Interactive RGB-D Image Segmentation Using Hierarchical Graph Cut and Geodesic Distance

Ling Ge, Ran Ju, Tongwei Ren, Gangshan Wu

State Key Laboratory for Novel Software Technology
Collaborative Innovation Center of Novel Software Technology and Industrialization
Nanjing University, Nanjing 210023, China
gelingnju@gmail.com, juran@smail.nju.edu.cn, {rentw, gswu}@nju.edu.cn

Abstract. In this paper, we propose a novel interactive image segmentation method for RGB-D images using hierarchical Graph Cut. Considering the characteristics of RGB channels and depth channel in RGB-D image, we utilize Euclidean distance on RGB space and geodesic distance on 3D space to measure how likely a pixel belongs to foreground or background in color and depth respectively, and integrate the color cue and depth cue into a unified Graph Cut framework to obtain the optimal segmentation result. Moreover, to overcome the low efficiency problem of Graph Cut in handling high resolution images, we accelerate the proposed method with hierarchical strategy. The experimental results show that our method outperforms the state-of-the-art methods with high efficiency.

Keywords: Interactive image segmentation, RGB-D image, Graph Cut, geodesic distance, scale space

1 Introduction

Image segmentation aims to partition an image into several parts automatically or with simple interactions. Compared to automatic segmentation [1], interactive image segmentation attracts much attention [2, 3] for its advantage in handling complex image content. It is widely used in many applications to simplify further processing, such as object dataset construction [4], image editing [5] and object image retrieval [6]. In interactive image segmentation, the segmentation problem is usually formulated as assigning a binary label to each pixel in an image to denote whether a pixel belongs to foreground or background [7]. Some hints are manual labelled to indicate parts of foreground and background, and the image is segmented by minimizing the defined energy functions.

As a representative of interactive image segmentation technology, Graph Cut [8] converts an image into a graph, in which each pixel is represented as a graph node and the adjacent nodes are connected with weighted edges. Then the segmentation problem is formulated as an energy minimization process, which can be solved using the min-cut algorithm. The main advantage of Graph Cut is that once the energy function is properly defined, it can provide a globally optimal solution with considering of both unary probability and smoothness.

Fig. 1. Comparison of segmentation results using different channels in RGB-D image. (a) Color channels with manual labels. (b) Depth channel. (c) Ground truth. (d) Segmentation result with Graph Cut on color channels. (e) Segmentation result using geodesic distance on depth channel. (f) Our result integrating color and depth cues.

Compared to the prosperity of segmentation research on traditional 2D image, little attention has been paid to interactive segmentation of emerging RGB-D image. Different to RGB image, RGB-D image provides an extra depth channel. Given a pixel in RGB-D image, we can obtain both its color information and its position in 3D spatial space. In this way, the pixels in an RGB-D image form a color point cloud with a certain spatial distribution, in which color channels offer color contrast and texture for distinguishing foreground from background and depth channel describes the geometrical characteristics of objects and scene. It provides quite different cues to RGB image segmentation, and makes interactive RGB-D image segmentation a novel and challenging problem. Fig. 1 shows an example of segmentation results comparison using different channels in RGB-D image. We can find that both color channels and depth channel provide partial information to distinguish between foreground and background, which leads to the inaccuracy of segmentation results (Fig. 1(d) and (e)). However, integrating color cue and depth cue can provide more accurate segmentation result, as shown in Fig. 1(f).

In this paper, we propose a novel interactive segmentation method for RGB-D image by integrating color cue and depth cue in a unified hierarchical Graph Cut framework. In the proposed method, RGB-D image is represented with the same graph representation as the original Graph Cut method, but differs in the image properties representation. Specifically, we utilize Euclidean and geodesic distance [9] as the dissimilarity metric for color and depth channels respectively. Compared to directly treating depth as a fourth channel of the input of Graph Cut [10], our method provides a better description of the spatial characteristics of the image content. Moreover, we accelerate our method with hierarchical strategy to efficiently handle high resolution RGB-D images. As the computational complexity of Graph Cut method is in proportion to the square of image pixel number [11], we extend Graph Cut to scale space to generate a primary segmentation result on the coarsest scale and then refine foreground boundary on finer scales, which can obviously reduce the number of graph nodes in segmentation.

We evaluate the proposed method on two public datasets [12, 13], and compare it with the state-of-the-art methods as well as manual labelled ground truths. The experiments show that the proposed method obtains better

segmentation results with little user interactions, and much more efficient than the other methods.

2 Related Work

We briefly review the relevant researches on image segmentation for RGB images and depth assisted image segmentation.

Interactive image segmentation. Interactive image segmentation takes manual labelled certain pixels as input and further segments an image by image content, such as color change and contrast. It can handle the images with complex structure with the assistant of manual interaction, which has been widely used in other techniques, such as mobile search [14] and social media analysis [15]. Graph cut [8] is one of the most representative methods. After converting an image to a graph, in which each pixel in image is represented as a graph node and the adjacent nodes are connected with edges, it formulates segmentation as a min-cut problem on the graph. Based on Graph Cut, GrabCut [16] uses an iterative strategy to improve the quality of segmentation. Random Walks method can also be utilized to perform segmentation [17]. Starting from each unlabelled pixel, the probability of a random walker will first reach one of the pre-labelled pixels is calculated and the pixel is labelled accordingly. This algorithm is also applicable on higher dimension but with lower time performance. Geodesic distance method for interactive segmentation [18] has been previously used in processing color cue. It uses star-convexity prior and replaces Euclidean rays with geodesic path to exploit the structure of shortest paths.

Noted that there are some similar multilevel strategies used in [19, 20] to accelerate Graph Cut. However, the application scenes of these methods are different and they are not suitable in processing RGB-D images.

Depth assisted image segmentation. There are several methods proposed to use depth channel to assist segmentation. [21] proposed two bi-layer segmentation methods for binocular stereo video: Layered Dynamic Programming and Layered Graph Cut. They applied color, contrast and stereo matching information to realize automatic foreground/background separation. It differs with our method in using depth information, namely stereo matching information directly and the usage scenario is limited. [22] proposed a method for modeling the background using time-adaptive, Gaussian mixtures in the combined input space of depth and luminance-invariant color. [23] used a stereo camera to extract human silhouettes indoors. It also utilized Graph Cut framework. Its contribution focused on object and background seed segmentation and depth assisted Graph Cut. These three algorithms take stereo images as input, which is different from ours, and neglect excavating the geometry of depth information. They all have special limitation for either scene or target scope. As to general segmentation method for RGB-D images, Julia *et al.* proposed a multi-label segmentation method for RGB-D images in [10]. However, they also

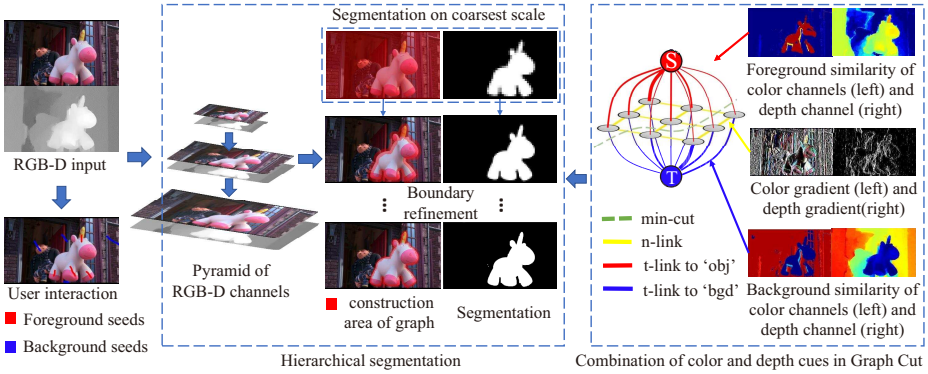

Fig. 2. An overview of the proposed approach.

considered the depth image as an additional data channel to put into Graph Cut framework directly and didn't take further process on depth information.

3 Interactive RGB-D Image Segmentation

Fig.2 shows an overview of the proposed method. For the input RGB-D images and user labels, we first construct the image pyramid on color channels and depth channel. Then Graph Cut is executed on the coarsest scale and the boundary refinements are performed using the processed segmentation result of the previous coarser scale as input. During the segmentation, color and depth are combined to compute the weight of terminal links and neighborhood links. Specifically we employ Euclidean distance and geodesic distance to measure the object likelihood for color and depth channels respectively. Finally, a high quality segmentation result is obtained on the finest scale with full resolution.

3.1 Preliminary of Hierarchical Graph Cut

For further analysis, we briefly review the principle of Graph Cut [8]. In Graph Cut, the original image I is represented as an undirected graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$. Here, $\mathcal{V}$ is the union of pixel nodes and two additional terminal nodes S and T, and $\mathcal{E}$ is the union of neighborhood links (n-links) and terminal links (t-links), which are the edges between adjacent pixel nodes and between pixel nodes and terminal nodes, respectively. With such representation, the graph $\mathcal{G}$ can be partitioned into disjoint regions by removing edges connecting them, which is formulated as a min-cut problem. The energy function is described as follows:

$$E(L) = \lambda R(L) + B(L), \tag{1}$$

where $R(L)$ is sum of penalties for assigning a certain pixel node p to foreground and background; $B(L)$ is the sum of penalties for discontinuity between adjacent

pixel nodes; L is the list of labels assigned to corresponding pixel nodes, whose value is *obj*(foreground) or *bgd*(background); λ is a balance coefficient which equals 5 in our experiments; $R(L)$ and $B(L)$ can be further defined as follows:

$$R(L) = \Sigma_{p \in I} R_p(l_p), \tag{2}$$

$$B(L) = \Sigma_{\{p,q\} \in \mathcal{N}} B_{\langle p,q \rangle} \delta(l_p, l_q), \tag{3}$$

$$\delta(l_p, l_q) = \begin{cases} 0, & if\ l_p = l_q \\ 1, & otherwise \end{cases}, \tag{4}$$

where $\mathcal{N}$ is the set of adjacent pixel nodes under a standard 8-neighborhood system; $R_p(L)$ indicates the possibility of pixel p to be labelled as a certain value of L; Boundary penalties $B_{\langle p,q \rangle}$ denotes the cost of cutting off the neighborhood links between adjacent node p and q.

For Graph Cut cannot efficiently handle high resolution images, a hierarchical strategy [19] is utilized to build a pyramid for the input image and construct graph on each scale separately. In this way, the total number of graph nodes and links are obviously reduced and the efficiency of Graph Cut is improved.

3.2 Scale Space Construction

There is a simple but meaningful phenomenon that the segmentation results on different scales of an image have similar appearance, and the difference of segmentation results on different scales occurs on the precision of boundaries. Based on this observation, we construct a scale space $\{I_0, I_1, \ldots, I_n\}$ for each RGB-D image I, which contains the color channel I^c and depth channel I^d. We obtain the primary segmentation result on the coarsest scale I_0 and refine the foreground boundary on the finer scales from I_1 to I_n.

For the original images may have quite different resolutions, it is not suitable to fix the number of scales n for different original images. A better solution is to restrict the scale of coarsest scale and set a proportion to the construction of adjacent scales. In this way, we can control the computation cost of the whole procedure in a acceptable scope. We adopt a self-adaptive strategy to determine the number of scales in scale space construction. A threshold φ is used to constrain the resolution of the coarsest scale. Obviously, too large φ will cause large computational cost to generate the initial segmentation result, but too small φ will result in serious content loss of the original image. In our experiments, we set φ to 50,000. Starting from the original image I, we down sample the current scale at a proportion of ρ to construct the next scale ($\rho = 0.25$ in our experiments). Once the pixel number of current constructed scale is below φ, construction of scale space is finished. We can figure out that $n = \lceil log_\rho \frac{\varphi}{|I|} \rceil$.

3.3 Integration of Color Cue and Depth Cue

On the coarsest scale I_0, we utilize Graph Cut [8] to generate the initial segmentation result s_0. Both RGB and depth cues are applied in Graph Cut.

As mentioned in section 3.1, regional term $R(L)$ and boundary term $B(L)$ in Equation (1) only takes color cue into consideration in typical Graph Cut framework in RGB segmentation method. In proposed RGB-D segmentation approaches like [10, 21–23], they take depth cue as an additional channel to add into Graph Cut directly. To further extract the spatial property of depth cue, we interprete depth information through geodesic distance [9]. Therefore, regional penalty $R_p(L)$ and boundary penalty $B_{\langle p,q\rangle}$ should be redefined.

We first rewrite regional penalty $R_p(L)$ in Equation (2) as the combination of color penalty $R_p^c(L)$ and depth penalty $R_p^d(L)$ with a balance coefficient α, which equals 1 in our experiments:

$$R_p(L) = R_p^c(L) + \alpha R_p^d(L), L \in \{obj, bgd\}, \tag{5}$$

where color penalty $R_p^c(L)$ of specifying a label l_p to pixel p is the color likelihood between p and the histograms of foreground and background color distributions:

$$R_p^c(L) = P(L = l_p|c_p) = \frac{P(c_p|(L = l_p))}{P(c_p|(L = obj)) + P(c_p|(L = bgd))}. \tag{6}$$

And depth penalty $R_p^d(L)$ is the ratio of geodesic distance, with geodesic distance from p to a specified region as the numerator and the sum of geodesic distance as denominator:

$$R_p^d(L) = \frac{D(p, l_p)}{D(p, obj) + D(p, bgd)}, \tag{7}$$

where $D(x, L)$ indicates the geodesic distance between x and label L, which is formulated as:

$$D(x, L) = \min_{y \in \Omega_L} d(x, y), \tag{8}$$

$$d(x, y) := \min_{C_{x,y}} \int_0^1 |G_d(x, y) \cdot C_{x,y}(p)|dp, \tag{9}$$

where $C_{x,y}(p)$ is a path connecting the pixels x, y; $G_d(x, y)$ is set as the gradient of greyscale.

We also rewrite boundary penalties $B_{\langle p,q\rangle}$ in Equation (3) with an ad-hoc function as follows:

$$B_{\langle p,q\rangle} \propto exp\Big(-\frac{(I_p - I_q)^2}{2\sigma_1^2} - \frac{\beta G_d(p, q)}{2\sigma_2^2}\Big), \tag{10}$$

where β is a balance coefficient, which is set to 1; σ_1 and σ_2 are two parameters to adjust the penalty, here $\frac{1}{2\sigma_1^2} = \frac{1}{2\sigma_2^2} = 0.0075$ in our experiments.

3.4 Upscaling Boundary Refinement

The initial segmentation result s_0, after an opening operation with a 3×3 element on it to avoid noise expansion, provides an approximate distribution of foreground and background of the original image. Based on it, we iteratively refine boundary area with Graph Cut from coarse to fine scale.

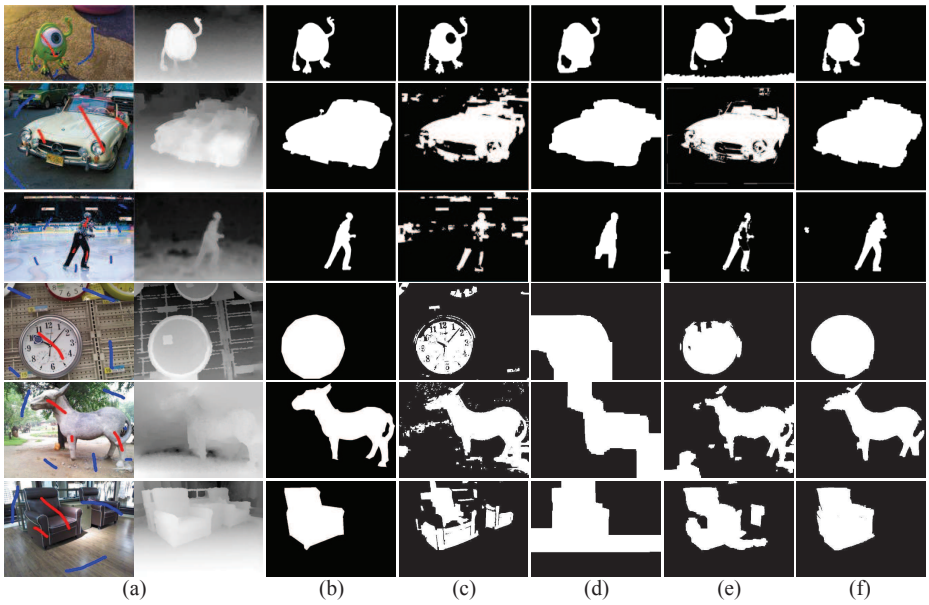

Fig. 3. Examples of segmentation results. (a) Color channels and depth channels of RGB-D images with user labels. (b) Ground truths. (c) Graph Cut. (d) Geodesic distance using depth only. (e) RGB-D segmentation using depth as a fourth channel. (f) Our method.

For the procedure of boundary refinement on each scale is similar, without loss of generality, we assume that boundary refinement is performed on the kth scale I_k with the initialization of the segment result $s_{(k-1)}$ of the $(k-1)$th scale. Fig. 2(d) shows the procedure of boundary refinement. First, dilation and erosion operations are performed on $s_{(k-1)}$ to determine the inner contour C_{in} and outer contour C_{out} of foreground, respectively. Then the processed $s_{(k-1)}$ is resized to the same size of I_k. The region inside C_{in} is settled as foreground, the region outside C_{out} is settled as background, and the rest part R_c is used to build a new graph for Graph Cut. The size of structuring element used in dilation and erosion operations on each scale is not fixed. A suitable size of structuring element should retain enough pixels in boundary area R_c to generate accurate foreground boundary and avoid too many pixels for efficiency. Here, we set the size of structuring element as $(k+3) \times (k+3)$ when dilate and erode s_k.

Graph Cut has been proved to have a complexity of $O(|C||\mathcal{E}||\mathcal{V}|^2)$ in the worst case [11], $|C|$ denotes the cost of min-cut which equals the total weight of removed edges. According to the result of the experiments, the complexity of refinement is only related with the final min-cut result C and the number of scales n in scale space. Coupled with the fact that the complexity of initial segmentation on I_0 can be seen as a constant, our method is proved practically faster than Graph Cut on one scale. The time cost of our method is presented in Section 4.3.

4 Experiments

To validate the performance of our method, we compare it with the methods only using color cue or depth cue and the method directly treating depth as a fourth channel of the input of Graph Cut. We also execute a time evaluation by comparing with other hierarchical methods.

4.1 Datasets and Experimental Settings

To quantitatively evaluate the quality of segmentation results, we use two datasets, NJU400 [12] and RGBD Benchmark [13] in our experiments, which provide 400 stereo image pairs and 1,000 RGB-D images with the corresponding depth maps and pixel-level manual-labelled ground truths, repectively. We treat each left image and its depth map of a stereo image as the color channels and depth channel of a RGB-D image, and compare the segmentation results with the provided ground truths.

The proposed approach is implemented by C++. All the experiments are carried out on a PC with a four-core 3.40GHz CPU and 8GB memory.

4.2 Segmentation Accuracy Evaluation

We first compare our method with the methods only using color cue or depth cue. We select three methods using color cue, including Graph Cut (GC) [8], GrabCut (GB) [16] and multi-level Graph Cut (MGC), for their effectiveness, and applying geodesic distance [9] on depth channel (GDD). Fig.3 illustrates some examples of segmentation results generated by different methods. As shown in Table 1, our method (HGG) obtains higher F_β ($\beta = 0.3$) than the compared methods. It demonstrates that both color and depth cues are beneficial to improve segmentation result.

We further compare our method with the method directly treating depth as a fourth channel of the input of Graph Cut method (RGBD) [10]. It shows that our method outperforms than RGBD method in precision, recall and F_β criteria. It shows that geodesic distance can extract geometry attributes of depth channel and provide better distance measurement.

We also compare the proposed method using hierarchical strategy (HGG) or not (GG). It shows that the application of hierarchical strategy slightly influences segmentation performance, but the performance of HGG is still better than other compared methods.

4.3 Running Time Evaluation

We evaluate the efficiency of the above methods on ten randomly selected images with the average resolution of two million pixels, and execute each method ten times to obtain its average running time of segmentation. All the methods are implemented by C++ and executed on the same platform. As shown in Table 2, out method is only slower than MGC for it contains additional computation for

Table 1. Comparison of segmentation accuracy of different methods.

	GC	GB	MGC	GDD	RGBD	GG	HGG
precision	0.7163	0.9361	0.7575	0.8542	0.8419	0.9272	0.8946
recall	0.7254	0.5558	0.7360	0.8921	0.7796	0.9032	0.9287
F_β	0.7184	0.8084	0.7524	0.8627	0.8267	0.9215	0.9022

Table 2. Comparison of processing time of different methods.

	GC	GB	MGC	GDD	RGBD	GG	HGG
time(s)	0.4340	5.6015	0.0828	32.0488	0.3423	32.2416	0.1131

geodesic distance on depth channel in its procedure, but obviously more efficient than other methods. Especially, our method is about 300 times faster than GG method while only having a slight decrease in segmentation accuracy.

5 Conclusions

In this paper, we propose an efficient hierarchical Graph Cut method for interactive RGB-D image segmentation using fusion of color and depth cues, which can generate high quality segmentation results and realtime interaction. Instead of directly using color channels and depth channel in a same mean, we utilize Euclidean distance on color channels and geodesic distance on depth channel, and fuse them in an unified Graph Cut framework. Moreover, to overcome the disadvantage in efficiency of Graph Cut, we accelerate the algorithm by using a hierarchical strategy which improves the efficiency about 300 times. The experiments show that the proposed method can fully utilize the characteristic of color and depth channels, therefore obtain a better performance to the state-of-the-art methods with high efficiency.

Acknowledgments. This work is supported by the National Science Foundation of China (No.61321491, 61202320), Research Project of Excellent State Key Laboratory (No.61223003), and National Special Fund (No.2011ZX05035-004-004HZ).

References

1. Li, S., Ju, R., Ren, T., Wu, G.: Saliency cuts based on adaptive triple threshoding. In: International Conference on Image Processing, IEEE (2015) 1–4
2. Nguyen, T.N.A., Cai, J., Zhang, J., Zheng, J.: Robust interactive image segmentation using convex active contours. IEEE Transactions on Image Processing **21**(8) (2012) 3734–3743
3. Delgado-Gonzalo, R., Chenouard, N., Unser, M.: Spline-based deforming ellipsoids for interactive 3d bioimage segmentation. IEEE Transactions on Image Processing **22**(10) (2013) 3926–3940

4. Cheng, M.M., Mitra, N.J., Huang, X., Torr, P.H.S., Hu, S.M.: Global contrast based salient region detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(3) (2014) 569–582
5. Ren, T., Liu, Y., Wu, G.: Image retargeting based on global energy optimization. In: IEEE International Conference on Multimedia and Expo, IEEE (2009) 406–409
6. Xu, X., Geng, W., Ju, R., Yang, Y., Ren, T., Wu, G.: Obsir: Object-based stereo image retrieval. In: IEEE International Conference on Multimedia and Expo, IEEE (2014) 1–6
7. Greig, D., Porteous, B., Seheult, A.H.: Exact maximum a posteriori estimation for binary images. Journal of the Royal Statistical Society. Series B (Methodological) (1989) 271–279
8. Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In: IEEE International Conference on Computer Vision, IEEE (2001) 105–112
9. Yatziv, L., Bartesaghi, A., Sapiro, G.: O(n) implementation of the fast marching algorithm. Journal of computational physics **212**(2) (2006) 393–399
10. Diebold, J., Demmel, N., Hazırbaş, C., Moeller, M., Cremers, D.: Interactive multi-label segmentation of rgb-d images. In: Scale Space and Variational Methods in Computer Vision. Springer (2015) 294–306
11. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(9) (2004) 1124–1137
12. Ju, R., Ge, L., Geng, W., Ren, T., Wu, G.: Depth saliency based on anisotropic center-surround difference, IEEE (2014) 1115–1119
13. Peng, H., Li, B., Xiong, W., Hu, W., Ji, R.: Rgbd salient object detection: a benchmark and algorithms. In: Computer Vision–ECCV 2014. Springer (2014) 92–109
14. Sang, J., Mei, T., Xu, Y.Q., Zhao, C., Xu, C., Li, S.: Interaction design for mobile visual search. IEEE Transactions on Multimedia **15**(7) (2013) 1665–1676
15. Sang, J.: User-centric social multimedia computing. Springer (2014)
16. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. **23**(3) (2004) 309–314
17. Grady, L.: Random walks for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(11) (2006) 1768–1783
18. Gulshan, V., Rother, C., Criminisi, A., Blake, A., Zisserman, A.: Geodesic star convexity for interactive image segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2010) 3129–3136
19. Lombaert, H., Sun, Y., Grady, L., Xu, C.: A multilevel banded graph cuts method for fast image segmentation. In: IEEE International Conference on Computer Vision, IEEE (2005) 259–265
20. Vaudrey, T., Gruber, D., Wedel, A., Klappstein, J.: Space-time multi-resolution banded graph-cut for fast segmentation. Springer (2008)
21. Kolmogorov, V., Criminisi, A., Blake, A., Cross, G., Rother, C.: Bi-layer segmentation of binocular stereo video. In: IEEE International Conference on Computer Vision and Pattern Recognition, IEEE (2005) 407–414
22. Harville, M., Gordon, G., Woodfill, J.: Foreground segmentation using adaptive mixture models in color and depth. In: IEEE Workshop on Detection and Recognition of Events in Video, IEEE (2001) 3–11
23. Ahn, J.H., Kim, K., Byun, H.: Robust object segmentation using graph cut with object and background seed estimation. In: International Conference on Pattern Recognition, IEEE (2006) 361–364