Automatic Refinement of Foreground Regions for Robot Trail Following

Mehmet Kemal Kocamaz & Christopher Rasmussen
University of Delaware
kocamaz@udel.edu, cer@cis.udel.edu

Abstract

Continuous trails are extended regions along the ground such as roads, hiking paths, rivers, and pipelines which can be navigationally useful for ground-based or aerial robots. Finding trails in an image and determining possible obstacles on them are important tasks for robot navigation systems. Assuming that a rough initial segmentation or outline of the region of interest is available, our goal is to refine the initial guess to obtain a more accurate and detail representation of the true trail borders. In this paper, we compare the suitability of several previously published segmentation algorithms both in terms of agreement with ground truth and speed on a range of trail images with diverse appearance characteristics. These algorithms include generic graph cut, a shape-based version of graph cut which employs a distance penalty, GrabCut, and an iterative superpixel grouping method.

1 INTRODUCTION

Vision-based trail finding and tracking can be considered as a form of road following [7, 13, 12, 5]. However, several factors make the computer vision task particularly hard, including indistinct borders, abrupt elevation changes, dead-ends and forks, sharply varying illumination conditions due to shadows, a wide range of trail materials and hence colors and textures, and the possibility of in-trail objects such as rocks, stumps, or grass. In our previous work [8], we described an efficient and robust approach to trail finding using a general model of color contrast and a triangular shape template. A shortcoming of the approach, however, was the approximation of the trail shape in its linearity and lack of local shape variation.

In this paper we describe a second stage of shape estimation in which the initial, rough shape is *refined* automatically, without user interaction. The contribution of this work is a survey, analysis, and comparison of several basic segmentation algorithms for this purpose, specifically graph cut [2], graph cut with distance maps [6], GrabCut [10], and a grouping method based on superpixel oversegmentation [4]. Although the focus here is on trail image data most relevant to mobile robot applications, we believe that the problem of automatically refining segmentations is a general one.

The rest of the paper is organized as follows. Section 2 outlines the basic segmentation methods. Section 3 describes some algorithmic changes we made to transform the GraphCut methods and GrabCut into automatic foreground refinement algorithms. Section 4 shows some results of the transformed methods on our data sets, compares their performances and the final segmentations of the techniques with ground-truth data. Finally, in Section 5, we summarize the algorithms and their results.

2 Review of Basic Segmentation Algorithms

Graph Cut: The segmentation algorithm proposed in [3, 1, 2] can convert an image segmentation task to a foreground/background labeling problem. Labeling problem is to assign a label L_i for each pixel i in the image. L_i is from the set of segmentation result, S_{result} = $(''background'', ''object'')$. $L = (L_1, L_2, ..., L_n)$, where n is the number of the pixels in the image, is the solution of the segmentation. To find the set L the following energy function can be solved:

$$E(L) = \sum_{i=0}^{n} R_i(L_i) + \lambda \sum_{\{i,j\} \in N} B(L_i, L_j) \quad (1)$$

In this energy function, $R_i(L_i)$ is called as *regional term* and $B(L_i, L_j)$ as *boundary term*. $\lambda \geq 0$ is a weight term to set the relative influence of *boundary term* versus *regional term* in the function. N is all unordered neigborhood pixel pairs.

Graph Cut with Distance Penalty: Standard Graph cut technique is capable of capturing areas similar to object of interest. As an addition to Graph cut method, [6] introduces the idea of using a distance penalty on pixels based upon the distance from the region of interest to bias the segmentation to remain in its area. The *Regional term* function in regular graph cut algorithm is changed as follows:

$$Map_{dist}(i) = \|i - Center(Object)\| \tag{2}$$

$$R_i("Object") = -lnPr(I_i|O) - \alpha(Map_{dist}(i)) \tag{3}$$

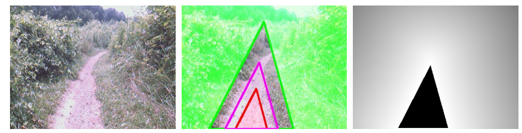
where $Map_{dist}(i)$ is the distance of a pixel in the image to the center of a given object priori and α is an additional weight to adjust relative distance penalty influence in *regional term* function.

GrabCut: The algorithm described in [10] is a powerful foreground/background segmentation technique. It extends the regular graph cut technique in several ways. First, the segmentation is performed on a given initial trimap *T*. The pixels are labeled as T_B for background pixels, T_F for foreground pixels and T_U for the pixels whose labels will be determined by GrabCut. Instead of one shot regular graph cut algorithm, iterative version of the graph cut optimization is combined with Gaussian Mixture Models of background and foreground regions. After each iteration of GrabCut, a border matting algorithm is performed around the object boundary and the colors of foreground pixels.

Superpixel grouping The algorithm in [4] is an efficient method to oversegment an image into self-similar regions. It defines a predicate for measuring the evidence for a boundary between two regions using a graph-based representation of the image. Their segmentation algorithm is constructed based on this predicate. Although this algorithm makes greedy decisions, it produces segmentations that satisfy global properties. In [9] we described an iterative method for grouping superpixels to maximize both shape and appearance contrast criteria which starts from initial triangular model $\hat{T}$. Superpixels outside this initial region may be added and superpixels inside may be removed in a non-parametric process that allows a wide effective range of shape deformations, including the introduction of "holes," or outlier subregions within the trail which may be obstacles.

3 Changes to GraphCut-based methods

In this section, we describe the changes we make on the basic algorithms outlined in Section 2 to use them as automatic foreground extraction methods for our image sets.



(a) (b) (c)

Figure 1. The image, its background and object models, and its distance map.

3.1 Obtaining Background and Foreground Models

Standard graph cut, graph cut with distance penalty and GrabCut algorithms require color information of background and foreground regions in the image. The weights of the edges in the segmentation graph G of these algorithms are set by provided background and foreground models, respectively M_B and M_F.

We obtain M_B and M_F from our previous work explained in [8]. Briefly, it fits an estimated shape S (here a *triangle* to the trail in the image. S may cover some part of the background or not include all the trail region. We scale down S and form S^-. S^- provides color information to construct M_F. Some background pixels in S may generally be placed near the border of S. Scaling down process eliminates those pixels and yields better true trail pixels to construct M_F. Simply, we use all the pixels in S^- and form M_F. To retrieve M_B, first S is scaled up and obtained S^+. All color information of the pixels outside of S^+ are employed in M_B. Scaling up and down factors are 60% and 40%, respectively. In Figure $1(b)$, the pixels colored as green form M_B, colored as red form M_F, the purple triangle corresponds to the initial estimated trail S.

The pixels in S^- are set as hard constraints in the segmentation graphs of the methods. No background pixels are provided as hard constraints to the algorithms. In the trimap of GrabCut method, unknown label is assigned to all pixels outside of S^-, and inside of S^- is considered as foreground.

3.2 Distance Map Construction

[6] uses a distance penalty function in its *regional term*. To set the penalty function of this method, we construct a distance map, Map_{dist}, and provide the map to the algorithm. Initial estimated priori S of foreground includes three important cues about the object: Color information, estimated shape of the object, and spatial information of it in the image. To obtain color models M_F and M_B, we used color information pro-

vided by S. Other two cues coming with S help to construct Map_{dist}. Figure $1(c)$ shows the distance map of the given image. Instead of using object center to calculate the distances as in [6], we use closest pixel to S. Equation 2 is changed as follows

$$Map_{dist}(i) = \|i - closestTo_i\| \quad (4)$$

where $closestTo_i \in S$ is closest pixel of S to pixel i in the image space. $\|i - closestTo_i\|$ is the distance between $closestTo_i$ and pixel i.

3.3 Removing Weak Components

The raw foreground mask generated by graph cut techniques typically contains some noisy, small and weakly-connected foreground regions, because our images can contain a non-homogeneous color distribution inside the foreground regions and [3] uses the color histogram to assign weights to terminal links. In order to clean up those regions, we first do morphological opening and closing and find the connected components. The largest region is taken as the final refined foreground region.

4 RESULTS

We measure the accuracy and efficiency of the algorithms described in Section 3 on a diverse set of trail images. The experiments are run on two set of images. Set-1 consists of the images collected from several trail image sequences taken by our robot platform as it was manually driven and Set-2 includes 30 images from the hiking trail, river and canyon sequences taken from the web. Images of Set-2 are available through the "Data/trail30" link at `http://nameless.cis.udel.edu`. The data sets are scaled to 320 by 240 as necessary. At regularly spaced intervals along trail sequences we have manually generated ground-truth segmentations. We compare the results with the ground-truth segmentations to quantify the accuracy of the refinement algorithms. We use the following polygon area overlap formula suggested by [11] to measure the overlap between the ground-truth segmentation and the result of the refinement methods:

$$Overlap(\mathcal{R}_1, \mathcal{R}_2) = A(\mathcal{R}_1 \cap \mathcal{R}_2)^2 / (A(\mathcal{R}_1)A(\mathcal{R}_2)) \quad (5)$$

where $\mathcal{R}_1$ and $\mathcal{R}_2$ are given two regions to calculate the overlap between them.

We have refined two set of images by using each described method in previous section. Then, we calculated median overlapping scores and average segmentation time of refinement methods and initial models obtained from [8] for two set of images. The overlap scores can be seen in Table 1. Average segmentation time of GrabCut, superpixel, graph cut, graph cut with distance penalty methods are 2.85, 4.95, 0.17 and 0.19 seconds, respectively. Since initial models may contain non-trail regions or not cover entire trail region, the median overlapping score of given initial models is not good in our first image set. Using regular graph cut method for refinement process improves the segmentation quality. However, incorporating distance map approach to graph cut performs better than regular graph cut in both two image sets. The results of the refinement methods are shown in Figure 2.

Algorithm	Median Overlap Score of Set-1	Median Overlap Score of Set-2
Initial Models [8]	0.663	0.732
Superpixel Grouping [4]	0.740	0.713
GrabCut [10]	0.631	0.738
Graph Cut [2]	0.730	0.737
Graph Cut(With Distance Map) [6]	0.783	0.760

Table 1. Comparisons of the Methods.

5 CONCLUSION

We have compared and analyzed the performance of several transformed methods which can refine automatically foreground regions for the purpose of robot trail following. These refinement methods are obtained by making some changes on graph cut, graph cut with distance penalty, GrabCut and super pixel segmentation algorithms. The refinement methods require to take initial information about the foreground region. These information contain cues about the color, shape and spatial position of the region in a given image. We construct initial color and shape models of the foreground and background regions by using these priori information. The performance of the algorithms are analyzed on several long sequences with diverse appearance and structural characteristics. Ground-truth segmentations are used to quantify performance where available.

References

[1] A. Blake, C. Rother, M. Brown, and P. Torr. Interactive image segmentation using an adaptive gmmrf model.

Figure 2. Trail refinement results of the methods. Rows show initial models, refinement results of the of the superpixel method, GrabCut, regular graph cut, and graph cut with distance penalty method, respectively. Left 4 columns include images from Set-1 and, right 2 columns contain images from Set-2.

European Int. Conf. on Computer Vision, 2004.

[2] Y. Boykov and G. Funk-Lea. Graph cuts and efficient n-d image segmentation. *Int Journal of Computer Vision*, 70:109–131, 2006.

[3] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *Proc. Int. Conf. Computer Vision*, 2001.

[4] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *Int Journal of Computer Vision*, 59(2), 2004.

[5] A. Huang, D. Moore, M. Antone, E. Olson, and S. Teller. Multi-sensor lane finding in urban road networks. *Robotics: Science and Systems*, 2008.

[6] J. Malcolm, Y. Rathi, and A. Tannenbaum. Tracking through clutter using graph cuts. *In British Machine Vision Conf. (BMVC)*, 2007.

[7] D. Pomerleau. RALPH: Rapidly adapting lateral position handler. *Proc. IEEE Intelligent Vehicles Symposium*, pages 506–511, 1995.

[8] C. Rasmussen, Y. Lu, and M. Kocamaz. Appearance contrast for fast, robust trail-following. *Proc. Int. Conf. Intelligent Robots and Systems*, 2009.

[9] C. Rasmussen and D. Scott. Shape-guided superpixel grouping for trail detection and tracking. *Proc. Int. Conf. Intelligent Robots and Systems*, 2008.

[10] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23:309–314, 2004.

[11] S. Sclaroff and L. Liu. Deformable shape detection and description via model-based region grouping. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(5), 2001.

[12] B. Southall and C. Taylor. Stochastic road shape estimation. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 205–212, 2001.

[13] C. Taylor, J. Malik, and J. Weber. A real-time approach to stereopsis and lane-finding. *Proc. IEEE Intelligent Vehicles Symposium*, 1996.