ROBUST SKYLINE EXTRACTION ALGORITHM FOR MOUNTAINOUS IMAGES

Sung Woo Yang, Ihn Cheol Kim and Jin Soo Kim
3-1-2, Agency for Defense Development, Yuseong P.O. BOX 35, Daejeon, Republic of Korea

Keywords: Skyline extraction, mountainous images, canny edge images, skyline candidate pixel.

Abstract: Skyline extraction in mountainous images which has been used for navigation of vehicles or micro unmanned air vehicles is very hard to implement because of the complexity of skyline shapes, occlusions by environments, difficulties to detect precise edges and noises in an image. In spite of these difficulties, skyline extraction is a very important theme that can be applied to the various fields of unmanned vehicles applications. In this paper, we developed a robust skyline extraction algorithm using two-scale canny edge images, topological information and location of the skyline in an image. Two-scale canny edge images are composed of High Scale Canny edge image that satisfies good localization criterion and Low Scale Canny edge image that satisfies good detection criterion. By applying each image to the proper steps of the algorithm, we could obtain good performance to extract skyline in images under complex environments. The performance of the proposed algorithm is proved by experimental results using various images and compared with an existing method.

1 INTRODUCTION

Skyline extraction is similar to a segmentation problem which partitions the image into the sky and non-sky areas. The skyline extraction in mountainous images is very useful in that we can obtain many local spatial features from skyline that hardly change even though time goes by. It is used for navigation of vehicles or micro unmanned air vehicles (Ettinger et al., 2002; Messi, 2003; Truchetel, 2006). And it can also be used for rendering cartographic data, rendering self-shadowing textures, accelerating flight simulation, visualizing scientific data, path planning to avoid detection etc (Stewart, 1998). But it is very difficult to extract skyline from mountainous images because of compexity and diversity of the skyline and influence by the noise caused by complex environments. Clouds, fog and backlight by the sun in an outdoor environment make the skyline ambiguous. And they also make it hard to extract skyline. Because of these difficulties, skyline extraction in mountainous images is one of the most difficult problems to solve in computer vision fields.
There are two approaches to skyline extraction. One is region-based approach which uses the characteristics of images that the sky often occupies the upper part of an image (Fang et al., 1993; Stein el al., 1992; Cozman et al., 1997, 2000). And the other is edge-based approach which uses the fact that skyline can be regarded as a boundary between two distinctive regions (Talluri and Aggarwal, 1992; Lie et al.,2005; Woo et al., 2005).
The approach proposed in Fang et al.'s work (1993) uses the threshold to find the skyline. They calculate the threshold using ten small sub-windows and the contrast of the image. After the threshold is determined, a vertical-line search from top to bottom is performed. Then, the pixels whose intensity is below the threshold are determined as skyline. Their approach has weak robustness when the image has complex environments, such as clear clouds above the skyline. Cozman et al.'s approach(2000) also uses vertical-line search. However, they use the smoothed intensity gradient image. This approach also has the drawback of weak robustness for complex environments. In the Stein et al.'s work (1992), they find the skyline using the segmentation method which segments the sky and the ground. But they don't mention the crucial segmentation step, and general segmentation methods have a limitation to find exact skyline. Talluri and Aggarwal(1992) use gradient value to extract the skyline based on edge-based approach. But their approach is too simplified to be applicable to complex images. A more practical approach is advocated by Woo et al. (2005) and Lie et al.(2005). They also use edge-based approach. In Woo et al.'s work, they use Dynamic Programming using contrast cost and

Woo Yang S., Cheol Kim I. and Soo Kim J. (2007).
ROBUST SKYLINE EXTRACTION ALGORITHM FOR MOUNTAINOUS IMAGES.
In *Proceedings of the Second International Conference on Computer Vision Theory and Applications - IFP/IA*, pages 253-257

homogeneity cost. In Lie et al.'s work, they use DP utilizing edge image, vertex cost and link cost. Unfortunately, DP has its limitation to extract skyline in that false skyline can be extracted due to the dependence on cost equation. A skyline candidate point is essential for using DP but the papers don't provide exact method to find it.

In this paper, we combine region-based and edge-based approach to extract skyline. The paper contains two fundamental contributions. The first is that this research presents the solution to the problem of extracting skyline using two-scale canny edge images. The second is the development of robust algorithm to extract skyline using the advantages of each canny edge images, the characteristics of the images and a proper linking algorithm. Detailed steps and experimental results are explained in the following sections.

2 TWO-SCALE CANNY EDGE IMAGES

Canny edge detector (Canny, 1986) is one of the most useful edge detectors in today's computer vision community and it is optimal in a mathematical sense. In spite of these good reputations, it is not adequate for skyline extraction. Because it has Localization-Detection Tradeoff (Trucco and Verri, 1998; Forsyth and Ponce, 2003). Good detection means the detector must minimise the probability of false positives. Good localization means that the detected edges must be as close as possible to the true edges. Optimal edge detector must satisfy both criteria but canny edge detector cannot improve both criteria simultaneously. It depends on the smoothing filter's scale: σ_x , σ_y and threshold values at the Hysteresis Thresholding step in canny edge detector: τ_h , τ_l . By adjusting these parameters, we can improve only one criterion, either good localization or good detection.

We cannot improve two criteria simultaneously. However, if we use each criterion in proper steps, we can obtain the same effect as improving both criteria simultaneously. So we make two-scale canny edge images. The first image satisfies good detection criterion. And the second image satisfies good localization criterion.

Figure 1 explains two sale canny edge images. We can obtain the center images of the Fig. 1 by setting the four parameters $(\sigma_x\ \sigma_y\ \tau_h\ \tau_l)$ as (4, 4, 20, 1) and the right images by setting them as (3, 3, 5, 0).

We can find out that the right images have more detailed edge segments than the left images. We will define the left image as High Scale Canny edge image and the right image as Low Scale Canny edge

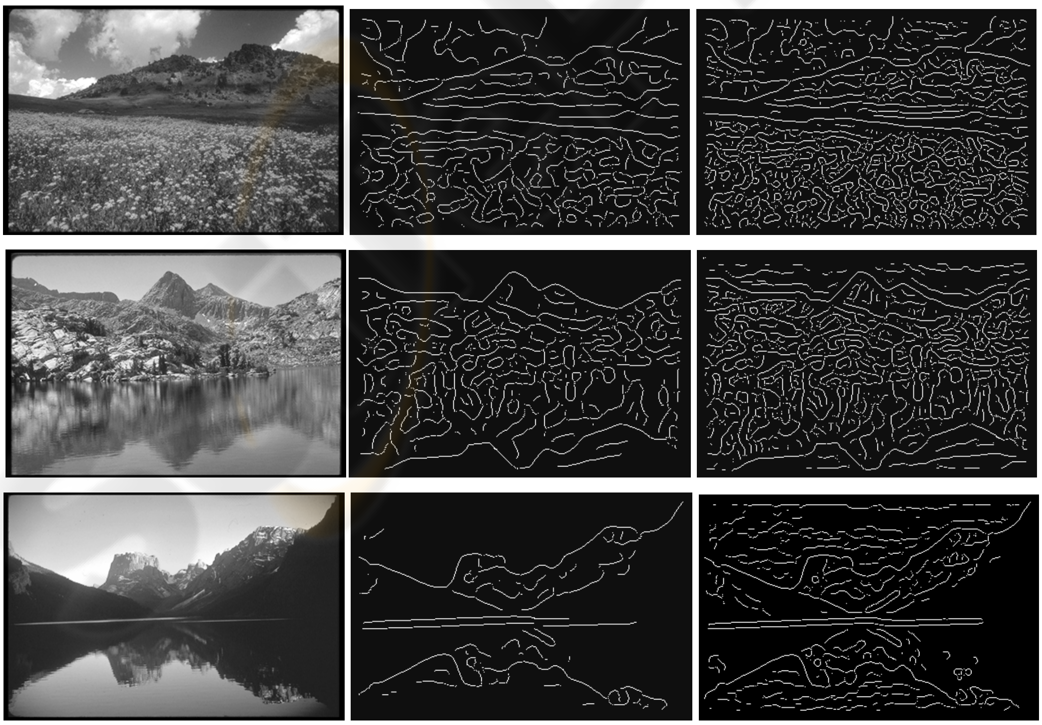

Figure 1: Three sample images in our image database and the corresponding two-scale canny edge images.

image. We will use High Scale Canny edge image for the processes which require good localization and Low Scale Canny edge image for good detection.

3 SKYLINE EXTRACTION

In this section, we introduce skyline extraction algorithm using two-scale canny edge images. There are two steps for skyline extraction. The first step is the seed selection and the second is skyline search. The seed is a definite point that we can think as a skyline candidate point. From this point, we can search the whole skyline using feasible search algorithm. In the subsequent sections, we explain detailed procedures.

3.1 Seed Selection

We need to find the skyline from many edge segments in the canny edge image. For this, we look for a definite skyline candidate point at first. We use the topological information that skyline in the mountainous image has a breakpoint at top of a mountain, and the location of the skyline in an image which may occupy the upper part of an image.

To find the breakpoint, we use a maximum point and two local minimum points of an edge segment. We define a maximum point as a point that has the highest y position in an edge segment. From the maximum point, we search an edge segment. A first point that has the minimum y position is the local minimum point. We can get two local minimum points from left and right side of the maximum point. We can make two lines from maximum point to left local minimum point, and from maximum point to right local minimum point. Joint angle between these two lines determines validity of the seed point. Considering the fact that the skyline usually pose in upper part of an image, we search the canny edge image from top to bottom and select a first maximum point of an edge segment that satisfies the specific joint angle as the seed. We process this step in High Scale Canny edge image. Fig. 2 shows the seed selection. Three white dots in the image mean left local minimum point, the maximum point and the right local minimum point.

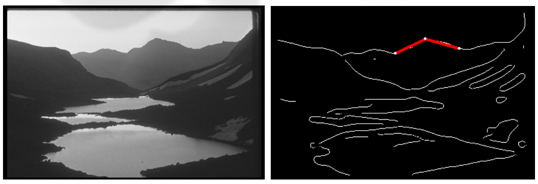

Figure 2: The example of seed selection.

We can observe advantages of adopting High Scale Canny edge image for finding the seed. High Scale Canny edge image only has strong edges that have high gradient values. So if there are not strong noises in the sky, for example thick clouds, there are not severe noises enough to disturb finding the seed in High Scale Canny edge image. If we use Low Scale Canny edge image, we may not find the appropriate seed. Fig.3 shows this situation. The first row is the original images. The second row is the seed found in High Scale Canny edge image and the third row is the seed found in Low Scale Canny edge image.

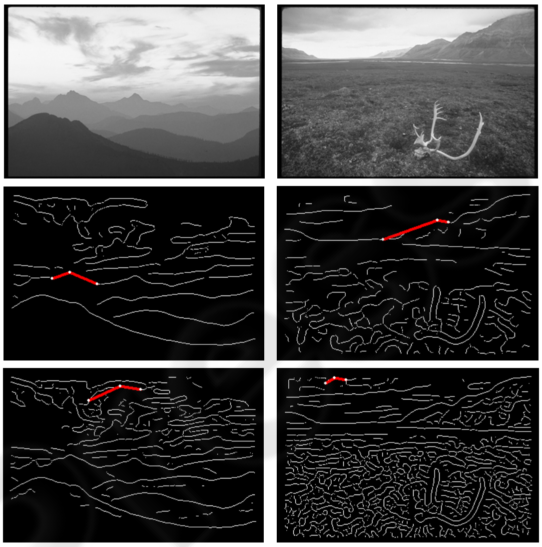

Figure 3: Seed selection using High Scale Canny edge image: the second row, and Low Scale Canny edge image: the third row.

The disturbance of the noises that exists even in High Scale Canny edge image can be overcome by a simple verification step. Using the found seed, we search the skyline. If the length of the extracted skyline is below specific threshold, we decide that this seed is not adequate, and find other seed from that position. In this way, we can find the pertinent seed.

3.2 Skyline Search

We start to search whole skyline from the seed found in previous stage. Skyline search is processed along an edge segment of the seed exists, and at first we search the right side of the seed. We search right 5-neighbor of the seed. If there is an edge pixel in this region, we set this point as a skyline candidate pixel. If there is no edge pixel, we search additional 9-neighbor of the seed. Fig. 4 explains skyline search algorithm. Black pixels are the skyline candidate pixels and dashed boxes show 5-neighbor and additional 9-neighbor of the skyline candidate points. Grey pixels in the dashed box are not selected as skyline candidate pixels because additional 9-neighbor search is followed by 5-neighbor search. If we find an edge pixel in 5-neighbor, we do not search additional 9-neighbor. This procedure makes the skyline search algorithm more accurate. If there is not a skyline candidate pixel, we search reasonable area around previous skyline candidate pixel. When there is an edge pixel, we continue the search process from this pixel. If we cannot find a pixel in this area, we think that the last skyline candidate pixel is the end of the skyline, and stop the search process. Search process is equally done on the left side of the seed.

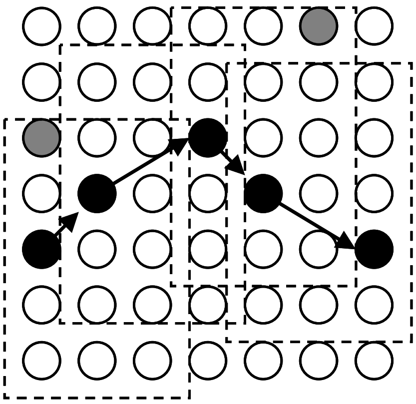

Figure 4: Skyline search using 5-neighbor and additional 9-neighbor of the skyline candidate points.

Skyline search is processed in Low Scale Canny edge image. It is not proper for the seed selection but for skyline search. Because Low Scale Canny edge image contains weak edge components that have weak gradient values as well as strong edge components. Fig. 5 explains the suitability of Low Scale Canny edge image for skyline search process. The second image is the result of the skyline search using High Scale Canny edge image, and the third image is the result using Low Scale Canny edge image. We can see that the skyline of the third image is more exquisite than that of the second image.

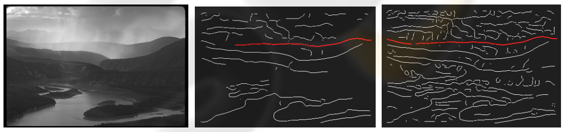

Figure 5: Skyline search using High Scale Canny edge image: the second image, and Low Scale Canny edge image: the third image.

4 EXPERIMENTAL RESULTS

In our experiments, 55 images acquired in different environments were tested. And we could extract skylines accurately in 50 images. We could extract a part of the skylines in the other 5 images. We can see that the proposed algorithm works accurately even when there are complex environments in the images. Some of the test images are shown here. The results are compared with Woo et al.'s, 2005 gradient based DP method. In Fig.6, the first column shows the original test images. The second column shows the seed using the proposed method, the third column shows the extracted skyline using the proposed method and the last column shows the extracted skyline using Woo et al.'s method. We can see that in Woo et al.'s method, it confuses the skyline and other lines that have large gradient. And it can not search the whole skyline accurately because of the noises, for example clouds and fog. But the proposed method still works well under the complex environments.

5 CONCLUSIONS

In this paper, we presented a new robust skyline extraction algorithm in mountainous images. The major contribution is the use of two-scale canny edge images and the development of new skyline extraction algorithm. Using topological information, location of the skyline in an image and a feasible search algorithm, we developed a new skyline extraction algorithm. By applying two-scale canny edge images to proper skyline extraction steps, we could overcome the defects of two images. We applied our algorithm to 55 images that have various complex environments. In about 90% of the images we could find exact skylines. The experimental results show the robustness of our algorithm under complex environments.

REFERENCES

Canny, J., 1986. A computational theory for edge detection. In *IEEE Trans. On Pattern Recognition and Machine Intelligence,* vol.26, no.6, pp.679-698.

Cozman, F., Krotkov, E., 1997. Automatic mountain detection and pose estimation for teleoperation of lunar rovers. In *Proc. Of the International Conference on Robotics and Automation,* pp.2452-2457.

Cozman, F. and Krotkov, E. and Guestrin, C., 2000. Outdoor visual position estimation for planetary rovers. *Autonomous Robots,* 9, pp.135-150.

Ettinger, Scott M., Nechyba, Michael C., Ifju, Peter G. and Waszak, Martin, 2002. Vision-guide flight stability and control for micro air vehicles. In *IROS '02, proc. IEEE/RSJ International Conference on Intelligent Robots and System,* pp.2134-2140.

Fang, M.., Chiu, M.-Y., Liang, C.-C. and Singh, A., 1993. Skyline for video-based virtual rail for vehicle navigation. In *Proc. IEEE International Sympos. On Intelligent Vehicles,* pp.207-212.

Forsyth, D. A. and Ponce, J., 2003. *Computer Vision A Modern Approach*, Prentice Hall, Upper Saddle River, NJ.

Lie, W.N., Lin, T.C.-I., Lin, T.-C. and Hung, K.-S., 2005.

SA robust dynamic programming algorithm to extract skyline in images for navigation. In *Pattern Recognition Letters,* 26, pp.221-230

Messi, Mark, et al., 2003. Vision chip flight stability and control for micro air vehicles, In *ISCAS '03, proc. IEEE International Conference on Circuit System,* pp.786-789.

Stein, F. and Medioni, G., 1995. Map-based localization using the panoramic horizon. *IEEE Trans. On Robotics and Automation,* 11(6), pp. 892-896.

Stewart, J.A., 1998. Fast horizon computation at all points of a terrain with visibility and shading applications. *IEEE Trans. Visual. Comput. Graphics,* 4(1), pp.82-93.

Trucco, E. and Verri, A., 1998. *Introductory techniques for 3-d computer vision*, Prentice Hall, Upper Saddle River, NJ.

Truchetel, F., et al., 2006. Attitude Measurement by Artificial Vision. *Measurement Science and Technology,* vol. 17, pp.101-110.

Talluri, R. and Aggarwal, J., 1992. Position estimation for an autonomous mobile robot in an outdoor environment. *IEEE Trans. Robotics and Automation,* 8(5), pp.573-584.

Woo, J.H., Kweon, I.S., 2005. Robust horizon and peak extraction for vision-based navigation. In *MVA'06 Proc. IAPR workshop on Machine Vision Applications*.

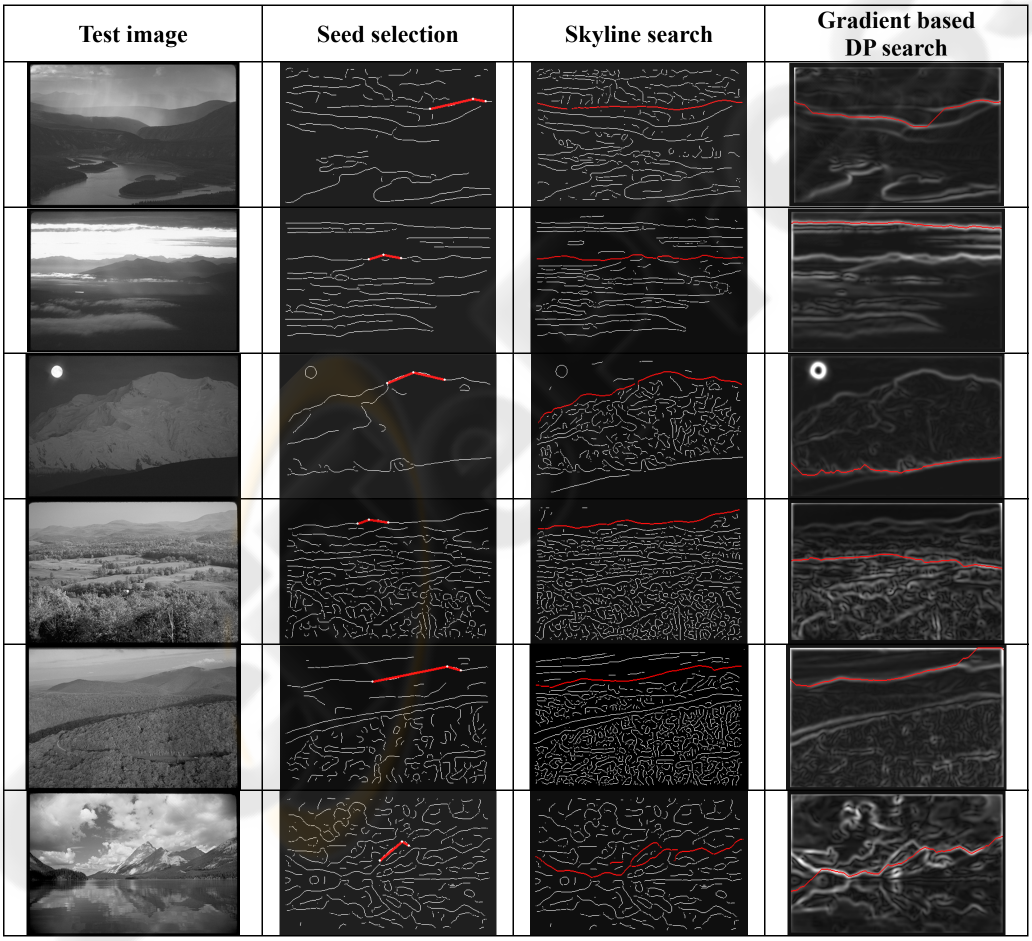

Figure 6: Experimental results for six chosen test images and the comparison with the results of Woo et al.'s method.