Wire Detection in Low-Altitude, Urban, and Low-Quality Video Frames

Joshua Candamo and Dmitry Goldgof
University of South Florida
{candamo, goldgof}@cse.usf.edu

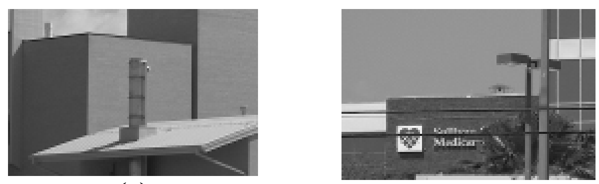
Abstract

We introduce a novel wire detection algorithm for use in low altitude urban aircraft reconnaissance. A line profile model is described and effectively used to discriminate wires from other linear patterns commonly found in urban scenes. The algorithm is able to cope with highly cluttered backgrounds, moderate rain and mist, and with no stabilization of camera. The studied domain is of particular interest to urban search and rescue and military reconnaissance operations. The algorithm's receiver operating characteristic curve is shown, based on a multi-site dataset with 10160 wires spanning in 5576 frames. Encouraging results show up to 37% detection improvement over a previously published baseline algorithm for comparable false alarm rates.

1. Introduction

In search and rescue and military operations, there are significant advantages of using small aircraft rather than humans in reconnaissance missions, especially in urban settings. Researchers have shown increasing interest in this domain [1][2][3]. However, small aircrafts have significant limitations of payload and electrical power that can be supported, which impose severe restrictions on the hardware that can be used on-board. Computer vision offers viable technology, due to the low size, weight, and power consumption of small vision-based sensors. Low altitude urban reconnaissance poses great challenges that are not typical of high altitude flight. Kurdila et al. [4] discussed some of the challenges faced in visual navigation of small autonomous aircraft in urban environments. As aircrafts become smaller, and more maneuverable, techniques for vision-navigation need to be adjusted in order to be used effectively in urban settings.

This paper focuses on images typical of urban reconnaissance missions. In this domain, small aircraft frequently fly within 15 meters of obstacles. Objects commonly found in urban areas are buildings, antennas, trees, poles, and wires as shown in Fig. 1. At close distance, wires appear in general as straight line segments. Gandhi et al. [5] and Kasturi et al. [6] have proposed viable wire detection strategies for flights at a distance from wires, in which wires appear as catenaries rather than straight lines. In [7] we described in detail the difficulties of dealing with low altitude urban scenery, in contrast to such domains.



(a) (b)

Figure 1. Typical urban scenes. (a) Building edges that appear similar to wires, and (b) wires in front of buildings

1.1. Wire model

In this work, we assume that wires encountered in urban reconnaissance flights are approximately horizontal (less than 45^0 with respect to the *x*-axis), and the typical wire thickness is between 1 to 3 pixels. Lines are described with the slope-intercept form:

$$y = mx + c \qquad \textbf{(1)}$$

A wire is described by the 3-tuple (w,m,c), where w (weight) is the number of pixels, m is the slope of the straight line that approximates the wire, and c is the y-intercept along the left edge of the image. Two wires are considered equal for detection purposes if their y-intercept is within 20 pixels and the angle θ with respect of the horizontal, i.e. $\theta=atan(m)$, is within 10^0.

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

2. Wire detection algorithm

The algorithm, as shown in Fig. 2, uses edge detection combined with morphological filters to create a feature map. Line fitting is performed for all connected components. Clutter effects are reduced by discarding lines based on a fitting error measure. Lines are kept if there is sufficient additional line support in the original edge map. Linear patterns commonly found in urban scenes are discarded using a novel profile analysis technique discussed in Section 2.2. Parameters used in the algorithm will be mentioned in each section, and also summarized in Section 4, including a brief description of the parameter learning strategy.

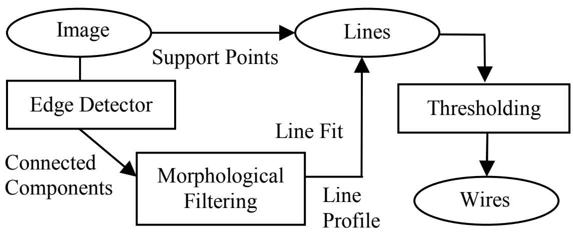
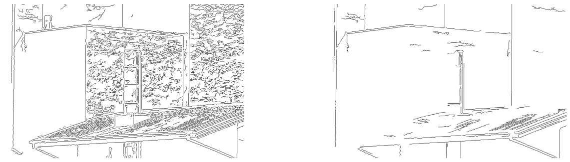


Figure 2. Algorithm flowchart

2.1. Edge detection and pre-processing

A significant challenge in urban scenery is to deal with the large variation of visual complexity across the domain. For this reason, we have previously trained Canny edge detector in [8] using a visual complexity measure of the scene. Clearly, this would require a robust representative training dataset. Instead, in this algorithm Canny edge detector uses adaptive thresholds as described in [9] and constant width of the Gaussian mask (σ=1), in order to minimize the effect of a potentially weak training dataset.

Starting with an edge map is consistent with the premise "*most of all wire pixels are edges, but most edges are not wire pixels*." However, we want a robust feature map that maximizes the number of true wire pixels, while minimizing the amount of information to be further processed. For this purpose, two morphological filters are applied: size, and eccentricity. Fig. 3, shows the effect of pre-processing on the original edge map. First, 8-connected component labeling is performed. Then, for each component the area and eccentricity is computed. A size filter based on the area is applied to eliminate small components, and eccentricity is used to eliminate components that do not look like lines. For simplicity, we will refer to edges that "survive" the filtering process as feature map.



(a) **(b)**

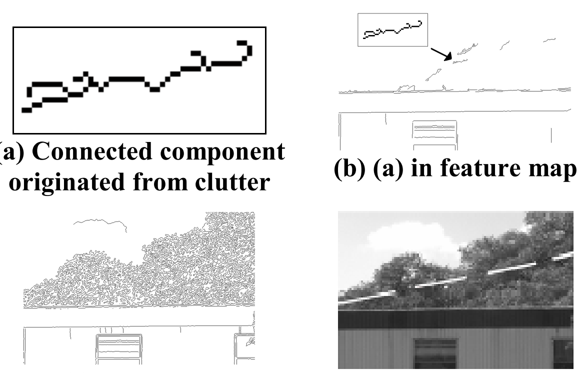
Figure 3. (a) The output of Canny edge detector and (b) morphological filtering on Fig. 1(a)

2.2. Line fitting and profiling

For each component, the relationship between pixel coordinates, i.e. all *(x,y)* pairs that form a component, is analyzed by fitting a straight line through regression. The best-fit line is the one which minimizes the squared error:

$$\sum_{(x,y)} (y-(mx+c))^2 \qquad \textbf{(2)}$$

Clearly, pre-processing is not perfect; thus, often components corresponding to clutter will appear on the feature map. Consequently, not all lines from components after pre-processing will be ultimately considered as wires. A fitting error threshold is introduced in order to minimize the effect of clutter. Additionally, there is a verification process that checks for support points along the fitted line on the edge map. The number of support pixels found is added to the initial component weight. For example, the connected component shown in Fig. 4(a) is not removed by the pre-processing but does not correspond to a true wire. In fact, the connected component is originated from the cluttered trees in the feature map (b), and has enough edge support in (c) to become a false positive (d) if not discarded by the fitting threshold.



(a) Connected component originated from clutter **(b) (a) in feature map**

(c) Edge Map **(d) Roof top and texture that look like wires**

Figure 4. (a) Connected component corresponding to tree clutter. (b) Feature map. (c) Edge map. (d) Corresponding best-fit line in the feature map.

Next, we introduce a line profile analysis that allows discrimination of wires from common urban linear patters. Line profile is defined by the pixel intensities at both sides of the line. As depicted in Fig. 5, the profile (using Steger's Asymmetric-bar model [11]) of the solid line is described by the two dashed lines. It is necessary to account for wire thickness as well as hardware and edge detector noise. Thus, the separation h is defined to guarantee the profile above and below the line will lie on the background, and not on top of the original line.

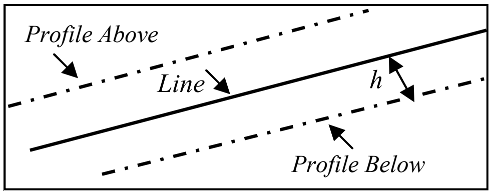


Figure 5. Line profile model

A wire in urban scenes will have a more "symmetric profile" than other thin lines common to this domain. A line profile maximum tolerance is defined in order to discard those lines with less "symmetric profiles". Fig. 4(d) illustrate how the rooftop or the texture of the building could be mistaken for a wire, even by humans.

2.3. Thresholding

Thresholds are defined per line, and based on the computed weight of each connected component plus the edge support. Let w_{max} be the maximum weight that a line would have if drawn in the current image. The threshold is the percentage of points of w_{max} that a line has to accumulate. For clarity, we will refer to this threshold as T_w later in this paper. This thresholding method has clear advantages compared to global thresholding approaches [6][7], which are prone to reject true positive short lines. In Fig. 6(a), a global threshold will most likely have difficulties correctly detecting the short wires; because other wires are several times longer. For completeness, Fig. 6(a-f) show detection results, and demonstrates how a T_w=0.6 is able to cope with different scene complexities. The bottom dashed white line in (f) is a false alarm caused by motion blur from a moving vehicle.

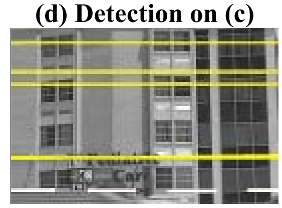
(a) Low clutter scene

(b) Detection on (a)

(c) Medium clutter scene

(d) Detection on (c)

(e) High clutter scene



(f) Detection on (e)

Figure 6. Sample detection results

3. Dataset

Urban reconnaissance aircraft data is scarce and hard to obtain, due to the significant risk involved with flying near obstacles within 15 meters. The dataset used is a multi-site dataset fully described in [7]. 1436 images with no wires (about 50% of data with no wires) were chosen randomly to be part of the test dataset and compute results. 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). The data was collected without using any stabilization hardware or software. Some videos were provided by search and rescue groups using helicopters and unmanned aerial vehicles. Most of the hardware used is unknown and used various types of compression. Ground truth lines were manually drawn, with a single straight line approximating each wire. There are a total of 10160 wires spanning in 5576 images in the dataset, with 1369 images selected randomly for training.

4. Parameters and results

Canny edge detector uses 3 parameters, which are selected dynamically as discussed in Section 2.1. There are 2 pre-processing parameters: size and eccentricity thresholds. Consistently with our previous work we use 30px and 0.99 respectively. The line profile tolerance, the line fitting error, and the profile separation were trained together using randomized hill climbing. In this paper the trained values are 32, 0.35, and 5 respectively. The weight threshold T_w is the control variable for the receiver operating curve (ROC) shown in Fig. 7. In ROC "baseline" refers to our previous work. For complete technical details the reader is referred to [7]. In a nutshell, the baseline is a wire detection algorithm that uses edge detection, morphological filtering, a windowed Hough Transform, and motion information to track wires in video. The present approach not only represents significant accuracy improvement but also requires fewer parameters than the baseline.

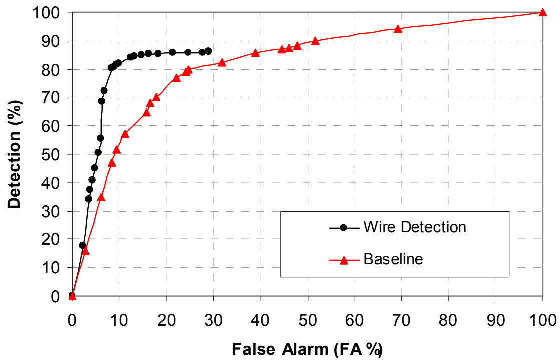


Figure 7. Wire detection ROC in dataset with wires

The ROC depicts the wire detection performance on part of the dataset containing images with wires. The false alarm (FA) rate is given by the number of false positives divided by the number of ground truth wires. Fig. 8, shows the number of FA per frame using 1436 test images with no wires.

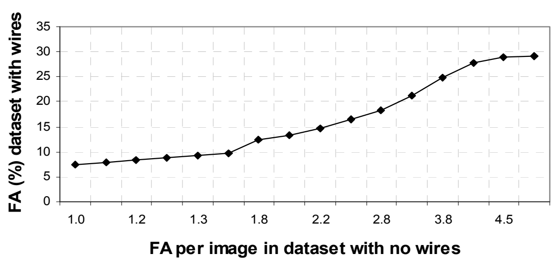


Figure 8. FA per image in dataset with no wires

The ROC detection performance shows statistical error of 0.05 and false alarm error of 0.15 with 95% confidence. Comparing the same FA rate, produced by the baseline and the proposed wire detection strategy, the average detection improvement is 20%, with a maximum improvement of 37% detection at 7% FA. Performance saturates when using all connected components, with 86% detection and 29% FA.

5. Conclusions

This work describes a novel wire detection algorithm for low altitude urban reconnaissance flight. The technique is capable of providing robust performance as shown by ROC analysis. The system is intended to support air vehicles operating in highly cluttered urban scenes, moderate weather conditions, and noise produced by low quality hardware.

The algorithm uses edge detection combined with morphological filters to create a feature map. Lines are fitted for each connected component. The fitting error is used to discard components generated by clutter. Lines are kept based on additional line support found on the original edge map. Linear patterns commonly found in urban scenes, such as rooftops, building textures, painting, and shadows can be often mistaken with wires. A novel line profile analysis is introduced, and effectively used to discriminate between wires and other similar linear patterns. A large and challenging multi-site low-quality dataset is used, with 10160 wire objects in 5576 images. Performance is determined by comparing the algorithm output with manual ground truth. Results compared to previous work, show that for false alarm rates less than 30%, the detection improvement is up to 37%.

6. Acknowledgements

This work was supported in part by the Center of Urban Transportation and Research (CUTR) at University of South Florida under Grant BS123456.

7. References

[1] L. Muratet, S. Doncieux, Y. Briere, J. Meyer, "A contribution to vision-based autonomous helicopter flight in urban environments," *Robotics and Autonomous Systems*, vol. 50, pp. 195-209, 2005.

[2] L. Singh, J. Fuller, "Trajectory generation for a UAV in urban terrain, using nonlinear MPC," *Proc. American Control Conf.*, pp. 2301–2308, 2001.

[3] C. Demonceaux, P. Vasseur, C. Pegard, "UAV Attitude Computation by Omnidirectional Vision in Urban Environment," *Rob. Automat.*, pp. 2017-2022, 2007.

[4] A. Kurdila, M. Nechyba, R. Prazenica, W. Dahmen, P. Binev, R. DeVore, R. Sharpley, "Vision-based control of micro-air-vehicles," *Conf. Decision and Control*, 2004.

[5] T. Gandhi, M.T. Yang, R. Kasturi, O. Camps, L. Coraor, J. McCandless, "Detection of obstacles in the flight path of an aircraft," *Aerospace and Electronic Systems*, vol. 39, no. 1, pp.176-191, 2003.

[6] R. Kasturi, O. Camps, Y. Huang, A. Narasimhamurthy, N. Pande, "Wire Detection Algorithms for Navigation," *NASA Tech Report*, 2002.

[7] J. Candamo, R. Kasturi, D. Goldgof, S. Sarkar, "Detection of thin lines using low quality video from low altitude aircraft in urban settings," *To appear Aerospace and Elec. Syst.*

[8] J. Candamo, R. Kasturi, D. Goldgof, S. Sarkar, "Vision-based on-board collision avoidance system for aircraft navigation," *Proc. of SPIE*, vol. 6230, 2006.

[9] J. Lu, J. Ren, Y. Lu, X. Yuan, C. Wang, "A Modified Canny Algorithm for Detecting Sky-Sea Line in Infrared Images," *Syst. Design App.*, vol. 2, pp. 289–294, 2006.

[11] C. Steger, "An unbiased detector of curvilinear structures," *Pattern Analysis and Machine Intelligence*, vol. 20, no. 2, pp. 113-125, 1998.