Horizon Detection Using Machine Learning Techniques

Sergiy Fefilatyev sfefilat@cse.usf.edu
Volha Smarodzinava smarodzi@cse.usf.edu
Lawrence O. Hall hall@cse.usf.edu
Dmitry B. Goldgof goldgof@cse.usf.edu

Department of Computer Science and Engineering, University of South Florida, 4202 E. Fowler Ave, Tampa, FL. 33620

Abstract

Detecting a horizon in an image is an important part of many image related applications such as detecting ships on the horizon, flight control, and port security. Most of the existing solutions for the problem only use image processing methods to identify a horizon line in an image. This results in good accuracy for many cases and is fast in computation. However, for some images with difficult environmental conditions like a foggy or cloudy sky these image processing methods are inherently inaccurate in identifying the correct horizon. This paper investigates how to detect the horizon line in a set of images using a machine learning approach. The performance of the SVM, J48, and Naïve Bayes classifiers, used for the problem, has been compared. Accuracy of 90-99% in identifying horizon was achieved on image data set of 20 images.

1. Introduction

Accurate horizon detection is essential for a number of applications such as port security, flight and navigation control, flow management, port safety, and protection of sanctuaries.

One of the previous approaches [1] used only image processing methods, which, however, performed very well in most cases. Another approach [2] was combining machine learning methods with morphology based operations, the Hough Transform and Expectation Maximization function to separate pixel distributions. However, their performance in identifying a horizon suffers when images are complicated with clutter such as a very cloudy or foggy environment, an uneven horizon line, and varying lighting conditions. Partly this is because features used in the approaches only included only intensities of the pixels in an image, which is not enough for identifying horizon in "hard" images. Method used in [3] utilizes texture information in images, and potentially is the most advanced among technique, but it was primarily developed for video sequences. In this paper we have investigated usability of different machine learning algorithms for horizon detection problem. Specific approach described in this paper allows the identification of a horizon line in single images for the most adverse conditions of the horizon by using machine learning to provide a general model. It uses texture information along with pixel colors for its feature set. The approach provides accuracy of identifying horizon line within 90-99%.

Detecting the horizon can be seen as finding a line which separates the image into two regions: sky and non-sky. Non-sky pixels usually belong to the ground or water surfaces, and, to be consistent with many papers which refer this problem, the non-sky pixels will be called further as *Ground* pixels. Thus, one of the objectives of the work in this paper was to attempt to successfully classify the pixels of the image into two major classes: either the *Sky* class or the *Ground* class, by using the selected classifiers.

Once we could classify each pixel into two classes, we could create a binary image, where a pixel with value of "0" belong to the *Ground* class and a pixel with a value "1" belong to the *Sky* class. Further in the paper such kind of an image will be referred as a black and white image where all the white pixels belong to the sky and all the black pixels belong to the ground. For this paper, an assumption was made that the line that would separate the sky and the non-sky pixels would be a straight line, and, thus, the other objective of the paper becomes finding a line, between black and white regions in the black-and-white picture, which would correspond to the actual horizon line in the original image.

2. Horizon Line Representation

Similarly to many image processing methods, like Hough transform, normal representation of a line was used (see Figure 1):

Proceedings of the 5th International Conference on Machine Learning and Applications (ICMLA'06)
0-7695-2735-3/06 $20.00 © 2006 IEEE

$$x\cos\Theta + y\sin\Theta = \rho \quad (1)$$

where Θ is the angle between x-axis and normal to the line and ρ is distance from the origin of the coordinate system to the line (see Figure 1 below)

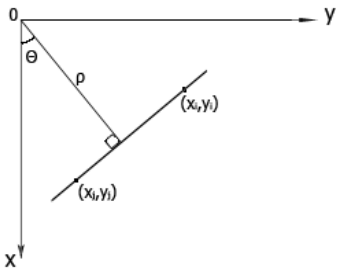


Figure 1. Normal line representation

It is important to observe that:

1. For every line, the angle Θ lies in [-π/2, π/2].
2. Distance ρ varies from 0 to D, where D is the length of the main diagonal in the image plane.

3. Feature Selection and Pixel Labeling

A superimposed horizon line was drawn manually as a ground truth over each image used for training and testing which gave ground truth values of Θ and ρ parameters of the horizon line. All the pixels that satisfied the inequality $x\cos\Theta + y\sin\Theta < \rho$ were assigned class "sky-pixel" and all the pixels that satisfy the inequality $x\cos\Theta + y\sin\Theta \geq \rho$ -class "ground-pixel".

Each pixel of the image was an individual instance in the dataset. We defined the following 21 attributes for each pixel (instance):

1. Intensity of red channel
2. Intensity of green channel
3. Intensity of blue channel

Texture measurements, described in [4], (for each of the three color channels) for a histogram of a small region of 10x10 centered around each pixel:

4-6. Mean $$m = \sum_{i=0}^{L-1} z_i p(z_i) \quad (2)$$

7-9. Standard deviation $$\sigma = \sqrt{\mu_2(z)} = \sqrt{\sigma^2} \quad (3)$$

10-12.Smoothness $$R = 1 - 1/(1+\sigma^2) \quad (4)$$

13-15. Third moment $$\mu_3 = \sum_{i=0}^{L-1} (z_i - m)^3 p(z_i) \quad (5)$$

16-18. Uniformity $$U = \sum_{i=0}^{L-1} p^2(z_i) \quad (6)$$

19-21. Entropy $$e = -\sum_{i=0}^{L-1} p(z_i)\log_2(z_i) \quad (7)$$

4. Horizon Line Detection

Once the results were produced by a chosen classifier we find the predicted class for each instance (pixel) in our dataset.

First, having information how each pixel is classified, a black-and-white image is produced. After that we need to find Θ and ρ of the line that best separates the white and black regions in a picture (see Figure 2).

Figure 2. Sky/Ground pixel classification

To determine the best separating line, we used the same expectation maximization function described in [1]. The following criterion, a 2-argument function of Θ and ρ, reaches a maximum, when the line corresponding to the arguments of this function (Θ and ρ) best separates the black and white regions.

Thus, this criterion is defined as follows:

$$J(\Theta,\rho) = \frac{1}{\Sigma_s + \Sigma_g} \quad (8)$$

where:

$$\Sigma_s = \frac{1}{(n_s)} \sum_{i=1}^{n_s} (x_i^s - \mu_s) \quad (9)$$

$$\Sigma_G = \frac{1}{(n_G)} \sum_{i=1}^{n_G} (x_i^g - \mu_g) \quad (10)$$

$$\mu_s = \frac{1}{n_s}\sum_{i=1}^{n_s} x_i^s, \quad \mu_g = \frac{1}{n_g}\sum_{i=1}^{n_g} x_i^g \quad (11)$$

$x_i^g - i^{th}$ pixel, which was identified as a ground-pixel (it is equal to 0 or 1 depending if it has white or black color).

$x_i^S - i^{th}$ pixel, which was identified as a sky-pixel (it is equal to 0 or 1 depending on whether it has white or black color).

μ_s - mean value of all pixels which where marked as sky.

μ_g - mean value of all pixels which where marked as ground.

After the criterion is defined, we search (with EM) the 2-dimensional parameter space (Θ and ρ) in order to find parameters Θ and ρ that correspond to the

maximum of the function. These parameters will then identify the predicted horizon line, and the results produced by the classifier can be visualized.

5. Experimental Results

We ran a number of tests using WEKA [5] for selected classifiers, trying to detect the ones that would produce higher accuracy. The following classifiers were chosen for the experiments:

1. SVM
2. J48
3. Naïve Bayes

During the experiments, the default options for SVM and Naïve Bayes were used. The J48 decision tree algorithm was used in 3 different combinations - default options, with the reduced error pruning setting, and with a confidence factor 0.005. The last 2 options were used to avoid overfitting of the training data.

The measure of accuracy of identifying the horizon line in an image was defined as the percentage of pixels classified correctly in the test image as compare to the total number of pixels in the image:

$$A = \frac{N_c}{N} \cdot 100\% \qquad (12)$$

where N_c is the number of correctly classified pixels, N – the total number of pixels in an image.

5.1. Identifying a Horizon Line in Similar Images

For the first experiment, 10 images were used, each with the resolution of 200x150

All of the images in the set were taken by the same camera, were fairly similar as far as lightning conditions, color, and the texture present are concerned.

A chosen default of 10x10 was used around each pixel to create the texture measurement.

During the experiment, we trained the classifiers on one image, and then tested on the remaining 9. We repeated the experiment 10 times, each time for a different training image, and averaged the results. The following 3 classifiers were used for the experiment:

1. SVM
2. Naïve Bayes
3. J48 , pruned by reducing the confidence factor to 0.005

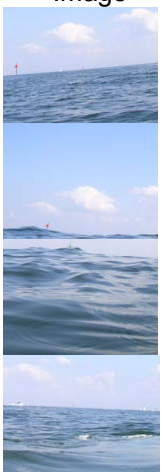
The results of pixel classification for the images in the first experiment are shown in Table 1. SVM was the best algorithm, based on the accuracy. Overall, all three chosen classifiers performed well.

Table 1. Accuracy for different classifiers in experiment#1.

Accuracy for image #	SVM	Naïve Bayes	J48, c = 0.005	Average
1	98.69%	98.40%	99.23%	98.77%
2	99.24%	97.96%	98.16%	98.45%
3	98.23%	95.02%	96.74%	96.66%
4	98.23%	98.72%	99.27%	98.74%
5	99.34%	97.66%	99.04%	98.68%
6	97.65%	96.88%	98.44%	97.66%
7	98.20%	95.97%	97.82%	97.33%
8	96.88%	94.02%	96.83%	95.91%
9	98.43%	98.92%	98.76%	98.70%
10	96.97%	93.91%	94.76%	95.21%
Classifier average:	98.19%	96.75%	97.91%	97.61%

Table 2 contains visualization for some of the results obtained during the experiment. The table contains original pictures, corresponding black and white images for sky/ground classification and the identified horizon line (in black) superimposed on the original image obtained for the SVM classifier.

Table 2. Visualization horizon line found for some images.

Original Image	Black and White Image	Detected Horizon
		

5.2. Identifying Horizon Line in "Hard" Images

For our second experiment, we used another 10 images, each with the resolution of 200x150.

The main difference in experiments is that we used more images for training. Out of all 10 images in the original set we used 9 to train the classifier and the remaining one to test. In such a way we tested all 10 images each time generating a new classifier.

For the second experiment, however, the images were distinctly different; all taken at a different time of the day, with different cameras, and with different lighting conditions (see Table 3). One can clearly notice a difference in the color of the water and the sky, texture patterns of the water and sky, and lighting conditions from one picture to another in the original set of images.

Table 3. Results visualization for the SVM classifier in the second experiment

#	Original Image	Black and White Image	Detected Horizon
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

Again, we used the same default of 10x10 window around each pixel to determine the texture measurement.

We also used the J48 classifier with another two options as opposed to the default settings - for one classifier we used reduced error pruning to avoid overfitting, for the other we set confidence factor to 0.005 from the default 0.25, also, in order to avoid overfitting and create a more compact tree. Hence, the full list of classifiers we used in the second experiment:

1. SVM
2. Naïve Bayes
3. J48
4. J48, used with Reduced Error Pruning
5. J48 with confidence factor 0.005

The results of pixel classification for the images in the second experiment are shown in Table 4.

Table 4. Accuracy for the second experiment.

Classifier used/ exp#	SVM	J48	J48 REP	J48 c=.005	Naïve Bayes	Avg.
Leave1	97.0%	94.8%	90.7%	94.80%	96.10%	94.8%
Leave2	98.8%	93.7%	94.9%	94.00%	82.00%	92.9%
Leave3	97.6%	94.8%	90.7%	94.80%	96.10%	94.8%
Leave4	84.0%	85.3%	85.8%	86.50%	90%	86.3%
Leave5	98.4%	97.9%	97.7%	97.90%	98.60%	98.1%
Leave6	99.5%	99%	98.4%	99%	99%	99.0%
Leave7	93%	84.5%	87.2%	86.30%	75%	85%
Leave8	90%	79.3%	86.7%	80.70%	73%	82%
Leave9	94.0%	98.2%	96.9%	99.60%	97%	97.2%
Leave10	74%	80.8%	78.2%	81.30%	64.20%	76%
Average	92.7%	90.8%	90.7%	91.4%	87.1%	90.6%

The original pictures, resulting black and white images for SVM classifier, and corresponding horizon line superimposed on the original images (in black) for the second experiment are show in Table 3.

The results of the second experiment showed some obvious failures. It appears that in image number 4, the sky reflection disrupted the accuracy of the classifier (SVM) and led it to believe that the reflection in the water was actually *Sky* pixels. To make matters worse, the slope produced in the black and white image resulted in a very "unusual" horizon line – a line with a large negative slope. This was the result of the assumption that the line representing the horizon that would best separate the sky pixels and the ground pixels would be a straight line.

Another interesting result was produced by image number 10. It appears that fog and scattered visibility caused the classifier to misinterpret the horizon line, which appeared right above the fog line. In images number 2, 7, and 8, the clouds in the sky were classified as belonging to class *Ground*. But, due to

the selection of the best separating line between the two regions, the proper horizon was detected.

The results of image number 6, where the wave of green water in the image had the potential of misleading the classifier. Yet, the classifier was able to correctly classify these pixels of the wave as *Ground* pixels.

6. Summary and Discussion

For the first experiment, the accuracies were much higher than in the second. The horizon line was detected without apparent failures. This is largely due to the fact that the images for training and testing were relatively similar (but far from being exactly the same) in color and lighting conditions.

For the second experiment we used images which were very different in lighting conditions, colors and texture patterns, so that training and test data were not similar in all cases; thus, the variety in images and an insufficient number of training instances for such diversity resulted in the lower accuracies and wrongly detected horizon lines.

Success in classifying image pixels depends on the amount of training and variety of data on which classifiers get trained. To improve overall performance, number of attributes can be increased by using different sizes of regions around each pixel while calculating texture measures. It is also possible to introduce attributes based on the relative position of a pixel, so that we could avoid situations when we encounter *Sky* pixels among *Ground* pixels and vice versa. In the future feature selection methods can be applied to reduce number of features and improve the algorithm performance. In addition, utilization of additional pre- and post-processing can further enhance the performance.

7. Acknowledgment

We would like to acknowledge Larry Langebrake of Center for Ocean Technology at the University of South Florida for suggesting the problem and providing some of the images.

8. References

[1] S.M.Ettinger, M.C. Nechyba, P.G. Ifju, M. Waszak, "Vision Guided Flight Stability and Control for Micro Air Vehicles", *Advanced Robotics*, vol. 17, no. 7, pp. 617-40, 2003.

[2] T.G. McGee, R. Sengupta, K. Hedrick, "Obstacle Detection for Small Autonomous Aircraft Using Sky Segmentation", *Proceedings of the IEEE International Conference on Robotics &Automation*, 2005

[3] S. Todorivic, M. Nechyba, "Sky/ground modelling for autonomous MAV flight", *Proc. of IEEE International Conf. on Robotics and Automation*, 2003.

[4] Gonzalez R.C., R.E. Woods, L.E. Steven, *Digital Image Processing using Matlab*", Pearson Prentice Hall, 2004

[5] Witten ,Ian H., Frank, Eibe, *Data Mining. Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Publishers, 2005