Pacific Graphics (2015)
N. J. Mitra, J. Stam, and K. Xu (Editors)

Short Paper

Efficient Interactive Image Segmentation with Local and Global Consistency

Hong Li[1] , Wen Wu[1] , Enhua Wu[1,2]

[1]Department of Computer and Information Science, University of Macau, Macau, China
[2]Chinese Academy of Sciences, BeiJing, China

Abstract

Interactive image segmentation models aim to classify the image pixels into foreground and background classes given some foreground and background scribbles. In this paper, we propose a novel framework for interactive image segmentation which builds upon the local and global consistency model. The final segmentation results are improved by tackling two disadvantages in graph construction of traditional models: graph structure modeling and graph edge weights formation. The scribbles provided by users are treated as the must-link and must-not-link constraints. Then the graph structure is modeled as an approximately k-regular sparse graph by integrating these constraints and our extended neighboring spatial relationships. Content driven locally adaptive kernel parameter is proposed to tackle the insufficiency of previous models which usually employ a unified kernel parameter. After the graph construction, a novel three-stage strategy is proposed to get the final segmentation results. Experimental results and comparisons with other state-of-the-art methods demonstrate that our framework can efficiently and accurately extract foreground objects from background.

Categories and Subject Descriptors (according to ACM CCS): I.4.6 [Image Processing and Computer Vision]: Segmentation—Pixel classification

1. Introduction

Image segmentation, which is described as extracting meaningful partitions from an image, is one of the most fundamental, well-studied but challenging problems in computer vision. In general, image segmentation models can be divided into two groups: automatic and interactive segmentation. There are many models in each group and [HCX*13] presents a very comprehensive review. In this paper, we only focus on interactive image segmentation models, in the sense that the users provide a partial labeling of the image.

Image segmentation is not easy because of many difficulties, such as noise pollution, illumination variation and background clutter, and so on. In the meanwhile, the segmentation results should also be insensitive to the seeds location and quantity in order to reduce the user effort. To confront all these difficulties, many approaches have been proposed in the literature with impressive results. Popular approaches which are related to our work include graph and region based models.

Graph based segmentation models can be roughly divided into two subgroups: graph-cut based models and random walk based models. Boykov and Jolly [BJ01] propose the first interactive graph-cut model. The user's provided foreground and background seeds are treated as source and sink nodes in graph respectively. The segmentation is performed by the min-cut/max-flow algorithm. It has been very popular because of its strong mathematical foundation provided by the MAP-MRF framework [GPS89]. Rother et al. [RKB04] propose an iterated graph-cut algorithm named GrabCut. It uses a Gaussian mixture model (GMM) to model the pixels colors' distribution and alternates between object estimation and GMM parameter estimation iteratively. Li et al. [LSTS04] also propose an improved (both in speed and accuracy) interactive graph-cut algorithm named Lazy Snapping. They adopt superpixels to construct the graph to reduce the computational cost. All these graph-cut based methods sometimes have the problem of short-cutting and it is usually caused by a lower cost along a shorter cut than that of a real boundary. To overcome this problem, Price et al. [PMC10] propose a geodesic graph cut method which takes geodesic distance (instead of Euclidean distance) into account. It outperforms previous graph-cut based methods

DOI: 10.2312/pg.20151279

when user's provided information separates the background and foreground feature distributions effectively.

Random walk based methods classify an unlabeled pixel via resolving a question: a random walker starts from one location, what is the most probable seed destination? Grady et al. [Gra06] regard the image segmentation as random walk on a graph and demonstrate that their method is more robust to noise, weak boundary detection and ambiguous region segmentation. However, it is very sensitive to the seeded points. Kim et al. [KLL08] propose a generative image segmentation algorithm by utilizing random walk with restart (RWR) which gives the walker two choices: randomly move to one of its neighbors with probability c or jump back to its initial seed point and restart with probability $1-c$. RWR algorithm can segment images with weak boundaries and textures more effectively, but its computational cost is very high because it demands large matrix inversion.

Region based methods can be categorized into two subgroups: region growing, region splitting and merging. Adams and Bischof [AB94] propose a fast and easily implemented method based on region growing. It iteratively add pixels in subregions near the foreground or background subregions to the active set and updates the seeds until all pixels in the image are assigned to a label. It generates unsatisfactory results when foreground and background have close color distribution. Both Maximal Similarity-based Region Merging (MSRM) [JLDC10] and Mating Attributed Relational Graph (MARG) [NGCJ*12] begin with superpixels. MSRM iteratively merges a region into a neighboring region which has the most similar color histogram and updates the histogram of newly merged region until there is no region to be merged. It has high overall computational complexity because it needs computing the histogram similarity in each iteration. MARG constructs two graphs: the input graph, which represents the input superpixels image; and the model graph, which is constructed by the labeled superpixels. Then the region merging is performed by matching these two graphs. This method needs many labeled pixels which is not impractical.

Almost all of these existing interactive segmentation systems provide users with an iterative procedure to add or remove scribbles to temporary results until they get the final satisfactory segmentation result. However, they can only get high precision segmentation results at the cost of high computational complexity or many carefully placed seeds. Obviously, these two disadvantages make their models impractical because the users usually require the system to respond quickly and update the corresponding result immediately for further refinement.

In order to overcome these shortcomings, we propose an efficient interactive image segmentation system that builds upon graph-based semi-supervised learning theory and superpixels. The input image is over-segmented into small homogeneous regions and the user provided scribbles are integrated with superpixels. Then we model the approximately k-regular sparse graph and form the affinity graph matrix using proposed content driven locally adaptive kernel parameter. The final segmentation is generated by a three stage strategy.

2. Efficient Interactive Image Segmentation with Local and Global Consistency

In this section, we first briefly introduce the learning with local and global consistency model in Section 2.1, then present the details of our proposed three stage interactive image segmentation framework in Section 2.2.

2.1. Learning with Local and Global Consistency

Graph-based semi-supervised models usually consist of two main parts: graph modeling and information inference. Given a set of n data points $X=\{x_1,x_2,...,x_q,...,x_n\}$, with each data $x_i \in R^m$, the first q points $\{x_1,x_2,...,x_q\}$ are labeled as the queries and the rest points $\{x_{q+1},...,x_n\}$ are unlabeled. The ranking algorithm aims to rank the remaining points according to their relevances to the labeled queries. Let $f: X \to R^n$ denotes a ranking function which assigns to each data point x_i a ranking value f_i. We can treat f as a vector $f=[f_1,f_2,...,f_n]^T$. y is an indication vector $y=[y_1,y_2,...,y_n]^T$, in which $y_i=1$ if x_i is a query, and $y_i=0$ otherwise.

Next, we define a graph $G=(V,E)$ on these data points, where the nodes V are dataset X and the edges E are weighted by an affinity matrix $W=[w_{ij}]_{n\times n}$. W is often obtained by applying the Gaussian kernel to a distance matrix:

$$w_{ij}=e^{-\frac{d^2(i,j)}{\sigma^2}} \tag{1}$$

where $d(i,j)$ denotes the distance between x_i and x_j and usually is computed via Euclidean distance between colors, σ decides the kernel size. The degree matrix is denoted as $D=diag\{d_1,d_2,...,d_n\}$, where $d_i=\sum_{j=1}^{n} w_{ij}$.

According to Zhou et al. [ZBL*03], cost function associated with the ranking function f is defined to be

$$Q(f)=\frac{1}{2}\Big(\sum_{i,j=1}^{n} w_{ij}\Big\|\frac{1}{\sqrt{d_{ii}}}f_i-\frac{1}{\sqrt{d_{jj}}}f_j\Big\|^2+\mu\sum_{i=1}^{n}\|f_i-y_i\|^2\Big) \tag{2}$$

where the regularization parameter $\mu>0$ controls the balance of the first term (smoothness constraint) and the second term (fitting constraint, containing labeled as well as unlabeled data.). The first term means that nearby points should have similar scores. Then the optimal ranking f^* of queries is computed by solving the following optimization problem:

$$f^*=\arg\min_f Q(f) \tag{3}$$

The solution of Eq. 3 can be denoted as

$$f^*=(I-\alpha S)^{-1}y \tag{4}$$

where I is an identity matrix, and $S = D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$ is the normalized Laplacian matrix, $\alpha = 1/(1+\mu)$. The detailed derivation can be found in [ZBL*03]. We denote this model as LGC for describing convenience.

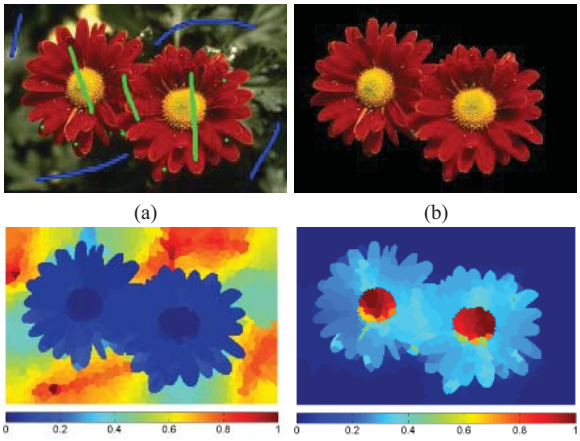

Figure 1: *Three Stage Interactive Segmentation. (a) Input image with labels. (b) Segmentation result. (c) Learning with background labels. (d) Learning with foreground labels.*

2.2. Efficient Interactive Image Segmentation via LGC

Above mentioned graph-based semi-supervised learning algorithm indicates that our interactive image segmentation framework should consist of two main parts: graph construction and information inference.

2.2.1. Labels Driven and Locally Adaptive Graph Construction

To better exploit the intrinsic relationship between data points, there are two aspects should be carefully treated in graph construction: graph structure and edge weights. We over-segment input image into small homogeneous regions using SLIC algorithm [ASS*12] and regard each superpixel as a node in the graph G.

For graph structure, we take the local smoothness cue (i.e., local neighboring superpixels are more likely to belong to the same object) into account and follow three rules. Firstly, each node is not only connected with its direct adjacent neighboring nodes, but also is connected with those nodes sharing common boundaries with its neighboring nodes. Secondly, the nodes labeled as foreground should be connected and the nodes labeled as background should also be connected. Thirdly, the labeled foreground and background nodes should not be connected. First rule models the graph as a k-regular structure by extended neighboring relationships and makes sure the graph structure being sparse. The rest two rules integrate the user's provided information into graph construction and destroy the k-regularity by treating the user provided scribbles as must-link and must-not-link constraints. However, the user provided constraints are much less than total amount of nodes and this makes the graph structure approximately k-regular.

After modeling the graph structure, the very core problem is to get the edge weight between any pairwise nodes given input data. Most models utilize the $L2$ distance based Gaussian kernel (See Eq. 1 for example) with unified kernel width parameter to define edge weights. However, choosing the optimal parameter σ is very challenging. So in this work, we proposal a locally adaptive kernel parameter based edge weights formation strategy, which can be defined as follows

$$w_{ij} = e^{-\frac{c_{ij}}{\sigma_{ij}}} \tag{5}$$

where c_{ij} denotes the Euclidean distance between superpixel region i and j in LAB color space.

The reason for this adaption is straightforward: a good choice of σ should pull intra-class objects together and push extra-class objects apart simultaneously. Different images have different feature representations and using a globally unified σ will not achieve this goal in most time. So we define our local content adaptive kernel width as

$$\sigma_{ij} = \underset{j\in\mathcal{N}(i)}{\Im}(c_{ij}) \tag{6}$$

where $\Im$ denotes the median operation, $\mathcal{N}(i)$ denotes neighboring nodes of superpixel i (all the nodes that have connections with node i).

Our constructed graph takes spatial relationship, user provided information and image content into account. It can exploit the intrinsic structure of input data more properly.

2.2.2. Three Stage Interactive Segmentation

In this section, we present details of our three-stage interactive image segmentation strategy.

Learning with Foreground Labels

We use the user labeled foreground seeds as queries and other nodes as unlabeled data. By this setting, we get the indicator vector y. The ranking scores are learned using Eq. 4. These ranking scores form a N dimensional vector, in which N stands for the number of superpixels (also is the total number of nodes of the graph). Every element in this vector gives the similarity of corresponding node to the foreground queries. Final foreground labels based ranking scores are defined as

$$RS_f(i) = \bar{f}^*(i) \tag{7}$$

where i is the superpixel index and $\bar{f}^*$ is the normalized f^* (in range of $[0,1]$).

Learning with Background Labels

In this stage, we form the indicator vector y by treating the user labeled background seeds as background queries. Then the ranking scores are computed according to Eq. 4

and are normalized into $[0,1]$. Final background labels based ranking scores are defined as

$$RS_b(i) = 1 - \bar{f}^*(i) \tag{8}$$

Notice that $\bar{f}^*$ are the ranking scores according to background queries, so we subtract them from one to get the corresponding foreground based scores.

Integration

When we get the foreground and background ranking scores, the next stage is to integrate them. In this work, we adopt a very simple strategy defined as

$$RS_f(i) = \mathcal{M}(RS_f(i) .* RS_b(i)) \tag{9}$$

where $.*$ stands for pixel-wise product, RS_f and RS_b are defined in Eq. 7 and Eq. 8 respectively, $\mathcal{M}$ denotes mean based thresholding operation defined by

$$\mathcal{M}(f_i) = \begin{cases} 1 & (f_i \geq \mu) \\ 0 & (f_i < \mu) \end{cases} \tag{10}$$

where μ is the mean value of $\{f_1, f_2, ..., f_N\}$.

Figure 1 illustrates this three-stage segmentation strategy. The detailed procedure can be found in Algorithm 1.

Algorithm 1 Efficient Interactive Image Segmentation

Input: Input image and user scribbles
1: Construct the graph as stated in section 2.2.1.
2: Form the foreground and background indicator vectors respectively according to user scribbles.
3: Get the ranking scores by Eq. 7 and Eq. 8 using corredponding indicator y.
4: Integrate the ranking scores and get the final segments using Eq. 9.
Output: Final segments.

3. Experiments and Analysis

To present the advantages over previous algorithms, we conduct qualitative and quantitative evaluations on the GrabCut dataset [RKB04] and some real natural images. Firstly, we will analyze the sensitivity of user scribbles. Then, we show the flexibility of our framework by extending it to single-line cutout problem.Thirdly, we show the segmentation comparisons of applying our method and other four methods: RWR [KLL08], GCPP [LSS09], NHL [KLL10] and CAC [NCZZ12] on some representative images. Finally, we report the running time of these models. The number of superpixels is set to be $N = 500$ in all the experiments. The balance weight α in Eq. 4 is set to be $\alpha = 0.99$ for all the experiments to put more emphasis on the label consistency like previous graph-based semi-supervised learning models usually did. We use green scribbles and blue scribbles to indicate the foreground and background regions respectively in all our experiments.

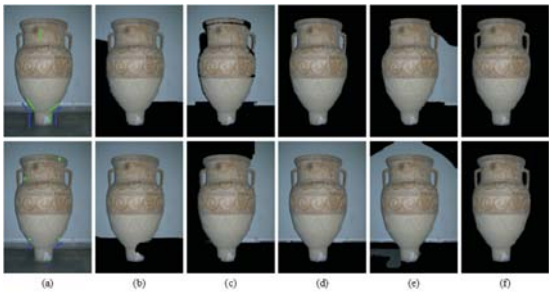

Figure 2: *User scribbles sensitivity comparison. (a) Input images with different user scribbles. (b) Results by RWR [KLL08]. (c) Results by GCPP [LSS09]. (d) Results by NHL [KLL10]. (e) Results by CAC [NCZZ12]. (f) Our results.*

3.1. Comparison of Scribbles Sensitivity

Through extensive experiments we find that the user scribbles play an very important role in the interactive image segmentation models. So a good interactive segmentation model should be insensitive to the locations and quantity of user scribbles. We demonstrate the user scribbles insensitivity of our method in Figure 2. We use less scribbles in bottom row and the scribbles are also placed in different locations in Figure 2(a). The corresponding segmentation results of RWR [KLL08], GCPP [LSS09], NHL [KLL10] and CAC [NCZZ12] are shown in Figure 2(b)-(e) respectively. Segmentation results of our method are shown in Figure 2(f). It can be seen that our method can get almost unchanged best segmentation results given user scribbles of different locations and quantities.

Figure 3: *Single-line cutout. Top row: Input images with single-line label (only foreground labels). Bottom row: Corresponding segmentation results.*

3.2. Single-line CutOut

Because we integrate the user scribbles into graph construction and also take spatial relationships into account, our proposed model can be easily extended to single-label segmentation problem in a straightforward manner. It only needs foreground labels to segment out the desired object. As

shown in Figure 3, it can get satisfying segmentation results using only single line interaction. This will definitely make the segmentation problem more convenient and interesting.

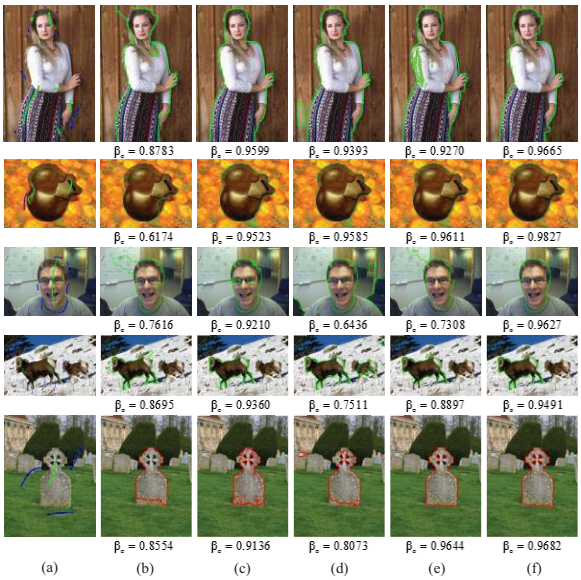

Figure 4: *Comparison of our model with other models. (a) Input images with labels. (b) Results by RWR [KLL08]. (c) Results by GCPP [LSS09]. (d) Results by NHL [KLL10]. (e) Results by CAC [NCZZ12]. (f) Our results.*

3.3. Qualitative and Quantitative Comparison

In Figure 4 and Figure 5, the segmentation results are generated by five algorithms including RWR [KLL08], GCPP [LSS09], NHL [KLL10], CAC [NCZZ12] and ours.

For qualitative comparison, we use same user scribbles to generate the segmentation results. Figure 4 and Figure 5 present fair comparisons of complicated images from the GrubCut dataset [RKB04].

For quantitative comparison, we use the normalized overlap β_o [SG07] to measure the similarity between segmentation result and ground truth quantitatively. It is defined as:

$$\beta_o = \frac{|S_f \cap G_f|}{|S_f \cup G_f|} \tag{11}$$

where S_f is the assigned foreground pixels of the segmentation result and G_f is that of ground truth. This value is presented below each segmentation result in Figure 4. Due to space limitation, we do not show this value in Figure 5.

As can be seen, RWR [KLL08] and GCPP [LSS09] can generally generate satisfactory segmentation results. However, RWR [KLL08] can only get good segmentation results when there are enough user scribbles to surround the desired object. This requirement makes their method inapplicable because it needs more user scribbles. For GCPP [LSS09], it will produce isolated regions (even dots) in bigger foreground regions as shown in fourth and last row of third column. CAC [NCZZ12] will also segment out background regions when the background and foreground have similar colors. NHL [KLL10] also has the problem of producing isolated regions and segmenting out background regions when the corresponding regions have no scribbles. On the other hand, our model consistently outperforms all other models.

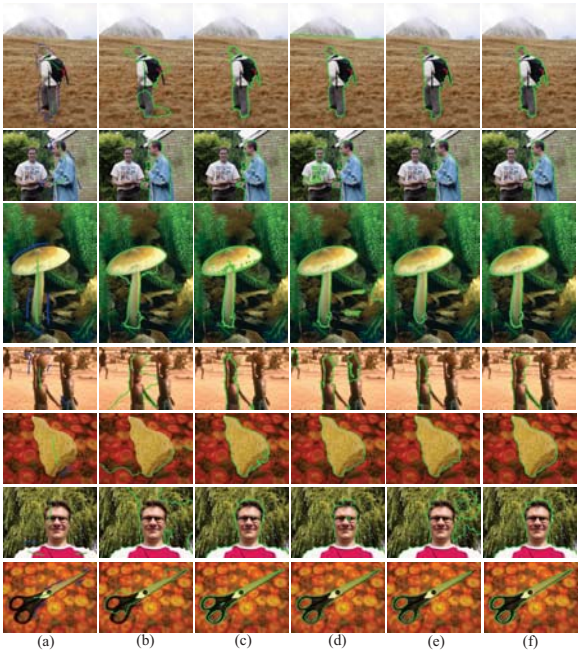

Figure 5: *Comparison of our model with other models. (a) Input images with labels. (b) Results by RWR [KLL08]. (c) Results by GCPP [LSS09]. (d) Results by NHL [KLL10]. (e) Results by CAC [NCZZ12]. (f) Our results.*

Model	Programming language	Time (in s)
NHL	MATLAB	48.79s
CAC	C++	2.8s
RWR	MATLAB	3.13s
GCPP	MATLAB/C++	2.1s
Ours	MATLAB	**1.98s**

Table 1: *Running time of different models.*

3.4. Running Time

The segmentation process should be very fast in order to let the users modify the segmentation results in a real time fashion. We conduct experiments on some representative images and report the mean running time of each model. All the experiments are carried out on a PC with an Intel Core i7 3.2 GHz processor and 16 GB of RAM. Table 1 illustrates the

running time of different models for segmentations on images with size of 640×480.

We can see from Table 1 that NHL [KLL10] needs the most time, it takes about fifty seconds to process an image. The rest four models including ours need almost same time to proceed. It's worth mentioning that our unoptimized MATLAB code only needs less than 2 seconds including over-segmentation computation time to segment the input image. The running time of our model can be sharply reduced by standard multi-cores methods due to the sparsity of our model in C++ implementation.

4. Conclusions and Further Work

In this paper, we propose a novel framework for interactive image segmentation, which generates accurate segmentation results with very fast respond to users' interactions. It is built upon a graph-based semi-supervised learning framework to rank similarities of unlabeled data points with respect to the labeled ones by exploiting the global and local consistency.

To better exploit the intrinsic structure of data, we firstly model the graph as a k-regular graph to take spatial relationships into account. Then we further enhance the graph structure by integrating users' provided scribbles and finally model the graph as an approximately k-regular sparse graph. To overcome the instability brought by the sensitivity of hyper-parameter, we propose a content driven locally adaptive kernel parameter to form the graph edge weights. A three-stage strategy is proposed to generate the final segmentation results. Our framework can also be easily extended to single-line cutout problem. Extensive experiments show that our model consistently outperforms other models both qualitatively and quantitatively. Last but not least, our framework has the least computational cost compared with other four models due to the sparsity of our constructed graph and usage of superpixels.

As future work, we consider three possible directions: multi-features, multi-scale and optimization. We only use color feature for now. There are other features that can be integrated into this framework to better differentiate different regions, such as texture and edge. We employ superpixels as our basic processing unit. The incorrect over-segmentation will affect the final segmentation result. This disadvantage can be overcome effectively by utilizing the multi-scale technique. We will further optimize the framework and consider parallelism to speed up the segmentation procedure.

Acknowledgments

The authors would like to thank the anonymous reviews for their valued suggestions which helped a lot to improve the manuscript. This work has been supported by NSF (National Natural Science Foundation of China, #61272326), the research grant of University of Macau (MYRG202(Y1-L4)-FST11-WEH), and the research grant of University of Macau (MYRG2014-00139-FST).

References

[AB94] ADAMS R., BISCHOF L.: Seeded region growing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 16*, 6 (Jun 1994), 641–647. 2

[ASS*12] ACHANTA R., SHAJI A., SMITH K., LUCCHI A., FUA P., SÜSSTRUNK S.: Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 34*, 11 (Nov 2012), 2274–2282. 3

[BJ01] BOYKOV Y., JOLLY M.-P.: Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on* (2001), vol. 1, pp. 105–112 vol.1. 1

[GPS89] 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. 1

[Gra06] GRADY L.: Random walks for image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 28*, 11 (Nov 2006), 1768–1783. 2

[HCX*13] HU S.-M., CHEN T., XU K., CHENG M.-M., MARTIN R.: Internet visual media processing: a survey with graphics and vision applications. *The Visual Computer 29*, 5 (2013), 393–405. 1

[JLDC10] JIFENG N., LEI Z., DAVID Z., CHENGKE W.: Interactive image segmentation by maximal similarity based region merging. *Pattern Recognition 43*, 2 (2010), 445–456. 2

[KLL08] KIM T. H., LEE K. M., LEE S. U.: Generative image segmentation using random walks with restart. In *Proceedings of the 10th European Conference on Computer Vision: Part III* (2008), ECCV '08, pp. 264–275. 2, 4, 5

[KLL10] KIM T. H., LEE K. M., LEE S. U.: Nonparametric higher-order learning for interactive segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (June 2010), pp. 3201–3208. 4, 5, 6

[LSS09] LIU J., SUN J., SHUM H.-Y.: Paint selection. In *ACM SIGGRAPH 2009 Papers* (2009), SIGGRAPH '09, pp. 69:1–69:7. 4, 5

[LSTS04] LI Y., SUN J., TANG C.-K., SHUM H.-Y.: Lazy snapping. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 303–308. 1

[NCZZ12] NGUYEN T. N. A., CAI J., ZHANG J., ZHENG J.: Robust interactive image segmentation using convex active contours. *Image Processing, IEEE Transactions on 21*, 8 (Aug 2012), 3734–3743. 4, 5

[NGCJ*12] NOMA A., GRACIANO A. B. V., CESAR JR R. M., CONSULARO L. A., BLOCH I.: Interactive image segmentation by matching attributed relational graphs. *Pattern Recogn. 45*, 3 (Mar. 2012), 1159–1179. 2

[PMC10] PRICE B., MORSE B., COHEN S.: Geodesic graph cut for interactive image segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (June 2010), pp. 3161–3168. 1

[RKB04] ROTHER C., KOLMOGOROV V., BLAKE A.: "grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 309–314. 1, 4, 5

[SG07] SINOP A., GRADY L.: A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on* (Oct 2007), pp. 1–8. 5

[ZBL*03] ZHOU D., BOUSQUET O., LAL T. N., WESTON J., SCHÖLKOPF B.: Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*. MIT Press, 2003, pp. 321–328. 2, 3