

# Extracting Lines in Noisy Image Using Directional Information

Jun Zhou and Walter F. Bischof  
Department of Computing Science  
University of Alberta  
Edmonton, AB T6G 2E8, Canada  
{jzhou,wfb}@cs.ualberta.ca

Arturo Sanchez-Azofeifa  
Department of Earth and Atmospheric Sciences  
University of Alberta  
Edmonton, AB T6G 2E3, Canada  
Arturo.Sanchez@ualberta.ca

## Abstract

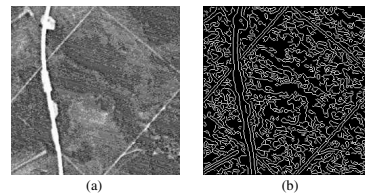
*Detection of lines in noisy image is not easy. When using Hough transform, multiple false peaks may be generated from collinear noisy edge points, which in turn may create false line segments. To overcome this problem, we introduce a method for detecting lines in noise based on directional information. Orientations are generated by Gabor filters to guide an anisotropic Gaussian filtering process, and they are also used in the peak selection in Hough transform. The experimental results show the effectiveness of this method.*

## 1. Introduction

Line detection is a fundamental task in computer vision. The Hough transform is the best known method for detecting lines [8, 9]. Usually, the Hough transform starts from edge detection based on the local differential properties of the image. Then the edge map is transformed from image space into parameter space, where collinear points in the image space correspond to peaks in the parameter space. Candidate points for line segments in the image space can be restored from these peaks. Analyzing and grouping these candidate points generates a final set of line segments.

Noise strongly affects line detection in the Hough transform. This is illustrated in the example in Figure 1(a), where the task is to find the four straight line segments that form a rectangle. These lines are roads through a forest as seen in an aerial photo. First, edge detection in this image is difficult because noise cannot be completely removed in the smoothing step. For example, the Canny edge detector [1], which often gives the best performance in edge detection, may not find a suitable scale to accurately detect and localize edges while efficiently removing the noise when the image is smoothed isotropically. This is shown in Figure 1(b) in which many edges are detected in the background forest area. Second, the noisy edges generate noisy peaks

in the parameter space. These noisy peaks may be mixed with a butterfly pattern (see below) of transformed line segments, leading to the generation of false peaks. It is difficult to distinguish true peaks from false peaks, which, in turn, prohibits correct grouping of candidate points into true line segments.



**Figure 1. Line detection in noisy image. (a) An aerial photo with lines to be detected in noisy background. (b) The Canny edge detection result on (a).**

Several methods have been proposed to solve these problems, such as anisotropic filtering in edge detection [10, 6], peak analysis in Hough transformation [11, 5], and grouping of directional features [2].

We propose a new method for effectively solving these problems using directional information. First, a Gabor filter is applied to the image to compute the dominant local orientation at each pixel. Then anisotropic Gaussian filtering is performed before edges are detected using the Canny edge detector. In this way, the noisy edges are reduced. Second, in the Hough transform, peaks are calculated from the edges and line segments are restored. Then a peak selection algorithm based on directional information is used to distinguish true line segments from the line segments generated from spurious noisy edges.

## 2. Anisotropic Filtering for Edge Detection

The purpose of anisotropic filtering is to effectively reduce the influence of noise while accurately detecting edges. In isotropic filtering, noise is smoothed in the same way in all directions. To reduce the side-effect of noise along edges or lines, one can take advantage of the directional property of these linear features. This requires fine-tuning the filter in the direction of the lines. To get the dominant local orientation, we used a Gabor filter. Then the image was smoothed anisotropically using the orientation map, followed by edge detection using the Canny edge detector.

### 2.1. Estimating Local Orientation

The Gabor filter is a bandpass filter [3]. A 2-D Gabor filter is a Gaussian envelope modulated by a complex sinusoidal carrier. An even-symmetric Gabor filter is given by [7]:

$$g_t(x, y) = \exp\left(-\left(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2}\right)\right) \cos\left(\frac{2\pi x}{t}\right) \quad (1)$$

where  $t$  defines the wavelength of the Gabor filter,  $(\sigma_x, \sigma_y)$  defines the scale of the Gaussian envelope along the  $x$ -axis and  $y$ -axis respectively.

We may change the orientation of Gabor filter to angle  $\theta$ , so that

$$g_{t,\theta}(x, y) = g_t(x', y') \quad (2)$$

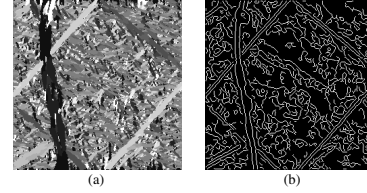
where  $x' = x \cos \theta + y \sin \theta$  and  $y' = -x \sin \theta + y \cos \theta$ .

To implement this even-symmetric Gabor filter, we need to determine parameters  $t$ ,  $\sigma_x$ ,  $\sigma_y$ , and  $\theta$ .  $t$  is a parameter related to the edge frequency, and was set to 5, while  $\sigma_x$ ,  $\sigma_y$  were both set to 3. To define the number of orientations, we set  $\theta$  to 16 directions uniformly distributed in  $[0, \pi)$ . The input image was filtered iteratively on these directions. The final orientation map takes the maximum response for each pixel over all directions. An example of the extracted orientation map of Figure 1(a) is shown in Figure 2(a). The dominant orientation at each pixels is displayed in different greylevels and is used to guide the anisotropic filtering of the original image.

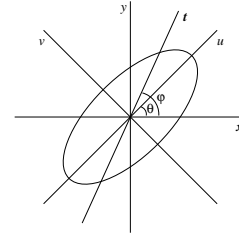
### 2.2. Anisotropic Filtering

A fast anisotropic Gaussian filter has been proposed by Geusebroek and Smeulders [6]. The oriented filter in two dimensions is given by the convolution of two Gaussian filters:

$$g(u) = \frac{1}{\sqrt{2\pi}\sigma_u} \exp\left(-\frac{u^2}{2\sigma_u^2}\right) \quad (3)$$



**Figure 2. Anisotropic filtering and edge detection. (a) Orientation map from Gabor filtering on figure 1(a). (b) Canny edge detection on anisotropically smoothed image.**



**Figure 3. Anisotropic Gaussian filter.**

and

$$g(v) = \frac{1}{\sqrt{2\pi}\sigma_v} \exp\left(-\frac{v^2}{2\sigma_v^2}\right) \quad (4)$$

where  $u = x \cos \theta + y \sin \theta$  and  $v = -x \sin \theta + y \cos \theta$ . Here,  $\theta$  is the orientation of the anisotropic Gaussian filter,  $x$  and  $y$  are the cartesian coordinates of the image pixels.

To facilitate the computation, the filter is transformed into image coordinates along the  $x$ -direction and  $t$ -direction, as shown in figure 3. The transformed filters are:

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left(-\frac{x^2}{2\sigma_x^2}\right) \quad (5)$$

and

$$g(t) = \frac{1}{\sqrt{2\pi}\sigma_t} \exp\left(-\frac{t^2}{2\sigma_t^2}\right) \quad (6)$$

where  $t = x \cos \phi + y \sin \phi$ . The relationship between  $\sigma_x$ ,  $\sigma_\phi$  and  $\sigma_u, \sigma_v, \theta$  is:

$$\sigma_x = \frac{\sigma_u \sigma_v}{\sqrt{\sigma_v^2 \cos^2 \theta + \sigma_u^2 \sin^2 \theta}} \quad (7)$$

$$\sigma_\phi = \frac{1}{\sin \phi} \sqrt{\sigma_v^2 \cos^2 \theta + \sigma_u^2 \sin^2 \theta} \quad (8)$$

$$\tan \phi = \frac{\sigma_v^2 \cos^2 \theta + \sigma_u^2 \sin^2 \theta}{(\sigma_u^2 - \sigma_v^2) \cos \theta \sin \theta} \quad (9)$$

In this way, a 2-D anisotropic Gaussian filter can be implemented as a convolution of two 1-D Gaussians in the  $x$ -direction and  $\phi$ -direction.

### 2.3. Edge Detection

Edge detection is performed by a Canny edge detector to generate a binary edge map. However, in the Gaussian smoothing step, the traditional isotropic smoothing is replaced by anisotropic filtering introduced above. The resulting edge map is shown in figure 2(b). Compared to Figure 1(b), the number of noisy edge points was reduced while the true edges were preserved. This is due to the fact that noise different from the dominant local orientation has been further smoothed out.

### 3. Line Detection using Hough Transform

Hough transform is performed on the edge map to transform the  $x - y$  coordinates of edge points into  $\rho - \theta$  space. We used the normal parameterization proposed by Duda and Hart [4]. A line in an image is given by

$$\rho = x \cos \theta + y \sin \theta \quad (10)$$

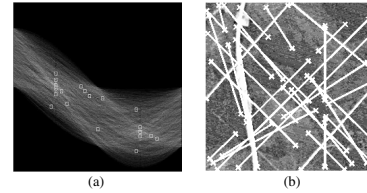
where  $\rho$  is the distance of the line to the origin, and  $\theta$  is the angle between the normal of the line and the  $x$ -axis, and  $(x, y)$  is any point on the line.

In the implementation of the Hough transform, the  $\rho - \theta$  space is normally quantized into cells. Assume that a line is mapped to a peak at point  $(\rho_0, \theta_0)$ , the collinear points on the line generate a pattern in the parameter space, and is often referred to as a butterfly [11]. The analysis on the butterfly is supposed to be restricted to a small neighborhood window  $(-w_\rho/2, w_\rho/2) \times (-w_\theta/2, w_\theta/2)$  around the peak  $(\rho_0, \theta_0)$ . The peaks in the neighborhood window can be grouped and removed in sequential peak selection [12].

One problem of this peak selection and grouping strategy is that noisy edge points may generate noisy peaks in the parameter space. For example, Figure 4(b) shows the lines restored from top 20 peaks in Figure 4(a), which was generated by Hough transform from Figure 2(b). Due to the noisy edge points, most extracted lines were false positive results and a true line was missing because it did not correspond to the top 20 peaks. Although we can increase the scale of the anisotropic Gaussian filter to further eliminate noise in edge detection, it usually cannot remove it completely, and, as a side-effect, may also remove correct line points. The noisy peaks can not be excluded without further analysis.

### 4. Peak Selection using Directional Information

An analysis of Figure 2 reveals that the orientation of line segments is helpful in peak selection. This is based on the fact that, in peaks mapped from lines, the collinear edge



**Figure 4. Hough transform and problems. (a) Top 20 Hough peaks detected from image 2(b). (b) Lines (bounded by crosses) correspond to the peaks in (a) using traditional peak selection.**

points that contribute to the peaks tend to have the same local orientation, while the noisy edge points do not. This directional information, combined with traditional line grouping method in Hough transform, can help to distinguishing true line segments from noisy ones. In this way, the peak selection can be considered as a classification procedure by a decision tree. Each node in the decision tree classifies the lines mapped from a peak in the parameter space using a certain feature.

We developed the following algorithm for the peak selection using the Hough peaks ( $P$ ) and the orientation map ( $O$ ) as input, and generating true line segments as output:

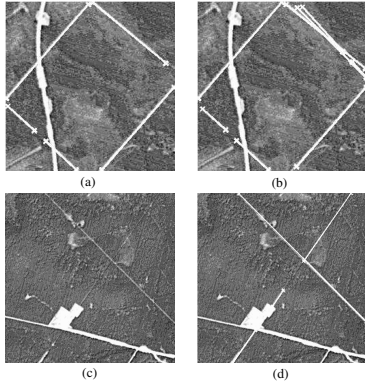
**Algorithm:** PEAKSELECTION( $P = \{p_1, \dots, p_n\}, O$ )

```

sort  $P$  according to the intensity
for each peak  $p_i$ 
  restore the contributing edge points  $EP_i$ 
   $\sigma_{p_i} \leftarrow$  average deviation on directions of  $EP_i$  from  $O$ 
  if  $\sigma_{p_i} > t_1$ 
    remove  $p_i$  and neighborhood peaks from  $P$ 
  else
    group points in  $EP_i$  with distance smaller than  $t_2$ 
    into line segments  $LS_i = \{ls_1, \dots, ls_m\}$ 
     $L_{LS_i} = \{l_1, \dots, l_m\} \leftarrow$  length of the line segments
    for each line segment  $ls_j$ 
      if  $l_j < t_3$ 
        remove  $ls_j$  from  $LS_i$ 
      end if
    end for
     $al_i \leftarrow$  average length of  $LS_i$ 
    if  $al_i > t_4$ 
      add  $LS_i$  into output
    end if
  end if
end for

```

The first step in the algorithm is the direction judgement which enables the peak suppression and removal of most



**Figure 5. Line (bounded by crosses) detection results. (a) Lines detected on figure 1(a) using proposed method. (b) Lines detected using isotropic filtering. (c) Another image sample. (d) Lines detected on (c) using proposed method.**

line segments composed of noisy edge points. The second and third step perform the point grouping, which excludes short noisy line segments. The last step removes noisy lines which are composed of short line segments that have not been successfully eliminated in the previous steps.

## 5. Experimental Results

We performed experiments on sub-images extracted from a  $7140 * 5940$  aerial photo with ground resolution of 5 meters. The system went through all the steps introduced in the previous sections. The standard deviations of the anisotropic Gaussian filter were set to  $\sigma_v = 1$  and  $\sigma_u = 3\sigma_v$ . The parameters of the Hough transform were set to  $\Delta\rho = 1$  and  $\Delta\theta = \pi/180$ . The thresholds in the peak selection and line grouping were set to  $t_1 = \pi/9$ ,  $t_2 = 20$ ,  $t_3 = 30$  and  $t_4 = 40$ , respectively. Due to space limitations, we can not include the detailed discussion of the parameter values that were used for this task.

The final result for Figure 1(a) is shown in Figure 5(a). The starting and ending points of the lines were marked by crosses. The results show that all target lines were detected successfully. Figure 5(d) shows another line detection result. Note that a short line segment is missing due to the weakness of the edges.

We compared the results of using isotropic and anisotropic filtering. In the former case, two noisy lines are detected, as shown in Figure 5(b). This is due to fact that too many noisy edge points have been detected in Figure 1(b). These edge points can not be completely removed in

the peak selection and thus generate false lines.

## 6. Conclusion

We have presented a line detection method based on directional information and Hough transform. The directional information is utilized in two stages. First, anisotropic filtering is used in the edge detection step. Each pixel in the image is smoothed according to their dominant local orientation extracted from an Gabor filter. Second, after the Hough transform, the peak selection and line segments grouping is guided by the same directional information. This method can successfully extract lines in noisy images.

## References

- [1] J. Canny. A computational approach to edge detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
- [2] M. Carreira, M. Mirmehdi, B. Thomas, and J. Haddon. Grouping of directional features using an extended Hough transform. In *Proceedings of the 15th International Conference on Pattern Recognition*, volume 3, pages 990 – 993, Barcelona, Spain, September 2000.
- [3] J. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America*, 2(7):1160–1169, 1985.
- [4] R. Duda and P. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.
- [5] Y. Furukawa and Y. Shinagawa. Accurate and robust line segment extraction by analyzing distribution around peaks in Hough space. *Computer Vision and Image Understanding*, 92(1):1–25, 2003.
- [6] J. Geusebroek, A. Smeulders, and J. Weijer. Fast anisotropic Gauss filtering. *IEEE Transactions on Image Processing*, 12(8):938–943, 2003.
- [7] L. Hong, Y. Wan, and A. Jain. Fingerprint image enhancement: algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):777–789, 1998.
- [8] J. Illingworth and J. Kittler. A survey of the Hough transform. *Computer Vision, Graphics, and Image Processing*, 44(1):87–116, 1988.
- [9] V. Leavers. Survey: Which Hough transform? *Computer Vision, Graphics, and Image Processing*, 58(2):250–264, 1993.
- [10] A. Paplinski. Directional filtering in edge detection. *IEEE Transactions on Image Processing*, 7(4):611–615, 1998.
- [11] T. Veen and F. Groen. Discretization errors in the Hough transform. *Pattern Recognition*, 14(1-6):137–145, 1981.
- [12] X. Zhang and H. Burkhardt. Grouping edge points into line segments by sequential Hough transformation. In *Proceedings of the 15th International Conference on Pattern Recognition*, volume 3, pages 672–675, Barcelona, Spain, September 2000.