IMPROVED GRAPH CUT SEGMENTATION BY LEARNING A CONTRAST MODEL ON THE FLY

Kevin McGuinness and Noel E. O'Connor

CLARITY: Centre for Sensor Web Technologies, Dublin City University, Ireland
{kevin.mcguinness, noel.oconnor}@dcu.ie

ABSTRACT

This paper describes an extension to the graph cut interactive image segmentation algorithm based on a novel approach to addressing the well known small cut problem. The approach uses a generative contrast model to weight interaction potentials. The model attempts to capture the expected changes in color between adjacent pixels in the unlabeled area of the image using the adjacent pixels in the user interactions as training data. We compare our approach to the standard graph cuts algorithm and show that the contrast model allows a user to achieve a more accurate segmentation with fewer interactions. We additionally introduce a variant of the approach based on superpixels that further enhances performance but reduces computational complexity to ensure instant feedback for optimal user experience.

Index Terms— Object segmentation, Interactive segmentation, Graph cuts

1. INTRODUCTION

Accurately segmenting objects from complex scenes generally requires incorporating some prior knowledge of the location and structure of the desired object. In some restricted domains this high-level information can be provided in the form of a prescribed model. For more general applications, like photo editing, this is not possible and typically interactive segmentation techniques are employed whereby a user provides some input to guide the segmentation by marking regions as foreground and background. This knowledge is used to guide the segmentation process, often providing feedback to the user facilitating iterative improvements in the segmentation result.

State-of-the-art interactive algorithms include approaches based on Geodesic distances [1], random walks [2], color models [3] and active contours [4, 5]. However, by far the most popular approach is the Graph Cuts approach [6] that formulates the segmentation problem using a MAP-MRF framework and uses the min-cut/max-flow algorithm [7] to find the minimum of an energy functional by embedding it in a graph. The energy functional depends on a pairwise interaction potential, to assign hard and soft data dependent constraints, and a data penalty term to encourage spatial consistency among labels assigned to neighboring pixels. A key limitation of the graph cuts approach, however, is the so-called small cut problem [8], or shrinking bias [9], which is attributable to a bias in the energy functional favoring small cuts in which less edge links are broken. This is particularly prominent in images with high-contrast textures where the accumulative cost of large cuts is high, and the cost of small cuts through textured regions is comparatively low.

This work is supported by the EU Project FP7 AXES ICT-269980, and by Science Foundation Ireland under Grant 07/CE/I1147 (CLARITY: Center for Sensor Web Technologies).

2. RELATED WORK

One method for tackling the small cut problem is to model the color of the object from the user interactions and use this to assign data penalties to the unlabeled pixels (e.g. [10, 11, 12]). Unfortunately, this method does not work well when the color distributions of the object region and background region are similar, resulting in disconnected regions and segmentations with poor spatial coherence. Several methods have been proposed to improve spatial coherence when using color models to assign data penalties. Rother et al. [11] use both an additional interaction mechanism (a rectangular selection), and iterated graph cuts to produce a more spatially consistent solution. Liu et al. [13] enforce spatial coherence as a post processing step. Vicente et al. [9] find approximate solutions with connectivity priors, and Gulshan et al. [14] use star convexity shape priors.

We propose a different approach: instead of using color models for the object and background and incorporating a shape/connectivity prior, we modify the interaction potentials to encourage larger cuts when appropriate. To achieve this, we use a generative model of contrast that captures information about colors that are expected to be found in adjacent pixels in the object and background regions. Experiments show that this produces a more accurate segmentation using fewer interactions. We additionally introduce a variant of the approach based on superpixels that further enhances performance but reduces computational complexity to ensure instant feedback for optimal user experience.

In the next section we briefly review the small cut problem before presenting our two novel contributions. Section 4

presents a comprehensive evaluation of both approaches in comparison to existing state of the art, before concluding the paper in Section 5.

3. PROPOSED APPROACH

The small cut problem The graph cuts algorithm for interactive segmentation is based on finding a binary labeling L of pixels in an image I that minimizes a energy functional that is a weighted combination of unary and pairwise terms. The energy functional can be written as:

$$E(L) = \sum_{p \in I} D_p(L_p) + \gamma \sum_{(p,q) \in \mathcal{N}} V_{p,q}(L_p, L_q), \quad (1)$$

where $D_p()$ is a data penalty function, and $V_{p,q}()$ is an interaction potential, or smoothness term, designed to encourage the spatial consistency of labels among neighboring pixels $(p, q) \in \mathcal{N}$. In general, the data term $D_p(L_p)$ should represent the cost of assigning label L_p to pixel p, and the smoothness term $V_{p,q}(L_p, L_q)$ the cost of assigning label L_p to pixel p and label L_q to pixel q. The neighborhood set $\mathcal{N}$ usually contains all pairs from the standard 4 or 8 pixel neighborhoods.

To represent hard constraints that result from user interactions, the data penalty term can simply be assigned a value larger than the sum over all interaction potentials associated with the relevant node [6]. This ensures that it is always more costly to assign a background label to a pixel marked as foreground (and vice-versa) than to assign different labels over the pairwise cliques associated with the node.

Various techniques have been proposed for setting the data penalty term for pixels that do not correspond to user interactions. One method is to simply to set these terms to zero. This encourages strong spatial consistency between neighboring pixels, but can often result in the small cut problem – since graph cuts tries to minimize the total edge weights of the cut, the resulting segmentation may be very small, particularly in high-contrast or highly-textured images, where the accumulated costs of longer cuts can be very high. The usual method of avoiding small cuts is to use non-zero data penalty terms for the unknown areas, setting these terms to equal the negative log probability of the pixel being part of a particular region given its value. The original formulation of interactive graph cuts proposes modeling the foreground and background regions from the user interactions using normalized grayscale histograms, and using the grayscale value of the pixel to estimate the negative log probability of the pixel given the histograms. For color images, data penalty terms are often set by using Gaussian mixture models to model the color distribution of the foreground and background regions [10, 11].

Unfortunately, using color models to set the data penalty terms often fails when the color distribution of the object is not sufficiently different from the color distribution of the background. Choosing a good γ parameter for Eq. (1) to balance the importance of the data penalty and interaction potential also becomes difficult when using such models [10]. This typically manifests as unwanted disconnected regions in the segmentation and poor spatial consistency.

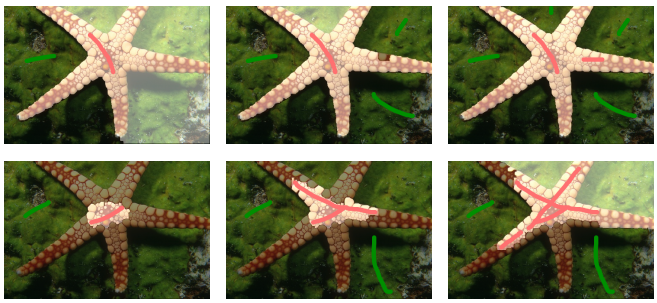

Fig. 1. Illustration of the proposed algorithm on the Corel starfish image. The top row shows the result of the proposed method after two, four, and six user interactions; the bottom row shows the result from standard graph cuts.

Proposed solution Our algorithm takes a different and novel approach to the small cut problem: instead of using color models for the object and background and incorporating some kind of shape or connectivity prior, we instead modify the interaction potentials to encourage larger cuts when appropriate. This allows us to avoid the spatial consistency issues that arise from using color models of the object and background, and the difficulties associated with choosing an appropriate value for γ [10]. To achieve this, we use a generative model of contrast that attempts to capture information about colors that are likely to be adjacent in the object and background regions. The remainder of this section describes the contrast model in detail.

The image contrast in the object and background are modeled as follows. Let F be the set of all pixels that have been marked as foreground by the user, and let P_F be the set of all pairs $\{(p, q) : p, q \in F, q \in \mathcal{N}_p\}$, where $\mathcal{N}_p$ is some neighborhood of pixel p[1]. Note that this set includes (q, p) if it includes (p, q). We define the directional contrast vector for (p, q) as:

$$C_{p,q} = [R_p, G_p, B_p, R_q, G_q, B_q], \quad (2)$$

where R_p, G_p, and B_p are the red, green, and blue channel values for pixel p, and R_q, G_q, and B_q are the same for pixel q[2]. Let $C_F = \{C_{p,q} : (p, q) \in P_F\}$ be the set of all such contrast vectors for the region marked as foreground by the user, and similarly C_B the set of all such vectors for the background.

We fit two separate Gaussian mixture models (GMMs) to C_F and C_B to model the distribution of directional contrast vectors in the known object and background regions. We

[1] For the remainder of the paper we assume a 4-connected neighborhood.

[2] We also tested the algorithm using the more perceptually uniform CIELAB color space, but found that it did not significantly affect accuracy. For computational simplicity, therefore, we use the RGB space.

use full covariance matrices with 6 centers for both GMMs and fit them using the standard EM algorithm for multivariate mixtures (see [15]), and use k-means (initialized with the `k-means++` method [16]) to find good initial centers. Pairwise interaction potentials are then set by evaluating the probability of the associated directional contract vector under both the foreground and background models and taking the maximum:

$$V_{p,q}(L_p, L_q, C_{p,q}) = [p \neq q] \max\{P(C_{p,q}|\theta_F), P(C_{p,q}|\theta_B)\}, \quad (3)$$

where $P(C_{p,q}|\theta_F)$ is the estimated joint probability of $C_{p,q}$ conditioned on the foreground contrast model θ_F, and $P(C_{p,q}|\theta_B)$ is the analog for the background contrast model θ_B.[3]

It is also possible to mix the contrast based interaction potentials from Eq. (3) with traditional smoothness terms. To do this, we treat the usual smoothness term as a weak prior on whether two labels belong to the same class and multiply this by the likelihood of the pixels under the contrast models to find a value for the interaction potential proportional to the posterior probability:

$$P(L_p = L_q|C_{p,q}) \propto P(C_{p,q}|L_p = L_q)P(L_p = L_q) \quad (4)$$

$$\propto P(C_{p,q}|\theta) \exp(-\beta||I_p - I_q||^2), \quad (5)$$

where we take

$$P(C_{p,q}|\theta) = \max\{P(C_{p,q}|\theta_F), P(C_{p,q}|\theta_B)\},$$

and β as a constant that combines information about the color variance σ^2 and a coefficient α attached to the prior: $\beta = \frac{\alpha}{2\sigma^2}$. In our experiments we found that a value of $\alpha = 0.05$ gave marginally superior results to using the contrast model alone. We also set $\sigma^2 = \mathbb{E}[||I_p - I_q||^2]$ as suggested in [11]. Although it is possible to also use an additional color model for the object and background regions with the proposed contrast model, in our experiments we simply set the data term to enforce the hard constraints given by the interactions. That is, we set $D_p(\text{“foreground”}) = K$ if $p \in F$ and $D_p(\text{“foreground”}) = 0$ otherwise, where K is assigned a value larger than the sum over all interaction potentials associated with the node. An analogous strategy is used to set the background data terms.

Superpixel-based variant The algorithm as described above takes approximately 2-5 seconds to process an interaction. One method for reducing computation time is to first pre-segment the image into a smaller number of compact superpixel regions [17]. The resulting segmentation can be viewed as a graph, in which the individual superpixels are the nodes and adjacent superpixels are connected via edges. Graph cuts based segmentation can be performed directly on this graph. Furthermore, assuming that the color variance of each superpixel is relatively low, which we expect to be true if we extract a sufficient number of superpixels, the directional contrast vector composed of the mean colors of adjacent superpixels can then be used to fit the contrast mixture models. The improvement in efficiency is two-fold: it is both less costly to fit the models and less costly to evaluate the probability of the directional contract vectors, since the superpixel graph will generally have fewer edges than the image lattice. Of course, generating the superpixel segmentation needs to be efficient to be used in interactive applications. The recently proposed SLIC superpixel algorithm [18] has been shown to deliver state-of-the-art performance [19], and critically, is ideal for interactive applications, being capable of producing a high-quality segmentation from moderately sized images in under a second [19]. Figure 2 shows an example of the superpixel variant of the algorithm on a typical image.

[3] Although the dependency of V on the color of pixels p and q is formally improper for an MRF, it usually works well in practice (e.g. [6, 12]).

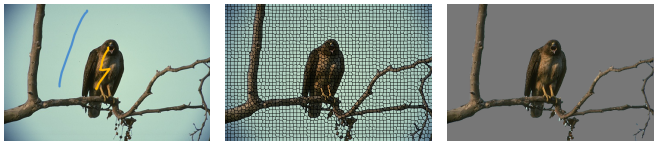

Fig. 2. Contrast cuts on a superpixel graph: user interactions (left), superpixels (middle); segmented foreground (right).

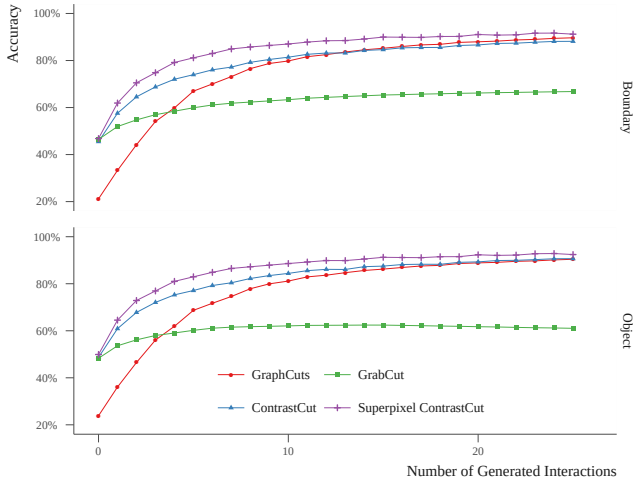

Fig. 3. Comparison of mean boundary accuracy (top) and object accuracy (bottom) over time for the proposed methods, graph cuts, and a single iteration of GrabCut.

4. EVALUATION

We evaluated our algorithm using the evaluation measures, dataset, and ground truth from [20] and using the stochastic user simulation framework proposed in [21]. The dataset consists of 96 images from the Berkeley segmentation dataset [22]

Table 1. Mean boundary and object accuracy (normalized AUC) over all iterations for each of the algorithms evaluated. The first 5 columns are variants of contrast-based algorithm: *RGB* and *CIELAB* refer to variants in the RGB and CIELAB spaces, *No Mixing* refers to a variant that does not mix the traditional smoothness term, and *Diagonal* uses GMMs with diagonal covariances.

	Superpixel (RGB)	RGB	CIELAB	No Mixing	Diagonal	GraphCuts	GrabCut
Boundary	**0.845**	0.793	0.790	0.789	0.769	0.752	0.625
Object	**0.860**	0.823	0.819	0.819	0.795	0.765	0.605

and 100 manually segmented objects with associated descriptions and ground truth. We used both the object and boundary accuracy measures proposed in [20] and the recommended automation strategy from [21] to generate the interactions[4].

We limited the evaluation to a maximum of 25 generated interactions and performed 5 simulation runs for each; accuracy leveled off after approximately 25 interactions and did not improve significantly thereafter. We computed object and boundary accuracy after each interaction, and averaged these values across runs to produce an accuracy value for each step.

Figure 3 shows object and boundary accuracy plotted against iteration (number of simulated interactions) for three algorithms: ContrastCut, the proposed algorithm; GraphCuts, the standard interactive graph cuts algorithm using only hard constraints for the data penalty term; and GrabCut, a single iteration of the GrabCut algorithm using Gaussian mixture models with six components and full covariance matrices to set the data penalty terms. Each algorithm operated in the standard RGB color space, and used the method suggested in [11] to set the variance parameter for the interaction potentials. For the GrabCut implementation we used a value of $\gamma = 100$ as the weighting term in Eq. (1). Clearly, the proposed algorithm provides a more accurate segmentation with fewer interactions. Compared to the standard interactive graph cuts, this difference is particularly prominent in the initial 10–15 interactions.

In our experiments, the superpixel algorithm was configured to produce approximately 2000 superpixels per image ($\approx$ 1.3% of the total number of pixels). With this number of superpixels, the contrast based algorithm can update the segmentation given a new set of user interactions in approximately 370 ms on a standard desktop PC, which allows for a smooth user experience. Furthermore, the superpixel segmentation need only be computed once for each image, so the time required for subsequent updates depends only on the structure of the superpixel graph: it is independent of the image size. Figure 3 shows the mean object and boundary accuracy profiles of the contrast based approach on a superpixel graph generated by SLIC. In addition to being substantially faster, the figure clearly shows that the superpixel variant also unambiguously outperforms the dense lattice variant of the contrast cut algorithm in terms of both object and boundary accuracy.

[4]We reimplemented the user simulation and evaluation framework in Python for our experiments; the code is available at `https://bitbucket.org/kevinmcguinness/python-ise`

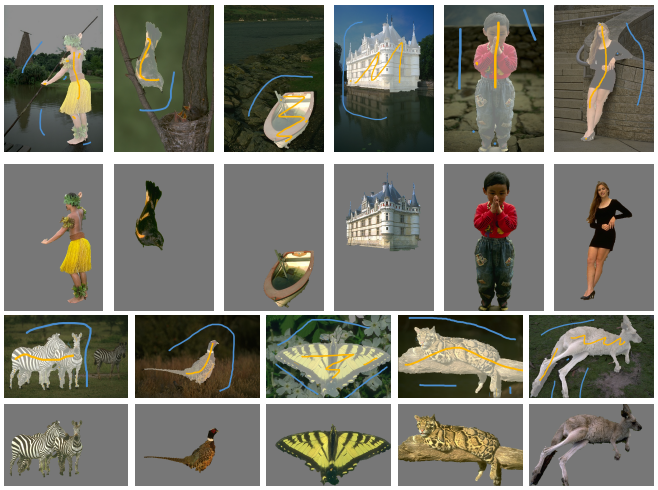

Fig. 4. Illustrative results for the superpixel-based algorithm: interactions and segmented region (top row); extracted foreground object (bottom row).

The mean boundary and object accuracy for the superpixel variant were found to be 0.845 and 0.86. Figure 4 illustrates the segmentation results on images from the BSDS 300 dataset. Table 1 summarizes the evaluation results.

5. CONCLUSION

We have presented an extension to the graph cuts method for interactive segmentation that helps address the small cut problem by using a generative contrast model to weight interaction potentials. Our evaluation has shown that the approach compares favorably against standard graph cuts based segmentation, particularly for the first 10–15 interactions. Using a superpixel graph in place of the dense pixel lattice leads to further improvements in both accuracy and computational cost. The superpixel variant outperforms both GrabCut and standard graph cuts, and is computationally efficient enough to allow for instant feedback in interactive applications. In the future we plan to consider in more detail the effect of the mixing coefficient and the effect of varying the number of superpixels on segmentation accuracy and performance, and investigate methods for reducing the computational cost of updating the GMM parameters, such as [23] or [24].

6. REFERENCES

[1] A. Criminisi, T. Sharp, and A. Blake, "Geos: Geodesic image segmentation," in *Proceedings of ECCV*, 2008, pp. 99–112.

[2] L. Grady, "Random walks for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006.

[3] G. Friedland, K. Jantz, and R. Rojas, "SIOX: simple interactive object extraction in still images," in *Proceedings of the International Symposium on Multimedia*, Dec. 2005, pp. 253–260.

[4] M. Jung, G. Peyré, and L. Cohen, "Non-local active contours," in *Scale Space and Variational Methods in Computer Vision*, vol. 6667 of *LNCS*, pp. 255–266. 2012.

[5] T.N.A. Nguyen, J. Cai, J. Zhang, and J. Zheng, "Robust interactive image segmentation using convex active contours," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3734–3743, Aug. 2012.

[6] Y. Boykov and M.P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images," in *Proc. of ICCV*, Jul. 2001, pp. 105–112.

[7] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.

[8] A.K. Sinop and L. Grady, "A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm," in *Proceedings of ICCV*, Oct. 2007, pp. 1–8.

[9] S. Vicente, V. Kolmogorov, and C. Rother, "Graph cut based image segmentation with connectivity priors," in *Proceedings of CVPR*, June 2008, pp. 1–8.

[10] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, "Interactive image segmentation using an adaptive GMMRF model," in *ECCV 2004*, vol. 3021 of *LNCS*, pp. 428–441. 2004.

[11] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, vol. 23, pp. 309–314, Aug. 2004.

[12] Y. Li, J. Sun, C.K. Tang, and H.Y. Shum, "Lazy snapping," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 303–308, Aug. 2004.

[13] J. Liu, J. Sun, and H.Y. Shum, "Paint selection," in *ACM SIGGRAPH 2009 papers*, 2009, pp. 69:1–69:7.

[14] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman, "Geodesic star convexity for interactive image segmentation," in *Proceedings of CVPR*, June 2010, pp. 3129–3136.

[15] S.J.D. Prince, *Computer Vision: Models, Learning, and Inference*, chapter Chapter 7, pp. 108–115, Cambridge University Press, 1st edition, 2012.

[16] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *Proceedings of the ACM-SIAM symposium on Discrete algorithms*, 2007, SODA, pp. 1027–1035.

[17] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proceedings of ICCV*, Oct. 2003, vol. 1, pp. 10–17.

[18] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels," Tech. Rep. 149300, EPFL, June 2010.

[19] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[20] K. McGuinness and N. E. O'Connor, "A comparative evaluation of interactive segmentation algorithms," *Pattern Recognition*, vol. 43, no. 2, pp. 434–444, Feb. 2010.

[21] K. McGuinness and N. E. O'Connor, "Toward automated evaluation of interactive segmentation," *Computer Vision and Image Understanding*, vol. 115, no. 6, pp. 868 – 884, 2011.

[22] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of Human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of ICCV*, Jul. 2001, vol. 2, pp. 416–423.

[23] Z. Zivkovic and F. van der Heijden, "Recursive unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 651–656, 2004.

[24] E. Hayman and J.O. Eklundh, "Statistical background subtraction for a mobile observer," in *Proceedings of ICCV*. IEEE, 2003, pp. 67–74.