Obstacle contour extraction method based on improved Grabcut algorithm

Jiateng Mao¹,² and Yueli Hu¹,²,³

¹ Shanghai Key Laboratory of Power Station Automation Technology, Shanghai 200072, China
² School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200072, China
³ E-mail: huyueli@shu.edu.cn

Abstract. Obstacle distance measurement has an important role in the design of mobile robots, which allows the robot to timely complete the movement steering to avoid damage caused by crushing. The accurate segmentation and edge detection of the target obstacle in the image is the premise of completing the measurement. In this paper, based on the classification and location of specific obstacles, a method of obstacle segmentation based on improved Grabcut algorithm is proposed, and the edge contour is extracted by Canny operator. The method classifies the image’s background pixels using a Laplacian Mixture Model and modifies the covariance matrix to change the probability density function of that model, thereby reducing the misclassification probability of the pixels. In addition, the method crops the input image in advance to reduce the running time of the algorithm. This paper also introduces a way to initialize the image’s Trimap, considering the visual angle of the sweeper. The above four improvements enable the method to accurately segment obstacle targets and extract contour curves. The accuracy of the method and the good real-time performance on the embedded platform are verified by comparison experiments.

1. Introduction

With the rise and development of Smart Home, more and more mobile intelligent sweeper robots appear in people's daily lives. In the design and manufacture process of these devices, the obstacle avoidance of the sweeper robot on the running route is the key to the successful operation of the machine. Once the sweeper robot fails to avoid obstacles on the floor, such as socks, slippers, data wires, etc., these objects are likely to be entangled into the sweeper and cause the drive to stop working, thus burning the motor. Nowadays, many researchers have trained neural networks models for objects detection, with good real-time performance. On this basis, in order for the intelligent sweeper robot to accurately determine the distance between itself and the obstacle, it needs to obtain the identified edge contour of the obstacle through algorithms.

At present, image segmentation and edge detection are mainly implemented in two ways. One is to use traditional algorithms in Computer Vision, and the other is to use deep learning and neural network models. Beucher et al. [1] proposed a watershed algorithm based on image region information. This algorithm regards the image as a topology of geodesy. The gray level of each pixel in the image is used to represent the altitude of the point. After initialization, a small hole is pierced in the surface of each local minimum value, and then the whole model is slowly immersed in water. As the immersion deepening, the influence of each local minimum value gradually expands outward, and the
dam is constructed at the confluence of the two collecting basins to form a watershed, thereby completing image segmentation. Boykov et al. [2] proposed the Graph Cuts algorithm, which uses a weighted undirected graph $G=<V,E>$ for each image. This algorithm relates the task of image segmentation with that of image’s minimization cut [3]. As long as the minimum value of the energy function is obtained, the optimal segmentation of the image is reached. He et al. [4] proposed the Mask R-CNN convolutional neural network in 2018. That network uses the Region Proposal Network to generate the region to be detected, and uses ROI-Align to match the features of Region of Interest with that of the original images, leading to more accurate object detection results. After training on the image dataset, it can perform object detection, semantic segmentation, and object classification at the same time, with high accuracy.

Although the above algorithms have a good performance in the field of image segmentation, they are not applicable for intelligent sweeper robots basing on embedded platforms. First of all, the noise in the image and the slight grayscale changes on the surface of the object may lead to excessive segmentation using the watershed algorithm. Secondly, the foreground and background model of the Graph Cuts algorithm is a gray histogram, resulting in the fact that the image information attained is less than that of the original RGB image. In addition, the algorithm minimizes the energy function by one time, so the accuracy of segmentation is low when the picture information is complicated. Thirdly, although Mask R-CNN has a high average accuracy (mAP) for object detection and semantic segmentation, the speed of prediction process is slow and cannot meet the real-time requirements. Figure 1 shows two flawed results by applying Graph Cuts and Watershed algorithm. The segmentation results of watershed are excessive, while that of Graph Cuts are inaccurate, which misclassify the pixels around the obstacle slipper.

![Figure 1. Flawed results by applying Graph Cuts and watershed algorithm](image)

### 1.1. The method in this paper and relative improvement

This paper uses the graph-based Grabcut algorithm [5] and improves the models and parameters inside to obtain a more accurate and faster image segmentation method. The improvements can be summarized by four facets. First, the input image is cropped before image segmentation, in order to reduce the running time of the algorithm. Secondly, Laplacian Mixture Model [6] is applied in the background model, rather than the original Gaussian Mixture Model [7]. Thirdly, the covariance matrix of the Laplacian Mixture Model is modified to adjust the probability density function, so that the classification of pixels would be more accurate. Additionally, the method does not use the rectangular box to initialize the Trimap $T$, but directly assigns values to the Mask image’s pixels according to the perspective of the sweeper robot and common features of obstacles. Such initialization could transmit more information to assist the method to finish Image segmentation.
2. Algorithm used in this paper

2.1. Principles of Grabcut Algorithm

Grabcut is an image segmentation algorithm based on graph theory. Unlike Graph cuts, Grabcut uses the Gaussian Mixture Model to determine the pixels of the image belonging to the foreground or background model. In addition, Grabcut algorithm will continuously improve the models of foreground and background through iterative segmentation and adjusting parameters, leading to more accurate determination of the pixels’ attribution.

Both the foreground and background Gaussian Mixture Models have K Gaussian components. In each iteration, pixels of the image will be assigned to a certain Gaussian component which has the highest probability. Similar to Graph cuts, the Grabcut algorithm has an energy function (1) consisting of a region term and a boundary term. U is the region term (2), (3), which characterizes the probability that a pixel belongs to the foreground or the background model. V is the boundary term (4), which represents the penalty of discontinuity between a certain pixel m and a surrounding pixel n.

\[ E(\alpha, k, \theta, z) = U(\alpha, k, \theta, z) + V(\alpha, z) \]  
(1)

\[ U(\alpha, k, \theta, z) = \sum_{n} D(a_n, k_n, \theta_n, z_n) \]  
(2)

\[ D(x) = \sum_{i=1}^{K} \pi_i g_i(x; \mu_i, \Sigma_i), \quad \sum_{i=1}^{K} \pi_i = 1 \text{ and } 0 \leq \pi_i \leq 1 \]  
(3)

\[ V(\alpha, z) = \gamma \sum_{(m,n) \in C} [\alpha_n \neq \alpha_m] \exp(-\beta\|z_m - z_n\|^2) \]  
(4)

Function g (5) is the probability density function of the multivariate Gaussian distribution. The parameters required by the region term function D, \( \pi, \mu, \Sigma \), constitute the parameter \( \theta \) of the Gaussian Mixture Model. Grabcut algorithm will initially classify the pixels according to the rectangular box or mask image provided by the user, and cluster them into K class (6) in the foreground model and the background model, according to the K-Means algorithm [8]. After the initialization, the algorithm would assign each pixel to the GMM model, and then perform segmentation by the max-flow/min-cut algorithm. At the end of every segmentation, some pixels may have new classification and the three parameters of the GMM model may be updated, making the segmentation result more accurate.

\[ g(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp \left[ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right] \]  
(5)

\[ K_n := \text{argmin}_n D_n(a_n, k_n, \theta, z_n) \]  
(6)

2.2. The shortcomings of applying the original Grabcut algorithm

During the experiment, the original Grabcut algorithm was first applied on the image data set we collected. However, the accuracy rate of segmentation was only 58.6% on the 1000 images, and the real-time requirement was not met. By looking into the algorithm’s source code and further exploration, the following three problems are founded:

(1) Image processing takes too long time

The Grabcut algorithm needs to classify each pixel of the image during each iteration. And the classification process needs to use the RGB value of pixels in the Gaussian Mixture Model to calculate the probability of pixels belonging to K different Gaussian components. What is more, the probability values obtained above need to be sorted, so traversing all the pixels will take a long time, leading to the poor real-time performance.

(2) Accuracy rate is low when background pixels are complicated

Generally K will be 5, which represents the number of Gaussian components in the foreground and the background models. Each component contains five colors, composed of RGB channels. Since the ROI area that needs to be segmented usually takes up only a small part of the entire picture, most of the pixels are classified into background model. In this case, the rich background pixels are likely to
cause the color of a certain Gaussian component of the background model to be close to one of foreground model’s components. As a result, some parts of the foreground pixels would be misclassified as background color, making the segmentation incomplete.

(3) The segmentation accuracy is low when the mask image is initialized only by a rectangular box.

There are two ways for Grabcut algorithm to implement initialization. The first is that a rectangular box is provided to indicate the possible foreground pixels. In other words, the outer pixels of that box are all classified as the background pixels \( T_{\text{background}} \) [9], and pixels inside the box \( T_{\text{unknown}} \) are all taken as the "possible foreground" pixels. The second way is that the user directly assigns certain values to the pixels of mask image, representing "absolute foreground pixels", "absolute background pixels", "possible foreground pixels" or "possible background pixels". Before the image is segmented by the algorithm, a convolutional neural network is used to predict the type and coordinates of the obstacle and to generate a predicted rectangular bounding box. Therefore, the rectangular box can be directly used in the initialization of mask image. However, although using the rectangular box for initialization is convenient and does not require additional assignment operations, the rectangular box simply divides all pixels into "absolute background" and "possible foreground" parts. Such division provides limited information that would cause obstacle in some images cannot be correctly segmented out.

3. Improvement of Grabcut Algorithm on Intelligent Sweeper robot

The following description will explain the specific improvement of the Grabcut algorithm in this method, including cropping the original images before segmentation, replacing the applied Gaussian Mixture Model of the background model with Laplacian Mixture Model, adjusting the background model’s covariance matrix, and initializing Trimap by the mask image rather than relying on the rectangular box.

3.1. Crop the image based on a bounding box provided by the neural network

Assuming that the height of the image is \( h \), the width is \( w \), and the number of iterations of the Grabcut algorithm is \( \text{IterCount} \). Each time the image is segmented, the mixture models of foreground and background need to perform \( h \times w \times \text{IterCount} \) operations on the pixels. In this case, if the input images are not cropped, it would take a long time to process one picture. Before the process of obstacle segmentation and edge detection, a neural network model MobileNet [10] is used to predict the location and classification of obstacles in images. In order to decrease the running time, the original image would be cropped after it obtaining the bounding box of the obstacle from convolutional neural network. For the purpose of maintaining relative integrity of the images, the image would be first resized to the original half, and then the height and width of the cropped images would be increased by 80 pixels. After the cropping process, the Grabcut algorithm would be executed to complete the segmentation task. Table 1 compares the time consumed by the original Grabcut algorithm and the later version, for images with different height and width.

| Case | Original height | Original width | Time consumed | Cropped height | Cropped width | Improved time consumed | Time saved percentage |
|------|-----------------|----------------|---------------|----------------|--------------|------------------------|-----------------------|
| Case 1 | 720             | 960            | 3710.9ms      | 127            | 199          | 159.2ms                | 95.7%                 |
| Case 2 | 1440            | 1080           | 9161.3ms      | 256            | 508          | 1578.8ms               | 82.7%                 |
| Case 3 | 720             | 540            | 2412.9ms      | 287            | 287          | 452.1ms                | 81.2%                 |

Taking Case1 as an example, the original image has a height and width of 720 and 960 pixels, respectively, which takes 3.7 seconds for the original algorithm to execute. The cropped image has a length and width of 127 and 199 pixels, respectively. It takes only 0.159 seconds for the improved algorithm to execute. In general, cropping the target obstacle images can reduce the running time of
the algorithm by about 80%, which greatly improves the real-time performance and speed up the segmentation.

3.2. **Directly assign the value to the mask image’s pixels, not use rectangular box only**

The Grabcut algorithm requires the user to interact with some information about foreground pixels. Although in this scenario the coordinates of the target obstacle are provided by the convolutional neural network and can be directly used to initialize Trimap T, such operation cannot provide enough information by only classifying the pixels inside the rectangular box as "possible foreground" and the pixels outside as "absolute background". Such initialization often causes the background pixels around the obstacles to be misclassified as foreground pixels. Therefore, we need to assign values to the pixels of mask image. Since the entire obstacle contour extraction method is done automatically, users are not able to initialize the mask image by themselves, indicating which region is "absolute foreground" and which region is "absolute background". Therefore, we would assign the pixels of mask image (Figure 2) according to the characteristics of the socks, slippers, data wires, etc., which are common obstacles on the floor.

![Figure 2. Mask image assignment (Trimap initialization).](image)

In Figure 2, the red portion is the bounding box of obstacles, provided by the convolutional neural network, and the area wrapped by the blue portion is the cropped image. The blue part, including the annular region and four small crosses on the vertices of the predicted rectangular box, will be assigned value of 0, meaning absolute background. When the obstacle is a slipper, only the upper two small crosses in the rectangular box retains because the slipper sole often occupies the whole width of that rectangle. The red portion will be assigned value of 3, meaning possible foreground. The yellow part in the center will be assigned value of 1, meaning absolute foreground.

During each iteration, Grabcut algorithm would re-evaluate the pixels’ attribution, which temporarily belong to possible foreground or possible background. Whenever segmentation is completed according to the maximum flow/minimum cut algorithm, Grabcut would update the pixels’ values of the mask image. It's important to note that the Grabcut algorithm would never change the status of pixels belonging to the absolute foreground and absolute background.

3.3. **Use Laplacian Mixture Model in image’s background model**

During the experiment, it was found that some misclassified background pixels were still not correctly classified after applying the previous two changes. So some attempts were made to further improve segmentation accuracy, like changing the foreground or background color model. Considering that the Gaussian Mixture Model is fitted by multiple Gaussian distributions, its probability density function is not steep. In other words, there is no obvious difference between probabilities of different colors. It is likely for some background pixels’ probability to be larger when calculated in foreground model, because the background colors are usually more complex than that of the foreground. As a result, those pixels are misclassified into foreground.
In order to solve this problem, it is necessary to increase the probability of these pixels calculated in the background model. Therefore, the Gaussian Mixture Model of the background is replaced by a Laplacian Mixture Model. The first thing to determine is the probability density function of the Laplacian Mixture Model, which is fitted by multiple Laplacian distributions. The probability density function of Laplacian distribution can be expressed by equation (7), and that of the Laplacian mixture model is expressed as (8).

\[
f_x(x_1, x_2, x_3) = \frac{2}{k} \frac{1}{(2\pi)^2 |\Sigma|^{0.5}} K_v(\sqrt{2x'^T\Sigma^{-1}x'})
\]

where: \(v=\frac{2-k}{2} \) and \(K_v \) is the modified Bessel function of the second kind

\[
P(x) = \sum_{i=1}^{K} \pi_i f_x(x; \mu_i, \Sigma_i), \quad \sum_{i=1}^{K} \pi_i = 1 \text{ and } 0 \leq \pi_i \leq 1
\]

Table 2 describes the probabilities of two previous misclassified pixels belonging to background. The parameter \(\mu_i\) represents the weights of color component I in background model, and \(\pi_i\) represent the possibility of pixel belonging to the color component I. Through comparison, the probabilities of these two pixels belonging to background get an obvious increment by applying Laplacian Mixture Model rather than Gaussian Mixture Model in background model. Such test data is conformed with theoretical analysis in last paragraph.

|        | \(u_1 \times p_1\) | \(u_2 \times p_2\) | \(u_3 \times p_3\) | \(u_4 \times p_4\) | \(u_5 \times p_5\) | \(\sum_{i=1}^{5} \mu_i \times p_i\) | Increased percentage |
|--------|-----------------|-----------------|-----------------|-----------------|-----------------|-------------------------------|------------------|
| Pixel 1| Gaussian         | 0.151           | 0.173           | 0.067           | 0.053           | 0.013                         | 0.457            |
|        | Laplacian        | 0.141           | 0.360           | 0.096           | 0.099           | 0.066                         | 0.762            |
| Pixel 2| Gaussian         | 0.226           | 0.182           | 0.014           | 0.023           | 0.010                         | 0.455            |
|        | Laplacian        | 0.492           | 0.273           | 0.022           | 0.034           | 0.021                         | 0.842            |

3.4. Modify the covariance matrix of Laplacian Mixture Model in background

In order to further increase the accuracy of the improved Grabcut algorithm in image segmentation, the covariance matrix [11] of the background Laplacian Mixture Model is modified. In essence, both modifying the covariance matrix and using Laplacian Mixture Model in background model aim to modify the probability density function of the background color model, and to make the shape steeper.

In order to enlarge the difference of probability between different colors, it is necessary to increase the eigenvalues of the covariance matrix. Since the covariance matrix of a Laplacian distribution reflects the relationship between three color channels of RGB and the variance of each color channel, the elements of the matrix cannot be directly modified. Instead, a matrix containing eigenvalues should be first obtained by linear matrix transformation, then the elements of this matrix would be multiplied by constant \(C\). After that, we could attain a new covariance matrix by inversely conducting that linear transformation. The covariance matrix can be expressed by the following formula (9). In order to make the operation more efficient, the formula (9) can be expressed by a matrix symbol formula (10), where \(V\) is a matrix whose columns are the eigenvectors of covariance matrix. \(L\) is a diagonal matrix, whose non-zero element corresponds to the eigenvalue. Therefore, the covariance matrix can be expressed as a function of eigenvectors and eigenvalues (11). After many attempts, we find that taking \(C\) as 4 can lead to optimal performance. Let \(L = V \times 4\) and perform an inverse linear transformation to obtain the new covariance matrix.

\[
\Sigma \tilde{\nu} = \lambda \tilde{\nu}
\]

\[
\Sigma V = VL
\]
4. Algorithm used in edge detection

After completing the image segmentation using the improved Grabcut algorithm, the Canny edge detection algorithm is applied to extract the contours of obstacles. The algorithm was developed by John F. Canny [12] in 1986. The following flow chart (Figure 3) is the implementation of algorithm in this paper. During implementation, we use a $5 \times 5$ Gaussian kernel to de-noise the segmented obstacle image and use the Sobel operator to calculate the gradient of the image. The edge selection is well implemented based on the fuzzy threshold method, which reduces the non-edge pixels in the output result.

\[
\Sigma = VLV^{-1}
\]  

(11)

5. Experiment results

The experiment results are based on the implementation of the Grabcut algorithm in OpenCV. The four changes mentioned in Chapter 3 are added by modifying the source code. The effect of cropping the pictures has been proposed in Table 1, and the specific assignment of mask image pixels’ value has been proposed in Figure 2. Since the method is supposed to enable the sweeper robot to successfully avoid the slippers, socks and data wires on the floor, these three obstacles will be used in the test. In Figure 4, the result of the improved Grabcut algorithm is showed by images with capitalized letter in label, and the other images are the results of the original Grabcut algorithm. Through the comparison, it is obvious that the improved method has better performance on obstacle segmentation. In flawed segmentation results such like a (3), b (3) and c (3), there are plenty of pixels being misclassified, leading to the inaccurate edge detection of those obstacles. In contrast, the corresponding results by applying improved method correctly extract the target obstacles, laying solid foundation for edge detection afterward. When the same group of images was used to test, the segmentation accuracy rate reached 91.4% (914/1000) with improved algorithm, which increased by 55.97% compared with the previous results.
Figure 4. Experiment results of original and improved Grabcut algorithm.
6. Other attempts that have not improved performance significantly

Other attempts had been made during experiments, but they did not have a significant or universal improvement on the results of image segmentation. First, the number of iterations of the Grabcut algorithm was increased. However, it has been found that more than 90% of the images have reached a stable state after 5 iteration cycles. Unless more detailed pixel information is offered about the foreground and background, the segmentation results will not change further. Secondly, the number of color clustering was increased in the initialization of color models. Nevertheless, since the target obstacle (foreground) does not have many types of colors, this measurement does not significantly improve the accuracy. Thirdly, the RGB images was converted into HSV color space before running the Grabcut algorithm. After the image segmentation is completed, the HSV image was converted back to the RGB image as the final result. Nonetheless, due to the fact that the obstacles are in different shapes, and the angle of camera sight changes from time to time with the movement of the sweeper robot, changes to the color space will have better segmentation effect for certain images, but this measurement is not universal and cannot improve the overall quality of image segmentation.

7. Conclusions

This paper explores the image segmentation and edge detection method for sweeper robots. After referring to many relative algorithms, the Grabcut and the Canny edge detection algorithm is used to finish the task. Following four improvements: (1) using the Laplacian Mixture Model in the background color model, (2) modifying the corresponding covariance matrix, (3) changing the initialization pattern of the Mask image regarding the perspective of sweeper camera, (4) cropping the image to reduce the operation time, make the improved Grabcut algorithm more accurate and efficient on the application of intelligent sweeper robot.

In the future research work, there are still two aspects could be improved. First, in images whose foreground pixels are similar to those around, some segmentation results are still defective. To improve the performance, it may be useful to pre-process the pixels, increasing the contrast and making their values more different. Secondly, since the sweeper robot is almost always in movement during work, many photos obtained by the camera would be blurred. Such low-quality images will cause difficulty in image segmentation and edge detection. Therefore, the scientific judgement of whether a picture is suitable for segmentation and how to capture clear images during movement are worthy to be further explored.

References
[1] Beucher S 1992 The watershed transformation applied to image segmentation SCANNING MICROSCOPY-SUPPLEMENT- p 299
[2] Boykov Y, Veksler O 2006 Graph Cuts in Vision and Graphics: Theories and Applications Boston: Springer pp 79-96
[3] Boykov Y, Kolmogorov V 2004 IEEE Transactions on PAMI 26(9) :1124-37
[4] He K et al. 2017 Mask r-cnn Proceedings of the IEEE international conference on computer vision pp 2961-2969
[5] Rother C, Kolmogorov V and Blake A 2004 ACM transactions on graphics 23(3): 309-314
[6] Amin T, Zeytinoglu M and Guan L 2004 IEEE Transactions on Multimedia 9(7): 1416-29
[7] Rasmussen, Carl E 2000 The infinite Gaussian mixture model Advances in neural information processing systems pp 554-560
[8] Hartigan J A, Wong M A 1979 Journal of the Royal Statistical Society. Series C (Applied Statistics) 28(1): 100-108
[9] Rheemann C et al. 2008 High resolution matting via interactive trimap segmentation IEEE Conference on Computer Vision and Pattern Recognition IEEE pp 1-8
[10] Howard A G et al. 2017 Preprint arXiv:1704 04861
[11] Carlson, Blair D 1988 IEEE Transactions on Aerospace and Electronic systems 24(4): 397-401
[12] Canny J 1987 IEEE Transactions on PAMI pp 679-698