

Automatic Determination of Runway Edges in Poor Visibility Conditions

Sri Satya Vardhan Gogineni, Zia-ur Rahman

Old Dominion University, Electrical and Computer Engineering Department,
Norfolk, Virginia 23529

ABSTRACT

The automatic detection of runway hazards from a moving platform under poor visibility conditions is a multi-faceted problem. The general approach that we use relies on looking at several frames of the video imagery to determine the presence of objects. Since the platform is in motion during the acquisition of these frames, the first step in the process is to correct for platform motion. Extracting the scene structure from the frames is our next goal. To rectify, enhance the details and to remove fog we perform multiscale retinex followed by edge detection on the imagery. In this paper, we concentrate on the automatic determination of runway boundaries from the rectified, enhanced, and edge-detected imagery. We will examine the performance of edge-detection algorithms for images that have poor contrast, and quantify their efficacy as runway edge detectors. Additionally, we will define qualitative criteria to determine the best edge output image. Finally, we will find an optimizing parameter for the detector that would help us to automate the detection of objects on the runway and thus the whole process of hazard detection.

Keywords: Runway detection, edge detection, poor visibility, Multi scale retinex

1. INTRODUCTION

One of the current research areas of interest to NASA's Aviation Safety Program is the automatic detection of external hazards on a runway under poor visibility conditions. The presence of unexpected objects on the runway can be a significant problem when landing an aircraft, especially in poor visibility conditions where the pilot's vision is considerably impaired. We define "poor visibility conditions" to include fog, smoke, haze and dim lighting. Since the pilot has to multi-task various operations while landing the aircraft, we do not want to make the process even more burdensome by adding the task of interpreting data from enhanced imagery to detect presence of potential hazards. Hence we propose to detect the hazards automatically.

This system is based on the Multi-scale Retinex image-enhancement method (MSR) (Section 2.1).¹⁻³ In addition to their exceptional image-enhancement capabilities, these methods show promise as a computational platform for higher level visual information processing. This is, in part, due to the intrinsic ability of these methods to produce "canonical" imagery by reducing the variability due to changing lighting conditions and exposure errors. The canonical image is a high quality, stable, visual rendition of an arbitrary scene,⁴⁻⁶ that forms the basis for sophisticated, vision-based, automation technology for future NASA aerospace missions. It provides robust pattern constancy, and hence facilitates higher level image processing that is needed to perform intelligent imagery analysis that can be used for external hazard detection and runway incursions. Image-enhancement and intelligent analysis are used to automate the process of hazard detection from sensor imagery to augment the pilot's situational awareness. A major part of this effort is to segment the runway from the rest of the image. To this end, we perform edge-detection on the image to find all the edges, and then analyze the edge-image to segment the runway from the rest. In this paper, we present a new parametric edge-detection method that allows robust detection of edges in poor visibility conditions.

Contact: SSVG: sgogi002@odu.edu; ZR: zrahman@odu.edu.

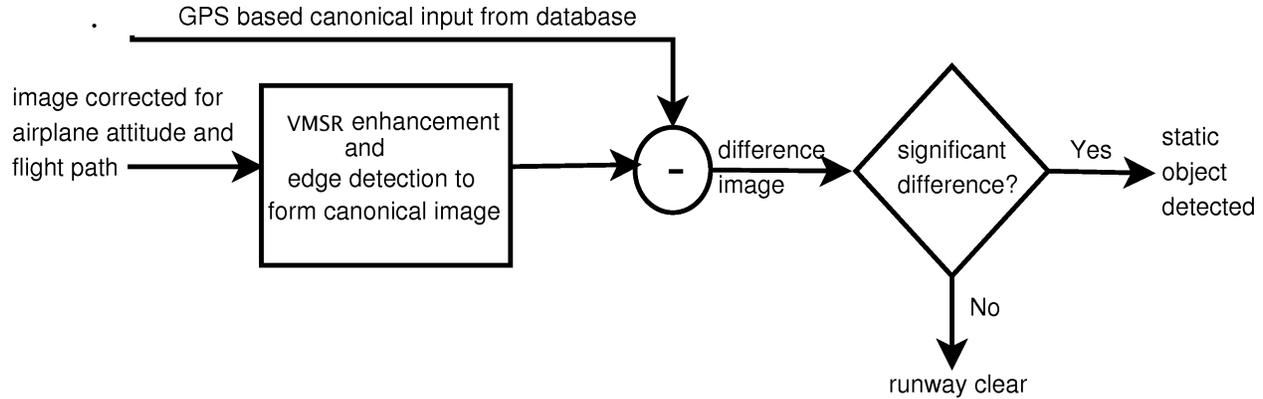


Figure 1: Illustration of the overall concept for detecting static runway hazards.

2. BACKGROUND

While making the pilot interpret imagery to detect hazards is a viable approach, it represents yet another task, among myriad tasks, that the pilot has to handle during the landing process. We propose a new technique that would automate the detection of objects on the runway and determine if they represent a hazard to landing. This technique combines image enhancement with smart edge analysis to detect objects on the runway. There are two main reasons for using image enhancement before edge extraction: the image enhancement operator

1. provides an illumination independent, and hence canonical, representation of the scene;
2. has the inherent capacity to enhance small signal differences and can, thus, bring out the detail in imagery even in poor visibility conditions.

We have, previously, used this method successfully to provide better-than-observer visibility under conditions of haze, fog, light clouds, and rain (Figure 4).^{6,7} The proposed approach for detecting static objects is shown in Figure 1. The VMSR enhanced imagery is further processed with an edge-detection operator to obtain an edge-only representation of the image—see Section 2.2 for details. This representation provides additional advantages:

1. The edge-only image is, in a sense, more canonical than the SRVR enhanced imagery. This is because the variations in shading that may still be present in VMSR processed imagery are virtually eliminated in the edge-only image.
2. The edge-only image is also more immune to variations in imagery due to sensor noise since insignificant edges such as those attributed solely to random noise are eliminated during the generation of the edge-only image.
3. The edge-only image requires less space to store in a database and also requires less time for comparison and retrieval.

To detect the presence of objects on the runway, the existing canonical edge representations of the runway* are compared with enhanced edge representations of the geometrically-corrected approach imagery as shown in Figure 1. The effect of applying the enhancement method is to make the imagery of the runway (almost) independent of the time of day, and the atmospheric conditions—whether it is sunny or cloudy—as discussed in Section 2.1. The impact of using an edge-only representation of the enhanced imagery is discussed above.

*Canonical representations of landing strips for different airports are retrieved from a database of canonical imagery.

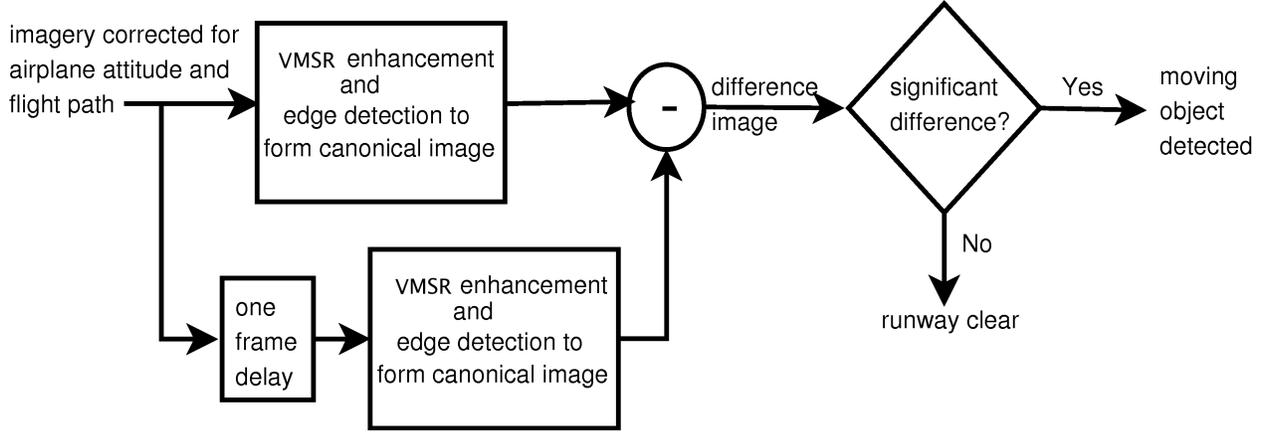


Figure 2: Illustration of the overall concept for detecting moving runway hazards.

Differences in the stored canonical representation and the computed representation show the presence of static objects in the scene.

Moving objects in the imagery are, in a sense, easier to detect than static objects. One does not require a stored canonical representation of the runway in order to detect moving objects which can be detected simply from inter-frame analysis of the imagery. Figure 2 shows how inter-frame differencing is used to detect moving objects. Two consecutive frames in the image stream are SRVR enhanced and represented using the edge-only form. The frames are co-registered on a common grid to correct for the motion of the airplane. This can be done by using the ancillary information from the GPS, and the airplane stabilization systems that record the attitude of the aircraft. Once the frames have been registered to a common grid, a simple differencing operation can be used to detect the presence of objects. Three or more frames can be used to determine if the objects are moving toward or away from the aircraft.

Once an object has been detected, its size can be determined from its two-dimensional projection on the image plane and the knowledge of geometry of the flight path and the orientation of the aircraft. The size of the object then determines whether it is a potential hazard. The eventual goal is to have this process perform in real-time and to flash potential hazards on whichever display the pilot is using. This can be used by the pilot either as a cue to take evasive action or as a cue to take a closer look at the indicated area thus making the detection of hazards less strenuous. The near-term goal is to develop the algorithms that can be used to (a) detect the presence of objects on the flight approach under various visibility and weather conditions, and (b) to determine the size of the object based upon the geometry of the flight path and the orientation of the aircraft.

2.1. Image enhancement

A fundamental concern in the development of resilient, vision-based, automation technology is the impact of wide-ranging extraneous lighting and exposure variations on the acquired imagery. This concern can be considerably ameliorated by the application of the (MSR) image-enhancement algorithm.^{1-3,8,9} The MSR is a non-linear, context-dependent enhancement algorithm that provides color-constancy, dynamic range compression and sharpening:

$$R_i(x_1, x_2) = \sum_{k=0}^{\kappa} w_k (\log (I_i(x_1, x_2)) - \log (I_i(x_1, x_2) * F_k(x_1, x_2))), \quad i = 1, \dots, N, \quad (1)$$

where I_i is the i^{th} spectral band of the N -band input image, R_i is the corresponding Retinex output, ‘*’ represents the (circular) convolution operator, F is a (Gaussian) surround function, and κ is the number of the



Figure 3. The image enhancement operator successfully compensates for changing illumination conditions and exposure errors. The camera aperture, shutter speed, and ISO setting were constant over this sequence.

scales. The Gaussian surround function is given by:

$$F_k(x_1, x_2) = a_k G_k(x_1, x_2) \quad (2)$$

$$G_k(x_1, x_2) = \exp(-(x_1^2 + x_2^2)/\sigma_k^2) \quad (3)$$

$$a_k = \sum_{x_1, x_2} G_k(x_1, x_2), \quad (4)$$

The σ_k are scale parameters that control the performance of the SSR: small σ_k lead to SSR outputs that contain the fine features in the image at the cost of color, and large σ_k lead to outputs that contain color information, but not fine detail.^{2,3}

In order to extract consistent scene structure from any image under widely varying scene and sensor conditions, one has to think in terms of transforming the image into a “canonical” representation that effectively eliminates such undesirable variability. The MSR has proven to be a powerful tool for doing just this. Because of its dynamic range compression and illumination independence properties, the MSR provides consistent rendering for imagery from highly diverse scene and sensor conditions. To expand the performance envelope of the MSR to handle narrow dynamic range images encountered in turbid imaging conditions such as fog, smoke, and haze, dim lighting conditions, or significant under- or over-exposures, we have developed a “smart” framework of visual quality measurements and enhancement controls that we call the Visual Servo^{7, 10, 11} (VS). The VS assesses the quality of the image in terms of brightness, contrast and sharpness, and controls the strength of the MSR enhancement. Figure 3 shows a sequence of images and its enhancement under visibility conditions that range from acceptable to unacceptable. The enhanced image provides useful information in every case regardless of the caliber of the original data.

Additionally, the enhancement can provide better-than-observer performance in many cases, especially when the obscuration is due to fog, rain, or light clouds in otherwise good illumination. Figure 4 shows the performance of the image enhancement operator on imagery acquired under hazy and cloudy imaging conditions. The enhancements were compared with the recollections of the observer about the extent to which he could discern features with the naked eye, or through the camera, at the time the image was acquired. In each case, according to the observer, the enhanced imagery provided more information than could be discerned either through the

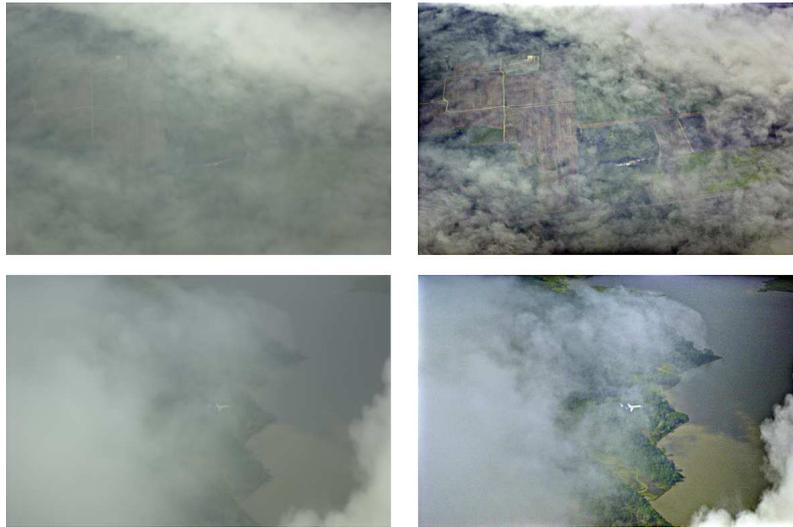


Figure 4. Images acquired with a Nikon Digital D1 camera during NASA Langley Research Center's FORESITE test flights. The enhancements provide better-than-observer visibility.

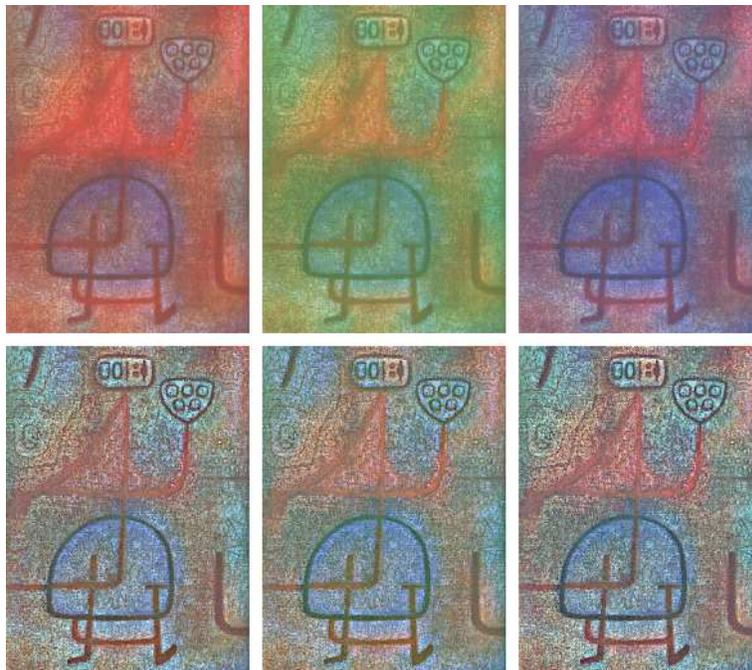


Figure 5. The impact of illuminant change was simulated by red, blue, and green shifting an image (top row). The MSR outputs are *almost* perfectly color constant (bottom row).

view-finder of the camera or with the naked eye. Although this is not a rigorous scientific test, it does justify laying the groundwork for further testing and analysis.

The image enhancement process also provides illumination independence, i.e., the output of the algorithm is (almost) independent of the type, or level, of illumination under which the image was acquired. This is especially critical for automatic classification and detection algorithms that rely on comparing imagery of the same scene at different times. The ability of the algorithm to produce images that are independent of the change in illumination conditions due to changing sun angle and atmospheric conditions considerably simplifies the automation process for detection and classification. Figure 5 shows an example illustrating the illumination independent output produced by the algorithm.

The fundamental problems relating to enhancement of still imagery have been addressed in Jobson, et al.^{1,2} and Rahman et al.³ Additionally, issues relating to enhancement of imagery under poor visibility conditions have been addressed in Jobson et al.^{4,7} and Woodell et al.^{6,11}

2.2. Edge Detection

Once the images have been enhanced and rectified, we perform edge-detection on the imagery to form a canonical representation of the imagery stream. As stated earlier, there are two major reasons for performing edge-detection:

1. edge-detection provides a certain degree of noise immunity since a considerable amount of noise can be suppressed by the proper edge-detection mechanism;
2. edge-detection reduces the overall complexity of the problem by reducing the degrees-of-freedom present in the original problem.

The Canny edge-detection operator¹² is considered by many to be the best edge operator available. It stands out from others in good localization of edges, and strong detection capabilities. It uses a technique called “non-maximum suppression” to take care of spurious edge-pixels, and “hysteresis”—which uses an upper and lower threshold—to determine the strength of the edges. A commonly used operator for edge-detection is the Sobel operator¹³ which uses a 3×3 kernel with a higher weight assigned to the center pixel to bring out the finer edges in the image. We initially used these operators are in our hazard detection procedure.

The main drawback of these, and other, edge-detection algorithms is that the strength of the edges in the output image cannot be controlled easily. While some of the algorithms are parameterized so that the strength of the edges can be adjusted, this is not an automatic process. We have developed an algorithm based on the edge-detection algorithm by Zhang et al.¹⁴ which uses an integer logarithm ratio of pixels. This concept of ratio provides better noise rejection than the above algorithms that use the difference operator. The integer logarithm ratio algorithm is implemented in the following steps:

Step 1: Smooth the image with a Gaussian filter to remove high frequency noise.

Step 2: Compute the natural logarithm of every pixel in the image:

$$\mathcal{L}(i, j) = \log[I(i, j)] \tag{5}$$

Step 3: Detect variations in x and y directions by taking difference of log values of the consecutive pixels.

$$\begin{aligned} L_x(i, j) &= |[\alpha \mathcal{L}(i, j)] - [(\alpha \mathcal{L}(i - 1, j))]| \\ L_y(i, j) &= |[\alpha \mathcal{L}(i, j)] - [(\alpha \mathcal{L}(i, j - 1))]|, \end{aligned} \tag{6}$$

where α is the parameter which alters the strength of the edges in the image.



Figure 6. (top-left) Original; (top-right) Sobel edge output; (bottom-left) Canny output Threshold ratio 3:1; (bottom-right) Canny output Threshold ratio 2:1

Step 4: Define a pixel as an edge pixel based on:

$$(L_x(i, j) = 0 \wedge L_x(i + 1, j) \neq 0) \vee (L_x(i, j) \neq 0 \wedge L_x(i + 1, j) = 0), \quad (7)$$

where \wedge and \vee represent the logical AND and OR operators, respectively. If we have a blob of pixels having same gray level value, then their log values will be same and hence $L_x(i, j)$ and $L_x(i + 1, j)$ both will be zeros. When the above condition specified in Equation 7 is not met the pixels will be suppressed in the edge output. Any variations in the neighboring pixels will be picked up as edges depending up the strength of the variations and the parameter α defined in step 3. Neighboring pixels with large variations in the gray level values constitute stronger edges compared to those with small variations.

Step 5: Repeat steps 3 and 4 in other required directions.

This algorithm was applied to the VMSR processed imagery but did not provide desired results: the edges that were produced were too thick so did not provide good localization information. A closer analysis revealed that the inherent fog or smoke in the image provided the necessary blurring to the image. Hence we got rid of the initial smoothing of the image. One way to bring out edge-information in low contrast, high brightness images is to examine the negative of the image. Our experiments on running the above algorithm on the negative of the image instead of the image directly proved to be very successful as can be seen by looking at the examples shown in Figures 7, 8 and 9.

As we stated earlier, the strength of the edges can be controlled by varying the value of α in Equation 6. Higher values of α classify more pixels as edge-pixels and produce thicker edges, while lower values of α have a

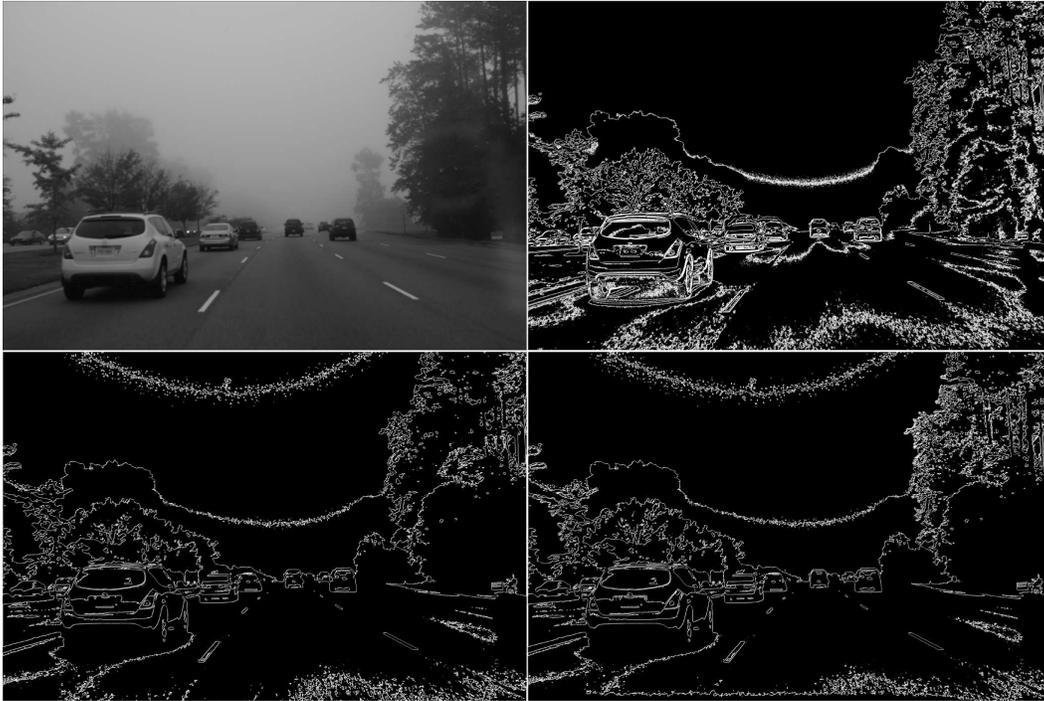


Figure 7. Example 1: (top-left) Original; (top-right) Integer logarithm edge output; (bottom-left) Using the negative of original Image ; (bottom-right) MSR on the negative of the image.



Figure 8. Example 2: (top-left) Original; (top-right) Integer logarithm edge output; (bottom-left) Using the negative of original Image ; (bottom-right) MSR on the negative of the image.

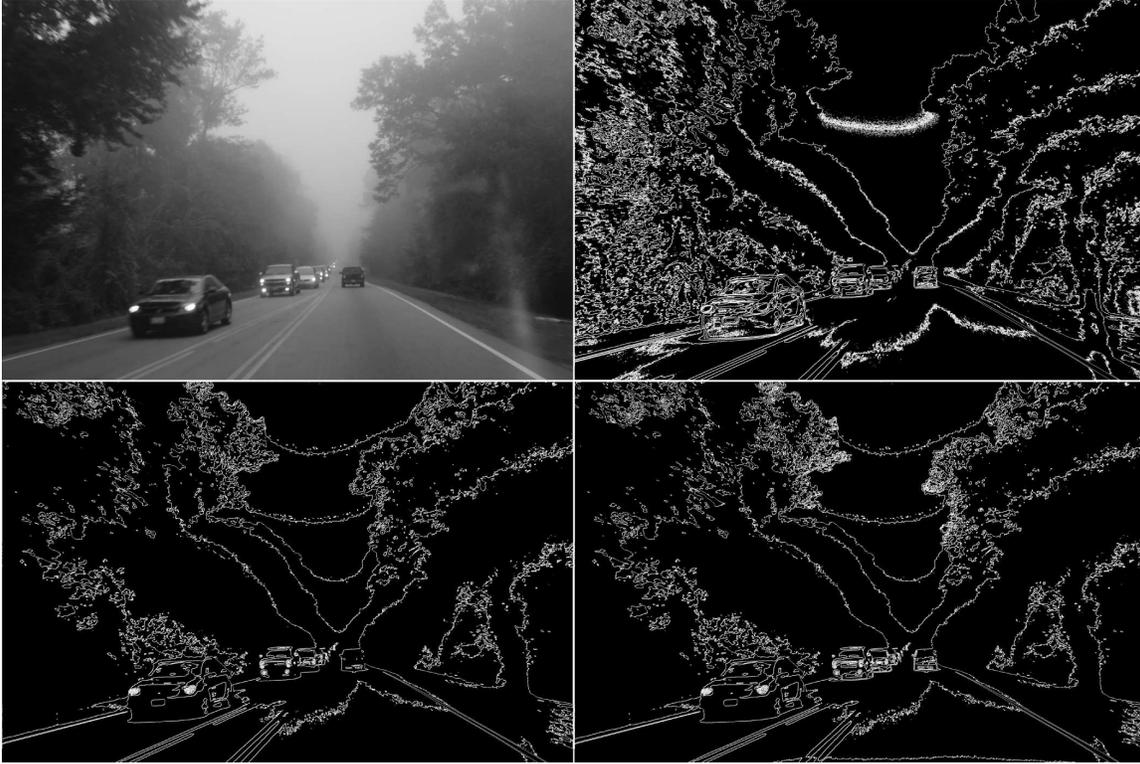


Figure 9. Example 3: (top-left) Original; (top-right) Integer logarithm edge output; (bottom-left) Using the negative of original Image ; (bottom-right) MSR on the negative of the image.

tendency to miss fine edge-detail though they preserve good localization information. We are currently examining methodology to automatically select appropriate values of α based upon image characteristics. Current results have been produced by manually selecting an “optimal” value of α . Figure 10 shows the impact of varying α on the edge-image. The results suggest that the optimal value of α probably lies between 5 and 8 for the type of imagery that we examined for detecting hazards in poor visibility conditions.

3. CONCLUSIONS

Various popular edge detection algorithms have been simulated and tested for hazard detection on runways under poor visibility conditions. Since we did not have access to images of runways acquired under poor visibility conditions, we used images of roads with fairly heavy traffic to test the algorithms. The ability to detect cars in poor visibility conditions gives us confidence that we should be able to detect hazards on a runway. The work presented here is in early stages and a completely automated hazard detection system has not yet been implemented. We are experimenting with the different operators needed to provide an automated system before integrating the whole.

We have presented a new edge-detection methodology that incorporates an integer logarithm algorithm produces with smart image enhancement to produce perfect edge detected outputs for poor visibility images. Instead of preprocessing the images we tried to make use of the inherent blurring available in the image resulting from poor visibility. An α parameter has been defined to automate edge detection in accordance to the strength of the edges required. Various combinations of image variants have been tried for best results. The outputs obtained using negative of an image followed by MSR processing gave favorable results. With the help of canonical inputs from GPS database and the output produced by the above algorithm we can successfully detect the presence of static and dynamic hazards on a runway under poor visibility conditions.

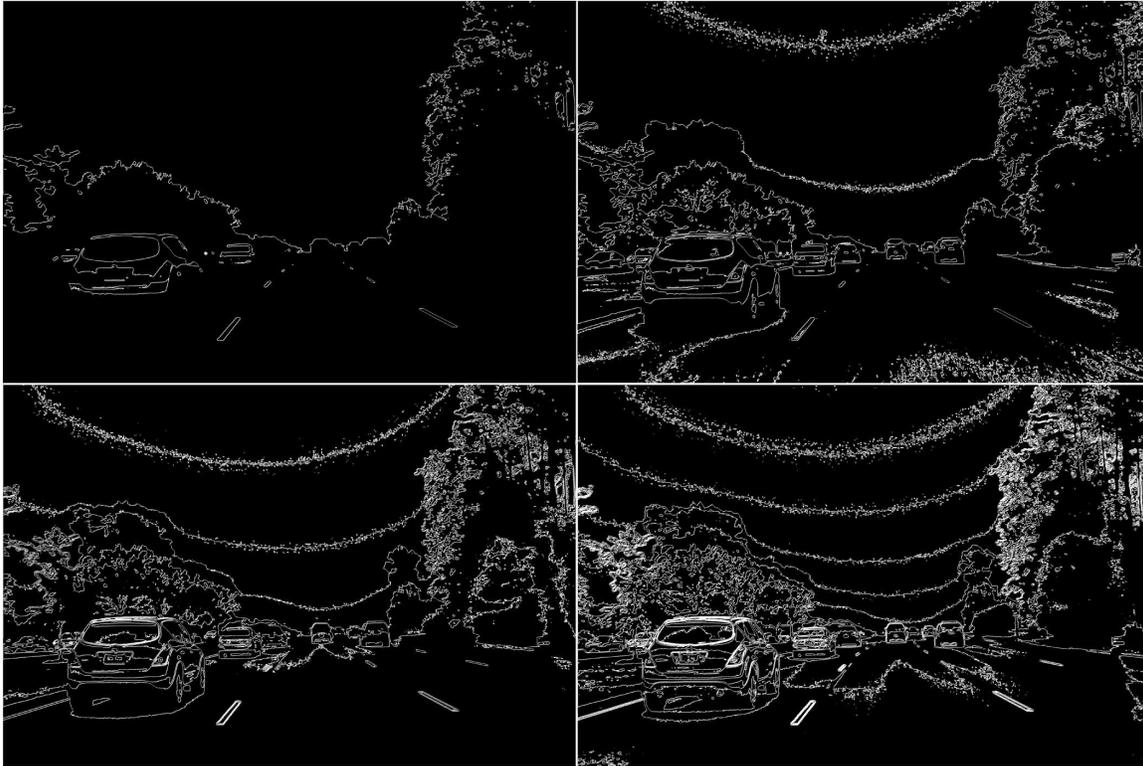


Figure 10: (top-left) (top-left) $\alpha = 1$; (top-right) $\alpha = 3$; (bottom-left) (top-left) $\alpha = 5$; (bottom-right) $\alpha = 8$.

ACKNOWLEDGMENTS

The authors wish to thank the NASA Aviation Safety Program for the funding which made this work possible. This work was supported under NASA cooperative agreement NNL07AA02A.

REFERENCES

1. D. J. Jobson, Z. Rahman, and G. A. Woodell, "Properties and performance of a center/surround retinex," *IEEE Trans. on Image Processing* **6**, pp. 451–462, March 1997.
2. D. J. Jobson, Z. Rahman, and G. A. Woodell, "A multi-scale Retinex for bridging the gap between color images and the human observation of scenes," *IEEE Transactions on Image Processing: Special Issue on Color Processing* **6**, pp. 965–976, July 1997.
3. Z. Rahman, D. J. Jobson, and G. A. Woodell, "Retinex processing for automatic image enhancement," *Journal of Electronic Imaging* **13**(1), pp. 100–110, 2004.
4. D. J. Jobson, Z. Rahman, G. A. Woodell, and G. D. Hines, "A comparison of visual statistics for the image enhancement of foresite aerial images with those of major image classes," in *Visual Information Processing XV*, Z. Rahman, S. E. Reichenbach, and M. A. Neifeld, eds., Proc. SPIE 6246, 2006.
5. Z. Rahman, D. J. Jobson, G. A. Woodell, and G. D. Hines, "Automated, on-board terrain analysis for precision landings," in *Visual Information Processing XV*, Z. Rahman, S. E. Reichenbach, and M. A. Neifeld, eds., Proc. SPIE 6246, 2006.
6. G. A. Woodell, D. J. Jobson, Z. Rahman, and G. D. Hines, "Advanced image processing of aerial imagery," in *Visual Information Processing XV*, Z. Rahman, S. E. Reichenbach, and M. A. Neifeld, eds., Proc. SPIE 6246, 2006.

7. D. J. Jobson, Z. Rahman, and G. A. Woodell, "Feature visibility limit in the nonlinear enhancement of turbid images," in *Visual Information Processing XII*, Z. Rahman, R. A. Schowengerdt, and S. E. Reichenbach, eds., Proc. SPIE 5108, 2003.
8. Z. Rahman, D. J. Jobson, and G. A. Woodell, "Resliency of the multiscale retinex image enhancement algorithm," in *Proceedings of the IS&T Sixth color Imaging Conference: Color Science, Systems, and Applications*, pp. 129–134, IS&T, 1998.
9. Z. Rahman, G. A. Woodell, and D. J. Jobson, "A comparison of the multiscale retinex with other image enhancement techniques," in *Proceedings of the IS&T 50th Anniversary Conference*, pp. 426–431, IS&T, 1997.
10. D. J. Jobson, Z. Rahman, G. A. Woodell, and G. D. Hines, "The automatic assessment and reduction of noise using edge pattern analysis in nonlinear image enhancement," in *Visual Information Processing XIII*, Z. Rahman, R. A. Schowengerdt, and S. E. Reichenbach, eds., Proc. SPIE 5438, 2004.
11. G. A. Woodell, D. J. Jobson, Z. Rahman, and G. D. Hines, "Enhancement of imagery in poor visibility conditions," in *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense IV*, E. Carapezza, ed., Proc. SPIE 5778, 2005.
12. J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(6), pp. 679–698, 1986.
13. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley, Reading, MA, 1993.
14. R. Zhang, G. Zhao, and L. Su, "A new edge detection method in image processing," in *IEEE International Symposium on Communications and Information Technology, 2005 (ISCIT 2005)*, pp. 445–448, 2005.