SEED IMAGE SELECTION IN INTERACTIVE COSEGMENTATION

Dhruv Batra, Devi Parikh
{batradhruv, dparikh}@cmu.edu
Carnegie Mellon University

Adarsh Kowdle, Tsuhan Chen
{apk64, tsuhan}@cornell.edu
Cornell University

Jiebo Luo
jiebo.luo@kodak.com
Eastman Kodak Company

ABSTRACT

Interactive image segmentation is a powerful paradigm that allows users to direct the segmentation algorithm towards a desired output. However, marking scribbles on multiple images is a cumbersome process. Recent works show that statistics collected from user input in a single image can be shared among a group of related images to perform interactive cosegmentation. Most works use a naive heuristic of requesting the user input on a random image from the group. We show that in practice, selecting the *right* image to scribble on is critical to the resulting segmentation quality. In this paper, we address the problem of *Seed Image Selection*, *i.e*., deciding which image among a group of related images should be presented to the user for scribbling. We formulate our approach as a classification problem and show that our approach outperforms the naive heuristic used by other works.

Index Terms— Interactive Cosegmentation, Interactive Image Segmentation, Object Cutout.

1. INTRODUCTION

Interactive image segmentation or Object-Cutout is a paradigm that enables users to direct the segmentation algorithm towards a desired output via interactions in the form of scribbles [2, 3, 9], or bounding boxes [11] around objects of interest. However, marking scribbles on multiple images is still a cumbersome process, and recent works [7, 12] have shown that statistics collected from user input in a single image can be shared among a group of related images to perform interactive cosegmentation. Cui *et al*. [7] learn local colour models and edge profile models from a fully segmented image to "transduce" segmentations on novel images. Schnitman *et al*. [12] use a patch-dictionary based method that learns patch-label costs from a fully segmented image to "induce" segmentations on novel images.

Clearly, re-using human effort to achieve segmentations on groups of images is a promising direction. However, we feel that past works ignore two key questions:

1. Which image in the group should be presented to the user for obtaining input (seed image)?
2. How much does the overall group segmentation accuracy depend on this choice of the seed image?

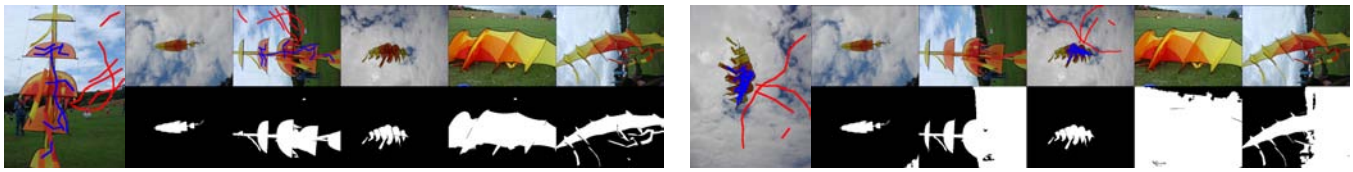
These are precisely the questions we address in this paper. To answer the second question, and to motivate the importance of the first one, we present experiments on a large collection of image groups collected from Flickr. We find that average segmentation accuracy for a group varies significantly with the image that was scribbled, thus making the selection of seed image extremely important (Fig. 1). We refer to this problem as *Seed Image Selection* in interactive cosegmentation. We formulate our approach as a classification problem, where the goal is to predict which image in the group would maximize the average segmentation accuracy for that group. We also find that the heuristic chosen by previous works [7,12] of arbitrarily picking an image from the group is a bad heuristic in practice, one that our method outperforms.

The rest of this paper is organized as follows: Section 2 describes our setup to answer the posed questions, including a description of the dataset and the segmentation algorithm; Sections 3 and 4 describe our analysis that emphasizes the importance of selecting the *right* seed image, and our approach for identifying this seed image in a group; Section 5 presents our experimental results demonstrating the improvement in cosegmentation accuracy we achieve, followed by conclusions and discussions in Section 6.

2. INTERACTIVE COSEGMENTATION

2.1. Dataset

In order to be able to make statistically significant inferences, we need a large segmentation dataset containing multiple groups of related images and pixel-level ground-truth annotations (to compute segmentation accuracies, and quantify answers to the posed questions). To the best of our knowledge no such dataset exists in public domain. We build a large cosegmentation dataset of groups of related images from the Flickr online photo collection, and manually segment (and label) all images. Our dataset consists of 38 groups, with 643 images in total. Examples of these groups are shown in various figures in this paper. To facilitate further research and allow for easy comparisons, this dataset (and annotations) will be made publicly available here [1].



(a) Scribbles (b) Images and corresponding segmentations (c) Scribbles (d) Images and corresponding segmentations

Fig. 1: Importance of Seed Image Selection: In (b,d) the first row shows a group of images with scribbles on image 2 (shown in a) and image 3 (shown in c). Blue scribbles denote foreground pixels, while red scribbles denote background pixels. The second row shows the segmentations achieved by these scribbles (white pixels denote forground). In this group, image 2 is the best image to scribble on, while image 3 is the worst, in terms of mean segmentation accuracies (as can be seen in the masks). This is intuitive because statistics learnt from image 3 do not contain information about grass, which is misclassified as foreground.

We now describe the segmentation algorithm we use for sake of completeness. It should be noted that this is a standard formulation following current trends in Object-Cutout [3, 7, 9], and hence we believe that the conclusions we draw from our results are generalizable to other cosegmentation setups.

2.2. Segmentation Algorithm

We cast our binary labelling problem as an energy minimization problem solved via graph cuts. We work with an oversegmentation of the image. The task is to label each superpixel as foreground or background. We construct a graph over these superpixels, where adjacent superpixels are joined by an edge. Associated with this graph is an energy which is a weighted combination of a data-term and an edge-term. We model the data-term as the negative log-likelihood of the features extracted at a superpixel given the class model. Our features are mean Luv colour features extracted over superpixels, and the class model is a Gaussian Mixture Model (GMM). The edge-term is modeled as a contrast sensitive Potts model using the learnt distances proposed by Batra *et al*. [2]. Finally, we use Graph-cuts to efficiently compute the MAP labels for all superpixels, using the implementation by Boykov *et al*. [4, 5, 8]. Example segmentations can be seen in Fig. 1.

2.3. Simulating User Scribbes

So far, we have described our segmentation algorithm given scribbles provided by a user. However, for the purpose of extensive analysis it is important to be able to perform automatic experiments without explicitly polling a human for these scribbles. Thus, we use the ground-truth segmentations to *simulate* user scribbles. Our scribble generation technique consists of sampling a starting point from the image (with all pixels having equal probability). A direction angle is randomly sampled such that it is highly correlated with the previous direction sample for the scribble, and a fixed-size step is taken along this direction to extend the scribble (as long as the scribble remains within the starting object bounds as provided by the ground truth segmentations). The reason for forcing correlated direction samples is to create smooth continuous scribbles similar to those that humans tend to provide. Examples scribbles can be seen in Fig. 1.

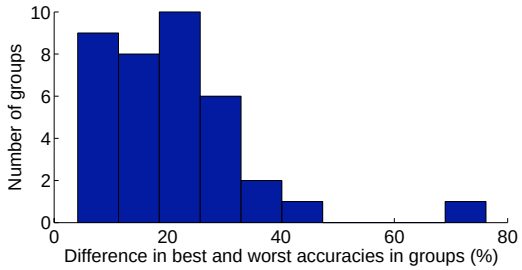


Fig. 2: Histogram of the difference between the best image accuracies and the worst image accuracies over all groups.

3. IMPORTANCE OF THE SEED IMAGE

In this section, we evaluate the importance of finding the *right* image to scribble on. Our experimental setup is as follows: for all the groups in our dataset, we cycle through the images in each group and generate scribbles for this image. Using these scribbles, data-term and edge-term are set up for all the images in this group and Graph-cuts are used to achieve segmentations. In this manner, average segmentation accuracies for a group are computed for each image scribbled. For each group we find the best and worst images to scribble on, *i.e*. the ones that resulted in the highest and lowest average group segmentation accuracies respectively. A large difference between the best and worst accuracy would indicate that for that particular group it is really important to pick the right image to scribble on, because that choice can have a strong impact on the group segmentation accuracies.

Figure 2 shows the histogram of the difference between the best and worst image accuracies for all the groups in our dataset. Notice that the histogram has a heavy tail. This confirms our hypothesis that significant number of groups have a large difference between the best and worst image accuracies. Clearly, this motivates the Seed Image Selection problem.

It is important to note that the inferences drawn are depen-

dent on the difficulty of the group being segmented. Clearly, if a group consists of successive frames from a video sequence, the choice of seed image is irrelevant. The higher the diversity in the images among a group, the more variation we would observe in the accuracies achieved by various seed images.

4. SEED IMAGE SELECTION

4.1. Feature Extraction

We pose the Seed Image Selection problem as a classification task. We extract the following features to describe an image.

Illumination histogram. One of the observations we made was that the presence of strong shadows across the image (Fig. 3(a)) often results in poor cosegmentation accuracies. To capture this intuition, we compute a 50-dim histogram of the gray scale image.

Hue, Saturation and Value entropy. The variety of colours in an image typically corresponds to the amount of useful information in the image, as seen in Fig. 3(b). We quantify this via a 3-dim vector holding the entropies of the hue, saturation and value marginal histograms. The more the number of colours present in an image, the higher the entropies in these distributions would be.

Gradient histogram. The distribution of the strength of edges is a good indicator of how interesting the image is in terms of the existence of several regions/objects in the image. To represent this, we compute a 20-dim histogram of the edge magnitudes across the image. Fig. 3(c) shows that the worst image in the group has very few strong edges as compared to the best image which has more variety in its content.

Scene Gist. The Gist features can help capture a holistic view of the overall scene layout (Fig. 3(d)) . We extract the 1280-dim Gist features [10] which captures the response of the image to gabor filters of different orientations and scales, along with the spatial layout of these responses over the image.

Segmentation histogram. Another indicator of the scene layout is the distribution of the sizes of segments in an image when run through an off-the-shelf segmentation algorithm. For instance, as seen in Fig. 3(e,f), there is a stark contrast in the distribution of sizes of segments found in these images. We use meanshift [6] for generating these segmentations.

4.2. Classification

We split our dataset into training groups and testing groups. We train a linear SVM using each of the n_f (=5) features described in Section 4.1 individually to classify the best image in a group from the worst image. At training time we do not consider the remaining images in a group, because multiple images in groups can be visually similar leading to close cosegmentation accuracies, as seen in Fig. 3(g), and including them during training would make the classification problem artificially hard.

During testing, each image from the test group is passed through these n_f SVMs, and their output scores are recorded. Let the score corresponding to image x_i and SVM (feature) f_a be μ_i^a. The best images in training groups were labeled as the positive class, and thus we expect μ_i^a to be higher for better images.

We are ultimately interested in a ranking of the m images in the test group, and in order to do so, we compute a quality measure for each image, by comparing it to every other image in the group. Each image x_i is assigned a quality measure

$$Q(x_i) = \sum_{j=1}^{m} \sum_{a=1}^{n_f} |(\mu_i^a - \mu_j^a)|_s \quad (1)$$

where $|t|_s$ is the sign function, *i.e.* $+1$ if $t > 0$, and -1 otherwise.

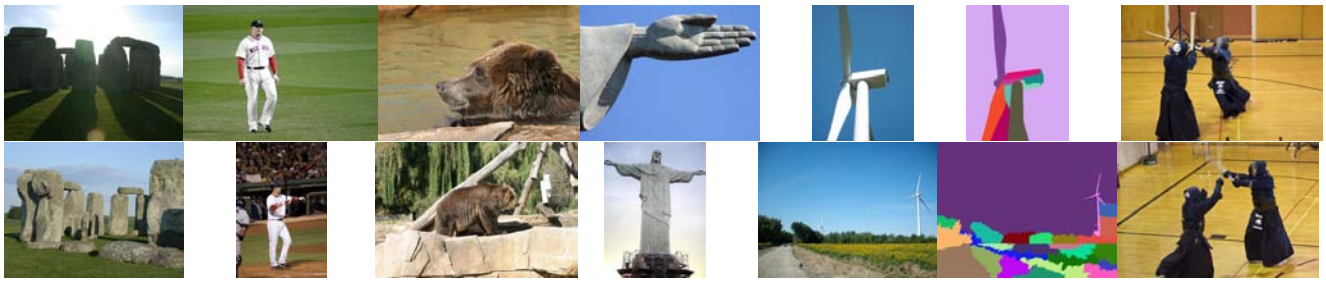
This effectively captures how many times the image x_i got voted as being better than other images in the group, among all features. The m images in a group are ranked by this measure, and the top ranked image is chosen as the seed image.

5. EXPERIMENTS AND RESULTS

For our experiments, we select m to be 5, and retain a random subset of 5 images from all groups. Since one of the 38 groups contained only 4 images, we work with the remaining 37 groups. We perform leave one out cross-validation on the groups. To understand the effectiveness of each of the individual features, we first report their corresponding image classification accuracies for identifying the best image from the worst. The results are shown in Fig. 4. It can be seen that all features hold some information to identify the best images from the worst ones (significantly outperforming chance, which would be 50%). It can be seen that the HSV entropies have the highest accuracy ($\sim$92%). This is understandable, especially since the segmentation algorithm uses colour features. All other features have similar accuracies ($\sim$76%).

To quantify the quality of the final ranking determined by our approach, we match our predicted ranks of images in the test groups, to the ground truth ranks (determined by sorting the average cosegmentation accuracies). We find that on average (across groups), we assign a rank of 2.14 to images that have a ground truth rank of 1. Moreover, the images that we select as rank 1, have, on average, a ground-truth rank of 2.11. In both cases, a random classifier would have an average rank of 3. Although this improvement in ranks may not seem significant, it should be noted that often groups contain more than one image that are "good" for scribbling and give similar segmentation accuracies, *e.g.* images shown in in Fig. 3(g).

The most relevant metric, for our application, is the gain in cosegmentation accuracies achieved by using our proposed Seed Image Selection algorithm, as compared to picking an image from the group at random, which is the heuristic used by previous works [7, 12]. These results are shown in Fig. 5.



(a) (b) (c) (d) (e) (f) (g)

Fig. 3: For columns (a-e), top and bottom rows show best and worst images from example groups, which motivate our choice of features (a) Illumination histogram (b) HSV colour histogram (c) Gradient histogram (d) Gist and (e) Segmentation histogram; (f) Segmentations of images shown in (e); (g) Two images from the group are very similar with similar segmentation accuracies making the ranking a slightly misleading metric. For details please see Section 4.1.

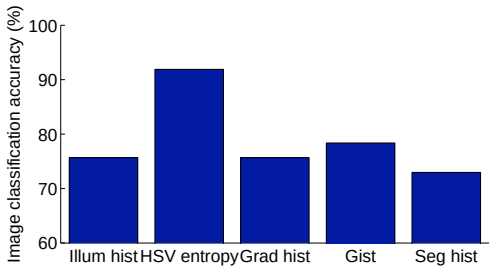


Fig. 4: The classification accuracies of each of our features in identifying best images in a group from the worst images.

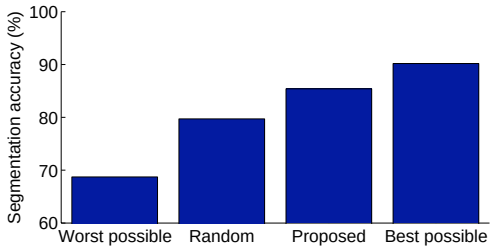


Fig. 5: The final cosegmentation accuracies.

We see that there is more than a 10% gap in the cosegmentation accuracies that can be achieved by scribbling on a randomly selected image (79.7%), and picking the best image in each group (90.2%). It should be noted that the best accuracy is the accuracy which would be achieved if an oracle were to label the best image in each group, and hence is the upper bound on what accuracy we can achieve. We can see that by scribbling on an image recommended by our system, we can fill more than half of this gap (at 85.4%).

6. CONCLUSION

We present the problem of *Seed Image Selection* in interactive cosegmentation, *i.e*. deciding which image in a group should be scribbled on. We collect a large dataset of image groups and manual pixel-level annotations. Our experiments on this dataset show that the group segmentation accuracies vary significantly with the choice of the seed image, and thus the heuristic used by previous works [7, 12] (*i.e*. randomly selecting an image) is a bad heuristic in practice. We formulate this Seed Image Selection problem as a classification problem, and show that we are able to outperform this naive heuristic. It is interesting to note that the improvement in segmentation accuracy is achieved without changing the underlying segmentation algorithm, simply by picking the *right* image to scribble, a question mostly overlooked by existing work in cosegmentation. Future work would involve predicting where in the image the segmentation algorithm should prompt the user for more scribbles, given a current set of scribbles.

Acknowledgement: The authors would like to thank Yu-Wei Chao for data collection and annotation.

7. REFERENCES

[1] D. Batra, A. Kowdle, K. Tang, D. Parikh, and T. Chen. `http://amp.ece.cornell.edu/projects/touch-coseg/`. Interactive Cosegmentation by Touch.

[2] D. Batra, R. Sukthankar, and T. Chen. Semi-supervised clustering via learnt codeword distances. In *BMVC*, 2008.

[3] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *ICCV*, 2001.

[4] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, 2004.

[5] Y. Boykov, O. Veksler, and R. Zabih. Efficient approximate energy minimization via graph cuts. *PAMI*, 20(12):1222–1239, 2001.

[6] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *PAMI*, 24(5):603–619, 2002.

[7] J. Cui, Q. Yang, F. Wen, Q. Wu, C. Zhang, L. V. Gool, and X. Tang. Transductive object cutout. In *CVPR*, 2008.

[8] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004.

[9] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *SIGGRAPH*, 2004.

[10] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[11] C. Rother, V. Kolmogorov, and A. Blake. "Grabcut": interactive foreground extraction using iterated graph cuts. *SIGGRAPH*, 2004.

[12] Y. Schnitman, Y. Caspi, D. Cohen Or, and D. Lischinski. Inducing semantic segmentation from an example. In *ACCV*, 2006.