

# Fusion of Edge-less and Edge-based Approaches for Horizon Line Detection

Touqeer Ahmad\*, George Bebis\*, Monica Nicolescu\*, Ara Nefian†, Terry Fong†

\*Department of Computer Science and Engineering

University of Nevada, Reno

{ahmad, bebis, monica}@cse.unr.edu

†NASA Ames Research Center

{ara.nefian, terry.fong}@nasa.gov

**Abstract**—Horizon line detection requires finding a boundary which segments an image into sky and non-sky regions. It has many applications including visual geo-localization and geo-tagging, robot navigation/localization, and ship detection and port security. Recently, two machine learning based approaches have been proposed for horizon line detection: one relying on edge classification and the other relying on pixel classification. In the edge-based approach, a classifier is used to refine the edge map by removing non-horizon edges. The refined edge map is then used to form a multi-stage graph where dynamic programming is applied to extract the horizon line. In the edge-less approach, classification is used to obtain a confidence of horizon-ness at each pixel location. The horizon line is then extracted by applying dynamic programming on the resultant dense classification map rather than on the edge map. Both approaches have shown to outperform the classical approach where dynamic programming is applied on the non-refined edge map. In this paper, we provide a comparison between the edge-less and edge-based approaches using two challenging data sets. Moreover, we propose fusing the information about the horizon-ness and edge-ness of each pixel. Our experimental results illustrate that the proposed fusion approach outperforms both the edge-based and edge-less approaches.

## I. INTRODUCTION

Horizon line detection is the problem of finding a boundary between sky and non-sky regions (ground, water or mountains) given a gray scale or color image. There are several applications of detected horizon including smooth navigation of unmanned aerial vehicles (UAVs) [17], [14], [23] and micro air vehicles (MAVs) [12], [13], [15], augmented reality [20], visual geo-localization and annotation of mountain/desert imagery [22], port security [16], and outdoor vehicle localization. Previous horizon line detection methods can be grouped into two major categories; (i) methods modeling sky and non-sky regions using machine learning [17], [12], [13], [16], [14], [15], [21] and (ii) methods relying on edge detection [19], [5]. Some attempts [1], [2], [20] have been made to combine these two approaches by eliminating non-horizon edges using classification; however, these attempts also fall under the second category as horizon detection is effectively based on edges. Earlier methods to horizon detection mostly suffer from the assumption that the horizon boundary is linear and hence are limited.

In the classical edge based approach for horizon line detection Lie et al. [5], an edge map is used to form a multi-

stage graph and Dynamic Programming (DP) is applied to extract the horizon line by finding the shortest path. This approach is based on the assumption that there exists a consistent edge boundary segmenting the sky and non-sky regions. This assumption, however, is not always true due to various environmental factors (e.g., clouds) and edge gaps. To deal with this issue, a gap-filling strategy has been proposed in [5]; however, it relies heavily on several parameters such as tolerance-of-gap (tog) and  $\delta$ . These parameters define a search region when an edge node cannot be connected to a node in the next stage of the graph. Edge gaps greater than the tolerance provided by the chosen parameter values, however, cannot not be filled. Moreover, Lie et al. [5] assume that consistent edges appearing in the upper half of the image belong to the horizon line. To enforce this assumption, they induce biases towards shortest paths in this region. This assumption is not always valid as there might be strong edges in this region due to environmental factors (e.g., clouds). By enforcing this constraint, consistent edge segments due to clouds could become part of the extracted horizon line since there is no way to discriminate between cloud edges and horizon edges.

Ahmad et al. [2] and Hung et al. [1] have extended the approach of Lie et al. [5] by eliminating non-horizon edges using classification. The key idea is building the multi-stage graph using only horizon classified edges instead of using all the edges. For this, a classifier is trained using features around key-points from horizon and non-horizon locations. Ahmad et al. [2] used SIFT descriptors [8] around the key-points and SVMs for classification [10] whereas Hung et al. [1] used SVMs with color information and the variance above and below a given key-point. Moreover, Ahmad et al. [2] speed-up horizon line detection by reducing the number of non-horizon edges by keeping only those edges which can survive over a range of Canny thresholds. In a related work [3], [18], they have experimented with various nodal costs for dynamic programming along with binary costs used in the traditional approach [5]. They have also experimented with using various texture features (e.g., LBP, SIFT, HOG) and their combinations; SIFT-HOG were found to outperform any other feature choice.

More recently, Ahmad et al. [4] have proposed an edge-less approach for horizon line detection. Their approach is motivated by the fact that conventional edge based methods [2], [1], [5] suffer from gaps and provide no confidence

about the likelihood of an edge point belonging to the horizon. Therefore, instead of applying dynamic programming on an edge map, they use a classification map. The classification map (termed as Dense Classifier Score Image or DCSI) is built by classifying each pixel as belonging to the horizon or not. Therefore, each pixel is associated with a classification score which reflects its likelihood of belonging to the horizon. The resultant DCSI hence is a dense map and does not contain any gaps. The DCSI is then used to form a multi-stage graph and DP is applied to find the horizon line.

In this paper, we compare edge-less horizon detection approach with edge based approaches in order to identify specific cases where each method fails. Furthermore, we propose a fusion strategy where the edge-less approach can benefit from edge-based approach. The key idea is boosting the likelihood of pixels which lie on edges. The resultant map is still a dense map, therefore, it does not suffer from gaps and parameter choices related to gap filling. The fusion strategy outperforms any individual approach on two challenging data sets.

The rest of paper is organized as follows. In section 2, we review the edge based approaches of Lie et al. [5] and Ahmad et al [2], [3], [18]. Section 3 reviews the edge-less approach of Ahmad et al. [4]. Section 4 presents the proposed fusion strategy while section 5 presents our experimental results. Finally, section 6 provides our conclusions and directions for future work.

## II. EDGE-BASED HORIZON LINE DETECTION

### A. Lie et al. [5]

Lie et al. [5] formulated the problem of horizon detection as a multi-stage graph search problem. Given an image, edge detection is applied first. The detected edges are then used to form a multi-stage graph and DP is applied to extract the horizon line by finding the shortest path. This approach is based on the assumption that the horizon is present in the upper half of the image. Specifically, given an image of size  $M \times N$ , edge detection is performed to compute a binary edge map  $I$  where 1 implies the presence of an edge pixel and 0 a non-edge pixel. The edge map is used to build an  $M \times N$  multi-stage graph  $G(V, E, \Psi, \Phi)$  where each pixel in the edge map corresponds to a graph vertex; a low cost  $l$  is associated with edge pixels while a very high cost (i.e.,  $\infty$ ) is associated with non-edge pixels as shown below:

$$\Psi(i, j) = \begin{cases} l, & \text{if } I(x, y) = 1. \\ \infty, & \text{if } I(x, y) = 0. \end{cases} \quad (1)$$

$\Psi(i, j)$  is the cost associated with vertex  $i$  in stage  $j$  (i.e.,  $v_{ij}$ ). The graph can be visualized as an  $N$  (columns) stage graph where each stage contains  $M$  nodes (rows). To deal with edge gaps, a gap filling approach was proposed. Given a node  $i$  in stage  $j$ , its neighborhood in the next stage  $j + 1$  is defined by a  $\delta$  parameter, that is, the number of nodes to which  $i$  could be connected in stage  $j + 1$ . The edges from  $i$  to its neighbors are associated with costs equal to the vertical absolute distance from it as shown in the equation below.

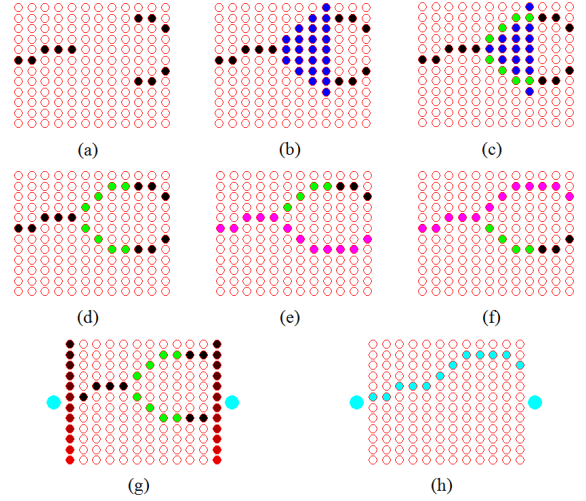


Fig. 1. Illustration of the horizon detection steps using the method of Lie et al. [5]

$$\Phi(i, k, j) = \begin{cases} |i - k|, & \text{if } I(i, j) = I(k, j + 1) = 1 \\ & \text{and } |i - k| \leq \delta \\ \infty, & \text{otherwise.} \end{cases} \quad (2)$$

If a node  $i$  in stage  $j$  cannot be connected to any node in stage  $j + 1$  within  $\delta$  neighborhood, a search window is defined using  $\delta$  and tolerance-of-gap (tog). Once an edge node  $k$  is found in the search window, the gap filling is performed by introducing dummy nodes between node  $i$  in stage  $j$  and node  $k$ . A high cost is associated with the dummy nodes introduced by the gap filling step.

Once the gaps have been filled with high cost dummy nodes, the cost of the nodes in stages 1 and  $N$  is increased based on the vertical position of the nodes as shown by the equation below:

$$\Psi(i, j) = \begin{cases} (i + 1)^2, & \text{if } j = 1 \text{ or } j = N \\ \Psi(i, j), & \text{otherwise.} \end{cases} \quad (3)$$

This enforces the assumption that the horizon line is present in the upper half of the image and hence biasing the DP solution towards a shortest path present in the upper half. Next, a source  $s$  and a sink  $t$  are added to the left of the left most stage (i.e., stage 1) and to the right of the right most stage (i.e., stage  $N$ ) respectively. A zero cost is associated with each one of them. The  $s$  node is connected with all the nodes in stage 1 and all the nodes in stage  $N$  are connected to node  $t$ . The horizon boundary is detected by finding the shortest path extending from node  $s$  to  $t$  using DP.

Figure 1 illustrates the steps of Lie et al. [5] for a sample image. An edge map is shown in Figure 1-(a) where black and white circles represent edge and non-edge pixels respectively. A search window (highlighted by blue circles) is shown in Figure 1-(b) for the edge node in stage  $j = 5$  using  $\delta = 1$  and  $tog = 4$ . Within the search window  $j + tog$ , two edge nodes are discovered which are then connected to node  $j$  by introducing

dummy nodes as shown in Figure 1-(c,d) (highlighted in green). So, there exist two equal cost paths 1-(e,f) in the resultant image, highlighted in magenta. However, the nodes in stage 1 and  $N$  are set to a higher cost associated with their vertical position; this is reflected by an increasing intensity in Figure 1-(g). Two nodes  $s$  and  $t$  (cyan) are then introduced, as described above, and DP is applied on this graph. As shown in figure 1-(e,f) the two paths have the same cost, however the bias introduced in 1-(g) would make the upper path of lower cost and DP will select this path due to the assumption of the horizon line being present in the upper half. However, it might be possible that the true horizon line is actually the lower one and that the upper edge segment was only due to some clouds.

### B. Ahmad et al. [2], [3], [18]

In [2] Ahmad et al. proposed the use of classification and Maximally Stable Extremal Edges (MSEEs) to reduce the number of non-horizon edges before applying dynamic programming. They have further extended their analysis in [3] where they investigated various texture features (e.g., SIFT[8], LBP[6], HOG[7] and their combinations) to train an SVM classifier. The combination of SIFT and HOG features resulted in the best false negative error rate. Here, we are mostly interested in the nodal cost formulations proposed in [3] rather than on the various texture features.

Although Ahmad et al. [3], [18] used gradient information as nodal costs, we limit our analysis to SIFT-HOG classified edges and SIFT-HOG classified scores since gradient information was found to perform poorly. In the first edge based formulation, Ahmad et al. [3] perform Canny edge detection using various threshold values and keeping only those edges which survive over a range of thresholds; we refer to these edges as Maximally Stable Extremal Edges (MSEEs). The number of non-horizon edges in the MSEE image  $E(x, y)$  is reduced using classification. The resulted edge map  $E_+(x, y)$  can be written as:

$$E_+(x, y) = \begin{cases} 1, & \text{if } E(x, y) = 1 \& \text{Class}[D(E(x, y))] = 1. \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

where,  $E(x, y)$  is the MSEE image,  $Class$  corresponds to the classifier and  $D$  is the descriptor around each edge pixel in  $E(x, y)$ . It should be noted that  $D$  corresponds to SIFT-HOG descriptor around the edge pixels. In this formulation, Ahmad et al.[3], [18] have used a binary cost for nodal initialization following Lie et al. [5]; in this case, equations 1 and 2 become:

$$\Psi(i, j) = \begin{cases} l, & \text{if } E_+(x, y) = 1. \\ \infty, & \text{if } E_+(x, y) = 0. \end{cases} \quad (5)$$

$$\Phi(i, k, j) = \begin{cases} |i - k|, & \text{if } E_+(i, j) = E_+(k, j + 1) = 1 \\ & \text{and } |i - k| \leq \delta \\ \infty, & \text{otherwise.} \end{cases} \quad (6)$$

In the second edge-based formulation, instead of using the binary nodal costs, the actual classification scores are used. This is to reflect that classification scores provide some

confidence about an edge pixel belonging to the horizon or not. The raw classification scores are normalized between 0 and 1. Nodal costs are then initialized by the normalized classification scores instead of setting all positively classified edges to a fixed cost. We have modified equation 5 to reflect this where  $\Omega$  is the classification score in the interval [0–1]. Since, DP is applied to find the shortest path in the graph, we have reversed the classification score values so that the smaller the classification score value is, the more likely that pixel is a horizon pixel.

$$\Psi(i, j) = \begin{cases} \Omega(E_+(x, y)), & \text{if } E_+(x, y) = 1. \\ \infty, & \text{if } E_+(x, y) = 0. \end{cases} \quad (7)$$

Figure 2 illustrates the steps for the first formulation (edge-based approach) for one of the query images 2-(a). Figure 2-(b) and 2-(c) show the output of Canny edge detector and MSEE edges respectively to highlight the reduction in the number of edges due to MSEE. Figure 2-(d) shows the horizon classified edges which are used to formulate the graph for DP. The detected horizon for the query image is shown in figure 2-(e) in red.

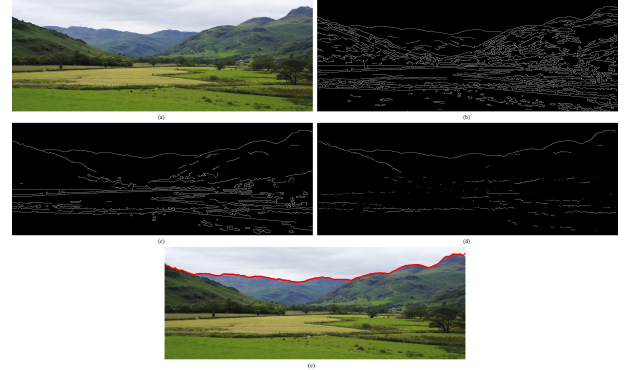


Fig. 2. Edge-based horizon line detection : (a) query image, (b) edges detected by Canny edge detector, (c) edges survived using MSEE, (d) horizon classified edges and (e) detected horizon (red boundary) using DP.

### III. EDGE-LESS HORIZON LINE DETECTION

In [4] Ahmad et al. proposed a machine learning approach for horizon line detection. Specifically, they apply DP on a classification map that associates with each pixel a classification score which can be interpreted as the confidence of the pixel being part of the horizon line. Unlike edge-based approaches where edge pixels are used to form the multi-stage graph, a dense graph is constructed where each pixel becomes a node. Once the source/destination nodes  $s/t$  have been added and the value of  $\delta$  has been decided, DP can be used to find the path that maximizes the sum of classification scores in the resultant dense graph. Since, the classification score is associated with each pixel, there exist no gaps in this approach.

For the edge-less approach, they have trained an SVM and a CNN classifier using normalized pixel intensities as features. The same positive and negative key-points as in [3] were used; intensity patches around them were used to train the classifier. Given a query image, a dense classification map  $D(x, y)$  is generated (i.e., DCSI) where a normalized classification score

is associated with each pixel. This classification score reflects the horizon-ness of each pixel in the image. Mathematically,

$$D(x, y) = \Gamma(V(x, y)) \quad (8)$$

where,  $\Gamma$  is the classifier and  $V(x, y)$  is the feature vector of normalized pixel intensities around the pixel  $(x, y)$ . Although the full DCSI can be used for horizon line detection, it was found that keeping only the  $m$  highest classification scores in each column does not compromise accuracy while reduces computations. The reduced DCSI is referred to as  $mDCSI$  [4]. The multi-stage graph corresponding to the  $mDCSI$  contains less vertices; as a result, fewer paths need to be considered when searching for the shortest path which results in considerable speedups. The  $mDCSI$  is computed as follows:

$$mD(x, y) = \begin{cases} D(x, y), & \text{if } x \in L(m)_y \\ l, & \text{otherwise; } l < 0. \end{cases} \quad (9)$$

where,  $mD(x, y)$  corresponds to the  $mDCSI$  and  $D(x, y)$  corresponds to the DCSI. If the  $x$ -th pixel (node) in column (stage)  $y$  belongs to the list  $L(m)_y$  of indices corresponding to the highest  $m$  classification scores, the classification score from  $D(x, y)$  is used; otherwise, the score is set to a low score  $l$  where  $l$  is smaller than the smallest score returned by the classifier. The  $M \times N$  graph  $G(V, E, \Psi, \Phi)$  can now be initialized with nodal costs from  $mD(x, y)$ .

$$\Psi(i, j) = mD(x, y) \quad (10)$$

The rest of the steps are the same as in edge-based detection, i.e., applying the DP on the resultant  $mDCSI$ . Figure 3 shows the steps for edge-less approach for the given image 3-(a). Figure 3-(b),(c) show the DCSI and  $mDCSI$  respectively. The detected horizon line is shown in 3-(d) in red.

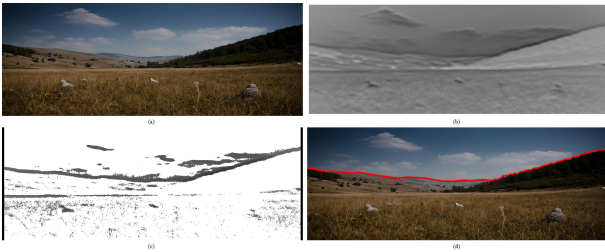


Fig. 3. Edge-less horizon line detection : (a) query image, (b) DCSI, (c)  $mDCSI$ , (d) detected horizon (red boundary).

#### IV. FUSION

Using machine learning for horizon line detection has shown to be very promising. Both machine learning based edge-less [4] and edge-based [3] approaches outperform the classical edge-based approach [5] by a high margin (see Table I). A quick look of table I reveals that edge-based method [3] outperforms edge-less approach [4]. To better understand the strengths and weaknesses of each method, we present below several reasons. Moreover, we propose a fusion strategy to improve horizon detection.

1) *Ground Truth Bias and Multiple Horizons*: While generating the ground truth, we used the results of edge detection to better localize the horizon line which favors edge-based methods. Moreover, sometimes there are more than one horizon lines in an image (e.g., lower mountains sitting in front of higher and more distant mountains); while generating the ground truth in those images, we chose the strongest segments found by the edge detector. When using edge-based methods for horizon detection, these segments are typically part of solution; however, this might not be the case for edge-less methods.

2) *Smoother Localization*: Edge-based solutions tend to be smoother while edge-less method are typically bumpy since DP tries to find the nodes with the least cost without imposing a smoothness constraint which is inherent in edge-based methods.

3) *Miss-Classifications*: Edge-less methods suffer more from miss-classifications compared to edge-based methods. This is because every pixel is classified in the case of edge-less methods while only edge pixels are classified in the case of edge-based methods.

Figure 4 shows several examples where edge-based horizon detection has outperformed edge-less horizon detection. Figure 5 provides more details.

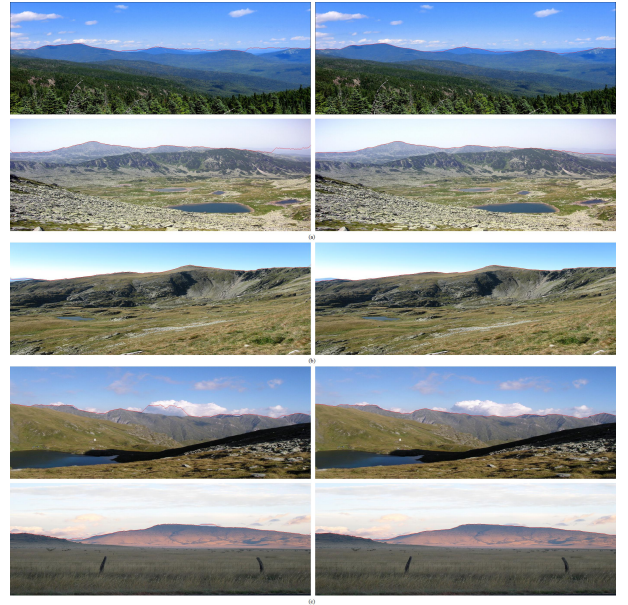


Fig. 4. Edge-less (left column) vs edge-based (right column) horizon detection results: (a) multiple horizons, (b) smoother localization, (c) miss-classifications. In each case, the edge-based approach has outperformed the edge-less approach.

##### A. Fusion

We discussed above several reasons favoring the edge-based approach. On the other hand, the main advantage of the edge-less approach is the lack of gaps in the DCSI map. Moreover, it can deal better with non-horizon edges such as cloud edges. To improve horizon detection, we propose a fusion strategy by combining edge-based with edge-less methods. First, the DCSI  $D(x, y)$  is generated for the query

## A. Data Sets and Quantitative Measure

To compare the performance of all the methods, we have experimented with two data sets : the Basalt Hills data set and the Web data set. The Basalt Hills data set is comprised of 45 handpicked images from a publicly available data set [11]. The images have been chosen such that they have considerable view point changes. The Web data set is based on 80 challenging images collected from Web and consists of various view point, seasonal and texture changes. The resolution of the images in both data sets is  $519 \times 1388$ . For training, we used only 9 images from the Basalt data set; 343 positive and 343 negative key points were chosen uniformly from the ground truth horizon locations and randomly from non-horizon edge locations. A  $16 \times 16$  image patch is taken around each of these key points. These image patches are used to compute the SIFT and HOG features for training the classifier in the case of the edge-based approach. The concatenated SIFT-HOG vector is used to get either a binary classification value for the current edge pixel or a normalized classification score depending on which nodal cost is used for the DP. In the case of the edge-less approach, we use the normalized pixel intensities for training the classifier.

To provide a fair comparison among different approaches, we have manually extracted the horizon line (ground truth) in all the images of our data sets. For evaluation, the detected and true horizon lines are compared by calculating a pixel-wise absolute distance  $S$  between them. For each column, the absolute distance between the detected and ground truth pixels is computed and summed over the entire number of columns in the image. The resultant distance is normalized by the number of columns in the image, yielding the average absolute error of the detected horizon line from the ground truth. Since nodes in a given stage are not allowed to be connected with nodes in the same stage, the true and detected horizon lines are bound to have the same number of columns/stages in the image/graph. Hence, there is a one-to-one correspondence between pixel locations in the true and detected horizons:

$$S = \frac{1}{N} \sum_{j=1}^N |P_{d(j)} - P_{g(j)}| \quad (13)$$

where  $P_{d(j)}$  and  $P_{g(j)}$  are the locations (rows) of the detected and true horizon pixels in column  $j$  and  $N$  is the number of columns in the test image. Table I shows the mean of the average absolute errors for all the images in each data set along with the standard deviation. The maximum and minimum average absolute errors are also shown for each data set to provide a more clear comparison for all the approaches.

## B. Results and Discussion

As shown in Table I, both edge-less and edge-based approaches using classification outperform the classical edge-based approach of Lie et al. [5] on both the data sets. The edge-based approach [3] performs better than the edge-less [4] approach due to the reasons mentioned earlier. The proposed fusion approach outperforms all other approaches since it benefits both from the presence of edges at a given image

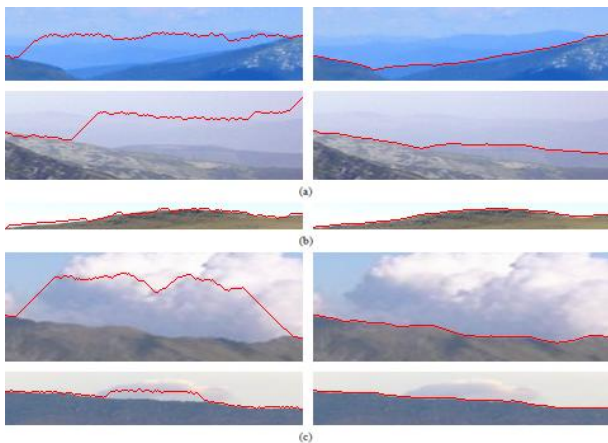


Fig. 5. Detailed segments from Figure 4 : (a) multiple horizons, (b) smoother localization, (c) miss-classifications.

image which provides the horizon-ness for each pixel. Second, edge detection is performed on the query image and the horizon-ness of each pixel is boosted if that pixel happens to lie on an edge. We have considered both Canny edges and MSEE edges in our experiments. Depending on whether Canny edges  $I(x, y)$  or MSEE edges  $E(x, y)$  have been used, equation 10 should be changed as illustrated by equations 11 and 12:

$$\Psi(i, j) = \begin{cases} D(x, y) + b, & \text{if } I(x, y) = 1 \\ D(x, y), & \text{otherwise;} \end{cases} \quad (11)$$

$$\Psi(i, j) = \begin{cases} D(x, y) + b, & \text{if } E(x, y) = 1 \\ D(x, y), & \text{otherwise;} \end{cases} \quad (12)$$

where,  $b$  is a constant added to strengthen the horizon-ness of a pixel. Once the nodal costs have been assigned, the rest of the steps (i.e., link costs and DP) are the same as described earlier. Figure 6 illustrates the steps of the proposed fusion strategy for a sample image. The DCSI 6-(b) is combined with MSEE edges 6-(c) to get the fused DCSI 6-(d) which is then used for finding the shortest path (horizon) by applying DP.



Fig. 6. Fusion based horizon detection: (a) query image, (b) DCSI, (c) MSEE Edges, (d) fused DCSI, (e) detected horizon (red boundary).

TABLE I. AVERAGE ABSOLUTE ERRORS

Approach	Basalt Hills				Web			
	$\mu$	$\sigma$	<i>min</i>	<i>max</i>	$\mu$	$\sigma$	<i>min</i>	<i>max</i>
Lie et al. [5]	5.55	9.46	0.53	49.31	9.15	17.92	0.38	93.02
SIFT+HOG Edges [3]	0.57	1.02	0	3.58	0.87	1.03	0.43	7.05
SIFT+HOG Scores [3]	0.41	0.81	0	3.08	0.97	1.57	0.38	12.19
SVM-mDCSI [4]	1.01	0.29	0.62	1.76	1.28	1.20	0.37	6.21
CNN-mDCSI [4]	0.75	0.23	0.42	1.28	1.41	1.49	0.27	10.79
Fusion: SVM-DCSI+MSEE Edges	0.73	0.32	0.48	2.07	0.85	0.89	0.35	5.05
Fusion: SVM-DCSI+Canny Edges	0.77	0.35	0.48	2.07	0.78	0.76	0.35	4.84

location and its horizon-ness. It should be noted that the Web data set is more challenging compared to the Basalt data set and the fusion strategy outperforms all other approaches on this data set. However, the average error for the fusion approach is higher for the Basalt data set compared to the edge-based approaches but its variance is much lower.

On Web data set, the fusion of the edge-less approach with Canny edges outperform the fusion of the edge-less approach with MSEE edges. This is due to the fact that although MSEE removes many non-horizon edges, it also removes some horizon edges. By visual inspection, it is clear that the difference in performance is minor and one can rely on MSEE edges for speed with very little loss of accuracy.

## VI. CONCLUSION

We have presented a detailed comparison between two recently proposed horizon line detection approaches based on machine learning. The first method uses classification to filter out non-horizon edges while the second method uses machine learning to estimate the horizon-ness of each pixel. Both approaches outperform the classical edge-based approach which suffers from faulty assumptions and edge gaps. It should be noted that our training set is comprised of only 9 images. The trained classifiers generalize very well to challenging examples in the Web data set. To further improve horizon detection, we have proposed a fusion strategy by combining the horizon-ness and edge-ness of each pixel. The proposed fusion strategy outperforms all earlier methods on the challenging Web data set. For future work, we plan to investigate the suitability of various horizon detection approaches for visual geo-localization and rover/UAV localization and navigation.

## ACKNOWLEDGMENT

This work is supported by NASA EPSCoR under Cooperative Agreement No. NNX11AM09A, and in part by NSF PFI.

## REFERENCES

- [1] Y. Hung, C. Su, Y. Chang, J. Chang, H. Tyan: Skyline Localization for Mountain Images. In Proceedings of International Conference on Multimedia and Expo (ICME), 2013
- [2] T. Ahmad, G. Bebis, E. Regentova, A. Nefian: A Machine Learning Approach to Horizon Line Detection using Local Features. In Proceedings of 9th International Symposium on Visual Computing (ISVC), 2013
- [3] T. Ahmad, G. Bebis, E. Regentova, A. Nefian, T. Fong: An Experimental Evaluation of Different Features and Nodal Costs for Horizon Line Detection. In Proceedings of 10th International Symposium on Visual Computing (ISVC), 2014
- [4] T. Ahmad, G. Bebis, M. Nicolescu, A. Nefian, T. Fong: An Edge-less Approach to Horizon Line Detection. In IEEE International Conference on Machine Learning and Applications, 2015 [Submitted]
- [5] W. Lie, T. C.-I. Lin, T. Lin, K.-S. Hung: A robust dynamic programming algorithm to extract skyline in images for navigation. Pattern Recognition Letters, **26**(2): 221–230, 2005
- [6] T. Ojala, M. Pietikainen, T. Maenpaa: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), **24**(7):971–987, 2002
- [7] N. Dalal, B. Triggs: Histograms of Oriented Gradients for Human Detection. In Computer Vision and Pattern Recognition (CVPR), 2005
- [8] D. G. Lowe: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (IJCV), **60**(2):91–110, 2004
- [9] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner: Gradient based learning applied to document recognition. PIEEE, **86**(11):2278–2324, 1998
- [10] C. Cortes, V. Vapnik: Support-vector networks. Machine Learning, **20**(3):273–297, 1995
- [11] A. V. Nefian, X. Bouysounouse, L. Edwards, T. Kim, E. Hand, J. Rhizor, M. Deans, G. Bebis, T. Fong: Planetary Rover Localization within Orbital Maps. In International Conference on Image Processing (ICIP), 2014
- [12] G. C. H. E. d. Croon, B. D. W. Remes, C. D. Wagter, R. Ruijsink: Sky Segmentation Approach to Obstacle Avoidance. In IEEE Aerospace Conference, pp. 1–16, 2011
- [13] S. M. Ettinger, M. C. Nechyba, P. G. Ifju, M. Waszak: Vision-Guided Flight Stability and Control for Micro Air Vehicles. In Proceedings of International Conference on Intelligent Robots and Systems (IEEE/RSJ), 2002
- [14] T. G. McGee, R. Sengupta, K. Hedrick: Obstacle Detection for Small Autonomous Aircraft Using Sky Segmentation. In Proceedings of International Conference on Robotics and Automation (ICRA), 2005
- [15] S. Todorovic, M. C. Nechyba, P. G. Ifju: Sky/Ground Modeling for Autonomous MAV Flight. In Proceedings of International Conference on Robotics and Automation (ICRA), 2003
- [16] S. Fefilyayev, V. Smarodzinava, L. O. Hall, D. B. Goldgof: Horizon Detection Using Machine Learning Techniques. In International Conference on Machine Learning and Applications (ICMLA), pp. 17–21, 2006
- [17] N. S. Boroujeni, S. A. Etemad, A. Whitehead: Robust Horizon Detection using Segmentation for UAV Applications. In IEEE 2012 Ninth Conference on Computer and Robot Vision (CRV), pp. 346–352, 2012
- [18] T. Ahmad, G. Bebis, E. Regentova, A. Nefian, T. Fong: Coupling Dynamic Programming with Machine Learning for Horizon Line Detection. International Journal on Artificial Intelligence Tools (IJAIT), 2015
- [19] B. Kim, J. Shin, H. Nam, J. Kim: Skyline Extraction using a Multistage Edge Filtering. World Academy of Science, Engineering and Technology, 2011
- [20] L. Porzi, S. R. Bulò, P. Valigi, O. Lanz, E. Ricci: Learning Contours for Automatic Annotations of Mountains Pictures on a Smartphone. In ACM/IEEE International Conference on Distributed Smart Cameras, 2014
- [21] A. P. Yazdanpanah, E. E. Regentova, A. K. Mandava, T. Ahmad, G. Bebis: Sky Segmentation by Fusing Clustering with Neural Networks. In Proceedings of 9th International Symposium on Visual Computing (ISVC). 2013
- [22] G. Baatz, O. Saurer, K. Koser, M. Pollefeys: Large Scale Visual Geo-Localization of Images in Mountainous Terrain. In Proceedings of European Conference on Computer Vision, pp. 517–530, 2012
- [23] S. Thurrowgood, D. Soccol, R. J. D. Moore, D. Bland, M. V. Srinivasan: A Vision Based System for Attitude Estimation of UAVs. In Proceedings of IEEE International Conference on Intelligent Robots and Systems, pp. 5725–5730, 2009