

BACKGROUND CUTOUT WITH AUTOMATIC OBJECT DISCOVERY

David Liu and Tsuhan Chen

Department of Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, U.S.A.
dliu@cmu.edu and tsuhan@cmu.edu

ABSTRACT

We present a novel approach to background cutout for image editing. We show how background cutout can be achieved without any user labeling. This is in contrast to current methods, where the user needs to label each image separately. Our method uses automatic object discovery methods to provide location and scale estimates of the object of interest; these estimates then provide seeds for initializing color distributions of a segmentation algorithm. We show that our approach can achieve similar performance as traditional methods that require users to specify for each image a bounding box of the target object.

Index Terms— Image segmentation, Unsupervised learning

1. INTRODUCTION

Image segmentation can be categorized into interactive and non-interactive. Interactive systems such as ‘Magic Wand’ [1] and ‘Intelligent Scissors’ [2] have practical importance in image editing. These systems start with a user specified region or rough contour and use texture or edge information to achieve segmentation. Recently, there has been improvements on further reducing the amount of required user interaction to achieve comparable segmentation performance. In the GrabCut method [3], only a rough bounding box is needed, which is a significant improvement over previous methods.

Suppose the user wants to segment the same type of object from a *set* of images, instead of a single image. In previous interactive systems, the user must specify the object within each image, which can be time consuming. If the target objects share certain characteristics, these characteristics can be shared across images. Hence it is possible to further reduce the required amount of human interaction.

In this work, we would like to investigate how well we can segment a *set* of images with *zero* mouse clicks. On a high level, this is achieved by the interaction between a method that provides rough bounding boxes of the target objects in each image, and a method that uses the bounding boxes as seeds to achieve foreground-background segmentation. The method

that provides bounding boxes will be called ‘automatic object discovery’, and the method that achieves segmentation is based on the ‘GrabCut’ method [3].

We will review image segmentation by GrabCut in Section 2. We will then detail the automatic object discovery method in Section 3, and its interaction with GrabCut in Section 4. This is followed by experiments in Section 5, and a summary in Section 6.

2. IMAGE SEGMENTATION BY GRAB CUT

The GrabCut method [3] is based on interactive graph cuts [4], which provides an energy minimization framework for segmenting a single image into foreground (object) and background. Hard constraints are obtained by the user who specifies certain pixels as foreground or background. Soft constraints incorporate both boundary and region information. Minimization is done using a standard minimum cut algorithm. The obtained solution gives the best balance of boundary and region properties among all segmentations satisfying the constraints.

More specifically, two Gaussian mixture models (GMM) are used to model the RGB color of each pixel z_n , one for the foreground and one for the background. A vector $\mathbf{k} = \{k_1, \dots, k_N\}$ assigns to each of the N pixels a unique GMM component, one component either from the background or from the foreground model, according to $\alpha_n = 0$ or 1. The energy function $E = U + V$ consists of a node potential $U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) = \sum_n D(\alpha_n, k_n, \underline{\theta}, z_n)$ where

$$D(\alpha_n, k_n, \underline{\theta}, z_n) = -\log \pi(\alpha_n, k_n) + \frac{1}{2} \log \det \Sigma(\alpha_n, k_n) \\ + \frac{1}{2} (z_n - \mu(\alpha_n, k_n))^T \Sigma(\alpha_n, k_n) (z_n - \mu(\alpha_n, k_n))$$

so that the parameters are

$$\underline{\theta} = \{\pi(\alpha, k), \mu(\alpha, k), \Sigma(\alpha, k), \alpha = 0, 1, k = 1 \dots K\}$$

and a smoothness potential

$$V(\underline{\alpha}, \mathbf{z}) = \gamma \sum_{(m,n) \in N} [\alpha_m \neq \alpha_n] \exp -\beta \|z_m - z_n\|^2$$

where $[\cdot]$ is the indicator function, N is the set of neighboring pixels, and γ and β control the strength of the smoothness term. Once the energy function is defined, the *segmentation mask* (i.e., the foreground-background identity α_n of each pixel) is found through the following steps:

1. User selects a bounding box by mouse clicking. Pixels outside of bounding box are marked as background ($\alpha_n = 0$). Pixels inside the box are marked as α_n unknown.
2. Computer creates an initial image segmentation, where all unknown pixels are tentatively placed in the foreground class and all known background pixels are placed in the background class.
3. Gaussian Mixture Models (GMMs) are created for initial foreground and background classes.
4. Each pixel in the foreground class is assigned to the most likely Gaussian component in the foreground GMM. Similarly, each pixel in the background is assigned to the most likely background Gaussian component.
5. The GMMs are thrown away and new GMMs are learned from the pixel sets created in the previous set.
6. The energy function is minimized to find a new tentative foreground and background classification of pixels (i.e., minimize the energy function E over α_n).
7. Repeat from Step 4 until convergence.

Convergence properties are discussed in more detail in [4][3].

Notice that, without Step 1, the system does not know the color characteristics of the background regions, and hence it cannot determine which regions are foreground and which are background.

3. AUTOMATIC OBJECT DISCOVERY

As mentioned in the introduction, if multiple images are to be segmented, and if these images contain the same type of object (call them the target objects) that the user wants to segment, then it is possible to analyze the region characteristics that consistently occur across images. It is the consistency that tells the foreground from the background apart. In this section, we will introduce such a method.

Topic models such as Probabilistic Latent Semantic Analysis (PLSA)[5], were originally used in the text understanding community for unsupervised topic discovery. In computer vision, topic models have been used to discover object classes, or *topics*, from a collection of unlabeled images. As a result, images can be categorized according to the topics they contain. In the context of unsupervised object detection, the

object of interest and the background clutter are the two *topics*.

Visual words (or textons) [6] are vector quantized local appearance descriptors from patches. Objects can be represented as collections of visual words. We will discuss in more detail in Section 5 on how the visual words are generated. Following the notations used in the text understanding community, $w \in W = \{w_1, \dots, w_{|W|}\}$ is the visual word associated with a patch, $z \in Z = \{z_{FG}, z_{BG}\}$ is a hidden variable that represents the *topic* (foreground or background) associated with a patch, and $d \in D = \{d_1, \dots, d_{|D|}\}$ is the index of the image associated with a patch.

PLSA assumes the joint distribution of d , w , and z can be written as $P(d, w, z) = P(d)P(z|d)P(w|z)$. PLSA is known for its capability of handling polysemy: if a visual word w is observed in two images d_i and d_j , then the topic associated with that word can differ in d_i and d_j : $\arg \max P(z|d_i, w)$ can be different from $\arg \max P(z|d_j, w)$. In other words, PLSA allows a visual word to have different meanings in different images.

We augment the PLSA model in the following way. We introduce an extra variable r in the graph. This variable is directly associated with the α values in GrabCut (Section 2). The idea is to use the segmentation mask produced by GrabCut to guide the automatic object discovery method. This sounds like the opposite direction of what we are seeking for: we wanted to use automatic object discovery to provide seeds for initializing the color distributions in GrabCut. But as we will see later, automatic object discovery and GrabCut are intimately connected, each feeding information to the other.

Figure 1 shows our proposed graphical model. The outer plate indicates the graph is replicated for each image, and the inner plate indicates the replication of the graph for each patch. The topic variable z is hidden. The r value for each patch is obtained by looking up the segmentation mask: if this visual word corresponds to a patch that is segmented by GrabCut as foreground, then $r = 1$; otherwise, $r = 0$.

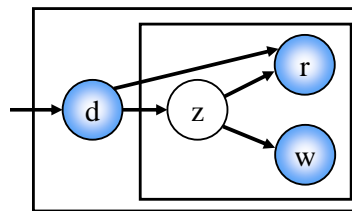


Fig. 1. The proposed graphical model for automatic object discovery.

The segmentation mask is correlated with the hidden topic z (foreground or background). This correlation is expressed in the graphical model as the link from z to r , or $P(r|z)$. We consider the r value of each patch as an additional feature r that is related to the hidden topic variable z . We learn the parameters using the EM algorithm:

E – step :

$$P(z|d, r, w) = k_1 P(z|d)P(w|z)P(r|z) \quad (1)$$

M – step :

$$P(w|z) = k_2 \sum_{d,r} m(d, w, r)P(z|d, r, w) \quad (2)$$

$$P(z|d) = k_3 \sum_{w,r} m(d, w, r)P(z|d, r, w) \quad (3)$$

$$P(r|z) = k_4 \sum_{w,d} m(d, w, r)P(z|d, r, w) \quad (4)$$

where k_1, \dots, k_4 are normalization constants, and $m(d, w, r)$ is a co-occurrence matrix that counts the triples (d, w, r) .

A typical distribution of $P(r|z)$ is shown in Figure 2. From this table we can see how the r value is correlated with z . The foreground topic z_{FG} strongly suggests that the GrabCut segmentation mask is also foreground at the corresponding position, while the ambiguity of the background topic is higher and does not as often correspond to GrabCut’s background. The automatic object discovery method is doing inference based on the co-occurrences of the visual words across images (as PLSA does), *and* also based on the segmentation mask returned by GrabCut. The EM algorithm figures out from data how to optimally make judgements from these two “sensors”: the GrabCut sensor (which provides $\{r\}$) and the appearance sensor (which provides $\{w\}$).

	r=0	r=1
z_{FG}	0.29	0.96
z_{BG}	0.71	0.04

Fig. 2. A typical $P(r|z)$.

4. AUTOMATIC SEEDING

In this section, we will use automatic object discovery to assign seeds without any user interaction. Here are the steps:

1. Automatic object discovery determines the topic of each visual word. In the first iteration, we do not know yet which topic corresponds to the foreground object. Use the topic whose positions of visual words has smaller variance as foreground. These visual words are called the foreground visual words.
2. Find a bounding box for each image: The location and scale of the box are the median and four times the standard deviation of the coordinates of the foreground visual words. We use the median as it is more robust to outliers than the mean.
3. Run GrabCut, except that using the computed bounding box instead of using user input. Get segmentation mask \underline{a} for each image.
4. Use segmentation mask to update the r values in automatic object discovery.
5. Repeat from Step 1 until convergence.

We found this iterative algorithm typically converges in three or four iterations.

Notice that, during automatic object discovery, information is flowing across all images because all images share the same $P(w|z)$ and $P(r|z)$ distributions, whereas during GrabCut, segmentation is done only locally within each image.

5. EXPERIMENTS

We use the Caltech face data set [7] to illustrate the process and results of our method. The method is general and can be applied to other object types as well. We randomly sample twenty images from the Caltech face data set, and resize them to 448×296 pixels.

In GrabCut we use patches found by the Watershed transformation [8] instead of raw pixels as basic units. This speeds up the processing. Using raw pixels can provide better segmentation, while the method stays the same.

In automatic object discovery the basic units are the visual words, which are created as follows. First, elliptical patches are detected by the Hessian Affine interest point detector [9]. We use a codebook size of 500 for quantizing the SIFT descriptors [10] of these patches into visual words. It is worth mentioning here that the SIFT descriptors, and hence the visual words, carry texture information, while the patches used in GrabCut carry color information. Hence our automatic background cutout method is utilizing information from both types of features.

Figure 3 demonstrates results. Figure 3(b) is the result of interactive GrabCut, which requires the user to specify a bounding box as shown in Figure 3(a). Notice that we use Watershed segmented patches instead of raw pixels for speed up, hence the result does not closely follow the object contour, but raw pixels could certainly be used instead. Figure 3(c) to (h) shows the result of our automatic method. The red and green crosses in (c)(e)(g) are the foreground and background visual words. Based on the foreground visual words (the red ones), a bounding box is calculated (not shown). Our method produces the results (d)(f)(h) after the first, second, and third iteration, respectively. The automatic result in Figure 3(h) is comparable to the interactive approach in Figure 3(b).

One of the reasons the result is not perfect is due to the way we calculate the bounding box. Our experience with GrabCut is that the result is quite sensitive to the preciseness of the bounding box, in terms of how close the box covers the

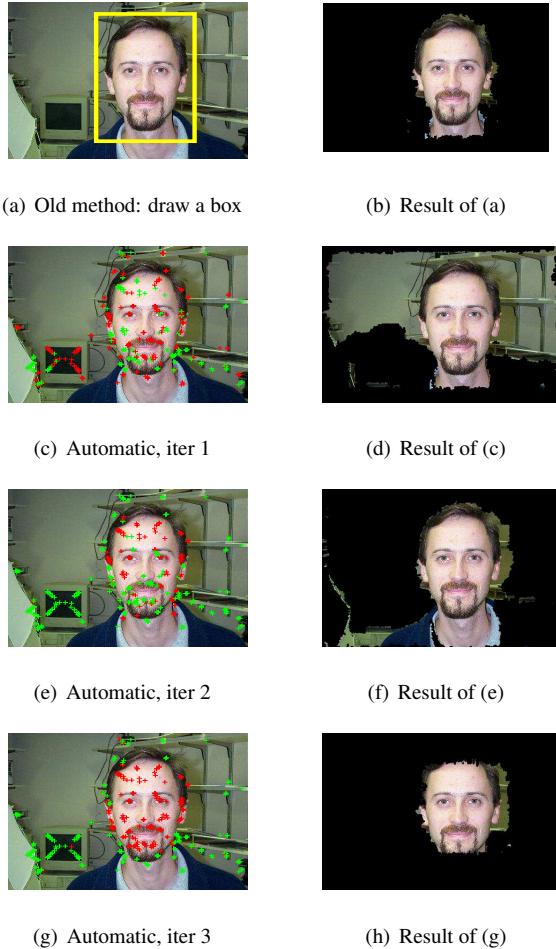


Fig. 3. See Section 5 for details.

object. If too much background is included, then the segmentation is rough. On the other hand, if the box is too small, then some part of the object will be cutout. For example, comparing Figure 3(e) and (g), since the visual words inside the human face are labeled more correctly in (g), the bounding box is tighter in (g) than in (e), resulting in the better results in (h) than in (f). We currently compute the bounding box using median and variance, but more sophisticated robust statistics might provide better boxes.

6. CONCLUSIONS AND FUTURE WORK

First, we have shown how background cutout can be achieved with zero user labeling. This is in contrast to current methods, where the user needs to label each image separately. We have shown that, by using the estimated foreground-background visual words in the automatic object discovery method, a bounding box can be automatically computed and used to initialize GrabCut. In return, the foreground-background segmentation mask of GrabCut can be used to update the features in au-

tomatic object discovery, and hence refining the foreground-background labeling of visual words in the next iteration. Second, our method integrates texture and color information in a novel way: automatic object discovery uses visual words, which captures texture around interest points; GrabCut uses color from patches found by Watershed transformation. Compared with previous automatic object discovery methods that only operate on sparse interest points [11], our method provides finer segmentation.

Since the visual words are relatively sparse, some regions of the foreground object might not be sufficiently covered by the visual words, resulting in biased estimates of the bounding boxes. This could be improved by using denser representations [12]. Future work also includes allowing the user to interact with the system *after* automatic segmentation to correct falsely segmented pixels. For high quality editing, border matting after segmentation is desirable.

7. ACKNOWLEDGEMENTS

This work is supported by the Taiwan Merit Scholarship TMS-094-1-A-049 and by the ARDA VACE program.

8. REFERENCES

- [1] *Adobe Photoshop 7*, Adobe Systems Incorp. 2002.
- [2] E. Mortensen and W. Barrett, “Intelligent scissors for image composition,” in *Proc. ACM Siggraph*, 1995.
- [3] C. Rother, V. Kolmogorov, and A. Blake, “Grabcut - interactive foreground extraction using iterated graph cuts,” in *Proc. ACM Siggraph*, 2004.
- [4] Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images,” in *IEEE Intl. Conf. Computer Vision*, 2001.
- [5] T. Hofmann, “Unsupervised learning by probabilistic latent semantic analysis,” *Machine Learning*, vol. 42, pp. 177–196, 2001.
- [6] J. Malik, S. Belongie, J. Shi, and T. Leung, “Textons, contours and regions: Cue integration in image segmentation,” in *IEEE Intl Conf. Computer Vision*, 1999.
- [7] M. Weber, *Caltech face dataset*, <http://www.vision.caltech.edu/archive.html>.
- [8] L. Vincent and P. Soille, “Watersheds in digital spaces: An efficient algorithm based on immersion simulations,” *IEEE Trans. PAMI*, vol. 13 (6), pp. 583–598, 1991.
- [9] <http://www.robots.ox.ac.uk/vgg/research/affine/>.
- [10] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Intl. Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.
- [11] D. Liu and T. Chen, “A topic-motion model for unsupervised video object discovery,” in *IEEE Conf. Computer Vision and Pattern Recognition*, 2007.
- [12] F. Jurie and B. Triggs, “Creating efficient codebooks for visual recognition,” in *IEEE Intl. Conf. on Computer Vision*, 2005.