Detection of the Vanishing Line of the Ocean Surface from Pairs of Scale-Invariant Keypoints

Sergiy Fefilatyev, Matthew Shreve, and Dmitry Goldgof

University of South Florida, 4202 E. Fowler Ave, Tampa, FL 33620
{sfefilatyev,mashreve,goldgof}@gmail.com

Abstract. In this paper, we propose an algorithm for estimating the vanishing line of a stochastically-textured plane in a single image taken by an uncalibrated perspective camera. As an example of such type of texture we take images of ocean surface for which existing methods of vanishing line detection from texture perform poorly. The proposed algorithm relies on finding pairs of similarly looking scale-invariant keypoints that are different in scale. The location of the vanishing line is estimated directly from those pairs of points by finding the vanishing line that represents the consensus of individually found vanishing points. We demonstrate the potential of the proposed method on a number of real images of ocean surface by estimating the horizon line using SIFT keypoints.

1 Introduction

We are investigating the method for estimating the affine geometry of a stochastically-textured plane from pairs of similarly looking scale-invariant keypoints. The motivation for this approach is the need for a method for inferring the attitude of a forward-looking camera attached to a highly-unstationary buoy platform surveying the ocean horizon [6]. It is not difficult to detect the horizon line in images when the horizon is indeed present in the field of view of the camera using traditional methods [4,12]. However, due to perspective effects on their appearance model, these methods fail to reliably identify the situation when camera points to the water regions only and no horizon in the image exists. By finding the orientation of the textured plane, and, thus, the camera's attitude, we are able resolve ambiguity of horizon line presence in maritime images.

Traditionally, methods for estimating the orientation of a plane are divided into two main categories: spectral [7,5] and structural [10,2] (although, the combinations of the two are also well represented [13,14]). The spectral category category of methods aims to estimate the orientation of a plane by finding the gradient of perspective distortion. Different spectral measurements are used to qualitatively estimate the gradient, making certain assumptions about the texture, such as homogeneity [3] or isotropy [15]. The methods from structural category rely on explicit identification of texture elements (textels) which then are used to find the geometric solution for the vanishing line.

In this paper, we chose the second path to find the orientation of a plane by identifying the vanishing line directly. However, in order to avoid a very fragile process of segmentation of textels, which are unique for every category of texture, we relied on pairs of similarly looking scale-invariant keypoints. Opposite to the traditional structural methods, the keypoints are only similarly looking within pairs, thus, each pair of

A. Petrosino (Ed.): ICIAP 2013, Part I, LNCS 8156, pp. 161–169, 2013.

keypoints may actually represent very different features of the texture. For example, in an image of the water surface one pair of such keypoints may come from tops of the ridges of sea waves, while another pair may come from hollow-looking depressions in the water. Thus, the homogeneity of appearance of structural elements is not required. Only the consensus of pairs, which are usually heterogeneous in appearance, is needed to find the orientation of the plane. In order to select the keypoints comprising a pair we chose Scale-Invariant Feature Transform (SIFT) [11].

The method works in several steps. First, the SIFT algorithm is used to detect candidate interest points within the image that are highly distinctive. Opposite to their original purpose of finding correspondences in *pairs* of images, these keypoints are, then, matched to other keypoints of different scales within the *same* image. Each potential pair consists of points that are similarly looking and are, necessarily, coming from different scales. Thus, we try to find similar features in the image that are located at different distances from the camera and, due to the foreshortening effect of perspective projection, are of different size in the image. We use the scales and positions of those keypoints in the pair to triangulate the position of an individual vanishing point in the image. Having identified a number of such vanishing points we estimate the position of the vanishing line (see Figure 1 (a)).

The appeal of such method is in its potential to estimate the plane's orientation in real images with very difficult stochastic textures. The method does not require prior knowledge of appearance of individual similar textels that comprise an image, avoiding the segmentation step of traditional structural methods.

2 Algorithm

The basic idea to identify the vanishing line of a plane under perspective projection comes from the properties of the intersection of two sets of parallel lines [9]. Each set of parallel lines intersects at the infinity point, which, under perspective projection, is a vanishing point, located on the vanishing line. In classical approaches of plane's orientation estimation, special structural elements (textels) are used with the assumption that the their sizes are the same and the lines drawn along their sides are parallel. We adopt a similar approach for vanishing point triangulation. However, instead of using pairs of textels we use pairs of scale-invariant keypoints. Such keypoints provide very intuitive locations, scale and orientations of keypoints which can be matched with high probabilities with other keypoints.

For geometrical computations, we make an assumption that each keypoint in the pair is of equal size under orthographic projection. Thus, under perspective projection, the position and sizes of two similar keypoints in the image can be used for computing a single vanishing point on a vanishing line. The size of matched keypoints is proportional to the scale where the keypoints are detected.

Two algorithms, SURF [1] and SIFT have been tested on a number of images of ocean surface. However, the results in this work are only reported for the SIFT algorithm.

The SIFT keypoints are only partially invariant to affine transformations [11] and the percentage of correctly matched points drops when significant affine transformation

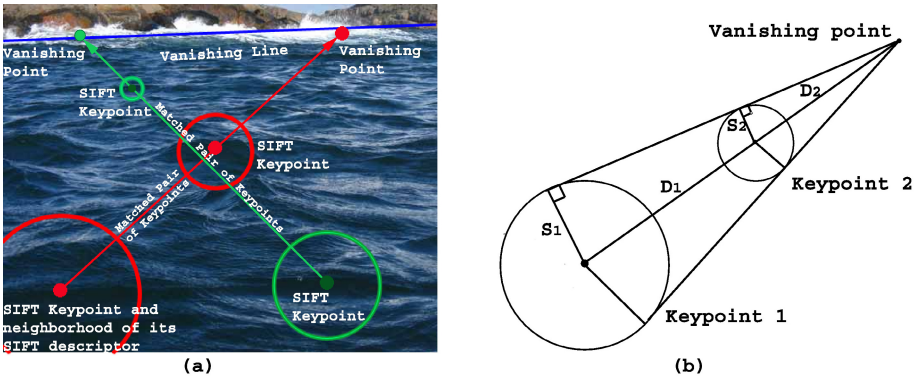

Fig. 1. Estimation of the vanishing line of the textured plane from pairs of matched SIFT keypoints. (a) Four SIFT keypoints with the regions (circles) under their descriptors are shown in the image. After matching by appearance of keypoints' descriptors two pairs are identified (each matching pair of keypoints is shown in its own color) and are used to estimate two vanishing points. The vanishing line is estimated through the locations of the vanishing points. (b) Geometry of vanishing point triangulation. Using the scales S_1 and S_2 on which the two matched keypoints were found and the distance D_1 between the keypoints it is possible to find the location of one vanishing point.

occurs. Thus, one may argue that it is impossible to compare keypoints of the same image features because they would be distorted by projective transformation. However, if keypoints in the matched pair are showing high similarity they are detected at locations in the image that are not distorted too much, otherwise the match would not have occurred.

Individual vanishing points computed from pairs of matched keypoints provide an imprecise location of true points on a line at infinity because of instability of the appearance of a stochastic texture. Another factor for imprecise localization of vanishing points comes from the fact that the scales on which the keypoints are found are selected from a limited number of scales in a scale pyramid (which depends on a number of layers and octaves), and, thus, do not represent a continuous range. However, a substantial number of such 'weak' vanishing points provides meaningful information about the vanishing line orientation. Some pairs of matched keypoints may not be pointing to the vanishing line at all, because the stochastic texture does provide the conditions for such pairs of keypoints to exist. However, these outliers can be removed from consideration using several geometrical constraints.

The following term is used to describe triangulation of a vanishing point. The vector that starts at the keypoints with bigger scale, goes through the keypoint with smaller scale, and ends at the vanishing point is called a *supporting vector* for the vanishing line (see Figure 1 (b)). The term reflects the idea of a feature in the image that supports the hypothesis of a specific vanishing line and is not related to *support vector* from the *Support Vector Machine* learning algorithm.

The constraints listed below are geometric rules used in order to estimate the locations of vanishing points and to filter out outliers:

1. A vanishing line of the plane can only be estimated if the matched keypoints in the image come from the texture of a surface plane. Thus, those keypoints that come from features of the image that do not belong to the textured plane should not be considered.
2. Keypoints in a matched pair should be of different scales. Difference in scale and the position of keypoints allows triangulation of a vanishing point.
3. Keypoints in a pair should be located far enough from each other that the regions under their descriptors do not intersect.
4. The vanishing point from a supporting vector with an angle to the gradient of perspective distortion (which is perpendicular to the vanishing line) above the maximal threshold, $A > T_A$, need to be disregarded from vanishing line computation. The reason behind it is that when the supporting vector is at a large angle to the gradient both keypoints from which the vanishing point is computed are located at approximately same distance from the camera. Thus, the scale change caused by perspective distortion is small and susceptible to error caused by the noise.
5. The supporting vector from a pair of matched keypoints should not point away from the estimated vanishing line. If, for a particular estimated vanishing line, the supporting vector points away, the vanishing point of such supporting vector should be considered an outlier.

The algorithm for estimating the vanishing line consists of two steps. First, the keypoints that are found in the image are matched to other keypoints in the image to generate vanishing points. The vanishing points are used to estimate the orientation of the vanishing line. Second, the estimated orientation of the vanishing line is used in a search for the position of the line that satisfies the geometrical constraints (4)-(5) and maximizes the optimization criterion.

2.1 Estimating Direction of Vanishing Line

The keypoints are selected by directly applying the SIFT keypoints detection algorithm on the raw image data. Once the keypoints are detected, their corresponding SIFT descriptors are computed and matched to the descriptors of other keypoints in that image. Since there are n^2 matches in the image with n keypoints, and most of them are of no interest, an effective constraint on the distance in the similarity space needs to be imposed to reduce the number of matches. In this work, we used a k-nearest neighbor in appearance space to select a 10 closest neighboring keypoints. The choice of selection of such a constraint as opposed to the radial distance in similarity space is based on the fact that some keypoints are very distinctive, and when unrestricted in number of neighbors, these points will dominate in the set of matches, skewing results for vanishing line.

In order to increase the accuracy of vanishing point localization, the number of scales in the scale pyramid is increased as compared to default values proposed in the description of SIFT algorithm [11] (see Table 1). The only indirect parameter during the SIFT keypoint localization used in our algorithm is the number of keypoints per image. This number drives the selection of other parameters for SIFT, such as the keypoint response threshold.

The matches selected from the previous step are checked against the constraints (1)-(3) listed above. The matches that are not consistent with the constraints are ruled out

from further consideration. The remaining matches are sorted by the similarity measure and the top 10% of the matches are used to generate (triangulate) vanishing points.

Figure 1 (b) shows the geometrical interpretation of triangulation of a vanishing point from a pair of matched keypoints. Let the distance between the locations of matched keypoints in the image be D_1. The radiuses for regions under the SIFT descriptors are proportional to the scales where the matched keypoints were found. If S_1 is the scale for the first keypoint with the bigger radius for SIFT descriptor, and S_2 is the scale for the second keypoint with the smaller radius, the distance D_2 from the second keypoint to the vanishing point is found as the following:

$$D_2 = \frac{D_1 S_2}{S_1 - S_2} \tag{1}$$

Using (1) the positions of all vanishing points are computed (see Figure 2). Having a set of vanishing points it is possible to estimate the line that best models the set using the sum of least squares as a optimization function. The line is found using the linear square regression. However, since the position of a vanishing point is subject to error in both dimensions, the orthogonal (or total) [8] linear least square fit is more appropriate than the regular linear square regression. The implementation of the algorithm used singular value decomposition to find the total least square solution on which the orthogonal linear least square fit is based, as described in [8]. The fit of the vanishing line to the points is done twice. First, the fit is performed on all vanishing points found in the previous step. After such initial computation, 20% of the vanishing points that have the biggest distance from the computed line are removed as outliers and the line parameters are recomputed on the remaining points. The found line is represented in a slope-intercept form $y = mx + b$, where m is the slope and b is the intercept.

The computed line can be used as the final estimate of the vanishing line. However, if the orientation of such line may be close enough to the orientation of the real vanishing line in the image, the positional accuracy can be improved.

2.2 Estimating Position of Vanishing Line

The nature of stochastic texture allows for conditions where matched SIFT features in an image look very similar, however, do not correspond to the features of texture what would be of the same size when they are imaged orthographically. Although the number of such matches is small compared to the number of correct matches, it is still important to filter them out from consideration in order to improve accuracy of the vanishing line estimation. To remove such outliers the geometrical constraints (4)-(5) are applied as described below.

For each supporting vector the angle between between it and the vector of the gradient of perspective distortion is computed and compared with the allowed threshold. The vanishing points of those supporting vectors that make an angle of more than T_a degrees from the gradient are removed from consideration for vanishing line position determination. In order to obtain the vector of perspective gradient the following procedure is applied. First, since the vector of perspective gradient is perpendicular to the vanishing line (as shown in [3]) the slope of such perpendicular is found as $m_\perp = \frac{1}{m}$,

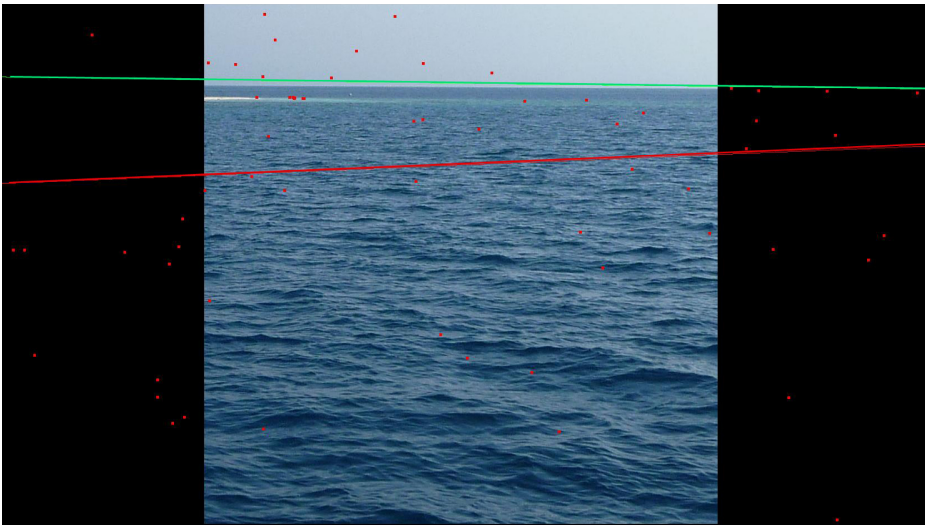

Fig. 2. Estimating direction of the vanishing line from a ensemble of vanishing points. The green line shows the ground truth. The red line shows the line estimated from the ensemble of vanishing points (red dots in the image) using total least squares algorithm. The vanishing points that are located far away from the estimated line are outliers.

where m is the slope of the vanishing line. Second, to convert the slope into a vector all the supporting vectors are projected onto the slope of the perpendicular and the projections are summed. The sign of such sum will indicate the direction along the slope as the direction of the gradient vector.

The orientation of the vanishing line found in the previous step is used in the search for the second parameter of the line - its position. The implementation of such search is done as following. The vanishing line in slope-intercept form is converted into normal representation of a line:

$$\rho = x \sin \Theta + y \cos \Theta \tag{2}$$

where ρ is the distance from origin of coordinate system to the estimated line and Θ is the angle between the normal to the line and x-axis. During the search , the Θ parameter is fixed, and the value of ρ is varied.

A special criterion is used in order to find the optimal ρ. The criterion minimizes the sum of absolute distances from the estimated line to the set of vanishing points from supporting vectors that make an angle of less than T_a degrees:

$$\rho = \arg_\rho min \sum_i D(L_{\Theta,\rho}, V_i) \tag{3}$$

where $L_{\Theta,\rho}$ is the vanishing line in a normal form parameterized by Θ and ρ, V_i is the vanishing point i, supporting vector of which makes an angle with the perspective gradient less than T_a degrees.

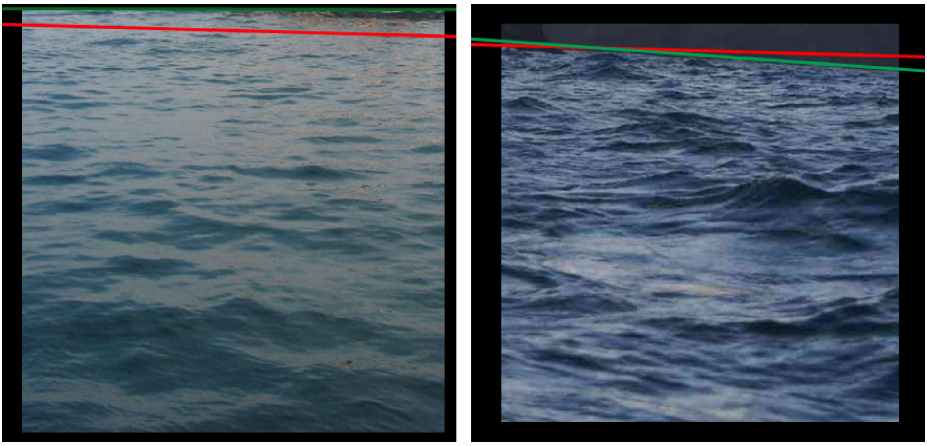

Fig. 3. Examples of vanishing line detection on real images of water surface. The green line shows the groundtruth. The red line shows the output of the algorithm.

3 Experiments

A special dataset of 83 real images of ocean surface was collected using Google Images search engine to evaluate the performance of the algorithm. Each image in the set contained a scene of ocean surface with significant perspective effect present. The images in the dataset were assigned the groundtruth and preprocessed according to the following rules. First, each image forms approximately a square in its dimensions so there is no bias in orientation due to the aspect ratio of the image. Second, each image was mostly showing ocean surface to minimize the chance that the keypoints come from the features of the image not related to the texture. The dataset of original 83 images was split into the $development\ set$ and the $test\ set$. The development set included 20 images and was used to tune the parameters of the algorithm. The test set included 53 images used to evaluate the performance.

The performance of the vanishing line estimation is evaluated in units directly related to the line parameters. The chosen metric is composed of average absolute error in the Θ angle and average absolute error in the normalized distance from the origin ρ_N. The normalized distance ρ_N is obtained for each image by dividing ρ of the evaluated line by the length of diagonal of the image. Such normalization allows comparison of errors in images of different sizes.

The Table 1 shows the values of parameters in the algorithm obtained experimentally in order to maximize performance on the development set. The performance of the algorithm with parameters from Table 1 was evaluated on the independent test set of 53 images. Table 2 shows the results of performance evaluation according to two setups. In the first setup, the vanishing line orientation and position are estimated in one step as described in Section 2.1 (thus, without correction of the position of the line). In the second setup, two step approach is applied: first, estimation of the orientation of the line, and then, correction of the position. Figure 3 shows some examples of images

Table 1. Values of parameters in the algorithm for vanishing line detection

Parameter name and description	Value
Number of layers in scale pyramid for SIFT	6
Number of octaves in each layer for SIFT	6
Number of SIFT keypoints	500
Length of SIFT descriptor vector	128
Number of nearest neighbors for each keypoints to consider when matching	10
Proportion of scales in a matched pair of keypoints to consider for supporting vector	1.5
Top percentage of matches by similarity distance to consider for vanishing line orientation	30%
Percentage of outliers in vanishing line orientation estimation	20%
Maximum angle deviation between the current gradient of perspective distortion and a valid supporting vector, T_a	45 degrees

Table 2. Performance of vanishing line estimation according to two setups

First Setup - Estimation of orientation and position of the vanishing line in one step

Average absolute error Θ, degrees	7.84
Standard deviation, error Θ, degrees	10.20
Average absolute error ρ_N	0.23
Standard deviation, error ρ_N	0.19

Second Setup - estimation of orientation and position in the first step, correction of position of the vanishing line in the second step

Error ρ_N	0.18
Standard deviation, Error ρ_N	0.15

with identified vanishing line. A good accuracy of vanishing line orientation estimation (average absolute error of 7.84 degrees) was achieved. The error in the line position estimation was 0.23 in the first setup, and 0.18 in the second. The images of the real stochastic texture are difficult to work with for all Shape From Texture methods, and, thus, the current performance is very promising. With structural methods, for example, it is impossible to define a textural element for such textures, which makes them inappropriate for the task. With spectral methods it is hard to define the scale which can be used to compare neighborhoods in the image to extract meaningful changes in the perspective gradient.

4 Conclusions

This paper describes a direct and simple method for estimating the vanishing line of a stochastically textured plane in real world images. As an example of such textured plane we used images of ocean surface. The described algorithm does not require presegmentation of textels, but uses the SIFT detector to find pairs of correspondent image

features satisfying special properties, and, thus, can be applied directly to raw images. In our initial experiments on a number of real world images of the ocean surface, the proposed algorithm demonstrates promising performance and has a potential for further improvement. Images and groundtruth used in the experiments will be available on our web-site.

References

1. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
2. Blostein, D., Ahuja, N.: Shape from texture: Integrating texture-element extraction and surface estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence 11(12), 1233–1251 (1989)
3. Criminisi, A., Zisserman, A.: Shape from texture: homogeneity revisited. In: Proceedings of British Machine Vision Conference, pp. 82–91 (2000)
4. Ettinger, S.M., Nechyba, M.C., Ifju, P.G., Waszak, M.: Vision-guided flight stability and control for micro air vehicles. In: International Conference on Intelligent Robots and Systems, vol. 3, pp. 2134–2140 (2002)
5. Farid, H., Kosecka, J.: Estimating planar surface orientation using bispectral analysis. IEEE Transactions on Image Processing 16(8), 2154–2160 (2007)
6. Fefilatyev, S., Goldgof, D., Shreve, M., Lembke, C.: Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system. Ocean Engineering 54, 1–12 (2012)
7. Galasso, F., Lasenby, J.: Shape from texture via fourier analysis. Advances in Visual Computing, 803–814 (2008)
8. Golub, G.H., Loan, C.F.V.: Matrix computations, vol. 3. Johns Hopkins Univ Press (1996)
9. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision, vol. 2. Cambridge University Press (2000)
10. Kwon, J., Hong, H., Choi, J.: Obtaining a 3-d orientation of projective textures using a morphological method. Pattern Recognition 29(5), 725–732 (1996)
11. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
12. McGee, T., Sengupta, R., Hedrick, K.: Obstacle detection for small autonomous aircraft using sky segmentation. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, ICRA 2005, pp. 4679–4684 (2005)
13. Ribeiro, E., Hancock, E.R.: Estimating the 3d orientation of texture planes using local spectral analysis. Image and Vision Computing 18(8), 619–631 (2000)
14. Ribeiro, E., Hancock, E.: Estimating the perspective pose of texture planes using spectral analysis on the unit sphere. Pattern Recognition 35(10), 2141–2163 (2002)
15. Witkin, A.: Recovering surface shape and orientation from texture. Artificial Intelligence 17(1-3), 17–45 (1981)