Stereo GrabCut: Interactive and Consistent Object Extraction for Stereo Images

Ran Ju, Xiangyang Xu, Yang Yang, and Gangshan Wu

State Key Laboratory for Novel Software Technology, Nanjing University, China
juran@smail.nju.edu.cn, xiangyang_xu@smail.nju.edu.cn, gswu@nju.edu.cn,
charlie.yang.nju@gmail.com

Abstract. This paper presents an interactive object extraction approach for stereo images. The extraction task on stereo images has two significant differences compared to that on monoscopic images. First, the segmentation for both images should be consistent. Second, stereo images have implicit depth information, which supplies an important cue for object extraction. In this paper, we generate consistent segmentation by putting the correspondence relationship in a graph cut framework. Besides, we leverage depth information, which is obtained by stereo matching, to give a pre-estimation of foreground and background. The pre-estimation is then used to generate accurate color models to perform a graph cut based segmentation. To simplify the user interaction, we supply an interface similar to GrabCut, which only needs the user to drag a compact rectangle in most cases. The experiments show our approach works fast and produces more satisfactory results than state-of-the-art.

Keywords: object extraction, graph cut, stereo matching, saliency

1 Introduction

Object extraction (segmentation) is a classical and widely studied problem which aims at finding an optimal binary segmentation for pixels in an image. The two categories of pixels are labeled as "Foreground (Object)" and "Background" respectively. The technique has a large number of applications in object tracking, instance recognition, image editing and so on. A lot of excellent methods have been proposed to solve the problem. And today as the world is generating more and more stereo images by mobile phones, stereo cameras and so on, it is urgently required to develop object extraction methods for stereo images. However, the traditional methods designed for monoscopic images can not be directly applied to stereo images for two reasons.

First, a stereo image is made up of two slightly different images and the segmentation should be consistent for both images. However, if we use traditional methods for each image separately, the consistency can't be guaranteed due to the difference between the two images. Also the workload will be doubled. We show an example in Fig. 1 (a).

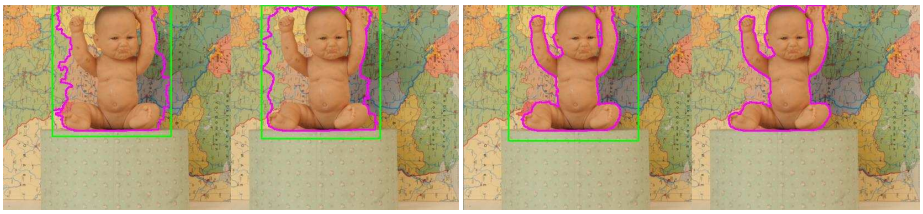


Fig. 1. Differences between object extraction on monoscopic images and stereo images. (a) GrabCut [1] is applied to the two images separately. The consistency can't be guaranteed due to the differences between the images and the user's inputs. (b) Stereo GrabCut is only applied to one image and produces consistent results. The segmentation results are improved owing to the depth guidance.

Second, stereo images have implicit depth information, which can be used as an important cue for segmentation. However, how to utilize the depth information effectively has not yet been well studied before. Additionally, the disparity map may be inaccurate and contain noise, which makes it difficult to analyze.

Our approach—Stereo GrabCut, takes the two points into consideration. For the first one, we perform stereo matching to establish a correspondence relationship between the two images to guarantee consistency. For the second point, we use depth saliency analysis to get a pre-estimation of foreground and background to improve the segmentation results. We show an example in Fig. 1.

To make the interaction easier for users, we design a user interface similar to GrabCut [1]. First, the user drags a compact rectangle around an object then we automatically generate an initial extraction result. Then, if the result needs to be improved, the user can scribble with a foreground and background brush for further editing. Generally our approach generates satisfactory results in the first step. Besides, the operating time is acceptable for interactions. Typically, computing the segmentation for a 320×240×2 stereo image takes 0.9 second.

2 Previous Work

For monoscopic images there have been a lot of successful interactive methods [2,3,4,1,5]. The graph cut based methods [4,1,5] model the segmentation task as a energy minimization problem. These methods make a best balance between data observation and smoothness. However, as we illustrated before, these methods can't be directly applied to stereo images.

Recently a few works have focused on consistent segmentation for stereo images [6,7,8]. Price et al. [6] proposed a consistent interactive object selection framework based on graph cut [4]. The method enforces consistency using dense stereo correspondence probability distributions. In [7] Tasli et al. used sparse corresponding to enforce consistency. These methods depend on the selection of initial seeds and thus are hard to master for amateur users. Besides,the depth information has not been used.

Leveraging depth information for object extraction has been introduced to some special purpose applications [9,10,11]. These methods focus more on automatic segmentation in particular domains and produce segmentation results for only one view. In contrast, our approach focus more on general purpose segmentation and preserving consistency.

The methods based on saliency analysis [12,13] for object extraction have attracted a large attention because the results are more natural and perceptual acceptable. While previous methods focus on color images, Niu et al. [13] proposed two methods to use stereo information for saliency analysis. One is based on global contrast [12] and the other is based on stereoscopic photography knowledge. However the priors are not usually satisfied which makes it limited in general purpose applications. To overcome the limitations, we utilize the users' inputs to guide depth saliency analysis and use the results to generate color models for graph cuts, which is similar to [14].

3 Approach

We give an overview of our approach in Fig. 2. We assume the input stereo images are rectified so that the corresponding pixels in both images are constrained in the same horizontal line. In step 1, the user drags a rectangle around the object. Then we perform stereo matching to get the disparity map. We compute the saliency map in the user selected area based on the disparity information and then give a pre-estimation of foreground and background. The pre-estimation is used to create the color models. Next we construct the graph model using the stereo image, the disparity map and the color models. At last we optimize the model by a max-flow/min-cut algorithm [15] to get initial segmentation results. If it is needed, in step 2, the user improves the details using a foreground and background brush. The color models, graph model and optimal flow are recomputed. Step 2 can be repeated until the user gets the satisfactory results.

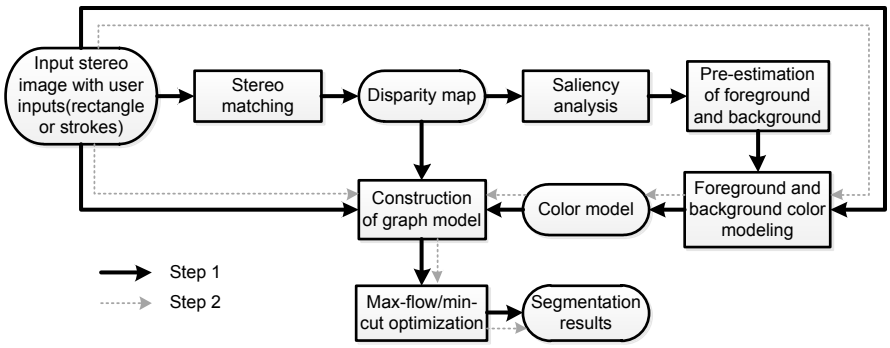


Fig. 2. Overview

3.1 Stereo Matching and Consistent Segmentation

The disparity map plays a key role in consistent segmentation and pre-estimation of foreground and background. However, today there still doesn't exist a perfect solution for stereo matching [16]. To overcome the problem, first we choose a robust and fast stereo matching algorithm called ELAS [17] to get the disparity map. The algorithm has been tested on the KITTI vision benchmark [18] which contains lots of challenging stereo pairs for autonomous driving task and performs excellent. Second, we don't require very accurate disparity maps for the following processes, which will be showed in the following.

Next we add the correspondence constraint to the global energy function:

$$\begin{aligned} E(A) = & \sum_{p \in P_l \cup P_r} R_p(A_p) + \lambda_B \sum_{\{p,q\} \in N_B} B_{\{p,q\}} \mid A_p - A_q \mid \\ & + \lambda_C \sum_{\{p_l, q_r\} \in N_C} C_{\{p_l, q_r\}} \mid A_{p_l} - A_{q_r} \mid \end{aligned} \tag{1}$$

where $E(A)$ is the global energy.

In the right of the formulation, the first two terms R and B are almost the same with [4]. The first term (region term) reflects the penalty for assigning "Object" or "background" to pixel p. This can be interpreted as how the intensity of a pixel p fits into a color model:

$$R_p(A_p) = -\log P(A_p|c_p) \tag{2}$$

where $P(A_p|c_p)$ is the probability of p assigned with A_p given its color c_p. In our approach, we use the Gaussian mixture model (GMM) for color modeling. $p \in P_l \cup P_r$ indicates all pixels in the left and right image. The second term measures the boundary properties between two neighboring pixels p and q. $\{p, q\} \in N_B$ indicates all neighboring pixels in the left and right image. The term indicates when p and q have similar colors they should be assigned with the same label and vice versa. The coefficient $B_{\{p,q\}}$ is defined as $\exp(-\beta|| \ c_p - c_q \ ||^2)$ in [4], where β is a constant controls the smoothing strength. The coefficient λ_B stands for a relative importance of boundary term to the other two terms.

The last term in (1) reflects the correspondence property between the two images of a stereo pair. Obviously, corresponding pixels should be assigned with the same label. And thus the correspondence term can be interpreted as a penalty for the mismatch between a pixel p_l in the left image and p_r in the right. In [6] $\{p_l, p_r\} \in N_C$ indicates all possible corresponding pixels. This adds to the graph model $N \cdot D$ links where N is the number of all pixels and D is the maximum disparity. In our approach we reduce the links by only building correspondence edges for consistently matched pixels satisfying $p_l - d_l(p_l) = q_r$ and $q_r + d_r(q_r) = p_l$, where $d_*(p)$ indicates the disparity of p. This improvement reduces the link number to less than N. However, the consistency will not be sacrificed for two reasons. First, the consistently matched pixels plays a role of "seeds" to propagate the segment label from one image to the other with high

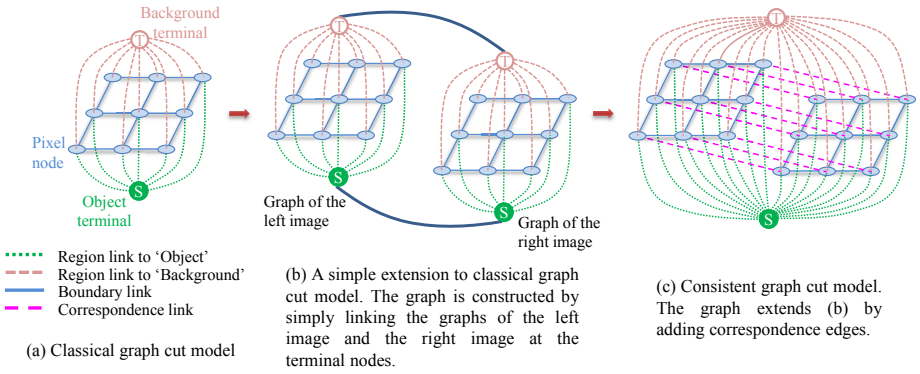


Fig. 3. Graph model comparison

confidence. Second, small errors in correspondence due to the inaccuracy of the disparity map will be recovered by neighboring pixels. The coefficient $C_{\{p_l,q_r\}}$ is defined as the support of p_l corresponding to q_r:

$$C_{\{p_l,q_r\}} = -\log E(p_l, q_r) \tag{3}$$

where $E(p_l, q_r)$ is the matching energy between p_l and q_r generated by ELAS. A lower energy indicates a higher confidence of correspondence. λ_C is a weight factor to balance the relative importance of correspondence term.

Now we construct the graph model of the energy function. The process may be interpreted as a transformation from the classical graph cut model [4] to our consistent graph cut model as shown in Fig. 3. First we construct a classical graph cut model for each image (see Fig. 3 (a)). The vertices are all pixels plus two terminal nodes called "Object terminal" and "Background terminal". Edges between pixel nodes are assigned weights with $B_{p,q}$. Edges linking pixel nodes with object or background terminal nodes are assigned weights with $R_p(\text{“}Object\text{”})$ or $R_p(\text{“}Background\text{”})$. Then we link the two graphs of the left and right image by connecting the terminal nodes (see Fig. 3 (b)). At last, we link the corresponding pixels between the two images and assign the linking edges with weights C_{p_l,q_r} (see Fig. 3 (c)). The graph model is optimized by a max-flow algorithm [15], which indicates a minimization to Equation (1). In step 2, the graph model is updated only for the region term and the user marked seed pixels are set as hard constraints [4], e.g. for pixels marked as "foreground", the weights of the edges linking to background terminal are set as 0 and those linking to object terminal are assigned with K ($K \gg B_{\{p,q\}}$).

3.2 Pre-estimation of Foreground and Background

The user input rectangle provides a rough segmentation of foreground and background named S_O and S_B. Next we use saliency analysis to give a finer segmentation because salient regions are more likely to attract the user's interests

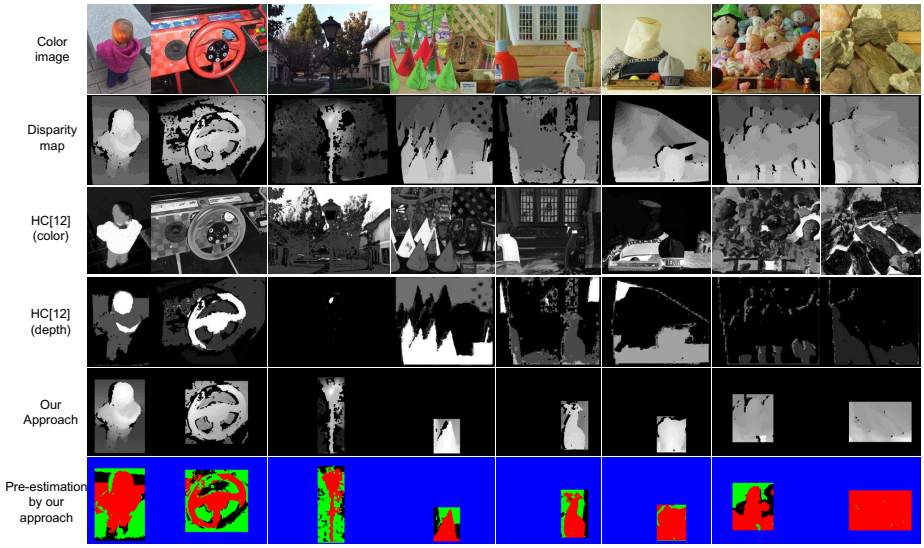


Fig. 4. Depth saliency and pre-estimation of foreground and background. The HC method [12] is applied to color and depth image separately (the 3rd and 4th row). The last row shows the pre-estimation of foreground and background. Red indicates probable "Object". Green indicates probable "Background". Black indicates "Unsure" and won't contribute to color modeling. Blue indicates stable "Background".

perceptually. We only use depth saliency analysis because color is easily confused by textures and thus unsuitable for general purpose segmentation.

We define the saliency value of a pixel as:

$$S(p_i) = \sum_{p_k \in S_B} |d(p_i) - d(p_k)|, \quad p_i \in S_O \tag{4}$$

where $d(p_i)$ indicates the disparity at p_i. The basic assumption is obvious: salient regions are more likely to differ from background. The assumption works well in most cases although it is not always true. Given a counter example, in the case the object is similar to background in depth, the saliency of the object is low everywhere. This degenerates to that using user selected rectangle as object segmentation without saliency analysis, which comes to no harm. Another counter case is that the interesting object is occluded by another object. However, in the following color modeling process the GMM can tolerate some noise so if the occluded region is small we can still produce acceptable results.

Our saliency analysis approach is similar to [12]. The difference is that the saliency value of a pixel is computed as the contrast over all image in their work, but in our approach the contrast is computed over the background, which is more targeted. The computation can be speeded up by a disparity histogram:

$$S_D(d(p_i)) = \sum_{0<d_k\leq D_{max}} f_{d_k} |d_k - d(p_i)|, \quad p_i \in S_O \tag{5}$$

where f_{d_k} is the number of pixels with disparity d_k. The time complexity can be reduced to $O(N_O)$ where N_O is the number of pixels inside the object rectangle. Note unstable pixels (zero disparity) are not computed. We show a few examples of the saliency map in Fig. 4. It can be seen that depth saliency maps are more robust than color saliency maps. Also our approach is more suitable for depth images owing to the utilization of users' guidance.

Next we use the saliency map to make a pre-estimation of foreground and background. We define the top 50% salient pixels as probable "Object" and bottom 20% as probable "Background". The rest are labeled as "Unsure" and not used in color modeling. The pixels outside the object rectangle are stable "Background". We show a few examples of pre-estimation in the last row of Fig. 4. The estimation is rough due to the inaccuracy of the depth map. Also a false example of our depth saliency assumption is shown in the last column. However, the pre-estimation gives a better training set for color modeling than the original rectangle segmentation and thus leads to a considerable improvement for the segmentation results.

We use GMM (Gaussian mixture model) for color modeling, which is the same as [1]. The parameters are initialized using K-means. Then we use the color model, the original color images and the disparity map to create a consistent graph model and perform segmentation as shown in Fig. 2 and Fig. 3.

4 Experiments and Analysis

Our experiment is made up of three parts. First we focus on consistency and accuracy. Next we demonstrate some user interaction examples. At last we show the running time. We compare our approach with two methods: GrabCut [1] and StereoCut [6], which represents the state-of-the-art method for monoscopic and stereo images respectively.

4.1 Dataset, input type and parameter settings

We use the open dataset introduced in [6] for evaluation. The dataset consists of 31 stereo images with groundtruth segmentations, 14 of them are introduced in [16] and have groundtruth disparity maps. We extend the dataset to 100 images covering more natural and daily life scenes. The images are downloaded from `http://www.flickr.com` and taken by a Fuji W3 stereo camera.

We define three kinds of input, rectangle, stroke, and rectangle plus stroke. The rectangle and rectangle plus stroke is used for GrabCut and our approach. Stroke is used for StereoCut. Each kind of input is performed both by machine and by user. In machine type, the input rectangle is a minimum window covering the object generated from groundtruth. The stroke is generated by skeletonization of the object and background. The two types are shown in Fig. 5. In user type, the users freely use the tools to finish the segmentation. Throughout our experiment the parameters are set as: $\{\lambda_B, \lambda_C, \beta, K\} = \{39, 50, 0.001, 1000\}$. Our approach is implemented in C++ and tested on a 2.4GHz Intel T8300 CPU with 2GB RAM.

Fig. 5. Input types. For both rectangle and stroke input, the white pixels indicate "Object" and black ones indicate "Background".

Table 1. Comparison of average consistency (in percentage).

Approach	Input by machine			Input by user		
	Rect	Stroke	Rect+Stroke	Rect	Stroke	Rect+Stroke
GrabCut	95.93	—	98.33	85.72	—	97.20
StereoCut	—	99.13	—	—	99.12	—
Stereo GrabCut	99.44	—	99.38	99.25	—	99.27

Table 2. Comparison of average accuracy (in percentage).

Approach	Input by machine			Input by user		
	Rect	Stroke	Rect+Stroke	Rect	Stroke	Rect+Stroke
GrabCut	81.52	—	94.85	74.01	—	95.02
StereoCut	—	91.49	—	—	94.44	—
Stereo GrabCut	87.36	—	96.62	85.89	—	98.16

4.2 Consistency and Accuracy Evaluation

Consistency is evaluated as $\frac{|C_l|+|C_r|}{|N_l|+|N_r|}$, where N_* is the set of pixels labeled as "Object" in the left or right image, and C_* indicates the set of pixels consistently labeled in N_*. $|*|$ indicates the number of pixels in the set. The pixels without corresponding pixels are not counted to both C_* and N_*.

We show the consistency evaluation in Table 1. The value is averaged over all images and all users. It can be seen that no matter what input type is, our approach and StereoCut performs much better than GrabCut. Especially in the user input type, as users' input rectangles usually differ between the two images of a pair, the consistency value of GrabCut drops down to only 85.72%. StereoCut can also produce highly consistent results but is much more time-consuming, which will be shown in Sect. 4.4. To demonstrate the consistency more clearer, we give a few examples of segmentation results in Fig. 6. It is very interesting that even small inconsistency in statistics appears very sharp in visual perception e.g. "Bowling" and "Dolls" using GrabCut with rectangle inputs (in the 5th row of Fig. 6). They get consistency value of 91.33% and 93.94% respectively but appear to be obviously inconsistent.

Next we evaluate accuracy using the intersection over union metric [19], which is defined as the ratio of intersection to union between the results and ground

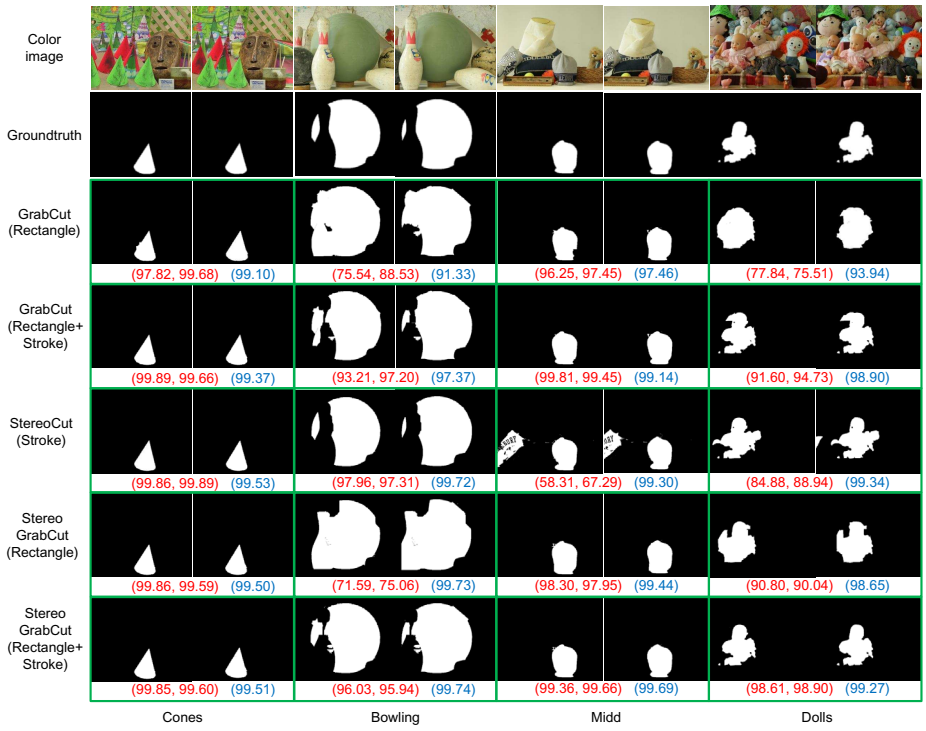


Fig. 6. Examples of GrabCut, StereoCut and our approach. The results are all generated in machine input type. The accuracy and consistency values are also shown below each image. Red numbers within parentheses indicate accuracy values for the left and right segmentation results. Blue numbers indicate consistency values.

truth. As shown in Table 2, in machine input type, when using stroke input all the three methods get high accuracy values. However, GrabCut and our approach both have initial segmentations and thus perform better than StereoCut. In user input type, our approach has made significant improvements to GrabCut. Especially in the first segmentation, our approach makes an improvement of 11.88% to GrabCut because users generally can't draw a minimum rectangle around the object. This is very meaningful for simplifying user interactions because our approach obtains good results at the first segmentation and thus users can focus more on improving small details, which makes our approach perform better than the other two (as shown in the last two columns of Table 2). A few examples are shown in Fig. 6. The stereo images named "Bowling" and "Dolls" are two failure examples of our depth saliency assumption. For "Bowling" the object is occluded by another object and for "Dolls" the object is mingled in background. However for "Dolls" our approach still produces suboptimal results because the GMM can tolerate some noise. For "Bowling" with very little interactions the error can be removed.

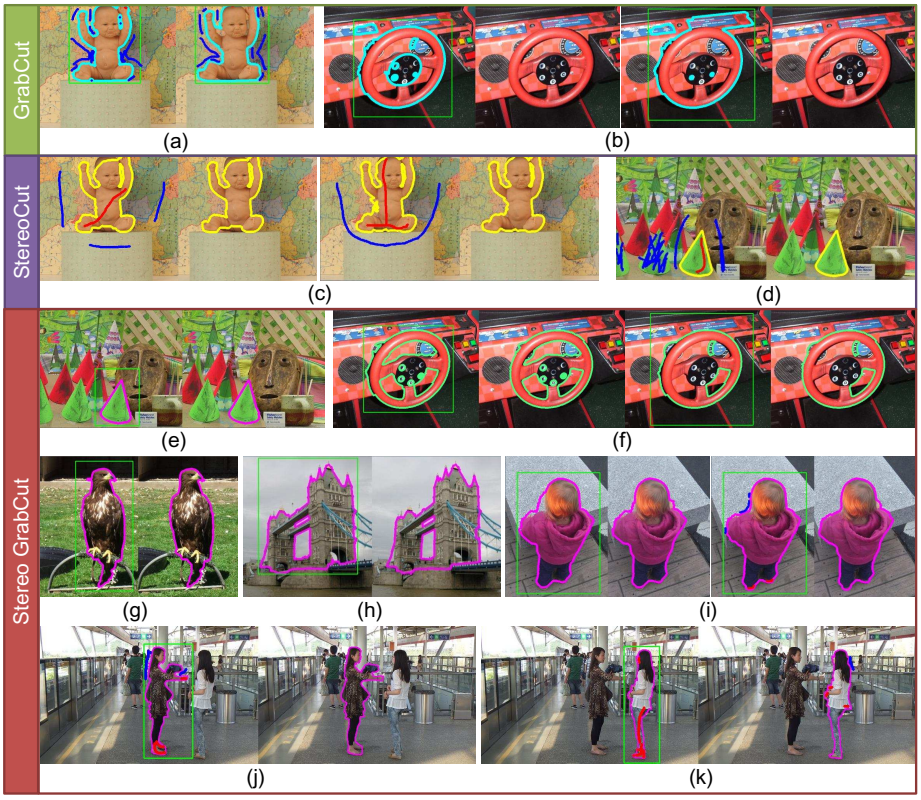
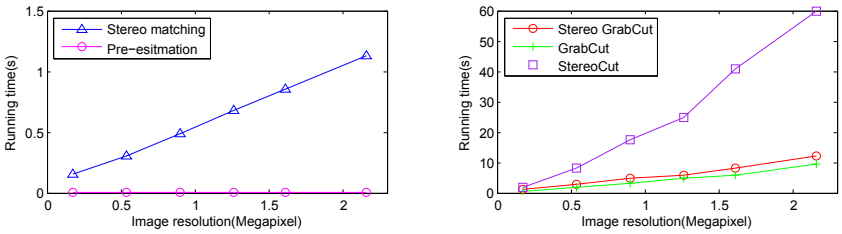


Fig. 7. User interaction examples.

4.3 User Interaction

We asked 10 amateur users to participate in our experiments. We taught them how to use the rectangle and stroke tools, and then show the groundtruth object of each image. Then each user finishes the tasks including 100 stereo images freely. Stereo GrabCut gets the best results with minimum time, which takes about 85 minutes totally. The total time increased to 120 minutes for StereoCut because it is hard to master and responses slowly. For GrabCut it increased to 190 minutes because the workload is doubled. We found the users tend to lose patience when repeating the jobs, especially in GrabCut (Fig. 7 (a)). Besides, users like to scribble a large area when the results are unsatisfactory (Fig. 7 (d)). In comparison, our approach doesn't require the users to draw professional seed labels as StereoCut (Fig. 7 (c)). Also our results do not have high dependence on the input rectangles as GrabCut (Fig. 7 (b) (f)). For most images our approach generates satisfactory results in the first step (see Fig. 7 (e) (g) (h)), which makes the users focus more on refining details (Fig. 7 (i)). Our approach can also handle multi-object extraction (Fig. 7 (j) (k)).



(a) (b)

Fig. 8. Running time. (a) Running time of stereo matching and pre-estimation. (b) Running time comparison of GrabCut [1], StereoCut [6] and our approach.

4.4 Running Time

We show the running time in Fig. 8. The time complexity of stereo matching and pre-estimation is almost linear. For a 2.16 Megapixels stereo image, stereo matching takes 1.13s and pre-estimation takes only 0.008s. The total running time of GrabCut, StereCut and our approach is shown in Fig. 8 (b). StereoCut is the slowest because it uses all possible matching pixels (ND) as correspondence terms. The running time of our approach is close to GrabCut but a little slower. This is mainly due to the expense of consistency forcing.

5 Conclusion and Future Work

We presented an interactive and consistent object extraction approach for stereo images based on graph cut, robust stereo matching and depth saliency analysis. Experiments show that our approach makes promising results in acceptable interaction time and outperforms the other two state-of-the-art methods.

In the future we plan to make efforts in the following directions. First, we will investigate more effective object estimation algorithms using depth information. We have shown the power of leveraging depth for object extraction by saliency analysis. We believe progress in depth analysis can effectively improve the object extraction results. Second, we will apply Stereo GrabCut to further applications such as object-based image editing etc. Also, we will extend our stereo image dataset and use Stereo GrabCut to generate groundtruth, which will be surely valuable for object extraction and recognition researches.

Acknowledgments. This work is supported by the Natural Science Foundation of China (No. 61021062), the 863 Program of China (No. 2011AA01A202) and National Special Fund (No 2011ZX05035-004-004HZ).

References

1. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. In: ACM Transactions on Graphics (TOG). Volume 23., ACM (2004) 309–314

2. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision **1**(4) (1988) 321–331
3. Mortensen, E.N., Barrett, W.A.: Intelligent scissors for image composition. In: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, ACM (1995) 191–198
4. Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In: Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. Volume 1., IEEE (2001) 105–112
5. Li, Y., Sun, J., Tang, C.K., Shum, H.Y.: Lazy snapping. ACM Transactions on Graphics (ToG) **23**(3) (2004) 303–308
6. Price, B.L., Cohen, S.: Stereocut: Consistent interactive object selection in stereo image pairs. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 1148–1155
7. Tasli, H.E., Alatan, A.A.: User assisted stereo image segmentation. In: 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2012, IEEE (2012) 1–4
8. Lo, W.Y., van Baar, J., Knaus, C., Zwicker, M., Gross, M.: Stereoscopic 3d copy & paste. In: ACM Transactions on Graphics (TOG). Volume 29., ACM (2010) 147
9. Kolmogorov, V., Criminisi, A., Blake, A., Cross, G., Rother, C.: Bi-layer segmentation of binocular stereo video. In: Computer Vision and Pattern Recognition, 2005. IEEE Computer Society Conference on. Volume 2., IEEE (2005) 407–414
10. Harville, M., Gordon, G., Woodfill, J.: Foreground segmentation using adaptive mixture models in color and depth. In: Detection and Recognition of Events in Video, 2001. Proceedings. IEEE Workshop on, IEEE (2001) 3–11
11. Ahn, J.H., Kim, K., Byun, H.: Robust object segmentation using graph cut with object and background seed estimation. In: Pattern Recognition, 2006. ICPR 2006. 18th International Conference on. Volume 2., IEEE (2006) 361–364
12. Cheng, M.M., Zhang, G.X., Mitra, N.J., Huang, X., Hu, S.M.: Global contrast based salient region detection. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 409–416
13. Niu, Y., Geng, Y., Li, X., Liu, F.: Leveraging stereopsis for saliency analysis. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 454–461
14. Fu, Y., Cheng, J., Li, Z., Lu, H.: Saliency cuts: An automatic approach to object segmentation. In: Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, IEEE (2008) 1–4
15. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. Pattern Analysis and Machine Intelligence, IEEE Transactions on **23**(11) (2001) 1222–1239
16. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision **47**(1-3) (2002) 7–42
17. Geiger, A., Roser, M., Urtasun, R.: Efficient large-scale stereo matching. In: Computer Vision–ACCV 2010. Springer (2011) 25–38
18. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR). (2012)
19. Lee, W., Woo, W., Boyer, E.: Silhouette segmentation in multiple views. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33**(7) (2011) 1429–1441