Saliency Cuts on RGB-D Images

Yuantian Wang, Lei Huang*, Tongwei Ren, Yunfei Zhang
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
wangyt@smail.nju.edu.cn,{leihuang,rentw}@nju.edu.cn,141250197@smail.nju.edu.cn

ABSTRACT

Saliency cuts aims to segment salient objects from a given saliency map. The existing saliency cuts methods focus on dealing with RGB images and videos, but ignore the exploration of depth cue, which limit their performance on RGB-D images. In this paper, we propose a novel saliency cuts method on RGB-D images, which utilizes both color and depth cues to segment salient objects. Given a saliency map, we first generate segmentation seeds with adaptive triple thresholding. Next, we extend GrabCut by combining depth cue, and use it to generate a roughly labeled map. Finally, we refine the boundary of the salient object adaptively, and produce an accurate binary mask. To the best of our knowledge, this method is the first specific saliency cuts method for RGB-D images. We validated the proposed method on the largest RGB-D image dataset for salient object detection, named NJU2000. The experimental results demonstrate that our method outperforms the state-of-the-art methods.

CCS CONCEPTS

• **Image Processing and Computer Vision** → Segmentation;

KEYWORDS

Saliency cuts, RGB-D image, depth-aware GrabCut, adaptive boundary refinement

ACM Reference format:
Yuantian Wang, Lei Huang*, Tongwei Ren, Yunfei Zhang. 2017. Saliency Cuts on RGB-D Images. In *Proceedings of ICIMCS '17, Tsingtao, Shandong, China, August 23-25, 2017,* 4 pages.
DOI: xx.xxx/xxx_x

1 INTRODUCTION

As a special task in object segmentation, saliency cuts aims to automatically segment salient objects from a given saliency map [8]. It can be used in numerous applications, such as object classification [2, 13], retrieval [11, 12, 21], social media analysis [17, 19, 24], and image annotation [20, 27]. Different to traditional object segmentation methods [25, 26, 28, 29], saliency map generated by saliency detection algorithm [10, 14] is the main input of saliency cuts. In contrast, original images or videos are ignored [1, 22] or used to improve refinement [6, 16].

ICIMCS '17, Tsingtao, Shandong, China

DOI: xx.xxx/xxx_x

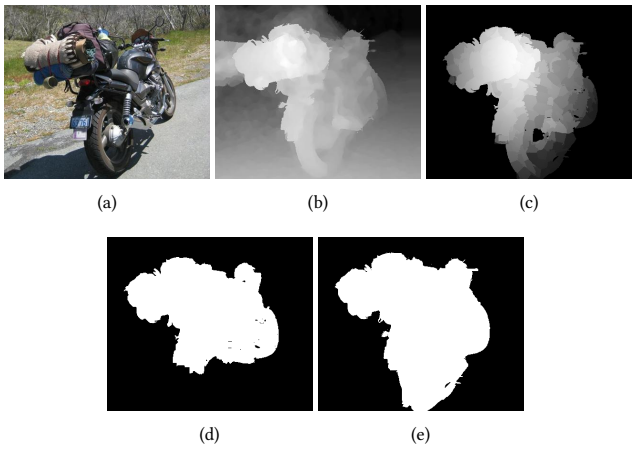

Figure 1: An example of the effect of depth cue in saliency cuts. The saliency cuts result (e) using color cue (a), depth cue (b) and saliency map (c) is better than the one (d) only using color cue and saliency map.

The existing saliency cuts methods are proposed to deal with RGB images and videos. For example, Otsu *et al.* produced segmentation results using thresholds from gray-Level histograms of saliency maps [22]. Achanta *et al.* segmented from the saliency value and luminance of saliency map [1]. Fu *et al.* generated saliency cuts results via professional labels [8]. Cheng *et al.* used a fixed threshold to binarize the saliency maps and produced results from iterative GrabCut calculation [6]. Li *et al.* fed segmentation seeds generated with adaptive triple thresholding method to GrabCut algorithm to produce segmentation results [16]. Banica *et al.* segmented video object via salient segment chain composition [3].

However, these methods ignore the exploration of depth cue, which prevent them to produce better performance on RGB-D images than on RGB images. Figure 1 shows an example of the effect of depth cue in saliency cuts. The saliency cuts result of motorcycle (Fig 1(d)) is incomplete using only color cue (Fig 1(a)) and saliency map (Fig 1(c)), because of the complexity and diversity of motorcycle's appearance in color cue. However, the motorcycle's appearance in depth cue (Fig 1(b)) is relatively simpler, which can help segmenting the motorcycle from the background. Hence, a possible improvement of saliency cuts is to combine color cue with depth cue to produce a better saliency cuts result (Fig 1(e)).

Based on the above observation, we propose a novel saliency cuts method on RGB-D images. Figure 2 shows an overview of the proposed method. We first use adaptive triple thresholding algorithm [16] to generate segmentation seeds from a given saliency map. Then, we feed the segmentation seeds to depth-aware GrabCut algorithm to generate roughly labeled map. Finally, we produce an accurate binary mask via adaptive boundary refinement. As

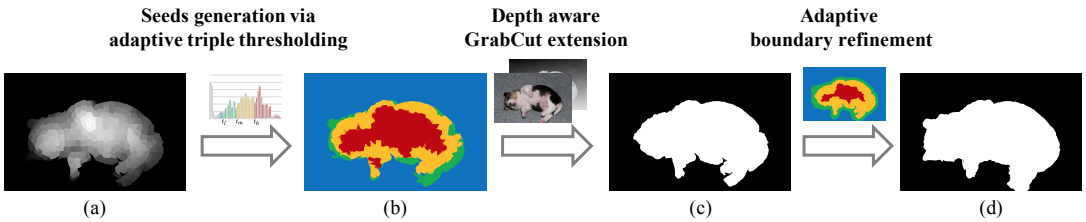

Figure 2: An overview of our proposed method. Given a saliency map (a), we first generate segmentation seeds (b) using adaptive triple thresholding. Next, we feed these segmentation seeds to depth-aware GrabCut to generate roughly labeled map (c). Finally, we refine the boundaries adaptively to produce accurate segmentation result (d).

far as we know, it is the first specific saliency cuts method on RGB-D images. We validated our method on the largest RGB-D image dataset for salient object detection, named NJU2000 [15]. The experimental results show that our method outperforms the state-of- the-art saliency cuts methods on RGB-D images.

Our contributions mainly include:

- We propose the first saliency cuts method on RGB-D images combining adaptive triple thresholding segmentation seeds generation, depth-aware GrabCut, and adaptive boundary refinement.
- We extend the GrabCut algorithm on RGB-D images via combining the color cue and depth cue.
- We validate our method on NJU2000 dataset, and our method is superior to the state-of-the-art methods.

2 OUR METHOD

2.1 Segmentation Seeds Generation via Adaptive Triple Thresholding

We generate segmentation seeds from saliency map using adaptive triple thresholding [16]. Assume the value range of saliency map is $[0, H]$, where H equals 255 in our experiments. t_l, t_m and t_h are the three thresholds used to divide saliency map M^s into four parts: $M^s = \Omega^s_{cb} \cup \Omega^s_{pb} \cup \Omega^s_{pf} \cup \Omega^s_{cf}$. Here, Ω^s_{cb}, Ω^s_{pb}, Ω^s_{pf} and Ω^s_{cf} denote certain background, probable background, probable foreground and certain foreground, which contain the pixels whose saliency values are in the value range of $[1, t_l]$, $[t_l+1, t_m]$, $[t_m+1, t_h]$ and $[t_h+1, H]$, respectively. Obviously, the intersection of each two in Ω^s_{cb}, Ω^s_{pb}, Ω^s_{pf} and Ω^s_{cf} is $\emptyset$. Assume n is the number of pixels on M^s, and n_{cb}, n_{pb}, n_{pf} and n_{cf} are the numbers of pixels on Ω^s_{cb}, Ω^s_{pb}, Ω^s_{pf} and Ω^s_{cf}, respectively. t_l, t_m, t_h are calculated as follows:

$$\{t_l, t_m, t_h\} = \arg\max\{\omega_{cb}\omega_{pb}(\mu_{cb}-\mu_{pb})^2 + \omega_{cf}\omega_{pf}(\mu_{cf}-\mu_{pf})^2\}, \tag{1}$$

where ω_{cb}, ω_{pb}, ω_{pf} and ω_{cf} are the weights of Ω^s_{cb}, Ω^s_{pb}, Ω^s_{pf} and Ω^s_{cf}, which equal $\frac{n_{cb}}{n}$, $\frac{n_{pb}}{n}$, $\frac{n_{pf}}{n}$ and $\frac{n_{cf}}{n}$, respectively; μ_{cb}, μ_{pb}, μ_{pf} and μ_{cf} are the average saliency value of Ω^s_{cb}, Ω^s_{pb}, Ω^s_{pf} and Ω^s_{cf}, respectively.

2.2 Depth-aware GrabCut

The segmentation procedure in GrabCut algorithm can be considered as a mini-cut problem [23]. We extend the energy function E of GrabCut by combining depth cue:

$$E = \alpha E'(L, K^c, \theta^c, Z^c) + (1-\alpha)E'(L, K^d, \theta^d, Z^d), \tag{2}$$

where L is the label set; K^c and K^d are the parameter sets of GMM model on color cue and depth cue; θ^c and θ^d are gray histogram of foreground or background on color cue and depth cue; Z^c and Z^d are the gray value sets of color cue and depth value set of depth cue; α is a parameter for combination, which equals 0.5 in our experiments; $E'(L, K, \theta, Z)$ is the energy function of color cue and depth cue, which is defined as follows:

$$E'(L, K, \theta, Z) = U(l_i, k_i, \theta, z_i) + V(L, Z), \tag{3}$$

where $U(l_i, k_i, \theta, z_i)$ is the data term; $V(L, Z)$ is the smooth term, which is calculated as follows:

$$V(L, Z) = \gamma \sum_{(m,n)\in C} [l_n \neq l_m] \exp -\beta Dis(z_m, z_n)^2, \tag{4}$$

where constant γ equals 50 [4]; C is the set of pairs of neighboring pixels; $\beta = (2\langle(z_m - z_n)^2\rangle)^{-1}$ and $\langle\cdot\rangle$ in β denotes expectation over an colorful image; $Dis(z_m, z_n)$ denotes the distance between pixels m and n.

Referring to [9], we use Euclidean distance $Dis^c(z_m, z_n)$ on color cue and geodesic distance $Dis^d(z^d_m, z^d_n)$ on depth cue, respectively, because geodesic distance can better extract the spatial property of depth cue. We define $Dis^c(z_m, z_n)$ as follows:

$$Dis^c(z^c_m, z^c_n) = ||z^c_m - z^c_n||, \tag{5}$$

where z^c_m and z^c_n are the gray value of pixel m and n on color cue, respectively, and define $Dis^d(z^d_m, z^d_n)$ as follows:

$$Dis^d(z^d_m, z^d_n) = \min\{\varphi_{m,n}\}, \tag{6}$$

where $\varphi_{m,n}$ denotes the distance of a path between pixel m and n, which is calculated as follows:

$$\varphi_{m,n} = \max_{i,j\in P_{m,n}} \{||z^d_i - z^d_j||\}, \tag{7}$$

where i and j are two neighbor pixels on path $P_{m,n}$; z^d_i and z^d_j are the depth value of i and j on depth cue.

Based on the above depth-aware GrabCut algorithm, we generate a roughly labeled map M^{rl}, which contains Ω^{rl}_{cb}, Ω^{rl}_{pb}, Ω^{rl}_{pf} and Ω^{rl}_{cf} with the similar definition to Ω^s_{cb}, Ω^s_{pb}, Ω^s_{pf} and Ω^s_{cf}, after we feed the segmentation seeds M^s.

2.3 Adaptive Boundary Refinement

To obtain more accurate salient objects, we adaptively refine the object boundaries generated by M^{rl} [18].

In order to avoid containing background in the segmented salient objects, we erode Ω_{cf}^{rl} as follows:

$$\Omega_{cf}^{rl'} = f_e(\Omega_{cf}^{rl}, \lambda_1 R(\Omega_{cf}^{rl})), \tag{8}$$

where $R(\Omega_{cf}^{rl})$ is the radius of circumcircle of Ω_{cf}^{rl}; λ_1 is a parameter, which equals 0.1; $f_e(\Omega, R)$ is a function to erode Ω with a radius R.

Ω_{pf}^{cl} is also updated as follows:

$$\Omega_{pf}^{cl'} = (\Omega_{cf}^{rl} \backslash \Omega_{cf}^{cl'}) \cup \Omega_{pf}^{rl}. \tag{9}$$

Meanwhile, to improve the completeness of the segmented salient objects, we dilate the foreground region, *i.e.*, the union of Ω_{cf}^{rl} and Ω_{pf}^{rl}, and refine probable background as the union of Ω_{pb}^{rl} and the newly covered region in dilation, which is defined as follows:

$$\Omega_{pb}^{cl'} = (f_d((\Omega_{cf}^{rl} \cup \Omega_{pf}^{rl}), \lambda_2 R(\Omega_{cf}^{rl} \cup \Omega_{pf}^{rl})) \backslash (\Omega_{cf}^{rl} \cup \Omega_{pf}^{rl})) \cup \Omega_{pb}^{rl}, \tag{10}$$

where $R(\Omega_{cf}^{rl} \cup \Omega_{pf}^{rl})$ is the radius of circumcircle of $\Omega_{cf}^{rl} \cup \Omega_{pf}^{rl}$, λ_2 is a parameter, which equals 0.1; $f_d(\Omega, R)$ is a function to dilate Ω with a radius R.

$\Omega_{cb}^{cl'}$ is also updated as follows:

$$\Omega_{cb}^{cl'} = \Omega_{cf}^{rl} \backslash \Omega_{pb}^{cl'}. \tag{11}$$

We re-feed the segmentation seeds $M^{cl'}$ to depth-aware GrabCut to generate the accurately labeled map M^{dl}, which contains Ω_{cb}^{dl}, Ω_{pb}^{dl}, Ω_{pf}^{dl} and Ω_{cf}^{dl}, and produce the accurate binary mask by defining the binary value of pixels in Ω_{cf}^{dl} and Ω_{pf}^{dl} as 1, denoting object, and defining the binary value of pixels in Ω_{cb}^{dl} and Ω_{pb}^{dl} as 0, denoting background, respectively.

3 EXPERIMENTS

3.1 Dataset and Experiment Settings

We validated our method on the largest RGB-D image dataset for salient object detection, named NJU2000, which contains 2, 000 RGB-D images with manually segmented salient object in ground truth [15]. Saliency maps are generated using Feng's method [7], because it is a state-of-the-art saliency detection method on RGB-D images.

All the experiments were conducted on a computer with 2.9GHz Intel Core i5 CPU and 8GB memory. The average processing time per image of our method is 2.12s. We apply the default settings of author suggestions for all the saliency cuts methods we used in our experiments.

3.2 Component Analysis

We first validate the effectiveness of three components in our method, namely adaptive triple thresholding segmentation seeds generation, depth-aware GrabCut, and adaptive boundary refinement.

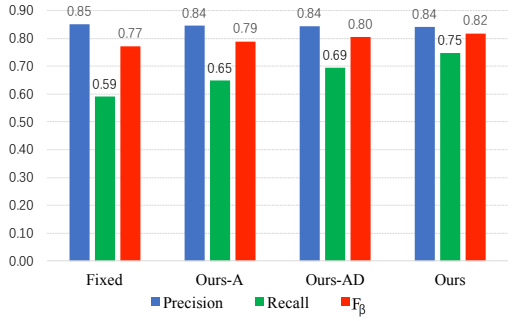

Figure 3: Effectiveness validation of different components in our method. Fixed, Ours-A, Ours-AD, and Ours are shown in Section 3.2.

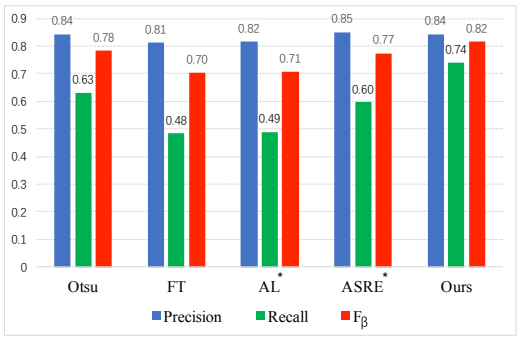

Figure 4: Effectiveness comparison of our method and four state-of-the-art saliency cuts methods, namely Otsu [22], FT [1], AL* [8], and ASRE* [6].

We compare our method with three baselines. *Fixed* denotes the method with segmentation seeds generation using fixed thresholds which uniformly divide saliency value range (*i.e.*, (t_l, t_m, t_h) equals $(64, 128, 192)$), original GrabCut and no boundary refinement, and add three components of our method in sequence to generate the comparison methods. *Ours-A* denotes the method with adaptive triple thresholding segmentation seeds generation, original GrabCut and no boundary refinement. *Ours-AD* denotes the method with adaptive triple thresholding segmentation seeds generation, depth-aware GrabCut and no boundary refinement. *Ours* denotes our proposed method, which uses adaptive triple thresholding segmentation seeds generation, depth-aware GrabCut and adaptive boundary refinement.

Figure 3 shows the precision, recall and F_β values of method Fixed, Ours-A, Ours-AD and Ours, here $\beta^2 = 0.3$ [5]. We can see that the recall and F_β value grow from method Fixed to Ours while precision value keeps relatively consistent. It indicates that each component in our method can help generating better saliency cuts results via improving the completeness of salient object segmentation.

3.3 Comparison with State-of-the-Arts

We also compare our method with four state-of-the-art saliency cuts methods, namely Otsu [22], FT [1], AL [8], and ASRE [6].

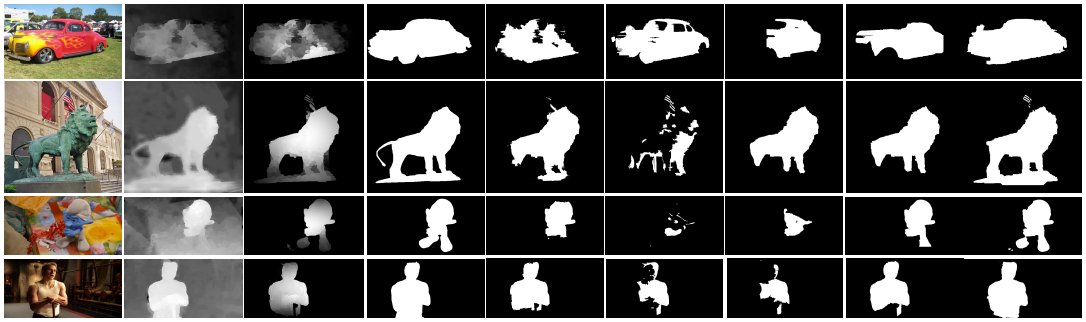

Figure 5: Examples of saliency cuts results of different methods. (a) Color cue. (b) Depth cue. (c) Saliency map. (d) Ground truth. (e) Otsu. (f) FT. (g) AL*. (h) ASRE*. (i) Ours.

Here, Otsu and FT use only saliency maps as input; AL and ASRE generate segmentation seeds from saliency maps, and feed segmentation seeds and RGB images to GrabCut algorithm. To make fair comparison, we extend AL and ASRE to AL* and ASRE* by replacing GrabCut with depth-aware GrabCut, because the later obtains better segmentation performance on RGB-D images.

Figure 4 shows the comparison results of five methods. We can see that our method outperforms other methods on F_β value, because it achieves the best balance between precision and recall. It indicates that our method segments the most complete and accurate salient objects in all five methods. Figure 5 shows some segmentation results generated by five saliency cuts methods on RGB-D images. It shows that our method produces the best segmentation results on various salient objects, such as car, animal, and person.

4 CONCLUSION

In this paper, we proposed the first saliency cuts method on RGB-D images, which utilizes segmentation seeds generation using adaptive triple thresholding , depth-aware GrabCut and adaptive boundary refinement. The proposed method was validated on NJU2000 dataset. The experimental results show that our method is superior to the state-of- the-art saliency cuts methods on RGB-D images.

5 ACKNOWLEDGMENTS

This work is supported by National Science Foundation of China (61321491, 61202320), National Undergraduate Innovation Project (G201610284069), and Collaborative Innovation Center of Novel Software Technology and Industrialization.

REFERENCES

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. 2009. Frequency-tuned Salient Region Detection. In *CVPR*. 1597–1604.
[2] T. Athanasiadis, N. Simou, G. Papadopoulos, R. Benmokhtar, K. Chandramouli, V. Tzouvaras, V. Mezaris, M. Phiniketos, Y. Avrithis, and Y. Kompatsiaris. 2009. Integrating Image Segmentation and Classification for Fuzzy Knowledge-Based Multimedia Indexing.. In *MMM*. 263–274.
[3] D. Banica, A. Agape, A. Ion, and C. Sminchisescu. 2013. Video Object Segmentation by Salient Segment Chain Composition. In *ICCV Workshops*. 283–290.
[4] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. 2004. Interactive Image Segmentation Using an Adaptive GMMRF Model. In *ECCV*. 428–441.
[5] A. Borji, M. M. Cheng, H. Jiang, and J. Li. 2015. Salient Object Detection: A Benchmark. *TIP* 24, 12 (2015), 5706–5722.
[6] M. M. Cheng, G. X. Zhang, N. J. Mitra, X. Huang, and S. M. Hu. 2011. Global Contrast Based Salient Region Detection. In *CVPR*. 409–416.
[7] D. Feng, N. Barnes, S. You, and C. Mccarthy. 2016. Local Background Enclosure for RGB-D Salient Object Detection. In *CVPR*. 2343–2350.
[8] Y. Fu, J. Cheng, Z. Li, and H. Lu. 2012. Saliency Cuts: An Automatic Approach to Object Segmentation. In *CVPR*. 1–4.
[9] L. Ge, R. Ju, T. Ren, and G. Wu. 2015. Interactive RGB-D Image Segmentation Using Hierarchical Graph Cut and Geodesic Distance. In *PCM*.
[10] J. Guo, T. Ren, L. Huang, and J. Bei. 2017. Saliency Detection on Sampled Images for Tag Ranking. *MULTIMEDIA SYST* (2017), 1–13.
[11] R. Hong, Z. Hu, R. Wang, M. Wang, and D. Tao. 2016. Multi-View Object Retrieval via Multi-Scale Topic Models. *TIP* 25, 12 (2016), 5814–5827.
[12] R. Hong, Y. Yang, M. Wang, and X. S. Hua. 2017. Learning Visual Semantic Relationships for Efficient Visual Retrieval. *TBD* 1, 4 (2017), 152–161.
[13] R. Hong, L. Zhang, C. Zhang, and R. Zimmermann. 2016. Flickr Circles: Aesthetic Tendency Discovery by Multi-View Regularized Topic Modeling. *TMM* 18, 8 (2016), 1555–1567.
[14] X. Hou and L. Zhang. 2007. Saliency Detection: A Spectral Residual Approach. In *CVPR*. 1–8.
[15] R. Ju, Y. Liu, T. Ren, L. Ge, and G. Wu. 2015. Depth-aware Salient Object Detection Using Anisotropic Center-surround Difference. *SPIC* 38, C (2015), 115–126.
[16] S. Li, R. Ju, T. Ren, and G. Wu. 2015. Saliency Cuts Based on Adaptive Triple Thresholding. In *ICIP*. 4609–4613.
[17] Z. Li and J. Tang. 2016. Weakly Supervised Deep Matrix Factorization for Social Image Understanding. *TIP* 26, 1 (2016), 276–288.
[18] J. Liu, T. Ren, Y. Wang, S. H. Zhong, J. Bei, and S. Chen. 2017. Object Proposal on RGB-D Images via Elastic Edge Boxes. *NEUCOM* 236 (2017), 134–146.
[19] L. Nie, M. Wang, Z. Zha, and T. S. Chua. 2012. Oracle in Image Search: A Content-Based Approach to Performance Prediction. *TOIS* 30, 2, Article 13 (May 2012), 23 pages.
[20] L. Nie, M. Wang, Z. Zha, G. Li, and T. S. Chua. 2011. Multimedia Answering: Enriching Text QA with Media Information. In *SIGIR*. ACM, 695–704.
[21] L. Nie, S. Yan, M. Wang, R. Hong, and T. S. Chua. 2012. Harvesting Visual Concepts for Image Search with Complex Queries. In *MM*. ACM, 59–68.
[22] N. Otsu. 2007. A Threshold Selection Method from Gray-Level Histograms. *SMC* 9, 1 (2007), 62–66.
[23] C. Rother, V. Kolmogorov, and A. Blake. 2004. "GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts. *TOG* 23, 3 (2004), 309–314.
[24] J. Sang, C. Xu, and J. Liu. 2012. User-Aware Image Tag Refinement via Ternary Semantic Analysis. *TMM* 14, 3 (2012), 883–895.
[25] J. Shi and J. Malik. 2000. Normalized Cuts and Image Segmentation. *TPAMI* 22, 8 (2000), 888–905.
[26] H. Song, Z. Liu, H. Du, G. Sun, Meur O Le, and T. Ren. 2017. Depth-Aware Salient Object Detection and Segmentation via Multiscale Discriminative Saliency Fusion and Bootstrap Learning. *TIP* PP, 99 (2017), 1–1.
[27] J. Tang, R. Hong, S. Yan, T. S. Chua, G. J. Qi, and R. Jain. 2011. Image Annotation by k NN-sparse Graph-based Label Propagation over Noisily Tagged Web Images. *TIST* 2, 2 (2011), 14.
[28] N. Xu, R. Bansal, and N. Ahuja. 2007. Object Segmentation Using Graph Cuts Based Active Contours. In *CVPR*. II–46–53 vol. 2.
[29] L. Ye, Z. Liu, L. Li, L. Shen, C. Bai, and Y. Wang. 2017. Salient Object Segmentation via Effective Integration of Saliency and Objectness. *TMM* PP, 99 (2017), 1–1.