Low Level Feature Detection based on Modified Ransac

Yingdi Xie† xieyingdi@ruri.waseda.jp
Jun Ohya † ohya@waseda.jp

1． Introduction

In computer vision research, all features in images can be coarsely classified into low-level features and high-level features. Low-level features can be extracted directly from the original images, whereas high-level feature extraction must be based on low-level features. Therefore, low-level features are important and critical components in a scene understanding system, the result of the extraction being the basis of the high level processes. Typical low-level features include edges [1], corners [2], and (to a lesser extent) ridges [3]. This paper concentrates on typical shape extraction.

Among previous literatures, the Hough Transform is recognized as a powerful tool in shape analysis. It is used to extract low level features, for example straight lines, and is useful despite noises and occlusions. However, the performance of the Hough Transform can be compromised due to the discrete nature of the image. Approximations are made to the true angle of a line, as digitally it will be represented by many short segments. It therefore becomes more difficult to extract the true angle from parameter space. In this paper, we propose a method that solves the above-mentioned problem of Hough Transform.

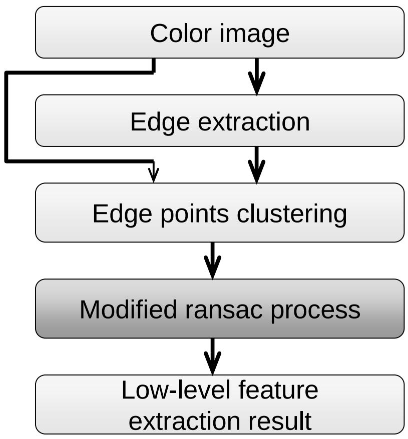

Figure 1. Block diagram of this method.

2. Computation strategy

As demonstrated in Fig. 1, our proposal for low-level feature extraction procedures includes the following steps: (1) Edge extraction; (2) Edge points' clustering; (3) Modified ransac process. We utilize the canny operator to detect edges. In order to decrease false low-level feature extraction, the edge points are then clustered against the colors of the original image and their corresponding gradient information. The detail of the clustering procedure is described in section 4. The grouped edge points are processed according to clustering result, by a modified ransac procedure. In this step, multiple low level features could be generated with the elimination of the noise points. The details are described in section 3. In section 5, experiment results for line extraction are illustrated.

† Waseda University, Global Information and Telecommunication studies

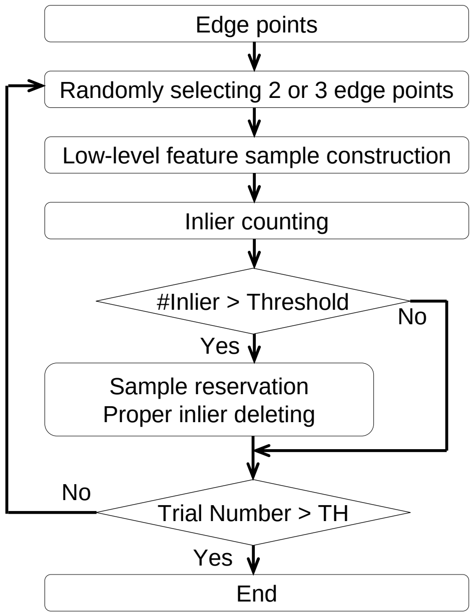

Figure 2. The computational logic of modified ransac for low-level feature detection

3. Modified ransac

In the previous work [4], a modified ransac strategy to extract multiple samples of a model under a noisy environment is introduced. The logical flow of the modified RANSAC is indicated by the following steps

(1) Randomly select a minimum data set $\boldsymbol{S_1}$, required to compute a required model from the universal set, denoted as $\boldsymbol{I}$.

(2) Compute a model $\boldsymbol{M_1}$ with set $\boldsymbol{S_1}$. Define, "proper inlier" and "Quasi-inlier" is the data whose distance to the computed model $\boldsymbol{M_1}$ is less than a predefined threshold $\boldsymbol{TH_1}$ and $\boldsymbol{TH_2}$ respectively, where $\boldsymbol{TH_1}<\boldsymbol{TH_2}$. Compute the proper inlier and quasi-inlier for all data within $\boldsymbol{I}$. Count their numbers. Denote the number of proper inlier as $\boldsymbol{N_{th1,1}}$, quasi-inlier as $\boldsymbol{N_{th2,1}}$ respectively. Notice, both proper inlier and quasi-inlier are treated as inlier to $\boldsymbol{M_1}$.

(3) If the inlier number is larger than the predefined threshold $\boldsymbol{TH_3}$ for validating the effectiveness of the computed model $\boldsymbol{M_1}$,

$$\boldsymbol{N_{th1,1}} + \boldsymbol{N_{th2,1}} > \boldsymbol{TH_3} \qquad (1)$$

Label the proper inliers with $\boldsymbol{M_1}$ and "proper" mark. The quasi-inliers are labelled with $\boldsymbol{M_1}$ and "quasi" mark. Update the universal data set $\boldsymbol{I}$ by deleting the proper inliers.

(4) Repeat step 2 and 3 until no valid model satisfying Eq. 1 can be found in a certain times of trial for the universal set ***I***, which keeps being updated. For the quasi-inliers, when compete occurs for their assignment to different labeling, treat them as the inliers of the model with smaller distance.

From above description, the above procedures reserve all desirable samples of a model in a noisy environment. Low-level features' extraction with the modified ransac processes according to the following steps. As demonstrated in Fig. 2, we randomly select minimal pixels that required constructing a sample of a certain low-level feature, for instance 2 points for line, three points for circle. A feature sample is constructed, and its validity is verified through counting the inlier number for all the edge pixels. The valid sample is recorded. Its corresponding proper inliers are labeled and then deleted from the edge points. Repeat the above step until the edge points are empty or the trial number meets pre-defined threshold.

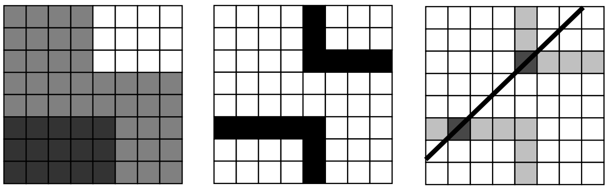

Figure 3. Left: Original image. Middle: Edge pixels which are labeled in black. Right: Example of wrong selection to build a line sample.

4 Edge points' clustering

With the modified ransac logic, we could successfully extract multiple low-level feature samples within noisy images. However, experiment shows that, processing with the raw data of edge pixels will generate redundant samples at a certain possibility. Moreover, the computational efficiency varies significantly according to the noise factor. As for the first defect, an example is shown in Fig. 3. Accordingly, we cluster the original edge points before modified ransac process. Weights and clustering result are assigned labeled to each edge point. Then, for each cluster the modified ransac process select points by a Gaussian sample with respect to the weights.

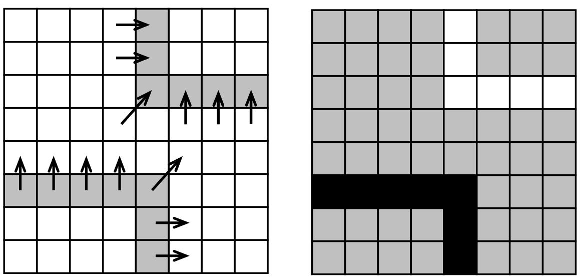

Figure 4. Left: gradient map for edge points. Right: clustering result example.

We adopt mean shift algorithm to achieve clustering. The computation is implemented in a seven dimensional feature space, which composes of $(r_p, g_p, b_p, r_{p\nabla}, g_{p\nabla}, b_{p\nabla}, grad_p)$. Where, (r_p, g_p, b_p) represents the color of edge pixel ***p***, $(r_{p\nabla}, g_{p\nabla}, b_{p\nabla})$ is the color of ***p***'s neighbor that in its gradient direction, $grad_p$ is the gradient direction for p. In Fig. 4, the left is the gradient map for edge points demonstrated in Fig. 3. The right figure is a clustering result example.

Processed with clustering, each edge point is labeled with a cluster index, and a weight which is the Euclidean distance from the point to the clustering centroid in feature space. Then, in stead of all the edge points, the modified ransac procedure starts with selecting points only from the same cluster with small weights. Therefore, the modified ransac becomes much more effective and efficient.

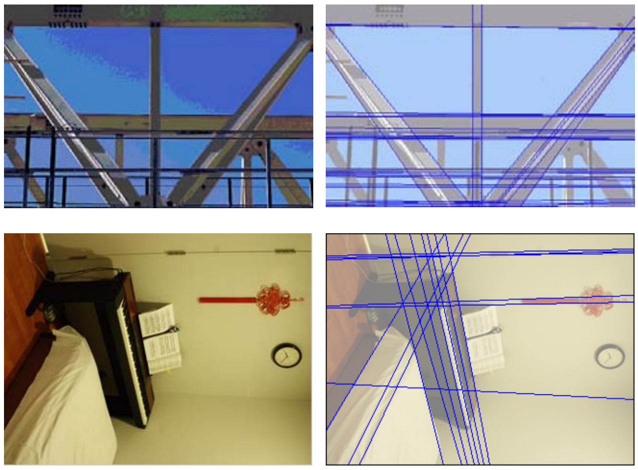

Figure 5. Experimental results for line extraction by modified ransac based method. Left: Original Images. Right: Line extraction results.

5 Experimental results and Conclusion

We conducted experiments on line detection. The results are demonstrated in Fig. 5, where the left column shows the original images, and the right column is the processing result. The results show that the proposed method is effective to detect low-level features.

In this paper, we propose a modified ransac based low-level feature detection method. It provides a solution to accuracy limitation of Hough Transform. Similarly, by providing the model equation, other low-level features within images, such as ellipse, circle, hyperbola etc, could also be extracted by this method.

References

[1] J. Canny, "A C*omputational Approach to Edge Detection*", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 8, No. 6, Nov 1986.

[2] C. J. Harris and M. Stephens. "*A combined corner and edge detector.*" Proc. 4th Alvey Vision Conferences, pages 147-- 151, 1988.

[3] R. Carmona, W. L. Hwang, B. Torr'esani, "*Multiridge Detection and TimeFrequency Reconstruction*," IEEE Trans. on Signal Processing, vol. 47, no. 2, February 1999.

[4] Yingdi Xie, Jun Ohya, "*A Modified RANSAC mechanism: Multiple models extraction algorithm*", IPSJ SIG-CVIM: Computer Vision and Image Media, Vol.2006, No.93, pp. 159-166, Sep.2006