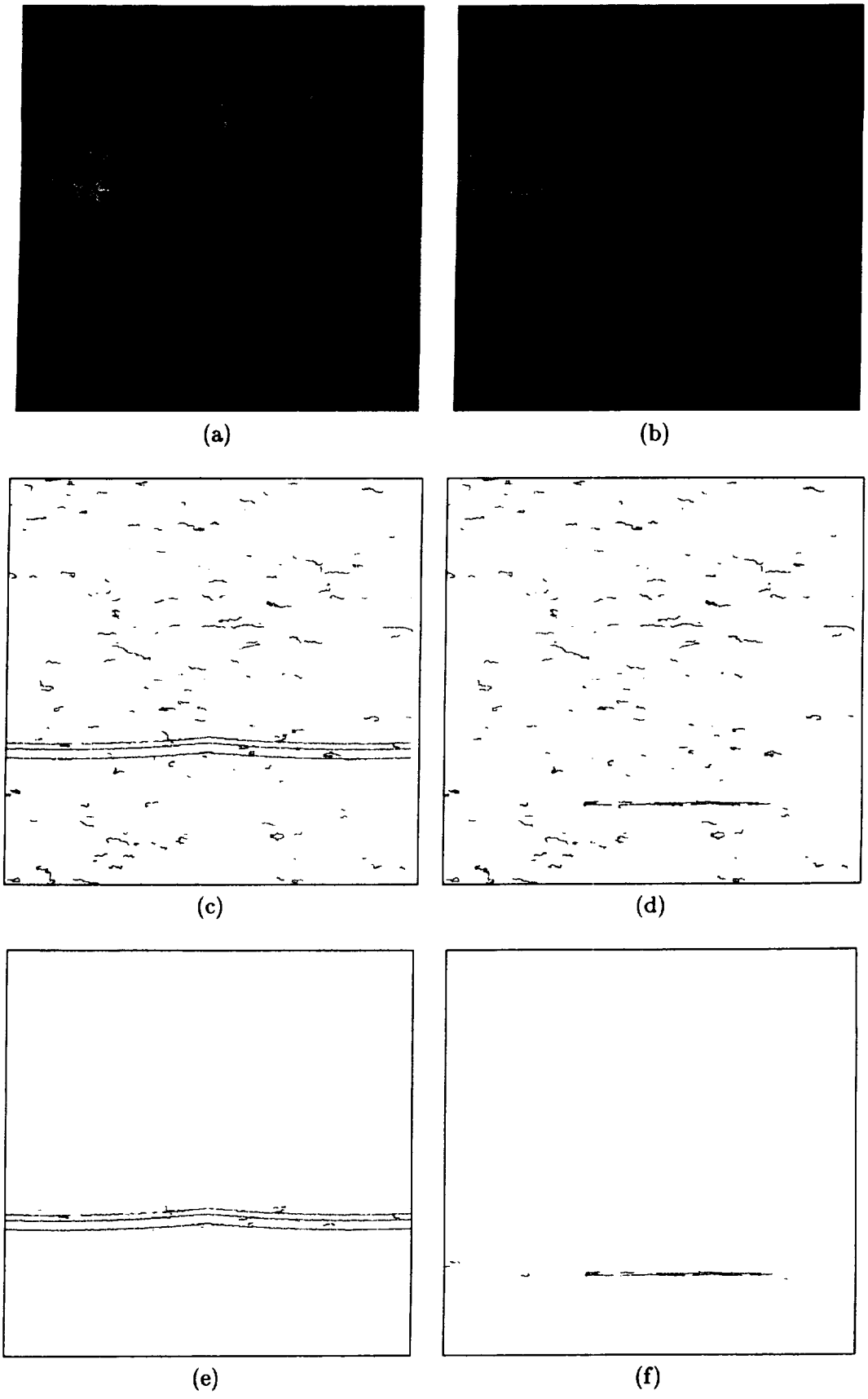


Figure 9: Three cables ($\phi = 18mm., 21.5mm., 45mm$) at 25 seconds to collision for a helicopter moving at (a) 100 km/h and (b) 400 km/h; (c) and (d) edges detected in (a) and (b) using Steger's algorithm;

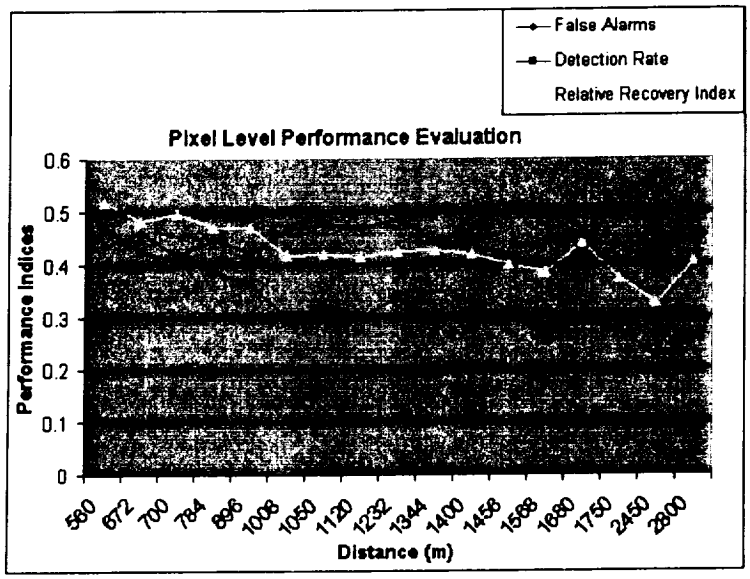


Figure 10: Pixel level indices (PDR, PFA, PRI) vs distance.

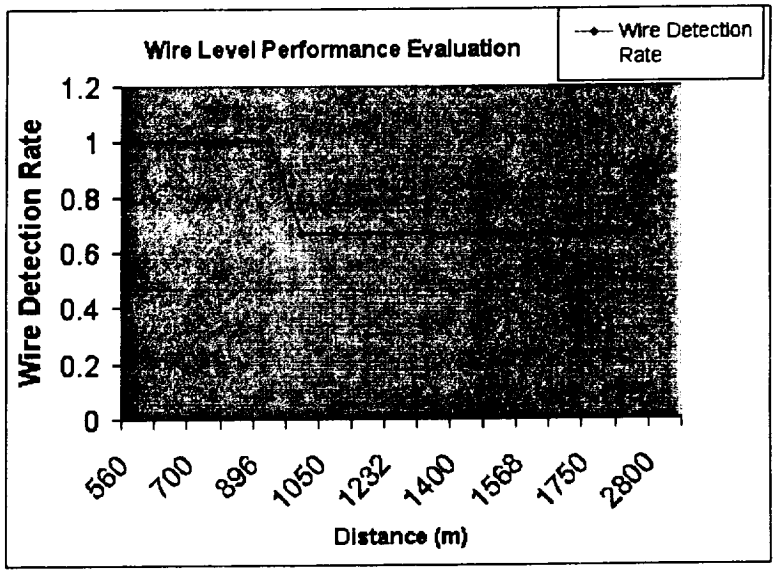


Figure 11: Wire level performance evaluation.

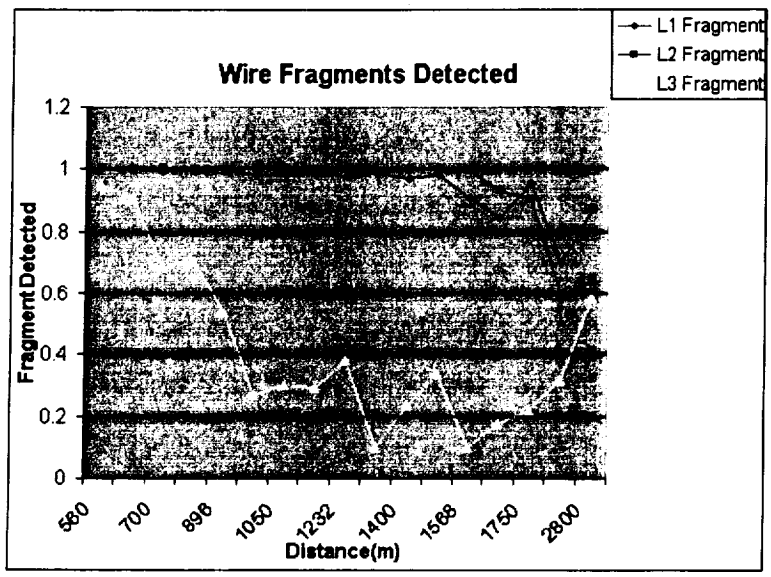


Figure 12: Wire level performance evaluation (fragments detected).

7 Summary and Conclusions

In this report we addressed the problem of obstacle detection for low-altitude rotorcraft navigation with emphasis on wire detection. A line detector with sub pixel accuracy proposed by Steger was identified from the published literature. Steger's algorithm was tested using a set of synthetically generated images combining real backgrounds with computer generated wire images. A set of performance indices at the pixel and the wire level were defined to evaluate the merits of the algorithm for the task at hand. The results of the experiments show that the algorithm can potentially detect wires, provided that they are not too thin or very far. It was also observed that the algorithm produces false alarms due to the severe image clutter. However, most of these false alarms can be successfully eliminated by using a simple – albeit time consuming – post processing such as a Hough transform that discards short lines.

Future research should explore 1) integration over time of the obtained results to detect very thin (or distant) wires and 2) use image context – i.e. search for wires near power poles.

References

[1] A. Busch. A common framework for the extraction of lines and edges. *Int. Archives of Photogrammetry and Remote Sensing*, 31 part B3:88–93, 1996.

[2] G. Chen and Y. H. H. Yang. Edge detection by regularized cubic b-spline fitting. *T-SMC*, 25:636–643, 1995.

[3] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach. Rigdes for image analysis. Technical Report TR93-055, Dept. of Computer Science, Univ. of North Carolina, Chapel Hill, 1993.

[4] M. A. Fischler, J. M. Tenenbaum, and H. C. Wolf. Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique. *Computer Graphics and Image Processing*, 15:201–223, 1981.

[5] T. Gandhi. *Image Sequence Analysis for Object Detection and Segmentation.* PhD thesis, The Pennsylvania State University, Dept. of Computer Science and Engineering, December 1999.

[6] D. Geman and B. Jedynak. An active testing model for tracking roads in satellite images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(1):1–14, 1996.

[7] A. Goshtasby and H. L. Shyu. Edge detection by curve fitting. *IVC*, 13:169–177, 1995.

[8] R. M. Haralick, L. T. Watson, and T. J. Laffey. The topographic primal sketch. *Int. Robotics Research*, 2(1):50–72, 1983.

[9] R. Jain, R. Kasturi, and B G. Schunck. *Machine Vision.* McGraw-Hill Inc., 1995.

[10] R. Kasturi, O. Camps, L. Coraor, T. Gandhi, and S. Devadiga. Detection of obstacles on runway using ego-motion compensation and tracking of significant features. Technical report, Dept. of Computer Science and Engineering, The Pennsylvania State University, 1996.

[11] R. Kasturi, O. Camps, L. Coraor, T. Gandhi, K. Hartman, and M. T. Yang. Obstacle detection algorithms for aircaft navigation. Technical report, Dept. of Computer Science and Engineering, The Pennsylvania State University, January 2000.

[12] I. S. Kweon and T. Kanade. Extracting topographic terrain features from elevation maps. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 59(2):171–182, 1994.

[13] J. B. A. Maintz, P. A. van den Elsen, and M. A. Viergever. Evaluation of ridge seeking operators for multimodality medical image matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(4):353–365, 1996.

[14] C. Steger. Analytical and empirical performance evaluation of subpixel line and edge detection. In Kevin J. Bowyer and P. Jonathon Phillips, editors, *Empirical Evaluation Methods in Computer Vision*, pages 188–210, Los Alamitos, California, 1998. IEEE Computer Society Press.

[15] C. Steger. An unbiased detector of curvilinear structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):113–125, Feb 1998.

[16] C. Steger. Subpixel-precise extraction of watersheds. In *7th International Conference on Computer Vision*, volume II, pages 884–890, 1999.

[17] C. Steger. Subpixel-precise extraction of lines and edges. In *International Archives of Photogrammetry and Remote Sensing*, volume XXXIII, part B3, pages 141–156, 2000.

[18] L. Wenyin and D. Dori. A protocol for performance evaluation of line detection algorithm. *Machine Vision Applications*, 9(5/6):240–250, 1997.