Detecting Wires in Cluttered Urban Scenes Using a Gaussian Model

Joshua Candamo, Dmitry Goldgof, Rangachar Kasturi and Sridhar Godavarthy
Computer Science and Engineering
University of South Florida, Tampa, USA
{candamo, goldgof, r1k, sgodavar}@cse.usf.edu

***Abstract*—A novel wire detection algorithm for use by unmanned aerial vehicles (UAV) in low altitude urban reconnaissance is presented. This is of interest to urban search and rescue and military reconnaissance operations. Detection of wires plays an important role, because thin wires are hard to discern by tele-operators and automated systems. Our algorithm is based on identification of linear patterns in images. Most existing methods that search for linear patterns use a simple model of a line, which does not take into account the line surroundings. We propose the use of a robust Gaussian model to approximate the intensity profile of a line and its surroundings which allows effective discrimination of wires from other visually similar linear patterns. The algorithm is able to cope with highly cluttered urban backgrounds, moderate rain, and mist. Experimental results show a 17.7% detection improvement over the baseline.**

I. INTRODUCTION

Wire detection plays an important role in many computer vision applications. Early wire detection work [1] focused on exploring possible methods for wire detection. These initial methods were designed for simple or uniform backgrounds. However, in low altitude reconnaissance videos, the background is non-uniform, highly cluttered, and detection is much more complex. Some of our earlier work has focused on urban reconnaissance [2][3], using a simple model of a line which does not take into account the line surroundings. In [4], we had proposed an algorithm that accounts for the surroundings of the wire, assuming the width of the wire is known a-priori.

The wire detection algorithm proposed in this paper uses a novel robust line profile model, which describes the intensity of the line and its surroundings with Gaussians. The model does not require the width of the wire to be known a-priori. The profile allows effective discrimination of wires from other visually similar linear patterns. Using the explicit line model and combining size, shape and geometrical principles, the algorithm is able to detect wires in heavily cluttered urban backgrounds. The algorithm in [4] is used as the baseline for evaluation. Experimental results are encouraging and show up to 17.7% wire detection improvement over the baseline.

The algorithm focuses on images typical of urban reconnaissance missions, including data with a wide range of conditions of weather, lighting effects, sensor noise, and scene complexity, without the use of temporal information or tracking.

II. WIRE DETECTION ALGORITHM

The proposed algorithm has several stages. The individual frames are subjected to line fitting to help reduce the number of probable line segments. Chain code histograms are used for noise reduction followed by a novel Gaussian based line profile estimation to discriminate wires from other lines. The discrimination is based on the premise that a wire has a uniform (local) background with symmetric profiles immediately above and below the wire. The results are further tidied up by a local weight thresholding which eliminates false positives. Finally we propose a scene correction technique to discard wires that are inconsistent with global statistics. A flowchart of the algorithm is given in Figure 1.

A. Pre Processing

Our goal is to create a robust feature map, which maximizes the number of pixels that belong to wires, while minimizing the number of pixels to be processed. The chosen feature map consists of a reduced edge map, similar to those used in previous wire detection algorithms [2][3][5]]. The feature map is computed as described in [4]. The noisy pixels in the feature map are minimized using 8-directional chain code histograms.

Chain code analysis has been used in closely related contexts to line pruning, such as general shape recognition [6][7], and proven effective for digital straight line segment recognition [8]. Each connected component is represented using a chain code as a sequence of directional codes from one pixel to the adjacent one. Directions are coded with integer values from 0 to 7, in a counterclockwise sense starting from the direction of the positive x-axis. The chain code histogram is given by the discrete function:

$$hist(v) = \frac{n_v}{n},\ \ v = 0,...,7 \quad \textbf{(1)}$$

Where n_v is the number of encoded pixels with v value in the chain, and n is the number of total pixels encoded. Only the pixels labeled with the code with the highest count in the histogram are considered for further processing. The relationship between those pixel coordinates is analyzed, by fitting a straight line.

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.114

IEEE computer society

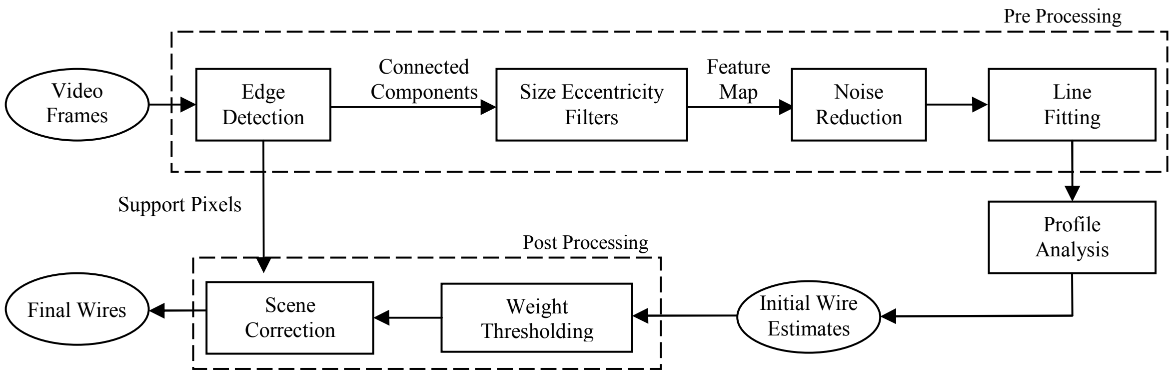
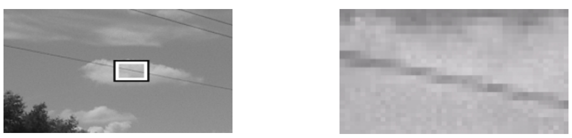


Figure 1. Flowchart of proposed algorithm.

B. Line Discrimination using Profile Analysis

Lines look like bright or dark thin regions surrounded, locally, by darker or brighter areas. For example, in Figure 2-b, the dark thin region in the middle represents the portion of the power line marked in Figure 2-a.



(a) (b)

Figure 2. (a) Original image, and (b) enlarged area corresponding to the rectangle shown in (a). The power line shows as a dark thin region. The clouds show as brighter areas on both sides.

Often, line detection methods consider cross-sections of lines to be bar-shaped. Steger [9] proposes the use of a realistic "rounded" parabolic profile. We propose the use of a more robust Gaussian-based profile model that is used to approximate the gray level distribution of the profile of a line. The profile is also used as a discriminative measure for different linear patterns.

Gaussian models have proven particularly effective in image segmentation [10] and background modeling [11]. One could characterize profile estimation as merely a segmentation problem with 3 regions, one region for the line and one region at each side surrounding the line. A simple online *K*-means approximation (*K*=3) can be used to solve this problem. The gray level distribution of a line of width $w=|w_2-w_1|$ measured along the vertical axis and both surrounding intensity regions are approximated by three Gaussians (G_i) with mean μ_i and variance$\sigma\sigma_{ii}$, i = 1, 2, 3 given by:

$$f_p(l)=\begin{cases} G_1(l,\mu_1,\sigma_1^2), & w_1 \le z \le w_2 \\ G_2(l,\mu_2,\sigma_2^2), & z > w_1 \\ G_3(l,\mu_3,\sigma_3^2), & z < w_2 \end{cases} \quad \textbf{(2)}$$

where z represents the pixel position and l represents the gray level intensity of the pixel. A method to estimate the Gaussian-based model described by Eq. 2 was proposed in [4] but is rather limited, as it requires the width of the lines to be known a-priori. Alternatively, we propose an iterative estimation algorithm.

For each line (potential wires), we compute the gray level distributions for the wire and surrounding regions by shifting the position of the line one pixel at a time (t = 0, 1, 2...) as:

$$y = mx + c \pm t \quad \textbf{(3)}$$

For each iteration, the normalized Euclidean distance [12] is used to determine to which region (Gaussians) each line corresponds to. We stop once all Gaussians have been computed. The estimated profile is finally used to discard asymmetric line profiles. In Figure 4-a, it is obvious how the rooftop of the building could be mistaken for a wire, even by humans and how the profile in Figure 4-b generated from the boxed region in Figure 4-a is sufficient to correctly discriminate the wires from similar linear patterns. The estimate of the wire's width w is computed using the number of update iterations.

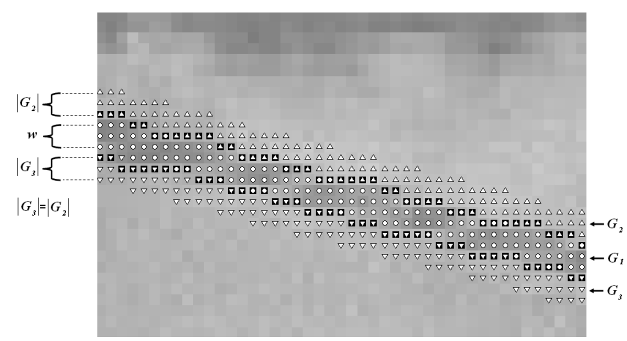


Figure 3. Profile estimation for the wire shown in Figure 2-b. The black squares represent edge pixels. The Gaussian model profile consists of 3 regions: the wire (shown in circles), and two other regions corresponding to the wire background (shown with straight and inverted triangles respectively).

There are three advantages to using an iterative algorithm as compared to the single line-based estimation in [4]. First, it does not require the width of the wires to be known a-priori. Second, it produces robust region estimates by using all available pixels from the wire and surrounding regions (instead of the 3 single lines modeling the wire and both surrounding regions). Third, it provides an estimate of the width of the wire.

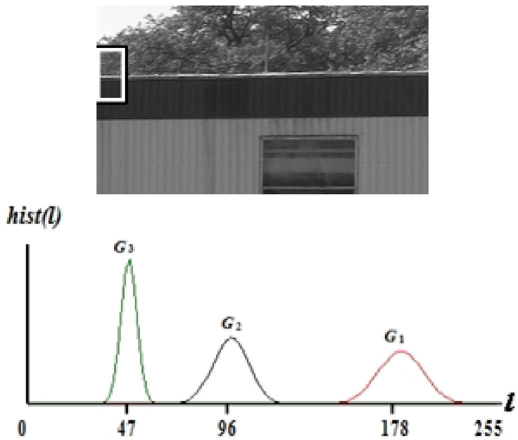


Figure 4. (a) Rooftop that can be easily mistaken for a wire (thin bright line at the top of the roof). (b) Profile corresponding to the boxed area shown in (a). The large separation of the means indicates the profile at both sides of the edge is quite

C. Post Processing

Post processing comprises of two stages: Weight Thresholding and Scene Correction. Weight thresholding is described in detail in [4]. The threshold is set to a percentage, determined as a function of the distance from the corner, of the maximum possible weight for a wire. This is based on the construct that wires that are further away from the corners have a greater length and hence a higher number of pixels than the wires that are closer to the corners. This approach helps by preventing faulty removal of short lines. The threshold for horizontal and vertical wires, however, is independent of distance, for obvious reasons. Candidates that do not clear the threshold are removed.

Next, we introduce Scene Correction, a procedure to eliminate lines that are inconsistent with global statistics, if such a statistic exists, and is useful in applications looking for lines that follow a group pattern, like power lines or barbwires. In the case of power lines, lines that do not satisfy the following equation are discarded:

$$|\,median\{m_s\} - m_i\,| \le \tan(\theta) \tag{4}$$

Where m_i is the slope of the wire candidate, ms is the set of slopes corresponding to all wire candidates that "survived" the weight thresholding, and θ is the angle of deviation required for a wire to be considered correctly detected. θ can be varied to account for factors such as wind, etc. Scene correction will not have any effect on scenes with only one or two wires.

III. RESULTS

The wire detection dataset [2] corresponds to 86 low-quality videos, with a total of 10160 instances of ground truth wires spanning in 5576 frames. The data is quite challenging, offering a great variety of scenes, cluttered backgrounds, lighting, and weather conditions (i.e. including light to moderate rain, and mist). Some videos were provided by search and rescue groups using helicopters and unmanned aerial vehicles. Ground truth lines were manually drawn, with a single straight line approximating each wire.

In order to establish a reasonable performance metric, and to present an empirical validation for the algorithm, we consider a wire to be correctly detected if found within an angle of 10◦ and y-intercept within 20 pixels of the ground truth [13].

Fitting accuracy was measured using 3942 connected components from the training data, by the coefficient of determination [14] expressed as a percentage. The fitting after noise reduction proved to be over 4% better in the average than fitting without noise reduction. The line weight threshold is the control variable for the proposed algorithm's Receiver Operating Characteristic (ROC) shown in Figure 5. In our analysis, performance per video shows statistical error of 0.82 for detection and false positive error of 1.01 with 95% confidence. There is a detection improvement of 17.7% with respect to the area under the ROC when compared to the baseline. Figure 6 depicts sample detection results.

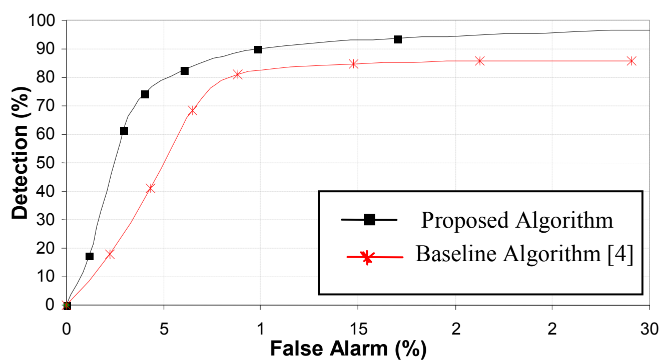


Figure 5. Wire detection performance ROC, comparing previously published algorithms and proposed method.

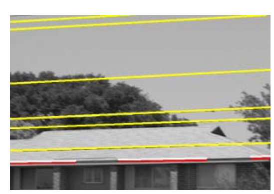


Figure 6. Sample result in an image with a complex background. Lines indicate detected wires and dashed lines indicate false positives.

IV. CONCLUSION

The presented algorithm outperforms all previous detection methods in our dataset. The wire detector uses an explicit line model, and combines size and shape filters, perceptual grouping, and geometrical principles to search for linear patterns. The algorithm uses edge detection combined with size and eccentricity filters to create a feature map. Edge noise is reduced using chain code histograms for connected components, and lines are fitted for each de-noised edge. An explicit line model using a Gaussian is used to approximate the wire's intensity profile surface. The profile analysis can be used to effectively discriminate between wires and other visually similar linear patterns such as rooftops, building textures, etc. Also, we introduce a scene correction technique useful in applications looking for lines that follow a group pattern, such as power lines or barbwires.

Future work should be focused on exploiting spatio-temporal relationships of wires across the videos and automatically detecting trees in backgrounds, which account for over 64% of the false alarms.

REFERENCES

[1] R. Kasturi, O. Camps, Y. Huang, A. Narasimhamurthy, N. Pande. Wire Detection Algorithms for Navigation, NASA Technical report, 2002.

[2] J. Candamo, R. Kasturi, D. Goldgof, S. Sarkar. Detection of Thin Lines using Low-Quality Video from Low-Altitude Aircraft in Urban Settings, IEEE Transactions on Aerospace and Electronic Systems, 45(3):937-949, July 2009.

[3] J. Candamo, R. Kasturi, D. Goldgof, S. Sarkar. Vision based on-board collision avoidance system for aircraft navigation, Proceedings of SPIE, 6230, 2006.

[4] J. Candamo, D. Goldgof. Wire detection in low-altitude, urban, and low-quality video frames, Nineteenth International Conference on Pattern Recognition, 2008.

[5] J. Lu, J. Ren, Y. Lu, X. Yuan, C. Wang. A Modified Canny Algorithm for Detecting Sky-Sea Line in Infrared Images, Sixth International Conference on Intelligent Systems Design and Applications, 2:289-294, October 2006.

[6] F. Arrebola, A. Bandera, P. Camacho, F. Sandoval. Corner detection by local histograms of contour chain code, Electronics Letters, 33(21):1769-1771, 1997.

[7] F. Sadjadi. Object recognition using coding schemes, Optical Engineering, 31:2580-2583, 1992.

[8] P. Bhowmick, B.B. Bhattacharya. Fast Polygonal Approximation of Digital Curves Using Relaxed Straightness Properties, IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(9):1590-1602, September 2007.

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

[10] C. Carson, S. Belongie, H. Greenspan, J. Malik. Blobworld: image segmentation using expectation maximization and its application to image querying, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(8):1026-1038, August 2002.

[11] D. Zhou, H. Zhang. Modified GMM background modeling and optical flow for detection of moving objects, IEEE International Conference on Systems, Man and Cybernetics, 3:2224-2229, 2005.

[12] S.Q. Han, Z.L. Pei, O.F. Wang, X.H. Shi. A Characteristic Sequences and Normalized Euclidean Distance Based Method for RNA Secondary Structures Comparison, IEEE International Conference on Bioinformatics and Biomedical Engineering, 222-225, 2007.

[13] R.H. Cuijpers, A.M.L. Kappers, J.J. Koenderink. Visual perception of collinearity, Perception and Psychophysics, 64(3): 392-404, 2002.

[14] N.J.D. Nagelkerke. A note on a general definition of the coefficient of determination, Biometrika, 78(3):691-692, 1991.