2012 Ninth Conference on Computer and Robot Vision

Robust Horizon Detection using Segmentation for UAV Applications

Nasim Sepehri Boroujeni, S. Ali Etemad
Department of Systems and Computer Engineering
Carleton University
Ottawa, Canada
{sepehri, etemad}@sce.carleton.ca

Anthony Whitehead
School of Information Technology
Carleton University
Ottawa, Canada
anthony_whitehead@carleton.ca

Abstract — **A critical step in navigation of unmanned aerial vehicles is the detection of the horizon line. This information can be used for adjusting flight parameters as well as obstacle avoidance. In this paper, a fast and robust technique for precise detection of the horizon path is proposed. The method is based on existence of a unique light field that occurs in imagery where the horizon is viewed. This light field exists in different scenes including sea-sky, soil-sky, and forest-sky horizon lines. Our proposed approach employs segmentation of the scene and subsequent analysis of the image segments for extraction of the mentioned field and thus the horizon path. Through various experiments carried out on our own dataset and that of another previously published paper, we illustrate the significance and accuracy of this technique for various types of terrains from water to ground, and even snow-covered ground. Finally, it is shown that robust performance and accuracy, speed, and extraction of the path as curves (as opposed to a straight line which is resulted from many other approaches) are the benefits of our method.**

Keywords- unmanned aerial vehicles; horizon; clustering; k-means.

I. INTRODUCTION

As the applications of automated and unmanned systems have grown in recent years, the need for precise and robust guidance and navigation systems has increased as well. Automated computer vision systems are aimed at taking over the role of the eyes and brain of an operator for controlling an automated vehicle. Automated cars, unmanned aerial vehicles (UAV), automated industrial systems, and visually impaired aid systems among many others are different examples of such systems.

Applications of UAVs range from traffic surveillance, monitoring of forests especially in fire situations, and transportation, to military. It has therefore received much attention in recent years [1].

UAV navigation, due to its sensitive nature and high cost in case of collision or failure, has been subject to extensive research [2]. Obstacle detection for UAVs is the first step towards robust navigation. This task, however, is extremely difficult for many reasons among which the following can be mentioned:

- the need for real-time response
- high speed of aerial vehicles
- existence of varying lighting conditions
- various existing types of terrains

In order to simplify the problem at hand, detection of the horizon line can be significantly beneficial for UAV obstacle detection and navigation systems. In addition to navigation applications, detecting the horizon line implies division of the scene into sky and ground. Accordingly, the ground can be labeled as one single obstacle for avoidance and thus the analysis can be focused on the sky portion alone. This will increase the speed or alternatively enable a more accurate analysis given the available time frame.

In this paper, the problem of horizon detection is studied. A new method is proposed based on the existence of a unique light field that occurs in imagery where the horizon is viewed. In the various data sets we have examined, all of them exhibit this light field effect at the horizon. Clustering the image is proposed which can be carried out very quickly and accurately. The proposed method can detect the horizon path not as a straight line but as the exact boundary between the sky and the ground regions. Different clustering methods can be employed in order to extract this light field. Among these methods, *k*-means and intensity-based are examined. Experiments indicate that both methods detect the horizon line precisely and this shows the effectiveness of the proposed technique both in terms of the approach and the tools used for the purpose at hand. Furthermore, the two clustering methods are compared for a better and complete analysis. A review on the related literature on horizon detection is presented in Section 2. Through Section 3, the methodology is described in detail. Section 4 presents the results and discussion on several issues including time requirements and speed. In section 5 the limitations of the method are discussed. Finally in Section 6 the concluding remarks and future work are presented.

II. RELATED WORK

Various literatures tackle the problem of horizon detection. Fefilatyev et al. [3] applied machine learning methods for this purpose. They first defined two segments for each scene, sky and ground. Subsequently all pixels in the scene were classified into either of these two segments. This classification was performed using support vector machines (SVM), decision trees, and Naive Bayes classifiers. The

978-0-7695-4683-4/12 $26.00 © 2012 IEEE
DOI 10.1109/CRV.2012.52

horizon was then assumed as the line that separates the sky pixels from ground ones.

In order to detect this line image processing methods such as Hough Transform [4] have been widely used. Zafarifar et al. [5] used a method based on edge and color detection for finding the horizon line in an image. They found the most significant line in the image by combining two different detectors: Canny edge detector and Hough Transform. In [6] Ettinger et al. illustrated the efficiency of vision based methods in horizon detection by evaluating bird's flights. For detecting the horizon, they focused on the variations that occur between ground and sky through calculating the covariance of images in three color channels.

Yuan et al. [7] proposed a method for detecting horizon in foggy images. In their method, an energy function in dark channel space was utilized for detecting the horizon. The results indicated the proper efficiency of the method in foggy conditions. In [8] Todorovic et al. made use of color and texture features in order to model the sky and ground parts in an image. Then these prior models were used to train the detection system. For representing texture and color information, complex wavelet transform (CWT), and hue and intensity were used. Finally Hidden Markov model Tree (HMT) was applied for the modeling purpose.

McGee et al. in [9] performed sky segmentation for small UAVs using an SVM. The pixels in an image were categorized into sky/ground groups based on their color in YCrCb color space. Applying the Hough Transform on the segmented image, they found several candidates for the horizon line. The horizon line was then defined as the minima in a cost function. In another approach Bao et al. [10] made use of orientation projection for robust horizon detection. Their method worked for images that were taken in bad conditions as well. Algorithmic optimization was applied for decreasing the computational time in the case of real time applications. They also used their results for computing navigation parameters of the Micro Air Vehicle.

Based on the aforementioned review on the techniques in the literature addressing detection of the horizon line, most techniques revolve around gradient and edge based approaches [5,7,8]. These methods often result in straight lines with particular slopes based on the orientation of the vehicle. One of the important drawbacks of most previous works in this field is that a single line is detected as the output. This means uneven horizon lines cannot be detected accurately. Furthermore, when the ground scene contains regions with high intensity gradients with respect to other regions, the detected horizon line may be affected. Others such as [3] and [9] which use machine learning techniques rely on a large corpus for accuracy and are typically slow. The method proposed in this paper robustly detects the exact horizon line while maintaining low computational time which makes it extremely robust and useful for practical applications.

III. METHODOLOGY

The proposed method employs light intensity gradients near the horizon line to detect the exact path of where the sky region meets the ground (or water). Whether sunset or sunrise has already occurred or not, a common element in many horizon scenes is a small and thin region slicing the image in two sections near the horizon path or region of interest (ROI). Figure 1 illustrates several instances where this region is clearly visible.

To detect the described regions near the exact horizon path in images, clustering is used as the main tool. By selecting the sufficient number of clusters, the ROI can be extracted for further analysis and obstacle detection purposes.

Prior to clustering, pre-processing is carried out. In order to minimize the effects of noise, sharp pixel intensity variations, and removal of false and invalid edges, the grayscale image is blurred. The Gaussian filter which is a low-pass filter (that employs the Gaussian function for blurring), is utilized for this purpose.

Clustering is carried out as the next step for detection of the ROI. Two types of clustering, among several others are employed for this purpose: intensity-based and *k*-means. Through the following sub-sections, each type is described and in Section 4, the performance of each type is illustrated along with their time requirements.

A. Intensity Clustering

Clustering based on the pixel intensities has been widely explored in the past [11,12] and is one of the most primitive and intuitive image processing techniques. While the color or intensity information in an image might seem naive, they are, in fact, extremely valuable and data-rich [12]. This is because pixels of an object that appear relatively close in an image are very likely to appear within a specific range of intensities. This type of segmentation is therefore used in many machine vision applications [12].

Intensity segmentation is usually performed by dividing the entire intensity spectrum into a fewer number of values through quantization. As shown in Eq. 1, the pixels containing similar quantized values fall within the same cluster.

$$I_c = floor\left(\frac{I_i}{255/n}\right) \quad \forall i \in image \tag{1}$$

In Eq. 1, I_c is the set of pixel intensities after clustering, I_i is the set of pixel intensities of the original image successive to grayscale conversion and blurring, and n is the number of clusters.

To further divide the image into distinct segments, similarly valued but non-connected clusters must be labeled as different clusters. Similarly colored objects in different regions of the scene as well as colors that appear alike when converted to grayscale can result in identically labeled yet non-connected clusters. To solve this problem, connectivity labeling is employed. If two parts in an image share a boundary with non-zero area they are considered connected. Through each iteration of the re-labeling process using this method, the image is converted to binary for each quantization level. The non-connected regions of that particular cluster (quantization level) are subsequently detected and re-labeled. The process of detecting and re-labeling of the clus-

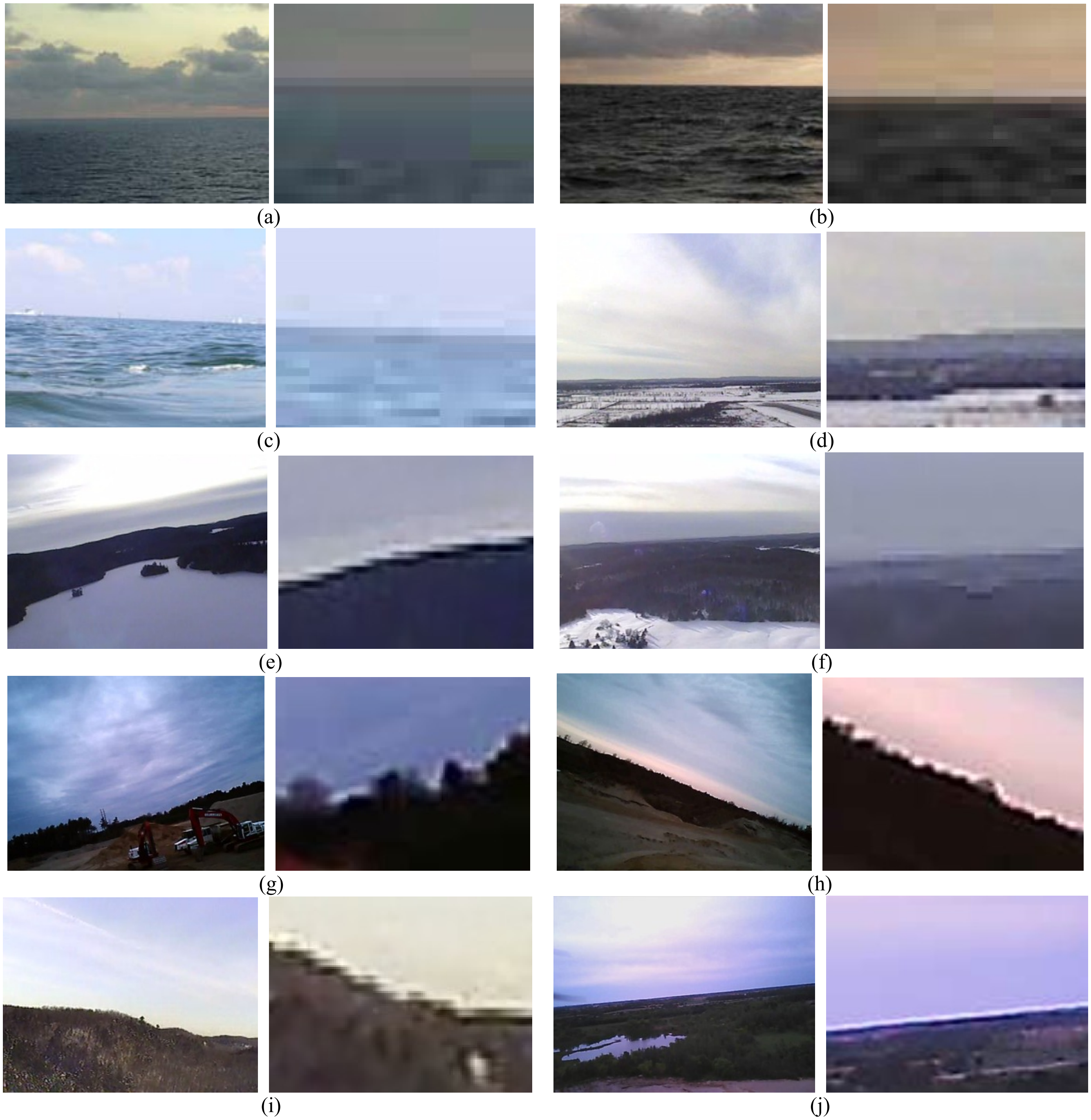


Figure 1. Horizon scenes showing regions with light intensity changes at the ROI: (a), (b), and (c) are from [3],(d) through (j) are courtesy of the National Research Centre of Canada dataset used for this research. The image on the left is the original, the image on the rights is the zoomed in image from the left that exhibits the high intenstiy light field indicating the locaiton of the horizon. It is particlulary clear in (e) (g) (h) (i) and (j).

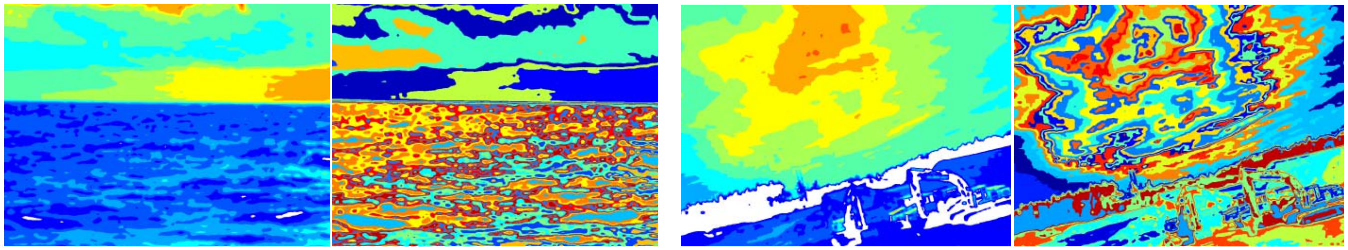
ters for acquiring a uniquely clustered output image is carried out using the union find algorithm [13,14].

The union-find algorithm consists of three major steps:

M(x): the make function which generates a new region with the only pixel x as the root.

F(x): the find function which follows the path from the pixel x to the root and returns the root as index of the path.

U(x,y): the union function which finds the indexes of x and y, respectively. Two regions are joined by connecting the corresponding roots.



(a) (b)

Figure 2. Clustering horizon scenes using intensity-based and *k*-means methods: (a) is the output for Figures 1-(b) [3], and (b) is the output for Figure 1-(g) (courtesy of the National Research Centre of Canada dataset used for this research). The images on the left have been clustered using the intensity-based technique while the one on the right is the output for the *k*-means technique. In both images and using either approach, the ROI can be seen as a distinct cluster.

The complete algorithm is described in pseudo-code through Algorithm 1.

Algorithm 1: Union-find algorithm

```
M(x): index(x) = x
U(x,y): connect(F(x),F(y))
connect(x,y):   if  x ≠ y
      then index(y) = x
F(x):   if  x ≠ index(x)
      then return F(index(x))
        else  return x
```

The resulting output of this process is an image with n number of clusters, each of which contains several labeled sub-clusters. As the final step of the clustering algorithm, the entire set of clusters and sub-clusters is relabeled from 1 to the sum of total sub-clusters available in the image.

B. *k-means Clustering*

k-means is a method for classifying a number of objects into a predefined number of groups/clusters. The classification is based on the similarity of each object to the center point of each cluster. In other words, the object which has the most similarity to the center point of a group is assigned to that cluster. The entire procedure is summarized below [12].

First a number of center points are selected randomly. Then each point is assigned to a cluster that its center point has smallest distance from the corresponding point. In the next step, the center points are updated. This updating is done by calculating the average of all current points in the cluster and considering the average as the new center point for that group. After that, all the points are divided into new clusters with new centroids. This process is continued until a certain convergence criterion is met. The equations below (Eq. 2 and Eq. 3) show how different points are assigned to different groups, and how the center points of clusters are updated.

$$C_i(t) = \{x_j : \| x_j - \mu_i(t) \| \le \| x_j - \mu_{i^*}(t) \| \} \quad \text{for all } i^* = 1,\dots,k \text{ and } j = 1,\dots,n \tag{2}$$

$$\mu_i(t+1) = \frac{1}{|C_i(t)|} \sum_{x_j \in C_i(t)} x_j \tag{3}$$

In Eq. 2 $C_i(t)$ represents the ith cluster at iteration t while x_j is the sample being placed in one of the clusters. In these equations, μ_i denotes the centroid of the cluster, calculated by Eq. 3. The number of clusters is denoted by k while n represents the number of points.

Finally, after clustering the image using the *k*-means method, the union-find algorithm is carried out to ensure unique clusters throughout the image similar to what was done with the intensity clustering technique.

C. *Cluster Detection*

After clustering, the cluster above the exact horizon path needs to be detected. Multiple investigations on clustered images show that the ROIs maintain a common characteristic. This common feature is minimization of the number of pixels that are members of the ROI cluster, stretching through the entire image from left to right. By increasing the number of clusters such that ROI cluster becomes thinner, this feature will become more apparent and robust. Eq. 4 and 5 formulate the criteria for extraction of the ROI where δ returns the number of pixels in cluster C, w is the width of the image, and X is the set of all horizontal indices of valid pixel coordinates.

$$\arg\min_i \; \delta(C_i) \tag{4}$$

$$\exists (x,y) \in C_i \Rightarrow \{1, w\} \subset X \tag{5}$$

IV. Results and Discussion

The proposed algorithm was implemented in MATLAB on a system with 4.0 GB of RAM and a CPU with a speed of 2.40 GHz. In order to evaluate the proposed method, several water and sky scenes from [3] have been employed along with images from the UAV data set of the National Research Centre of Canada (NRC). Figure 2 illustrates clustering of the images using both the described intensity-based and *k*-means methods. Clearly, the ROIs have been detected as a distinct cluster in both methods. However, the *k*-means method requires many more clusters for the horizon to be detected precisely.

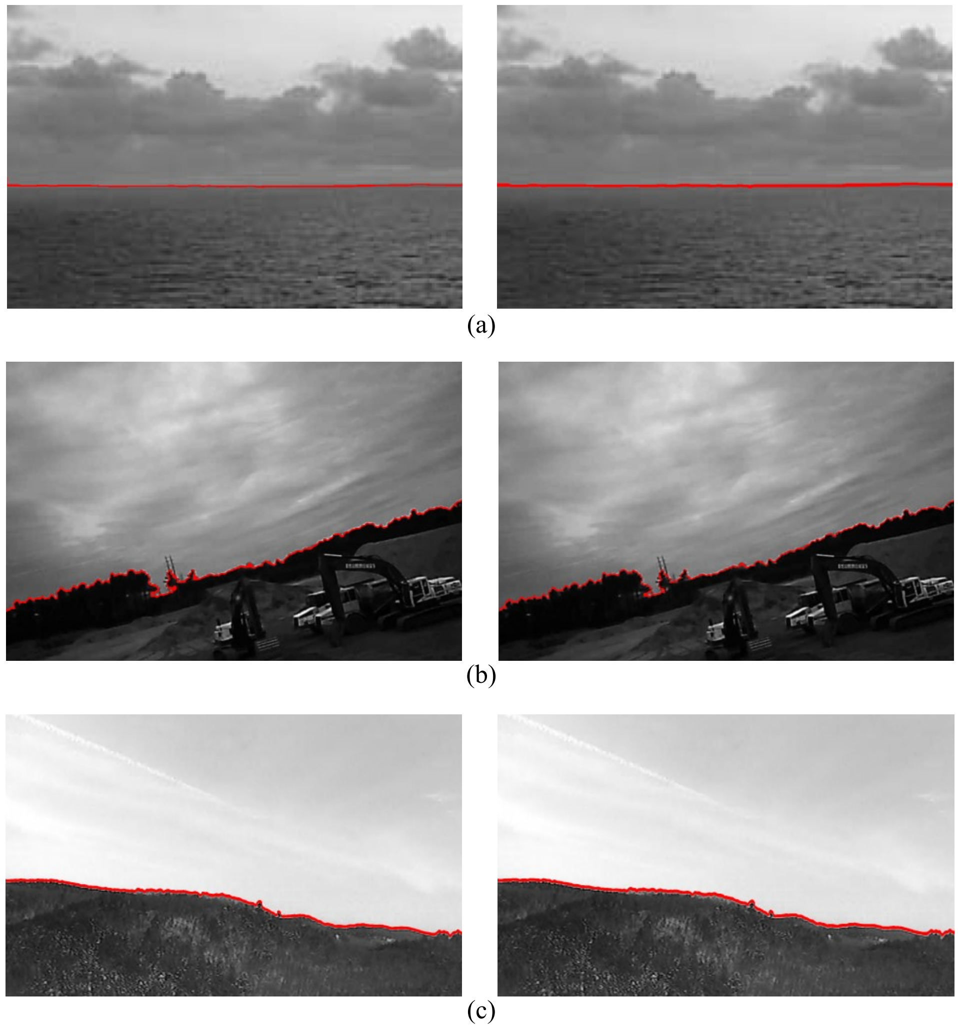


Figure 3. Horizon detection using both intensity-based (left) and k-means (right) clustering: (a) is from [3],(b) through (c) are courtesy of the National Research Centre of Canada dataset used for this research.

Figure 3 illustrates the final output of the algorithm performed on three of the images illustrated in Figure 1. The remaining 6 are presented in Figure 4. The results clearly show that the exact ROI is extracted. Furthermore, the two methods are clearly very similar in terms of the final output. This is due to the fact that a region, mostly quite thin, is visible right above the horizon in which there is a sudden intensity change. The method executes robustly for different terrain types as can be seen in Figures 3 and 4. It should also be noted that this technique outperforms the original approach used for Figures 1 (a) to (c) used in [3]. Furthermore, we applied the proposed method on several videos from the National Research Centre of Canada dataset. Similar to the figures shown earlier, the horizon paths were detected accurately and as expected. The technique using intensity clustering, when applied on 160 frames of UAV videos, performed with an error rate of 1.2%.

It is illustrated that our method succeeds in various scenery conditions. Scenes with different terrains and light conditions all demonstrate the described light field, and therefore, the horizon paths are robustly extracted. Furthermore this method outperforms other methods which are based on color and edge features in cases like Figures 4(d) and 4(e) where there are significant edges or color variations other than the main horizon line. Other methods may detect incorrect lines as the horizon, while focusing on the mentioned light field will result in correct outcome.

To further analyze the performance of each method, Figure 5 illustrates the time requirements for each type of clustering vs. the number of clusters. While usually less than 15 clusters are required to single out the ROI, the algorithms are tested for up to 30 clusters. Moreover, linear approximations for each method illustrate the rate of speed decrease for each clustering method. This figure shows that the slope of *k*-

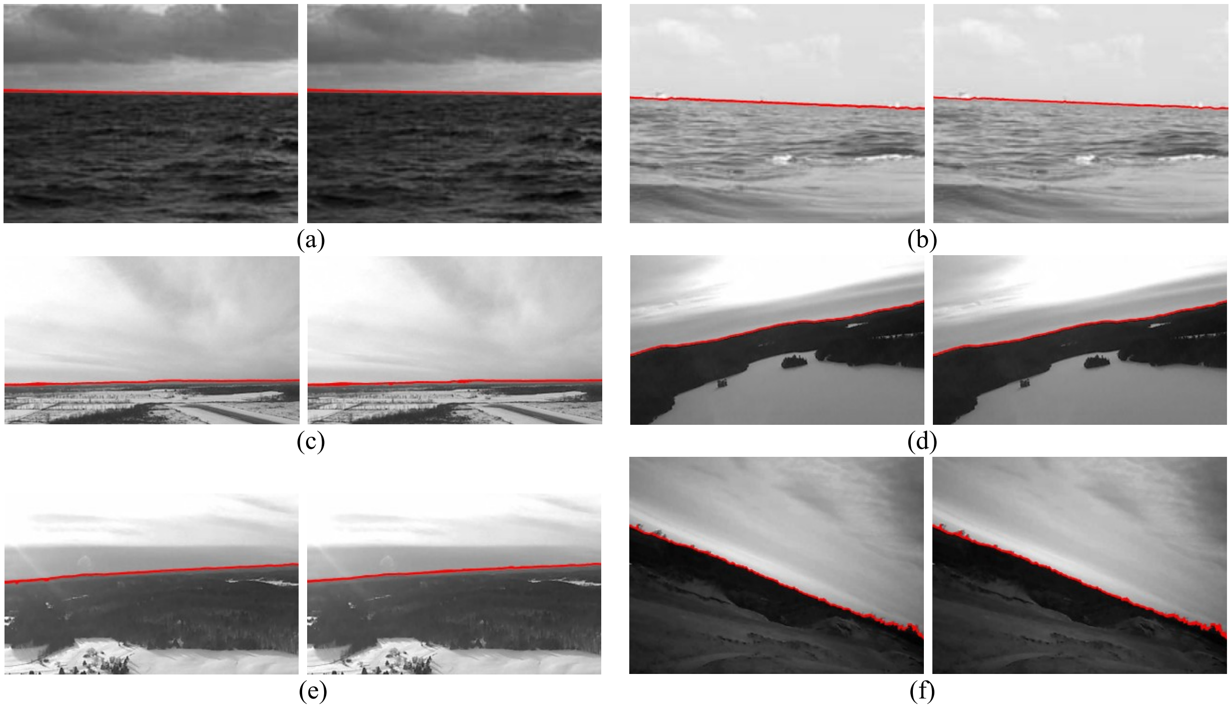


Figure 4. Horizon detection using both intensity-based (left) and k-means (right) clustering: (a) and (b) are from [3],(c) through (f) are courtesy of the National Research Centre of Canada dataset used for this research.

means to be approximately 7 times greater than that of intensity-based clustering (0.7 vs. 0.1). These results are found when segmentation is carried out on Figure 1-(e), but very similar graphs are acquired for other images as well. In general the average computational time for intensity-based clustering is less than 1.5 seconds in MATLAB while for *k*-means it is approximately 10 seconds.

The number of clusters is a critical parameter in the proposed method. As the numbers of clusters are increased, the area of each cluster, as well as the cluster containing the ROI, decreases. It is therefore important to compute the optimum number of used clusters so that the horizon path is detected and as thin as possible. Increasing the number of clusters beyond a certain point, however, will have significant impact on time which can be avoided. More importantly, over-clustering will result in the ROI cluster to be broken and therefore lost prior to detection. In general, *k*-means provides similar output but requires much more time because of its algorithm and also because in most cases it requires more clusters for the ROI to be detected. It is therefore logical for intensity-based technique to be adapted as the more practical technique.

V. Performance and Limitations

The proposed method was applied on numerous images with various types of scenes including cloudy (Figures 1 (a) and (b)), sunny (Figures 1 (c) and (i)), and sunset (Figure 1 (h)) sky types. The ground section was also very variable through the tested images. Water scenes captured with different angles (Figures 1 (a), (b) and (c)), soil (Figures 1 (h) (i), and (j)), snow and ice (Figures 1 (d), (e), and (f)), and soil with various objects in the field of view (Figure 1 (g)) have all been assessed. The system performs robustly in all cases and the ROI was detected accurately as a continuous line with all the necessary curves, hills, and valleys.

Based on the 3D orientation of the vehicle as well as the landscape, different types of ROIs can be assumed. A diagram for this general classification is illustrated through Figure 6. Among the 12 situations shown in this figure, (a) through (h) are detectable using our algorithm. The situation shown in Figure 6 (a) which is the general diagram for Figures 1 (a) to (d), and (f) has been tested with the ROI successfully detected. A situation such as Figure 6 (b) or its mirrored image can simply be detected by rotating or flipping the scene accordingly. The ROI in Figure 6 (c) which resembles cases such as Figures 1 (e), (g) and (j) has been detected successfully as well. Cases such as Figure 6 (d) can again be dealt with through rotating or flipping of the image. Situations such as Figure 6 (e) which are similar to Figures 1 (h) and (i) can also be successfully processed as illustrated earlier. The ROI in scenes such as Figure 6 (f) can be detected through required rotations and flipping to achieve the previous situation. Finally general situation shown through Figures 6 (g) and (h) are special cases of Figure 6 (c) and (e) which have already been discussed and shown to pose no difficulty for the proposed system.

In the end, cases illustrated in Figures 6 (i) through (l) require modification in the detection algorithm (Eqs. 4 and 5).

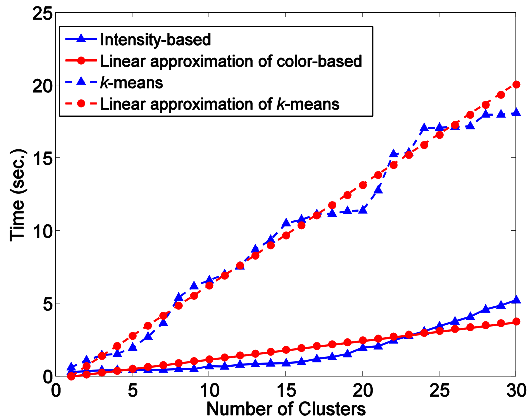


Figure 5. Time requirements for hoziron detection using intensity-based and *k*-means clustering vs. the number of clusters.

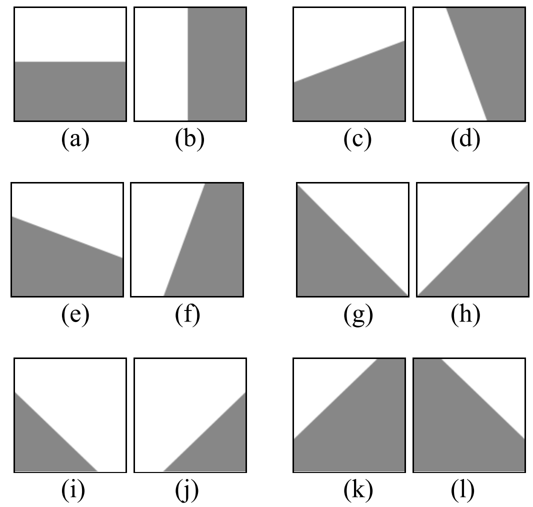


Figure 6. General situations for horizon paths. The proposed method performs robustly on cases such as (a) through (h) while (i) through (l) require modification on the cluster detection process.

By adjusting Eq. 5 detection of such cases will become possible which we intend to implement for future work.

VI. Conclusion and Future Work

A novel method for horizon detection was proposed that exploits the discovered property of the existence of a dominant light field between the sky and the ground right at the horizon path. This dominant light field can be found using clustering as the main tool. Two different clustering techniques, intensity-based and *k*-means, were employed to illustrate the extractability of the mentioned field. By selecting the number of clusters within a particular range, the horizon path was extracted very accurately. This can be used for many UAV applications such as obstacle detection and navigation. The proposed technique performed robustly for various types of terrain and scenes showing the horizon.

It was shown that the intensity-based method was significantly faster than the *k*-means method and should be easily implementable in real time.

For future work, we will apply the proposed method on more video sequences and implement it using a lower level language in real time. Moreover, more investigations on determining the ideal number of required clusters based on the input will be carried out to further automate and formulate the process. Finally as discussed in Section 5, we intend to modify Eq. 5 to enhance the system for performing on cases such as that shown in Figure 6 (i).

References

[1] S. Park, D.H. Won, M.S. Kang, T.J. Kim, H.G. Lee, and Kwon, S.J. "RIC (robust internal-loop compensator) based flight control of a quad-rotor type UAV," IEEE /RSJ International Conference on Intelligent Robots and Systems, pp. 3542–3547, 2005.

[2] L. Jian, and L. Xiao-min, "Vision-based Navigation and Obstacle Detection for UAV," International Conference on Electronics, Communications and Control, pp. 1771–1774, 2011.

[3] S. Fefilatyev, V. Smarodzinava, L. O. Hall, and D. B. Goldgof, "Horizon Detection Using Machine Learning Techniques," International Conference on Machine Learning and Applications, pp. 17–21, 2006.

[4] S. R. Deans, "Hough Transform from the Radon Transform," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-3(2), pp. 185–188, 1981.

[5] B. Zafarifar, H. Weda, and P. H. N. de With, "Horizon Detection Based on Sky-Color and Edge Features," Conference on Visual Communications and Image Processing, vol. 6822, pp. 682220 1-9, 2008.

[6] S. M. Ettinger, M. C. Nechyba, P. G. Ifju, and M. Waszak, "Vision-Guided Flight Stability and Control for Micro Air Vehicles," IEEE /RSJ International Conference on Intelligent Robots and Systems, vol. 3, pp. 2134 –2140, 2002.

[7] H. Z. Yuan, X. Q. Zhang, and Z. L. Feng, "Horizon detection in foggy aerial image," International Conference on Image Analysis and Signal Processing, pp. 191– 194, 2010.

[8] S. Todorovic, M. C. Nechyba, and P. G. Ifju, "Sky/ground modeling for autonomous MAV flight," Proceedings of IEEE International Conference on Robotics and Automation, vol. 1, pp. 1422–1427, 2003.

[9] T. G. McGee, R. Sengupta, and K. Hedrick, "Obstacle Detection for Small Autonomous Aircraft Using Sky Segmentation," Proceedings of International IEEE Conference on Robotics and Automation, pp. 4679 –4684, 2005.

[10] G. Q. Bao, S. S. Xiong, and Z. Y. Zhou, "Vision-based horizon extraction for micro air vehicle flight control," IEEE Transactions on Instrumentation and Measurement, vol. 54(3), pp. 1067 - 1072, 2005.

[11] M. Bajaj, and J.A. Lay, "Image Indexing and Retrieval in Compressed Domain Using Color Clusters," IEEE Symposium on Computational Intelligence in Image and Signal Processing, pp. 271–274, 2007.

[12] M. Mignotte, "Segmentation by Fusion of Histogram-Based k-Means Clusters in Different Color Spaces," IEEE Transactions on Image Processing, vol. 17(5), pp. 780–787, 2008.

[13] C. Fiorio, and J. Gustedt, "Two linear time Union-Find strategies for image processing," Theoretical Computer Science, vol. 154(2), pp. 165–181, 1996.

[14] M. Tamminen, and H. Samet, "Efficient octree conversion by connectivity labeling," Proceedings of the 11th annual conference on Computer graphics and interactive techniques, 1984.