Computer Vision and Image Understanding 191 (2020) 102879

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Horizon line detection using supervised learning and edge cues ☆

Touqeer Ahmad [a,*], George Bebis [a], Monica Nicolescu [a], Ara Nefian [b], Terry Fong [b]

[a] Department of Computer Science and Engineering, University of Nevada, Reno, 1664 N. Virginia Street, Reno, 89557, NV, USA
[b] NASA Ames Research Center, Moffett Field, 94035, CA, USA

ARTICLE INFO

Communicated by Nikos Paragios

MSC:
41A05
41A10
65D05
65D17

ABSTRACT

Traditionally, edge detection has been extensively employed as the basic step for the horizon line detection problem. However, generally such methods do not discriminate between edges belonging to horizon boundary and others due to clouds or other natural phenomenon. Additionally, most edge based methods suffer more in the presence of edge gaps. To address these issues, we propose an edge-less horizon line detection approach based on pixel classification, hence not relying on edge information. The key idea is formulating a multi-stage graph using classification maps, instead of edge maps, where each node cost reflects the likelihood of pixel belonging to the horizon boundary. The shortest path is found in the formulated multi-stage graph using dynamic programming which conforms to the detected horizon line. We demonstrate the performance of the proposed approach on two challenging data sets and provide comparisons with two edge-based methods: one relying on edge detection while the other based on edge classification. Overall, the proposed approach achieves comparable performance against carefully crafted edge based formulations. A by-product of the edge-less approach is its capability of associating a confidence level with the found solution, which can be used to confirm the presence or absence of a horizon line in a given image. The method is also capable of dealing with partial horizon line in an image. To further improve the detection performance, we propose a fusion strategy which combines both edge-based and edge-less information. Extensive evaluations, including a publicly available data set, illustrate the superiority of the proposed fusion approach.

1. Introduction

Segmenting a gray scale or color image into sky and non-sky regions is coined as horizon line detection or skyline extraction problem. It is a challenging problem due to non-linear boundaries, background clutter due to clouds, fog, mist etc. and extreme variations in non-sky regions. Depending upon the non-sky regions (which could be water, city buildings, mountains or plain fields etc.) imaged in the picture, skyline or horizon line detection finds many applications. Horizon or skyline detection has been adopted for smooth navigation of unmanned aerial vehicles (UAVs) (Boroujeni et al., 2012; McGee et al., 2005; Thurrowgood et al., 2009; Grelsson et al., 2015; Hou and Li, 2015; Di et al., 2012) and micro air vehicles (MAVs) (d. Croon et al., 2011; Ettinger et al., 2002; Todorovic et al., 2003), augmented reality (Porzi et al., 2014), rover localization (Boukas et al., 2014; Cozman and Krotkov, 1997; Cozman et al., 2000), visual geo-localization and annotation of mountain/desert imagery (Chen et al., 2015; Saurer et al., 2016; Baatz et al., 2012; Liu and Su, 2014; Tzeng et al., 2013; Baboud et al., 2011), port security and ship detection (Fefilatyev et al., 2006; Gershikov et al., 2013; Kruger and Orlov, 2010; Kong et al., 2016), outdoor vehicle localization (Gupta and Brennan, 2008; Ho and Chakravarty, 2014; Dumble and Gibbens, 2012, 2015) and autonomous vehicle navigation (Neto et al., 2011). Previous methods to horizon line detection can be categorized into two major groups; (i) methods modeling sky and non-sky regions using classification (Boroujeni et al., 2012; d. Croon et al., 2011; Ettinger et al., 2002; Fefilatyev et al., 2006; McGee et al., 2005; Todorovic et al., 2003; Verbickas and Whitehead, 2014), and (ii) methods relying on edge detection (Kim et al., 2011; Lie et al., 2005; Shen et al., 2013). Recently, some attempts have been made to combine these two ideas by eliminating non-horizon edges using classification e.g. Hung et al. (2013), Ahmad et al. (2013) and Porzi et al. (2014).

1.1. Horizon line detection review

Most of the earlier horizon detection methods are based on the assumption that the horizon forms a linear boundary; Hough transform was used to find the line parameters subject to some cost function. In Ettinger et al. (2002), authors proposed using the horizon line for flight stability and control of MAVs. Their horizon detection approach is based on the assumptions that the horizon is linear and it segments the

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.cviu.2019.102879.

* Corresponding author.
E-mail address: tahmad@nevada.unr.edu (T. Ahmad).

https://doi.org/10.1016/j.cviu.2019.102879
Received 27 August 2018; Received in revised form 4 September 2019; Accepted 26 November 2019
Available online 6 December 2019
1077-3142/

image into two regions of significantly different appearance (sky and non-sky). Using RGB color information, the sky and ground regions are modeled using Gaussian distributions. Since it was assumed that the sky and ground regions follow a Gaussian distribution, which is not always valid; Todorovic et al. (2003) proposed a general statistical image modeling framework to build prior models for sky and ground. Unlike the work in Ettinger et al. (2002), they found both color and texture to be critical for building priors. They used color (Hue, Intensity) and texture (Complex Wavelet Transform (CWT), magnitude only) to train a Hidden Markov Tree (HMT) model using the expectation maximization (EM) algorithm. The posterior likelihoods for two classes at different scales are fused together and Bayesian segmentation is performed to separate the sky and non-sky regions. McGee et al. (2005) used sky segmentation as an obstacle detection tool for small scale UAVs. They trained a Support Vector Machine (SVM) classifier based on YCbCr color information to classify pixels into sky and non-sky regions. Morphological erosion and dilation were applied on the resultant binary image to rectify misclassifications. Next, they used Hough Transform on the border pixels to cast votes for candidate linear horizon boundaries. The approach of Fefilatyev et al. (2006) is also based on the horizon boundary being linear; it uses color and texture features such as mean intensity, entropy, smoothness, uniformity etc. to train an SVM, a J48 and a naive Bayes classifier. Their experiments are limited to two sets of ten images each and their method fails to detect good linear horizons for two out of ten images due to reflection of water and presence of fog respectively. In Liu and Su (2014), authors have proposed a sensor fusion approach to estimate the horizon line using a textured Digital Elevation Map (DEM), an airport model, GPS, AHRS and vision sensors. Their objective was to estimate an accurate linear horizon boundary from an aircraft in low visibility conditions; their approach does not generalize to non-linear horizons.

d. Croon et al. (2011) extended the features used in Fefilatyev et al. (2006), Ettinger et al. (2002), Todorovic et al. (2003) and McGee et al. (2005) by including corner-ness, grayness and Fisher Discriminant features to train shallow decision trees. Their approach was tested in the context of MAVs for obstacle avoidance and is able to detect non-linear horizon boundaries. The fusion-based approach of Yazdanpanah et al. (2013, 2015) combines the output of a Neural Network (NN) with K-means clustering and is based on the same texture features such as in d. Croon et al. (2011) and Fefilatyev et al. (2006). Their system is based on various heuristics and parameter settings that might not generalize well to different data sets. In Boroujeni et al. (2012), they also rely on clustering for horizon line detection. Their method is based on the assumption that a dominant light field exists between sky and non-sky regions right above the horizon. They have investigated K-means and intensity based clustering to find this light field in various images. In principle, the assumption about the presence of a light field should be validated under different seasonal conditions or geographical locations and the data set being used in Boroujeni et al. (2012) is not general enough to justify such a strong assumption. Thurrowgood et al. (2009) used horizon detection for UAV attitude estimation. In their approach, a projection onto a single line in the RGB color space is found by minimizing the overlap of the sky and non-sky classes. This is somewhat similar to the Fisher Discriminant used by Ettinger et al. (2002) yet computationally less expensive. In Neto et al. (2011), authors proposed a robust horizon line detection algorithm using Otsu segmentation and Hough transform for real-time autonomous navigation. Braun and Singhof (2015) proposed a seed growing algorithm where the top row of the image is assumed to belong to the sky. This sky region is then grown based on the assumption that the distance of the current pixel's brightness from a local brightness mean is smaller than a fixed percentage of the global standard deviation. This method is heavily dependent on the percentage factor. Moreover, the results reported are based on a very small image set (18 images) from the Switzerland data set (Baatz et al., 2012) and no quantitative comparison with the ground truth is provided.

Dusha et al. (2007) used horizon line information with optical flow for attitude estimation of fixed-wing air-crafts. In their approach, edge detection is performed separately in each smoothed color channel. The detected edges from each channel were combined in a single map and the horizon line was found using Hough voting. Shen et al. (2013) proposed an edge-based hierarchical approach for horizon line detection where coarse-level detection was performed first, followed by fine-level adjustments. They successively performed Canny edge detection and Hough voting on a low pass image to find the strongest lines. Using the five highest peaks in the Hough space, the best line was chosen based on average edge strength. The straight line chosen is refined, essentially becoming non-linear, using edge position and strength information. The results reported are based on synthetic images generated from Google maps which do not reflect if the idea generalizes to real images. Gershikov (2014) provides a comparison between gray scale and color based horizon detection methods which rely on edge detection and Hough transform. The most prominent method belonging to the second category is that of Lie et al. (2005) where horizon line detection is formulated as a graph search problem. Their approach relies on edge detection and assumes a consistent edge boundary between sky and non-sky regions. The detected edge map is represented as a multi-stage graph where each column of the image becomes a stage in the graph and each edge pixel becomes a vertex. The shortest path, extending from the left-most column to the right-most column, is then found using DP. It should be mentioned that the assumption that the horizon boundary is a consistent edge boundary is rarely true in practice due to environmental conditions (e.g., clouds, fog, mist) and edge gaps. To address the issue of gaps, Lie et al. (2005) have proposed a gap-filling step which highly depends on the choice of certain parameters. Moreover, they assume that the edges in the upper half of the image belong to the horizon boundary and hence introduce a bias to find the horizon solution in that region. However, the edges in this region may very well be due to the presence of clouds. Shen and Wang (2013) proposed a simple gradient magnitude based sky segmentation approach. By redefining the energy function proposed by Ettinger et al. (2002), their approach is applicable to general curves instead of a linear boundary.

Ahmad et al. (2013) and Hung et al. (2013) have independently extended the approach of Lie et al. (2005) by introducing a classification step to remove non-horizon edges. This is performed by training a classifier using features from horizon and non-horizon key-points. The multi-stage graph is then built using only horizon classified edge pixels. Ahmad et al. (2013) used SIFT descriptors (Lowe, 2004) around the key-points and an SVM classifier whereas Hung et al. (2013) used an SVM classifier with color information as well as the variance above and below key-points. In addition to using classification; Ahmad et al. (2013) apply a preprocessing step to remove unstable edges by keeping only those edges which survive a large number of Canny thresholds. In a related approach, Porzi et al. (2014) reduce the number of edges by thresholding the output of the Sobel detector; a Random Ferns classifier was also used to classify the remaining edge pixels into contour and non-contour edges. Ahmad et al. (2013) extended their work in Ahmad et al. (2014, 2015c) by investigating various textural features and nodal costs. Verbickas and Whitehead (2014) compared Convolution Neural Networks (CNN) with SVMs and Decision Trees using a similar set of features to those in Fefilatyev et al. (2006); using an extended data set, CNNs demonstrated superior performance. More recently, some attempts have also been made to adapt deep neural nets for skyline segmentation e.g. Porzi et al. (2016).

With the massive availability of geo-tagged imagery and increased computational power, geo-localization or geolocation has captured a lot of attention from researchers in computer vision and image retrieval communities. Significant progress has been made in urban environments with stable man-made structures and geo-referenced street imagery of frequently visited tourist attractions (Hays and Efros, 2008; Zheng et al., 2009; Zamir and Shah, 2010). Recently some attempts

have been made towards geo-localization of natural/mountain scenes which is more challenging due to changed vegetation, lighting and seasonal changes and lack of geo-tagged imagery. Typical approaches for mountain/natural geo-localization rely on mountain peaks and valley information, visible skylines, ridges or combinations of all three (Stein and Medioni, 1995; Chippendale et al., 2008; Baboud et al., 2011; Baatz et al., 2012; Tzeng et al., 2013; Porzi et al., 2014; Fedorov et al., 2014; Chen et al., 2015; Ozcanli et al., 2016; Saurer et al., 2016). Skyline has been established to be a robust natural feature for mountainous images which can be matched with the synthetic skylines generated from publicly available terrain maps e.g. Digital Elevation Models (DEMs). Hence, the very first step in the geolocation pipeline for mountainous regions is to find the skyline in the given query image. Most of the solutions for mountainous geo-localization rely on **user-in-the-loop** methods for skyline extraction where a user is required to mark/correct portion of the skyline (Baatz et al., 2012; Tzeng et al., 2013; Chen et al., 2015; Saurer et al., 2016). Making a truly autonomous horizon/skyline detector would definitely advance this research dimension.

1.2. Paper outline

Some components of this work have previously been presented in Ahmad et al. (2015b,a). This work provides a more detailed analysis and evaluation of different skyline detection methods and their comparison. However, this is not a review paper as we propose a new skyline detection method based on a fusion strategy and further propose to use our framework for couple of sub-problems. The major components of the paper are summarized below:

1. In this work, an edge-less horizon line detection approach based on Dynamic Programming (DP) is proposed; the key idea is building the multi-stage graph from a dense classification map instead of an edge map. Typically edge maps contain gaps and do not distinguish between horizon and other edges. To address these issues, we propose classifying each image pixel (or pixels within a region of interest) as belonging to the horizon or not. This is performed by training a classifier using both horizon and non-horizon pixels using a small set of training images. The resulting classification map contains no gaps and provides information about the likelihood of each pixel belonging to the horizon line; we refer to the classification map as Dense Classifier Score Image (DCSI). The horizon line can then be extracted by finding a path that maximizes the sum of classification scores using DP. We have experimented with SVMs (Cortes and Vapnik, 1995) and CNNs (LeCun et al., 1998) for classification, using normalized pixel intensities as features.
2. We provide a detailed comparison of our proposed edge-less approach with an existing edge-based approach to better understand the strengths and weaknesses of each method. Using two challenging data sets, we evaluate the performance of each approach and identify specific cases where one outperforms the other. Finally, we propose a fusion strategy which boosts the performance of the edge-less approach using edge information. The fusion approach, which has been tested on an additional challenging data set (Baatz et al., 2012), outperforms each of the methods alone.
3. Next, we demonstrate the capability of our formulation to detect absence of horizon boundary and detection of partial horizon lines. In many applications where the horizon line is used for rover/robot localization and navigation, it is important not only to detect the horizon line but also to report a confidence measure of the detection. This is useful in many cases, for example, when the horizon line in not visible in the image. A by-product of the proposed edge-less/fusion approach is a confidence measure corresponding to the normalized sum of classification scores along the path found by dynamic programming. Using a Bayesian approach, we can determine whether the path found corresponds to the horizon line or some other irrelevant path. This could also be used as a validation step where the method misses the actual horizon line and finds another solution instead. The statistical measure along the found path can be used to reject such faulty detections. Moreover, we demonstrate how the proposed approach can be adapted to handle partially visible horizon lines. This is quite useful since localization is still feasible, at least from a theoretical point of view, using partially visible horizon lines. To the best of our knowledge, this is the first study to address these issues of non-visible, partially visible horizon lines and verification of a detected horizon.
4. To measure the performance of different horizon line detectors, we adopt a more accurate measure compared to the ones reported in the literature. All performance comparisons reported in this study are based on the absolute average error between the solution found and ground truth. Previously, detection results were reported in the form of a percentage (e.g., what percentage of the horizon line was detected) which does not provide a true quantitative evaluation of horizon line detection methods. The percentage metric does not take into account how far the detected skyline pixels are from the ground truth skyline pixels. This is why we opted to use absolute average error for presenting our detection results. It should be noted that Hung et al. (2013) also reported the absolute average error. Our results indicate sub-pixel detection error for two of our data sets using the proposed fusion approach. Although not compared explicitly on the same data set, Hung et al. (2013) have reported an average error of more than 3 pixels using a large training set (333 images). In contrast, our training set contained only 9 images which were captured under the same imaging conditions and geographical location. It should be noted that the images used in their experiments were 240 pixel wide whereas the images used in our experiments were 1388 pixels wide, resulting in horizon profiles with much more variation.

The rest of paper is organized as follows. In Section 2, we review the edge-based approach based on the methods of Lie et al. (2005) and Ahmad et al. (2013, 2014, 2015c). Section 3 describes the edge-less approach. Section 4 provides a detailed comparison of edge-less and edge-based methods. In Section 5, we discuss the fusion of edge-based and edge-less information and provide experimental results to illustrate its performance. Sections 6 and 7 describe how the problems of detecting the absence of horizon or a partial horizon line can be addressed as a by-product of our framework. Finally, Section 8 presents our conclusions and provides directions for future research.

2. Edge-based horizon detection

In this section, we review the edge-based approach and its extensions for horizon line detection; these methods form a basis of comparison with the proposed edge-less approach.

2.1. Lie et al. (2005)

Lie et al. (2005) have formulated the problem of horizon detection as a multi-stage graph search problem. Given an $M \times N$ image, edge detection is performed first to compute a binary edge map $E(i,j)$ where 1 implies the presence of an edge pixel and 0 a non-edge pixel.

$$E(i,j) = \begin{cases} 1, & \text{if } (i,j) \text{ is an edge pixel.} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

The edge map $E(i,j)$ is used to build an $M \times N$ multi-stage graph $G(V, E, \Psi, \Phi)$ where each pixel in the map corresponds to a graph vertex v_{ij} in the graph; a low cost l is associated with edge pixels

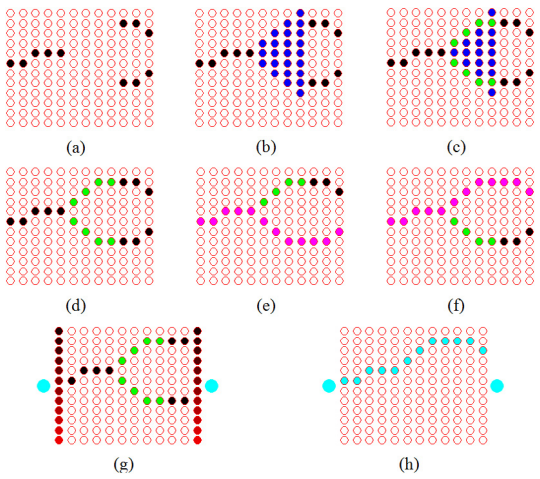

Fig. 1. Illustration of the horizon detection steps based on the method of Lie et al. (2005). Section 2.1 provides the details. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(vertices) while a very high cost (i.e., ∞) is associated with non-edge pixels (vertices) as shown below:

$$\Psi(i,j) = \begin{cases} l, & \text{if } E(i,j) = 1. \\ \infty, & \text{otherwise.} \end{cases} \tag{2}$$

$\Psi(i,j)$ is the cost associated with vertex i in stage j (i.e., v_{ij}). It should be noted that use of ∞ reflects a node with a high numeric cost. The graph can be visualized as an N stage graph where each stage contains M vertices. To deal with edge gaps, they have proposed a gap filling process. Given a node i in stage j, its neighborhood in the next stage $j+1$ is defined by a δ 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:

$$\Phi(i,k,j) = \begin{cases} |i-k|, & \text{if } E(i,j) = E(k,j+1) = 1 \\ & \text{and } |i-k| \leq \delta \\ \infty, & \text{otherwise.} \end{cases} \tag{3}$$

If a node i in stage j cannot be connected to any node in stage $j+1$ within δ distance, a search window is defined using two parameters: δ and tolerance-of-gap (tog). If an edge node k is found within the search window, gap filling is performed by introducing dummy nodes between node i in stage j and node k within the search window j+tog. A high cost is associated with dummy nodes introduced by the gap filling step. It should be noted that ∞ in the above equation emphasizes that a node can only b connected to nodes (in next stage) which have an absolute vertical distance less than or equal to the set threshold δ.

Once the gaps are filled with high cost dummy nodes, the cost of the nodes in stages 1 and N is increased based on their vertical position according to the equation below:

$$\Psi(i,j) = \begin{cases} (i+1)^2, & \text{if } j = 1 \text{ or } j = N \\ \Psi(i,j), & \text{otherwise.} \end{cases} \tag{4}$$

This enforces the assumption that the edges present in the upper half of the image belong to horizon and hence biases the DP solution towards shortest paths present in the upper half. Next, two nodes, 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 while the t node is connected with all the nodes in stage N. A shortest path is then found extending from node s to t using DP which conforms to the detected horizon boundary.

Fig. 1 illustrates the steps of Lie et al. (2005) for a sample image. An edge map is shown in Fig. 1-(a) where black and white circles represent edge and non-edge pixels respectively. A search window (highlighted by blue circles) is shown in Fig. 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 Fig. 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 Fig. 1–(g). Two nodes s and t (cyan) are then introduced, as described above, and DP is applied on this graph. As shown in Fig. 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. It should be noted that any gap bigger than the search window defined by j+tog would not be filled and hence there might still be gaps depending upon the specific parameter values used.

Generally, the edge-based methods do not make any distinction between edges that belong to the horizon or those belonging to some other artifacts e.g. clouds. During gap-filling, such methods can connect these two types of edges — which can result into solutions where some part belongs to the horizon and other part belongs to a cloud. Section 4.2 and Fig. 7 provides explicit examples of failures of post-processing where clouds have been shown to be confused as part of solution in Lie et al. (2005) method.

2.2. Ahmad et al. (2013, 2014, 2015c)

In Ahmad et al. (2013), authors used Maximally Stable Extremal Edges (MSEEs) and classification to reduce the number of non-horizon edges before applying DP. In their analysis (Ahmad et al., 2014), they have investigated various texture features (e.g., SIFT Lowe, 2004, LBP Ojala et al., 2002, HOG Dalal and Triggs, 2005 and their combinations) for training 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 Ahmad et al. (2014, 2015c) rather on the various texture features. The interested reader should consult (Ahmad et al., 2013, 2014, 2015c; Porzi et al., 2014). In their first formulation, the nodal costs were defined using gradient information. This is a simple extension of the binary edge map approach of Lie et al. (2005) and does not require training a classifier. In this case, DP was used to maximize the sum of gradient magnitudes along a path. To ensure good continuity, they enforce the constraint that the difference between gradient magnitudes of adjacent pixels is minimized. The gradient magnitude at each pixel of the input image $I(i,j)$ is computed as follows:

$$\nabla(i,j) = \Gamma[I(i,j)] \tag{5}$$

where, Γ is a function which takes a gray scale image I as input and returns the gradient magnitude image ∇. Next, the difference of the gradient magnitude image $d\nabla(i,j)$ is computed. Since a node i in stage j can be connected to as many nodes in stage $j+1$, as defined by the δ parameter, several gradient magnitude difference images need to be generated. The equation below shows the gradient magnitude difference image assuming that the nodes are connected at the same level between stages:

$$d\nabla(i,j) = |\nabla(i,j) - \nabla(i,j+1)| \tag{6}$$

The normalized (i.e., between 0 and 1) gradient magnitude and gradient difference images are combined through a weighted average as shown in the equation below:

$$G_r(i,j) = w_1 * d\nabla(i,j) + (1-w_1) * (1-\nabla(i,j)) \tag{7}$$

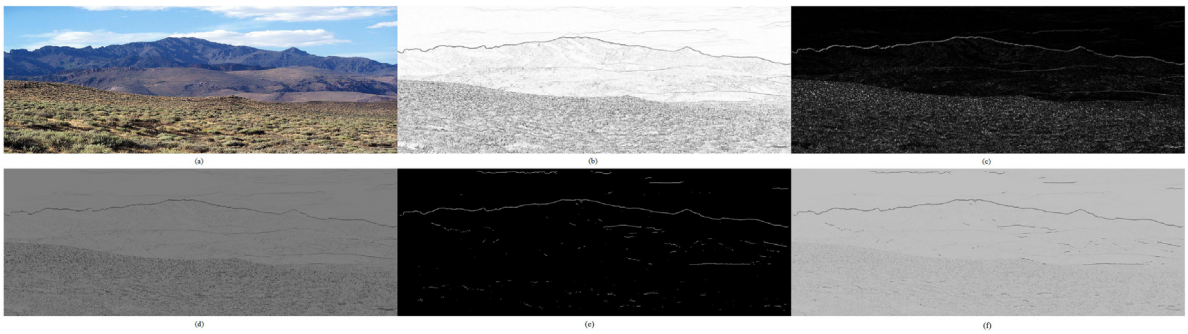

Fig. 2. Visualization of various types of information used in the formulation of different nodal costs (Ahmad et al., 2015c): (a) original image, (b) gradient magnitude image, (c) difference of gradient magnitude image, (d) weighted average G_r, (e) classification score image and (f) classification score image + weighted average G_r.

where w_1 is the weight parameter; in our experiments, we have set $w_1 = 0.5$. Note that since DP solves a minimization problem, we have used the difference $(1 - \nabla(i,j))$ in the equation above. The weighted average G_r was used as the nodal cost:

$$\Psi(i,j) = G_r(i,j) \tag{8}$$

The link costs may be initialized using Eq. (3). Next, Ahmad et al. (2013, 2015c) considered MSEEs and classification as a way to filter out non-horizon edges. Given a query image, they first apply Canny edge detection using various thresholds and keep only those edges which remain stable over a range of thresholds. The idea was inspired by MSER (Matas et al., 2002) and is coined as MSEE. This step reduces the number of non-horizon edges since horizon edges tend to be stronger. The MSEE map $E_m(i,j)$ is further refined by classifying each MSEE pixel (i,j) as horizon or non-horizon:

$$C(i,j) = \begin{cases} 1, & \text{if } (i,j) \text{ pixel is classified as horizon.} \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

As mentioned earlier, Ahmad et al. (2013, 2014, 2015c) experimented with various texture features and their combinations for classification; among them, the SIFT–HOG combination yielded the lowest false negative rate. The edge map $E_+(i,j)$, comprising of the horizon classified MSEE edges, was used to define the nodal costs in the context of DP. Specifically, the edge map comprising of positively classified MSEE edge points can be expressed as follows:

$$E_+(i,j) = \begin{cases} 1, & \text{if } E_m(i,j) = 1 \text{ and } C(i,j) = 1. \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

One way to define the nodal costs is by using the binary costs based on $E_+(i,j)$; this changes Eqs. (2) and (3) as follows:

$$\Psi(i,j) = \begin{cases} l, & \text{if } E_+(i,j) = 1. \\ \infty, & \text{otherwise.} \end{cases} \tag{11}$$

$$\Phi(i,k,j) = \begin{cases} |i-k|, & \text{if } E_+(i,j) = E_+(k,j+1) = 1 \\ & \text{and } |i-k| \le \delta \\ \infty, & \text{otherwise.} \end{cases} \tag{12}$$

An alternative way to define the nodal costs is by using the actual classification scores. This is motivated by the fact that classification scores provide some confidence about an edge pixel belonging to the horizon or not. First, the raw classification scores are normalized between 0 and 1. Then, the nodal costs are initialized by the normalized classification scores $S(i,j)$ instead of setting all positively classified edges to a fixed cost. Eq. (11) is modified accordingly to reflect this idea where $S(i,j)$ is the normalized classification score. It should be noted that since DP solves a minimization problem, the classification score values need to be reversed (i.e., the smaller the classification score value is the more likely is that the pixel is a horizon pixel).

$$\Psi(i,j) = \begin{cases} S(i,j), & \text{if } E_+(i,j) = 1. \\ \infty, & \text{otherwise.} \end{cases} \tag{13}$$

In their last formulation, they have combined classification scores with gradient information. By combining Eqs. (7) and (13), the nodal costs can be defined as follows:

$$\Psi(i,j) = w_2 * S(i,j) + (1 - w_2) * G_r(i,j), \tag{14}$$

where, w_2 is a weight parameter; we have set $w_2 = 0.5$ in our experiments. Fig. 2 shows visualizations of the various images used to initialize the nodal costs in DP (Ahmad et al., 2015c).

3. Edge-less horizon detection

Using classification to assess the horizon-ness of an edge pixel has proven very useful by discarding the edges belonging to non-skyline regions. However, discarding non-horizon edges does not necessarily address the issue of edge gaps. To deal with this issue, we propose classifying all pixels (or pixel within a region of interest) as horizon or non-horizon pixels. We refer to the classification map obtained as Dense Classifier Score Image (DCSI). The DCSI is used to form an $M \times N$ multi-stage graph where DP is applied to find the horizon line. Specifically, once we have introduced the source/destination nodes s/t and decided on the value of δ, any shortest path finding algorithm can be used to find the path that maximizes the sum of classification scores. We will later show that the number of nodes per stage can be significantly reduced by only considering the pixels with the m highest classification scores where m is a parameter; we refer to this reduced map as mDCSI map. Using fewer nodes per stage does not affect accuracy while it speeds up computations considerably. Fig. 3 illustrates the main steps of the proposed approach. The proposed approach does not rely on edge detection, therefore, it does not require performing gap filling or introducing dummy nodes. Moreover, we do not force the nodes in stages 1 and N to be associated with their vertical position since the assumption of the horizon line being present in the upper half of the image could be violated as demonstrated in Fig. 1. Our only assumption is that the horizon line extends from left to right.

3.1. Pixel classification

For classification, we have experimented with two classifiers: SVM (Cortes and Vapnik, 1995) and a CNN (LeCun et al., 1998). Each classifier is trained using horizon and non-horizon image patches from a set of training images where the horizon line has been extracted manually (ground truth). Specifically, for each training image, we select N points uniformly from the ground truth; an equal number of points is randomly selected from non-horizon locations. We take a 16×16 image patch around each sampled point and normalize the pixel intensities between -1 and 1; the resulted 256-D vector is used for training the classifiers. For the CNN classifier, we use an architecture comprising of 2 Convolution(C)–Sub-sample(S) layers followed by fully connected layers; in principle similar to LeNet (LeCun et al., 1998). The first C–S layer is comprised of 4 levels with a convolution(C) mask of 5×5 and

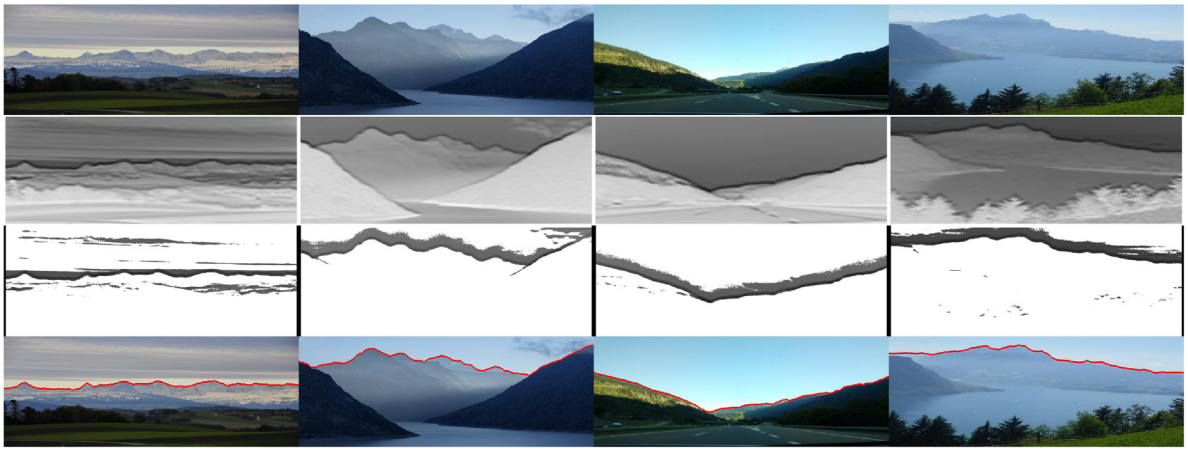

Fig. 3. Main steps of the edge-less horizon line detection approach. Sample test images from the Switzerland data set (Baatz et al., 2012) (row1), respective DCSIs (row2), mDCSIs (row3), and detected horizon lines (row4, highlighted in red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

a sub-sampling(S) mask of 2×2. The second C–S layer is comprised of 8 levels with a C mask of 3×3 and an S mask of 2×2. We have only used 9 images for training the classifiers with 343 positive (horizon) and 343 negative (non-horizon) examples extracted from each image (about 6000 image patches). It should be mentioned that the number of training examples and key points were the same with those used to train the SVM classifier for the edge-based approach described in Section 2.2. However, unlike (Ahmad et al., 2013, 2014, 2015c), we use normalized pixel intensities instead of local texture features; since CNN is being compared.

We should note that the CNN network considered in our work is rather small. This is due to the fact that we are focused here to provide a general framework for skyline detection problem and addressing the underlying issues and not much concerned with comparative performance of CNN against SVM which has previously been explored in Verbickas and Whitehead (2014). By including both classifiers, we want to highlight that we have proposed a general framework and the classification model being used is an open choice depending on applications. Since, our network is small, we have used a fixed learning rate and Xavier weights for initialization.

3.2. Dense Classifier Score Image (DCSI)

Once the classifiers have been trained, the DCSI can be generated for a given test image. For each pixel location (i, j) in the test image, a 16×16 patch of pixel intensities around that pixel is extracted. The normalized intensities are then used to form a 256-D vector, which is fed to the classifier. The classification score is then associated with that pixel location (i, j). Classification scores are normalized in the interval [0, 1]; the resultant scores form the DCSI which is denoted as $S(i, j)$. In essence, $S(i, j)$ can be interpreted as a probability map which reflects the likelihood of a pixel belonging to the horizon line; similar to semantic segmentation. This dense classifier score image can directly be used to initialize the nodal costs under the DP framework as shown in Eq. (15). Fig. 3 shows the DCSIs for various sample images.

$$\Psi(i, j) = S(i, j) \tag{15}$$

The difference between Eqs. (13) and (15) should be noted. In Eq. (13), the classification score is used to initialize only those nodes which are MSEE edges and have been positively classified as horizon edges whereas in Eq. (15), all the nodes have been initialized with the normalized classification scores and no edge information has been used. Also in the former case SIFT–HOG features are extracted around (i, j) location while in the lateral case normalized pixel intensities are used.

3.3. Reduced Dense Classifier Score Image (mDCSI)

Although the full DCSI can be used to initialize the nodal costs as in Eq. (15), we have found that keeping only the m highest classification scores in each column does not compromise accuracy while reducing computations. This is because the highest classification scores are typically concentrated within a small band around the horizon line. We refer to the reduced DCSI as $mDCSI$. The multi-stage graph corresponding to the mDCSI contains fewer vertices; as a result, fewer paths need to be considered when searching for the shortest path which results in considerable speedups. In our experiments, we have found that by keeping the highest 50 classification scores yields accurate horizon line detections.

$$S_+(i, j) = \begin{cases} S(i, j), & \text{if among m highest scores in col. j} \\ c, & \text{otherwise} \end{cases} \tag{16}$$

where, S_+ and S correspond to mDCSI and DCSI respectively. If the ith pixel (node) in column (stage) j is among the m highest classification scores for column j, the classification score from $S(i, j)$ is used; otherwise, the score is set to a very low score c. Fig. 3 shows examples of the respective mDCSIs.

3.4. Nodal and link costs

In the approaches described earlier (Hung et al., 2013; Ahmad et al., 2013, 2014, 2015c; Lie et al., 2005), the edge map is used to form the multi-stage graph and gap filling is an essential step in extracting the horizon line. The proposed edge-less approach does not require any gap filling since it does not rely on edge maps. The mDCSI is used to create an $M \times N$ graph $G(V, E, \Psi, \Phi)$ with node costs initialized using $S_+(i, j)$:

$$\Psi(i, j) = S_+(i, j) \tag{17}$$

Since the resulted graph is a dense graph, each node i in stage (column) j is connected to three nodes i, $i-1$ and $i+1$ in stage(column) $j+1$ (i.e., $\delta = 1$). The link costs are initialized accordingly, by adjusting Eq. (3),

$$\Phi(i, k, j) = \begin{cases} |i - k|, & \text{if } |i - k| \leq \delta \\ \infty, & \text{otherwise.} \end{cases} \tag{18}$$

Since, the horizon line might not always appear in the upper half of the image, we do not initialize the nodes in stages 1 and N proportional to their vertical position. Two dummy nodes, s and t, are introduced to the left of stage 1 and to the right of stage N respectively as described in Section 2.1. The edge weights from s to every node in stage 1 and from every node in stage N to node t are set to zero. Fig. 4 shows sample images from two data sets and the detected horizon lines found by the proposed approach overlaid in red.

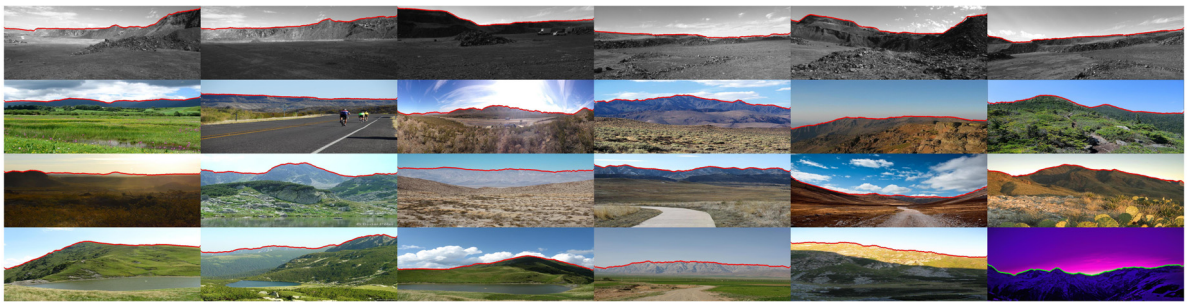

Fig. 4. Sample results illustrating our horizon line detection approach: Basalt Hills data set (row1) and Web data set (row 2 through 4). Detected horizon lines are highlighted in red/green. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. Comparison of edge-less and edge-based approaches

4.1. Data sets and evaluation metric

To evaluate the performance of the edge-less approach, we have experimented with two different data sets: the Basalt Hills data set (Nefian et al., 2014) and the Web data set (Ahmad et al., 2015c). The Basalt Hills data set is a subset of a larger data set which was generated by placing cameras on an autonomous robot navigating through Basalt Hills (Nefian et al., 2014). We have chosen 45 images from this data set with considerable viewpoint and scene changes. The Web data set consists of 80 mountainous images that have been randomly collected from the web. This data set is quite challenging as it includes viewpoint, geographical and seasonal variations. For training, we used only 9 images from the Basalt Hills data set. In particular, we used 343 positive and 343 negative key points from horizon and non-horizon locations. The same key points and experimental settings were also used in Ahmad et al. (2015c). A 16 × 16 block was used around the key points and the normalized intensities [−1,1] were used as features since the CNN classifier operates on image intensities directly; this is in contrast to Ahmad et al. (2015c) where SIFT–HOG features were used to train an SVM classifier assuming 16 × 16 patches. The resolution of all images in our data sets is 519 × 1388. Sampling in this manner results into about 6k image patches.

To quantitatively evaluate the performance of the proposed approach, we have manually extracted the horizon line (ground truth) in all the images of our data sets (Basalt & Web). The detected and true horizon lines are then compared by calculating a pixel-wise absolute distance S between them. First, we compute the absolute distance between the detected and ground truth pixels in each column; the total distance is then computed by summing up the errors over the columns. The resultant distance is normalized by the number of columns, yielding the average error between the detected and true horizon lines. In our experiments, we allow each node i in stage j to be connected to nodes $i-1$, i and $i+1$ in stage $j+1$. Since nodes in a particular stage are not allowed to be connected to other 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 horizon lines:

$$S = \frac{1}{N}\sum_{j=1}^{N} |P_{d(j)} - P_{g(j)}| \tag{19}$$

where $P_{d(j)}$ and $P_{g(j)}$ are the positions (rows) of the detected and true horizon pixels in column j and N is the number of columns in the test image. Fig. 4 shows some representative results of our horizon detection approach using images from the Basalt Hills and Web data sets. Table 1 shows the average (μ) and standard deviation (σ) of the absolute error for all the images in our data sets, both for the SVM and CNN classifiers. For comparison purposes, we also provide results based on the method of Lie et al. (2005) and Ahmad et al. (2015c) formulation based on only gradient information.

Table 1
Average absolute errors using edge-based (traditional, without classification) and edge-less approaches.

Approach	Basalt Hills				Web			
	μ	σ	min	max	μ	σ	min	max
Lie et al. (2005)	5.55	9.46	0.53	49.31	9.15	17.92	0.38	93.02
G_r (Ahmad et al., 2014, 2015c)	3.99	6.35	0.18	31.33	11.86	26.81	0.15	121.48
SVM-mDCSI	1.01	0.29	0.62	1.76	1.28	1.20	0.37	6.21
CNN-mDCSI	0.75	0.23	0.42	1.28	1.41	1.49	0.27	10.79

It should be noted that while Intersection over Union (IoU) is conventionally opted for segmentation methods, we have chosen a different metric to measure the performance of different skyline detection methods. This is because metrics like IoU could be misleading for our problem. Since, we do not want to measure the number of misclassified sky and non-sky pixels but rather want to measure how close is a detected skyline to the ground truth skyline. That is why, we have chosen average absolute vertical pixel distance which has also been previously used by Ahmad et al. (2014, 2015c) and Hung et al. (2013). Our chosen metric provides a fine granularity to measure the pixel-wise distance compared to IoU. For example, in Fig. 6 our chosen metric is more descriptive (i.e., solution is off by an average of five pixels in the top image) whereas IoU would focus on measuring misclassified pixels and could result in change of digit at or after third decimal place.

4.2. Discussion

Our experimental results (Table 1) illustrate that the proposed approach outperforms the traditional approach of Lie et al. (2005) based on edge maps and Ahmad et al. (2015c) based on gradient information i.e. without any classification. Specifically, both the average error and standard deviation of the traditional approach are much higher than the proposed approach based on SVM or CNN classifiers. To better understand the performance of the traditional approach, we have identified specific examples where it fails to detect the true horizon line or it misses parts of it (e.g., horizon edges might not be strong enough or stronger edges might exist close to the horizon line due to various environmental effects such as clouds, mist, fog etc.). Although (Lie et al., 2005) have proposed a gap filling approach by introducing dummy nodes with high costs, this does not always work well, for example, when gaps are long and edges from clouds are close to the horizon line. In these cases, it is likely that the DP approach might find a low cost path by taking an alternative path. Fig. 5 (row 1) shows two examples where the method of Lie et al. has failed to find a good solution due to edge gaps and the presence of edges due to clouds; the proposed method was able to find the true horizon line with high accuracy in both cases (row 2). Fig. 6 shows detailed sub-images of the left column images of Fig. 5 for better visualization.

Another reason affecting the performance of Lie et al. is the underlying assumption about the edges in upper half of the image being

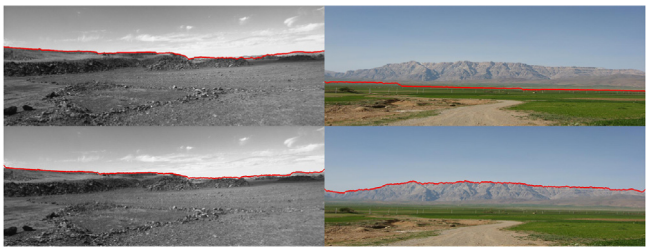

Fig. 5. Examples illustrating: [row 1] missing the horizon line or parts of it due to edge gaps (Lie et al.), and [row 2] detecting the true horizon line using our approach (SVM).

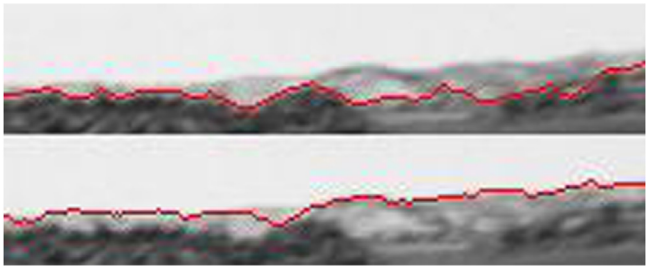

Fig. 6. Detailed sub-images of the left column images of Fig. 5.

Fig. 7. Examples illustrating: [row 1] missing parts of the horizon line due to the assumption that the horizon line is close to the top of the image (Lie et al.), and [row 2] detecting the true horizon line using the proposed approach (SVM-mDCSI).

due to horizon boundary. When clouds are present in an image, this assumption might cause a portion of the true horizon to be missed if the true horizon line is below the clouds. Fig. 7 shows some examples where the approach of Lie et al. has found solutions consisting of both horizon line segments as well as cloud edge segments. Our approach, on the other hand, was able to find the correct solution for these cases as it does not make such assumptions.

Comparing the two classifiers used in our experiments, the CNN classifier outperforms the SVM classifier on the Basalt Hills data set while the SVM classifier outperforms the CNN classifier on the Web data set. This indicates that the features found by the CNN classifier might not generalize well to different data sets. Fig. 8 shows some representative DCSI results using the SVM and CNN classifiers. It is worth noting that the CNN classifier provides a crispier DCSI, having a narrower band around the true horizon line as compared to the DCSI produced by SVM while both trained on the same sized image patches. We would like to highlight that a deeper and well-optimized CNN network probably will outperform SVM by a bigger margin. However, our focus here is to demonstrate that the proposed skyline detection framework is general and different classifier can be used suitable for different applications.

In an effort to better understand why the edge-less approach sometimes finds poor solutions, we have identified two main reasons. The first reason is due to not allowing nodes in some stage to connect with nodes in the same stage but only with nodes in the next stage. In the

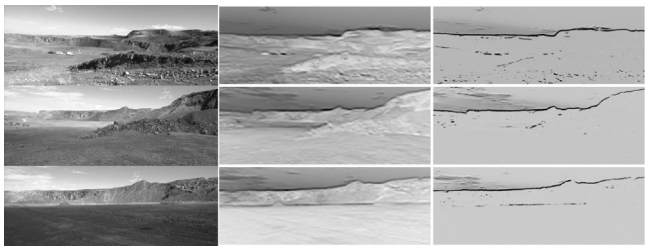

Fig. 8. Test images (column1), corresponding SVM-DCSIs (column2) and CNN-DCSIs (column3).

Fig. 9. [row 1] effect of not allowing node connections within the same stage; [row 2] solution obtained by allowing node connections within the same stage.

multi-stage graph formulation of Lie et al. a node in stage j is only allowed to be connected with nodes in stage j+1; this, however, is problematic when the horizon line has high slope (i.e., steep peaks); Fig. 9 (row 1) shows an example (zoomed image segment). This issue can be easily rectified by allowing nodes in some stage to be connected both with nodes in the same and next stages. Fig. 9 (row 2) shows the solution obtained by allowing connections between nodes in the same stage. Allowing connections within the same stage, however, increases time requirements.

The second most important reason affecting the performance of the proposed approach is due to using a very small set of training images (i.e., only 9 images from the same data set). Fig. 10 shows some examples where the proposed method has failed to find good solutions. This issue can be addressed by increasing the size of the training set and making it more versatile. Careful analysis of our results on the Web data set shows that the proposed approach failed to find a good solution in 9 out of the 80 images due to using a small training set; where a good solution is defined as to have the average absolute error below 1.5 pixels. Removing these images from the data set improves the average error of our approach using the SVM classifier from 1.2854 to 0.9227 while the variance is reduced from 1.1988 to 0.3637 i.e. a sub-pixel accuracy is achieved for 90% of the images.

5. Fusion of edge-based and edge-less approaches

5.1. Edge-less versus edge-based approaches

For comparison purposes, Table 2 compares the proposed edge-less approach with the edge-based approaches using edge classification. As

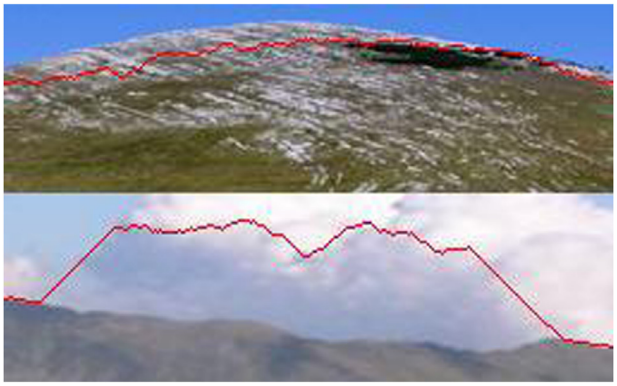

Fig. 10. Examples illustrating the inability of the proposed method to find a good solution due to using a small training data set.

Table 2
Average absolute errors using edge-based (traditional, with classification) and edge-less approaches.

Approach	Basalt Hills				Web			
	μ	σ	min	max	μ	σ	min	max
SIFT+HOG Edges (Ahmad et al., 2014, 2015c)	0.57	1.02	0	3.58	0.87	1.03	0.43	7.05
SIFT+HOG Scores (Ahmad et al., 2014, 2015c)	0.41	0.81	0	3.08	0.97	1.57	0.38	12.19
SIFT+HOG Scores + G_r (Ahmad et al., 2014, 2015c)	0.43	0.81	0	3.08	1.30	3.98	0.38	34.95
SVM-mDCSI	1.01	0.29	0.62	1.76	1.28	1.20	0.37	6.21
CNN-mDCSI	0.75	0.23	0.42	1.28	1.41	1.49	0.27	10.79

it is evident from Tables 1 and 2, using machine learning for horizon line detection is very promising. Both machine learning approaches outperform the classical edge-based approach (Lie et al., 2005) by a high margin. A quick look of Table 2, however, reveals that non-classical edge-based approaches (Ahmad et al., 2015c) outperform the edge-less approach by a small margin. In an effort to better understand and explain these results, we have identified several reasons which are discussed below. Moreover, to leverage the strengths and weaknesses of each approach, we propose a fusion strategy.

5.1.1. 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 the edge-based methods when computing the error. 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 edge segments. When using edge-based methods for horizon line detection, these segments are typically part of solution; however, this might not be the case for edge-less methods. To sum it up, the ground truth is more biased towards edge-based approaches.

5.1.2. Smoother localization

Edge-based solutions tend to be smoother while edge-less method are typically bumpy. This is because DP tries to find a path with a low cost without imposing any smoothness constraints on the solution which favors edge-based methods again.

5.1.3. 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 a much smaller number of edge pixels are classified in the case of edge-based methods.

Fig. 11 shows several examples where edge-based horizon detection has outperformed edge-less horizon detection. Fig. 12 provides more details.

Fig. 11. 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 as outperformed the edge-less approach.

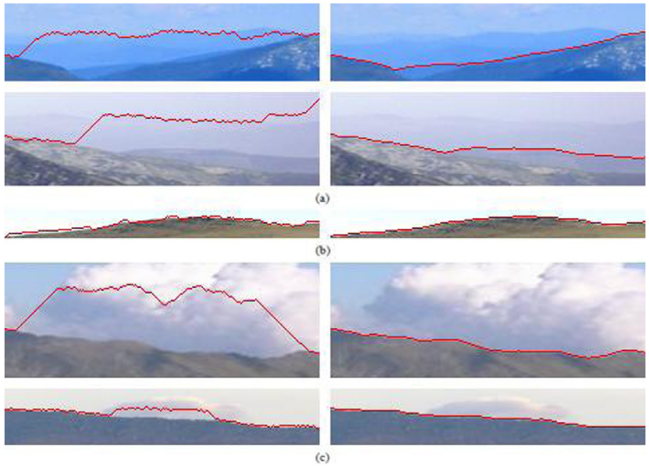

Fig. 12. Detailed segments from Fig. 11: (a) multiple horizons, (b) smoother localization, (c) miss-classifications.

5.2. Fusion

We discussed in the previous section several reasons favoring edge-based methods. On the other hand, the main advantage of the edge-less approach is that the DCSI map contains no gaps. To improve horizon line detection, we propose fusing information from edge-based and edge-less methods.

The fusion of gradient information with pixel classification scores is a natural extension of Eq. (14) assuming that both G_r and S are dense maps. Fusing edge information with pixel classifications can be performed in two steps. First, the DCSI map (i.e., $S(i,j)$) is generated for the query image which provides the horizon-ness for each pixel. Second, edge detection is performed on the query image; then, the horizon-ness of each pixel is boosted if that pixel happens to be an edge pixel. We have considered both Canny and MSEE edges in our experiments. Depending on whether Canny edges (i.e., $E(i,j)$) or MSEE edges (i.e., $E_m(i,j)$) are used, Eq. (15) needs to be modified as follows:

Fig. 13. Fusion based horizon detection: (a) query image, (b) DCSI, (c) MSEE Edges, (d) fused DCSI, (e) detected horizon (red boundary). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

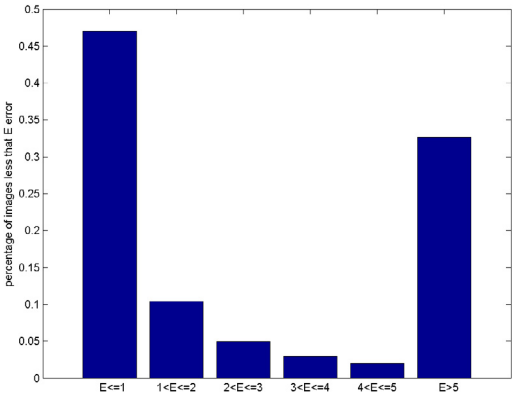

Fig. 14. Percentage distribution of average absolute error across images in Switzerland data set (Baatz et al., 2012): 45% horizons detected with sub-pixel error while 67% solution with an average absolute error of less than 5-pixels.

$$\Psi(i,j) = \begin{cases} S(i,j) + b, & \text{if } E(i,j) = 1 \\ S(i,j), & \text{otherwise;} \end{cases} \tag{20}$$

$$\Psi(i,j) = \begin{cases} S(i,j) + b, & \text{if } E_m(i,j) = 1 \\ S(i,j), & \text{otherwise;} \end{cases} \tag{21}$$

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. Fig. 13 shows various steps of the proposed fusion strategy. The DCSI 13–(b) is combined with MSEE edges 13–(c) to obtain the fused DCSI 13–(d) which is then used for finding the shortest path (horizon) by applying DP.

Table 3 provides a quantitative comparison of our fusion formulations; for completeness, we have also included the best results of the edge-based and edge-less approaches from Table 2. As Table 3 shows, using gradient information tends to harm the overall accuracy which is in agreement with our earlier results using gradient information (Table 1). This is because there are edges, due to clouds, with strong magnitudes near the horizon which become part of the DP solution.

Fig. 15. Examples of faulty detection with average error more than five pixels: sample images (column 1 & 3), solution found by fusion approach (column 2 & 4).

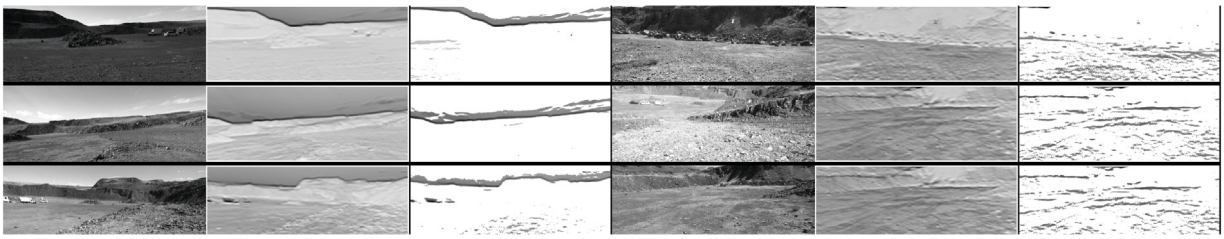

Fig. 16. Absence of horizon affecting the mDCSIs, causing them to have high scores for found shortest paths: Column 1 and 4 showing images with and without horizons respectively, respective SVM-DCSIs (columns 2 & 5) and mDCSIs (column3 & 6) show no continuity for non-horizon images.

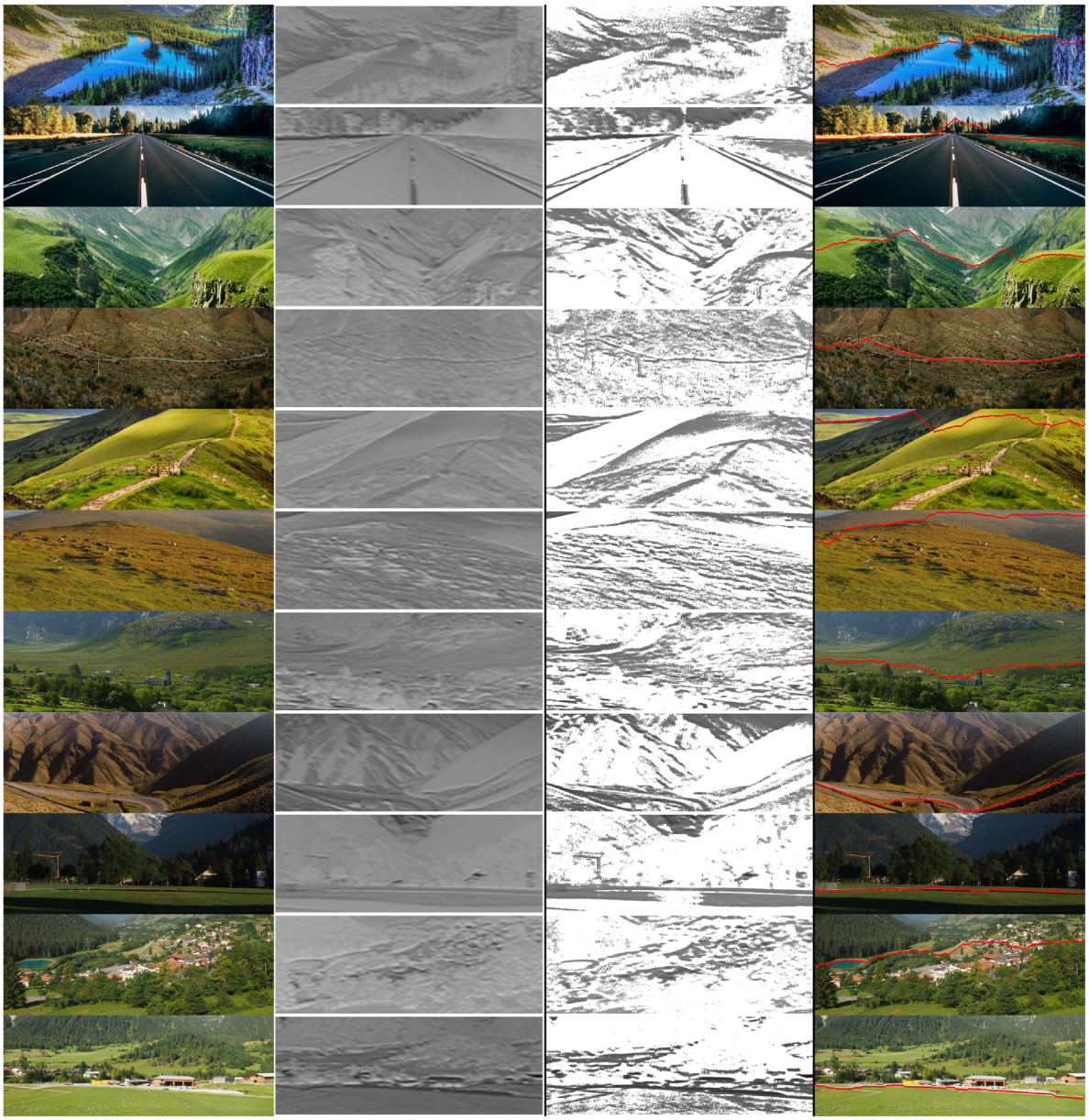

Fig. 17. Examples of absence of horizon line detection: sample images (column 1), respective DCSIs (column 2) and mDCSI (column 3) and faulty solutions found by DP (column 4, highlighted in red) which would be identified by the Gaussian classifier as faulty detections. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Ignoring the strength of edges (i.e., gradient magnitude) by simply boosting the horizon-ness of edge pixels is more effective in excluding cloud edges from becoming part of the detected horizon line. It is interesting to note that using Canny edges for fusion is slightly better on the Web data set than using MSEE edges; this is because some horizon edges might not survive during the extraction of MSEE edges. Overall, the proposed fusion approach (**SVM-DCSI+Canny Edges**) has outperformed all edge-based and edge-less formulations.

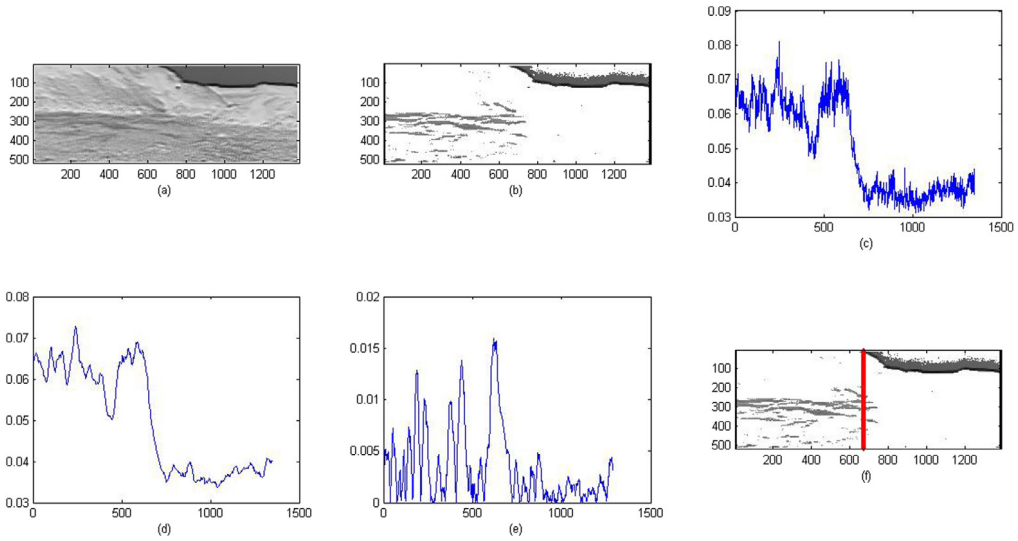

Fig. 18. Steps towards determining the start/end point for partial horizons: (a) DCSI for an image with partial horizon, (b) mDCSI, (c) average classification scores for each column, (d) smoothed averages, (e) peak corresponding interest point and (f) found end point marked by vertical red bar.

Table 3
Average absolute errors using fusion of edge-based (No Classification) and edge-less information.

Approach	Basalt Hills				Web			
	μ	σ	*min*	*max*	μ	σ	*min*	*max*
SIFT+HOG Edges (Ahmad et al., 2014, 2015c)	0.57	1.02	0	3.58	0.87	1.03	0.43	7.05
SVM-mDCSI	1.01	0.29	0.62	1.76	1.28	1.20	0.37	6.21
SVM-DCSI+G_r	0.60	0.29	0.17	1.31	4.86	15.98	0.14	98.46
SVM-DCSI+MSEE Edges	0.73	0.32	0.48	2.07	0.85	0.89	0.35	5.05
SVM-DCSI+Canny Edges	0.77	0.35	0.48	2.07	0.78	0.76	0.35	4.84

Table 4
Percentage verification accuracies for different 1d/2d Gaussian classifiers for the good-ness of Baatz et al. (2012) data set.

Gaussian classifier	TP	TN	FP	FN
1D : μ vector	92.42	50.75	49.25	7.57
1D : σ vector	93.18	82.09	17.91	6.81
2D : μ & σ vectors	89.39	82.09	17.91	10.60

5.3. Further evaluation

To further test the fusion approach, we have considered the Switzerland data set (Baatz et al., 2012). This data set is comprised of more than 200 mountainous images with considerable viewpoint, terrain and weather variations. The ground truth has been made available by the authors of Baatz et al. (2012). Our fusion strategy has achieved a sub-pixel average error in 90 out of the 203 images and less than 5 pixels error in 67% of the images. Fig. 14 shows the distribution of absolute average error for the Switzerland data set. It should be emphasized that these images are very challenging, never seen by the trained classifier, and captured under different seasonal conditions and geographical locations. It is worth mentioning that Baatz et al. (2012) have reported that human interaction was necessary for almost half of these images in order to extract the horizon line in their localization experiments. Also, they did not report the average errors for the detection; so it is not clear how good the detection was for half of the images where horizon was detected without any human involvement. Fig. 20 shows some examples from the Switzerland data set, along with the ground truth and detected horizon lines using our fusion method (SVM-mDCSI+Canny Edges) while Fig. 15 shows some examples of faulty detections with average absolute error beyond five pixels.

It should be noted that in most of these cases DP found a solution which corresponds to discontinuity caused by another mountain sitting in front of a distant (less discontinuous) horizon or snowy mountains having more confidence than the actual horizon. The fusion approach did not get any benefit from edges because both faulty and actual horizon has edge support where as the solution has more confidence due to classifier compared to the actual horizon. It should be noted that our training set of nine images does not contain a single snowy image. We expect the accuracy to increase a lot if snowy examples are added to the training set.

5.3.1. Verification

The distribution of the nodal costs along the DP solution (i.e. detected horizon) provides a metric about the good-ness of the solution. We exploit this information to verify if a found solution is actually a good/acceptable solution (error $\leq$ 5 pixels) or a faulty one (error $>$ 5 pixels). This verification can be of great interest; for example in visual geo-localization or planetary rover localization; knowing that the found horizon solution is a faulty one would save the computations in the subsequent steps of localization pipeline i.e. DEM rendering and horizon matching etc. Specifically, we compute the mean and standard deviation of the nodal costs for the pixels belonging to the detected solution by DP. The mean and/or standard deviation measure the divergence of the nodal costs for the nodes belonging to horizon solution. The intuition being that: for a faulty solution the nodal costs along the path would diverge and hence result in comparatively big values for mean and standard deviation. A simple 1d/2d Gaussian classifier can then be trained for the verification of found solutions. The computation for mean (μ_d) and standard deviation (σ_d) are shown in the equations below,

$$\mu_d = \frac{1}{N}\sum_{j=1}^{N} \Psi(P_{d(j)}, j) \tag{22}$$

$$\sigma_d = \sqrt{\left(\frac{1}{N}\sum_{j=1}^{N}\left[\Psi(P_{d(j)}, j) - \mu_d\right]^2\right)} \tag{23}$$

where $P_{d(j)}$ is the node (row) index for jth stage (column) in the detected path, N is the number of stages in the graph for DP and cost Ψ follows from Eq. (20).

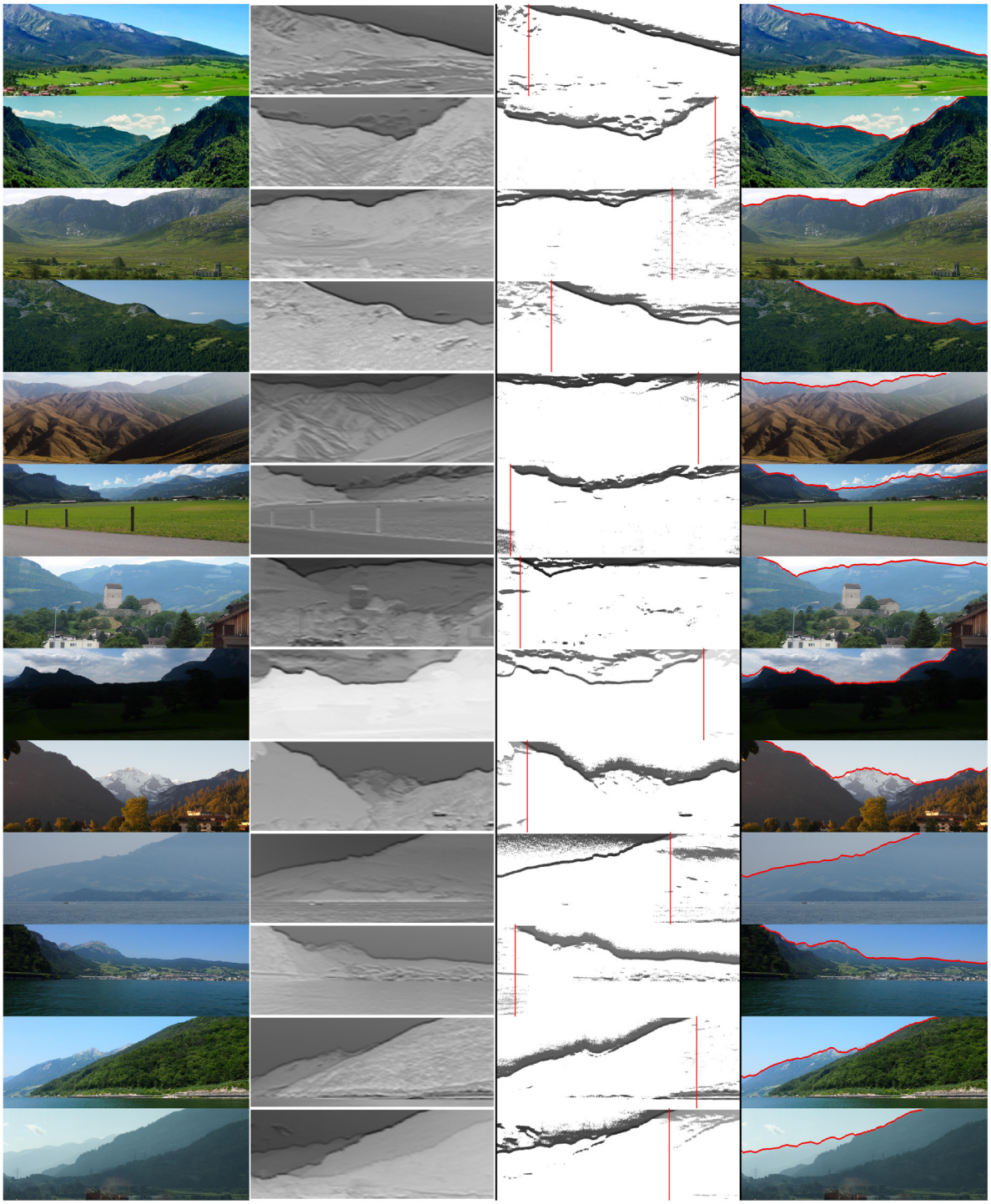

Fig. 19. Examples of partial horizon line detection: sample images (column 1), respective DCSIs (column 2) and mDCSI (column 3) with determined start/end points (marked by vertical red bars) and partial horizons found by DP (column 4, highlighted in red).

For verification of horizon solutions found for Baatz et al. (2012) data set, we first computed the μ_d and σ_d for all the images in the Web data set and formulated 1d Gaussian classifiers based on means, standard deviations vectors alone and a 2d Gaussian classifier based on both mean and standard deviation vectors. These classifiers are then used to verify if a found solution is good or faulty based on computed mean and deviation for each solution from Switzerland data set (Baatz et al., 2012). Table 4 shows the percentage false positive, false negative, true positive and true negative rates for each of these 1d/2d classifiers used to verify solutions from Baatz et al. (2012) data set. To the best of our knowledge this is the first attempt towards measuring the good-ness of a found horizon solution.

6. Non-horizon line detection

The solution verification introduced in previous section can further be extended for verifying the presence/absence of horizon at all. In various applications, including rover navigation, it is important to detect the horizon line with high confidence. In the DP formulation, however, a shortest path solution will always be found irrespective

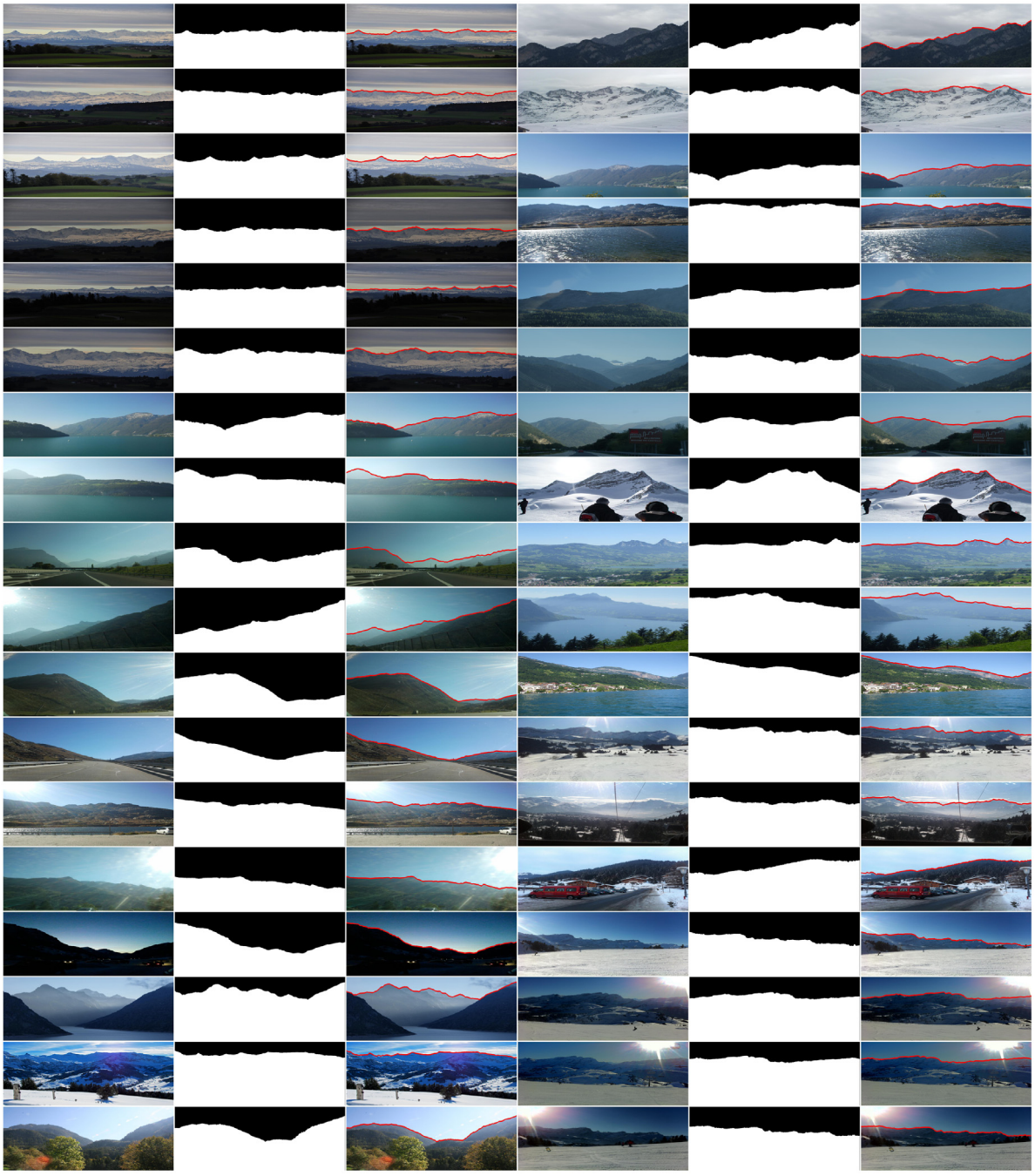

Fig. 20. Results of the proposed fusion on sample test images from Switzerland data set (Baatz et al., 2012): query images (columns 1 & 4), ground truth segmentations (columns 2 & 5) and found horizon lines (columns 3 & 6, highlighted in red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of whether the horizon is actually present in the query image or not. To the best of our knowledge, this problem has not been addressed before in the literature either. To determine the presence/absence of the horizon line, the mDCSI map could be used again under the same framework identified in previous section. Specifically, a narrow band of pixels with high horizon-ness (i.e., classification scores) is typically found around the true horizon line; the absence of such a narrow band would indicate that the horizon line is not visible as shown in Fig. 16. In this case, the shortest path found would have much higher cost than a typical solution where the cost is defined as the sum of classification scores along the path found. A simple Gaussian classifier can then be used to verify if the solution found reflects absence of horizon in the query image same as used to verify the good-ness of detected solution in previous section.

Using a collection of 40 mountainous images from web where horizon line was not visible, we were able to confirm that Gaussian classifier based on Web data set correctly classified these solutions to be faulty. Some of these examples along with their DCSIs, mDCSIs and found solutions are shown in Fig. 17. It should be noted that how DP is trying to find a continuous path e.g. along the mountains, pathways and roads. The DCSIs generated are based on the original proposed edge-less approach and not that of fusion and hence the respective nodal costs used for the Gaussian classifier.

7. Partial horizon line detection

Previous studies have not addressed the issue of partial horizon line detection i.e. when segment of the horizon is missing and horizon not necessarily extends from left most column to right most. When flying and ground robots move at various angles or steep terrains, however, it is often the case that only a partial horizon line is visible. From a theoretical point of view, the presence of a partial horizon line provides sufficient information in a number of tasks, for example, in robot localization or visual geo-localization. The problem of partial horizon detection boils down to the problem of determining where to place the start and sink nodes in the DP formulation. In the general case, it is assumed that the horizon line extends from the left most column to the right most column of the image which might not always be true as shown in Fig. 19-row1. The mDCSI map can be exploited to determine the end points of the horizon boundary. Given a query image, the mDCSI map is computed as described earlier. Then, we compute an average classification score for each column in the mDCSI map and apply smoothing (based on averaging) using a 1x3 window. After smoothing, the start/end points of the horizon line are determined by local maxima detection. Fig. 18 shows a sample DCSI image containing a partial horizon from the Basalt Hills data set (a), the respective mDCSI (b), the average and smoothed classification scores (c, d), the end point detected through local maxima detection (e), and the found end point marked on the mDCSI (f). We have experimented with 40 partial horizon images collected from web and were able to successfully find the partial horizon in 35 of them. Some representative results of partial horizon line detection are shown in Fig. 19 along with the end points found by our method. It should be mentioned that very accurate detection of the start/end points is not necessary and that extracting a portion of the horizon line would be sufficient for various applications. We are again demonstrating these results based on edge-less approach/nodal costs and not the fusion strategy, although that can be equally applicable.

8. Conclusion

We have presented an edge-less horizon line detection approach where classification scores are used within a DP framework. This formulation differs from earlier DP based solutions where edges are used to construct a multi-stage graphs which suffers from edge-gaps. We have trained SVM and CNN classifiers using normalized pixel intensities to generate DCSI/mDCSI images. A comparative analysis between the proposed and other edge based methods has been provided using two challenging data sets. Moreover, a fusion strategy was proposed where edge information is used to further enhance the horizon-ness confidence. The fusion strategy is shown to outperform all other formulations and is further tested on an additional challenging test set. The fusion strategy is further tested on an extensive more challenging data set with reasonable performance given the limited training. We demonstrate various by-products of our proposed approach, for example statistical measures along the detected path are used to determine the good-ness of the solution and hence determining the acceptability of solution for lateral stages of geo-localization pipeline. We also show how the proposed approach/fusion can be used to determine the absence of horizon in a given image and to find a partial horizon line. For future work, we plan to investigate the suitability of using the horizon line for ground and flying robot localization. We would like to point that, due to its time complexity and hybrid nature, current proposed framework is not suitable for real-time applications and this is an area of our current research efforts.

Acknowledgments

This work is supported by NASA EPSCoR, USA under Cooperative Agreement No. NNX11AM09A, and in part by NSF PFI (USA). We would like to thank (Baatz et al., 2012) for making their data set available.

References

Ahmad, T., Bebis, G., Nicolescu, M., Nefian, A., Fong, T., 2015a. An edge-less approach to horizon line detection. In: Proceedings of 14th International Conference on Machine Learning and Applications (ICMLA), pp. 1095–1102.

Ahmad, T., Bebis, G., Nicolescu, M., Nefian, A., Fong, T., 2015b. Fusion of edge-less and edge-based approaches for horizon line detection. In: Proceedings of 6th International Conference on Information, Intelligence, Systems and Applications (IISA).

Ahmad, T., Bebis, G., Regentova, E., Nefian, A., 2013. A machine learning approach to horizon line detection using local features. In: Proceedings of 9th International Symposium on Visual Computing (ISVC).

Ahmad, T., Bebis, G., Regentova, E., Nefian, A., Fong, T., 2014. An experimental evaluation of different features and nodal costs for horizon line detection. In: Proceedings of 10th International Symposium on Visual Computing (ISVC).

Ahmad, T., Bebis, G., Regentova, E., Nefian, A., Fong, T., 2015c. Coupling dynamic programming with machine learning for horizon line detection. Int. J. Artif. Intell. Tools 24 (4), 1–19.

Baatz, G., Saurer, O., Koser, K., Pollefeys, M., 2012. Large scale visual geo-localization of images in mountainous terrain. In: Proceedings of European Conference on Computer Vision (ECCV), pp. 517–530.

Baboud, L., Cadik, M., Eisemann, E., Seidel, H.-P., 2011. Automatic photo-to-terrain alignment for the annotation of mountain pictures. In: Computer Vision and Pattern Recognition (CVPR).

Boroujeni, N.S., Etemad, S.A., Whitehead, A., 2012. Robust horizon detection using segmentation for UAV applications. In: IEEE 2012 Ninth Conference on Computer and Robot Vision (CRV). pp. 346–352.

Boukas, E., Gasteratos, A., Visentin, G., 2014. Localization of planetary exploration rovers with orbital imaging: a survey of approaches. In: International Conference on Robotics and Automation Workshops (ICRAW).

Braun, D., Singhof, M., 2015. Automated silhouette extraction for mountain recognition. In: Proceedings of 27th GI-Workshop, pp. 18–23.

Chen, Y., Qian, G., Gunda, K., Gupta, H., Shafique, K., 2015. Camera geolocation from mountain images. In: Proceedings of 18th International Conference on Information Fusion, pp. 1587–1596.

Chippendale, P., Zanin, M., Andreatta, C., 2008. Spatial and temporal attractiveness analysis through geo-referenced photo alignment. In: Proceedings of IEEE International Geoscience and Remote Sensing Symposium, pp. 1116–1119.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20 (3), 273–297.

Cozman, F., Krotkov, E., 1997. Automatic mountain detection and pose estimation for teleoperation of lunar rovers. In: Proceedings of International Conference on Robotics and Automation (ICRA), pp. 2452–2457.

Cozman, F., Krotkov, E., Guestrin, C., 2000. Outdoor visual position estimation for planetary rovers. Auton. Robots 9(2), 135–150.

d. Croon, G.C.H.E., Remes, B.D.W., Wagter, C.D., Ruijsink, R., 2011. Sky segmentation approach to obstacle avoidance. In: IEEE Aerospace Conference. pp. 1–16.

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition (CVPR).

Di, L., Fromm, T., Chen, Y., 2012. A data fusion system for attitude estimation of low-cost miniature UAVs. J. Intell. Robot. Syst. 65(1), 621–635.

Dumble, S.J., Gibbens, P.W., 2012. Horizon profile detection for attitude determination. J. Intell. Robot. Syst. 68 (3), 339–357.

Dumble, S.J., Gibbens, P.W., 2015. Efficient terrain-aided visual horizon based attitude estimation and localization. J. Intell. Robot. Syst. 78 (2), 205–221.

Dusha, D., Boles, W., Walker, R., 2007. Attitude estimation for a fixed-wing aircraft using horizon detection and optical flow. In: Proceedings of Digital Image Computing Techniques and Applications (DICTA), pp. 485–492.

Ettinger, S.M., Nechyba, M.C., Ifju, P.G., Waszak, M., 2002. Vision-guided flight stability and control for micro air vehicles. In: Proceedings of International Conference on Intelligent Robots and Systems(IEEE/RSJ).

Fedorov, R., Fraternali, P., Tagliasacchi, M., 2014. Mountain peak identification in visual content based on coarse digital elevation models. In: Proceedings of 3rd International Workshop on Multimedia Analysis for Ecological Data, pp. 7–11.

Fefilatyev, S., Smarodzinava, V., Hall, L.O., Goldgof, D.B., 2006. Horizon detection using machine learning techniques. In: International Conference on Machine Learning and Applications (ICMLA). pp. 17–21.

Gershikov, E., 2014. Is color important for horizon line detection?. In: International Conference on Advanced Technologies for Communications. pp. 262–267.

Gershikov, E., Libe, T., Kosolapov, S., 2013. Horizon line detection in marine images: Which method to choose?. Int. J. Adv. Intell. Syst. 6 (1,2).

Grelsson, B., Felsberg, M., Isaksson, F., 2015. Highly accurate attitude estimation via horizon detection. J. Field Robotics.

Gupta, V., Brennan, S., 2008. Terrain based vehicle orientation estimation combining vision and inertial measurements. J. Field Robotics 25 (3), 181–202.

Hays, J., Efros, A.A., 2008. IM2GPS: estimating geographic information from a single image. In: Computer Vision and Pattern Recognition (CVPR).

Ho, N., Chakravarty, P., 2014. Localization on freeways using the horizon line signature. In: Proceedings of International Conference on Robotics and Automation (ICRA).

Hou, J., Li, B., 2015. An improved algorithm for horizon detection based on OSTU. In: Proceedings of International Conference on Intelligent Human-Machine Systems and Cybernetics, pp. 414–417.

Hung, Y., Su, C., Chang, Y., Chang, J., Tyan, H., 2013. Skyline localization for mountain images. In: Proceedings of International Conference on Multimedia and Expo (ICME).

Kim, B., Shin, J., Nam, H., Kim, J., 2011. Skyline Extraction using a Multistage Edge Filtering. World Academy of Science, Engineering and Technology.

Kong, X., Liu, L., Qian, Y., Cui, M., 2016. Automatic detection of sea-sky horizon line and small targets in maritime infrared imagery. Infrared Phys. Technol. 76, 185–199.

Kruger, W., Orlov, Z., 2010. Robust layer-based boat detection and multi-target-tracking in maritime environments. In: Proceedings of International Waterside Security Conference.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient based learning applied to document recognition. PIEEE 86 (11), 2278–2324.

Lie, W., Lin, T.C.-I., Lin, T., Hung, K.-S., 2005. A robust dynamic programming algorithm to extract skyline in images for navigation. Pattern Recognit. Lett. 26 (2), 221–230.

Liu, W., Su, C., 2014. Automatic peak recognition for mountain images. In: Advanced Technologies, Embedded and Multimedia for Human-Centric Computing.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. (IJCV) 60 (2), 91–110.

Matas, J., Chum, O., Urban, M., Pajdla, T., 2002. Robust wide baseline stereo from maximally stable extremal regions. In: Proceedings of British Machine Vision Conference (BMVC), pp. 384–396.

McGee, T.G., Sengupta, R., Hedrick, K., 2005. Obstacle detection for small autonomous aircraft using sky segmentation. In: Proceedings of International Conference on Robotics and Automation (ICRA).

Nefian, A.V., Bouyssounouse, X., Edwards, L., Kim, T., Hand, E., Rhizor, J., Deans, M., Bebis, G., Fong, T., 2014. Planetary rover localization within orbital maps. In: International Conference on Image Processing (ICIP).

Neto, A.M., Victorino, A.C., Fantoni, I., Zampieri, D.E., 2011. Robust horizon finding algorithm for real-time autonomous navigation based on monocular vision. In: International Conference on Intelligent Transportation Systems.

Ojala, T., Pietikainen, M., Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) 24 (7), 971–987.

Ozcanli, O.C., Dong, Y., Mundy, J.L., 2016. Geo-localization using volumetric representations of overhead imagery. Int. J. Comput. Vis. 116 (3), 226–246.

Porzi, L., Bulo, S.R., Ricci, E., 2016. A deeply-supervised deconvolutional network for horizon line detection. In: ACM Conference on Multimedial.

Porzi, L., Bulo, S.R., Valigi, P., Lanz, O., Ricci, E., 2014. Learning contours for automatic annotations of mountains pictures on a smartphone. In: ACM/IEEE International Conference on Distributed Smart Cameras.

Saurer, O., Baatz, G., Koser, K., Ladicky, L., Pollefeys, M., 2016. Image based geo-localization in the alps. Int. J. Comput. Vis. (IJCV) 116 (3), 213–225.

Shen, Y., Krusienski, D., Li, J., Rahman, Z., 2013. A hierarchical horizon detection algorithm. IEEE Geosci. Remote Sens. Lett. 10 (1), 111–114.

Shen, Y., Wang, Q., 2013. Sky region detection in a single image for autonomous ground robot navigation. Int. J. Adv. Robot. Syst. 10.

Stein, F., Medioni, G., 1995. Map-based localization using the panoramic horizon. IEEE Trans. Robot. Autom. 11 (6), 892–896.

Thurrowgood, S., Soccol, D., Moore, R.J.D., Bland, D., Srinivasan, M.V., 2009. A vision based system for attitude estimation of UAVs. In: IEEE International Conference on Intelligent Robots and Systems, pp. 5725–5730.

Todorovic, S., Nechyba, M.C., Ifju, P.G., 2003. Sky/ground modeling for autonomous MAV flight. In: Proceedings of International Conference on Robotics and Automation (ICRA).

Tzeng, E., Zhai, A., Clements, M., Townshend, R., Zakhor, A., 2013. User-driven geolocation of untagged desert imagery using digital elevation models. In: Computer Vision and Pattern Recognition Workshops (CVPRW).

Verbickas, R., Whitehead, A., 2014. Sky and ground detection using convolutional neural networks. In: International Conference on Machine Vision and Machine Learning (MVML).

Yazdanpanah, A.P., Regentova, E.E., Mandava, A.K., Ahmad, T., Bebis, G., 2013. Sky segmentation by fusing clustering with neural networks. In: Proceedings of 9th International Symposium on Visual Computing (ISVC).

Yazdanpanah, A.P., Regentova, E.E., Muthukumar, V., Bebis, G., 2015. Efficient terrain-aided visual horizon based attitude estimation and localization. Int. J. Comput. Appl. 121 (10).

Zamir, A.R., Shah, M., 2010. Accurate image localization based on google maps street view. In: European Conference on Computer Vision (ECCV).

Zheng, Y., Zhao, M., Song, Y., Adam, H., Buddemeier, U., Bissacco, A., Brucher, F., Chua, T., Neven, H., Yagnik, J., 2009. Tour the world: building a web-scale landmark recognition engine. In: Computer Vision and Pattern Recognition (CVPR).