Int J Comput Vis (2011) 93: 273–292
DOI 10.1007/s11263-010-0415-x

Interactively Co-segmentating Topically Related Images with Intelligent Scribble Guidance

Dhruv Batra · Adarsh Kowdle · Devi Parikh · Jiebo Luo · Tsuhan Chen

Received: 6 May 2010 / Accepted: 8 December 2010 / Published online: 7 January 2011

Abstract We present an algorithm for *Interactive Co-segmentation* of a foreground object from a group of related images. While previous works in co-segmentation have focussed on unsupervised co-segmentation, we use successful ideas from the interactive object-cutout literature. We develop an algorithm that allows users to decide what foreground is, and then guide the output of the co-segmentation algorithm towards it via scribbles. Interestingly, keeping a user in the loop leads to simpler and highly parallelizable energy functions, allowing us to work with significantly more images per group. However, unlike the interactive single-image counterpart, a user cannot be expected to exhaustively examine *all* cutouts (from tens of images) returned by the system to make corrections. Hence, we propose *iCoseg*, an automatic recommendation system that intelligently recommends where the user should scribble next. We introduce and make publicly available the largest co-segmentation dataset yet, the CMU-Cornell iCoseg dataset, with 38 groups, 643 images, and pixelwise hand-annotated groundtruth. Through machine experiments and real user studies with our developed interface, we show that iCoseg can intelligently recommend regions to scribble on, and users following these recommendations can achieve good quality cutouts with significantly lower time and effort than exhaustively examining all cutouts.

Keywords Interactive segmentation · Co-segmentation · Scribbles · Energy minimization

D. Batra (✉)
Carnegie Mellon University, Pittsburgh, PA, USA
e-mail: batradhruv@cmu.edu

A. Kowdle · T. Chen
Cornell University, Ithaca, NY, USA

A. Kowdle
e-mail: apk64@cornell.edu

T. Chen
e-mail: tsuhan@ece.cornell.edu

D. Parikh
Toyota Technological Institute, Chicago, IL, USA
e-mail: dparikh@ttic.edu

J. Luo
Eastman Kodak Company, Rochester, NY, USA
e-mail: jiebo.luo@kodak.com

1 Introduction

If there is one thing that the growing popularity of photo-sharing website like Flickr and Facebook (4 and 10 Billion photos respectively, as of Oct. 2009) has taught us—it is that people love taking photographs. Consumers typically have several related pictures of the same object, event or destination, and this rich collection is just waiting to be exploited by vision researchers—for something as simple as building a collage of all the foregrounds to something more sophisticated like a complete 3D model of a particular object. In many such tasks, it would be useful to extract a foreground object from all images in a group of related images. This co-segmentation of foreground objects from multiple related images is the goal of our paper.

Most existing works on co-segmentation (Rother et al. 2006; Mukherjee et al. 2009; Hochbaum and Singh 2009) can be described as unsupervised co-segmentation techniques. The common central idea in these approaches is to formulate co-segmentation as an energy minimization problem. A single energy function is defined over all pixels in all images in a group. This energy function consists of the usual MRF smoothness prior that encourage smooth

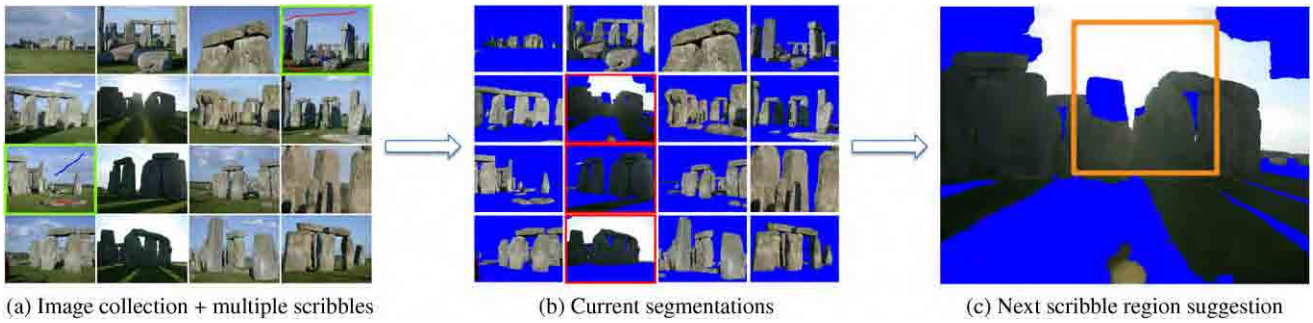


Fig. 1 Overview of iCoseg: (**a**) shows a group of Stonehenge images, and foreground/background scribbles on two images (with *highlighted borders*); (**b**) shows cutouts using these scribbles. A naïve interactive co-segmentation setup would force a user to examine all cutouts for mistakes, and then iteratively scribble on the worst segmentation to obtain better results. Cutouts needing correction are shown with *highlighted borders*. (**c**) Shows the region prompted for more scribbles by iCoseg, thus avoiding exhaustive examination of all cutouts by users

segmentations in each image, and importantly, a histogram matching term, that penalizes dissimilarity between foreground histograms of images in the group. Thus, by minimizing this energy function, these methods jointly segment (or co-segment) all images in the group such these co-segmentations are both smooth and have similar appearance in the foreground regions. In practice, these methods typically work with a pair of images with similar (sometimes nearly identical) foreground, and unrelated backgrounds (e.g. the "Stone-pair" in Fig. 2). This property is necessary because the goal of these works is to extract the common foreground object *automatically*, without any user-input. Due to the nature of our application (i.e. multiple images of the same event or subject), our images typically do not follow this property (see Fig. 2). Hence, without user-input, the task of extracting the foreground object "of interest" is ill-defined.

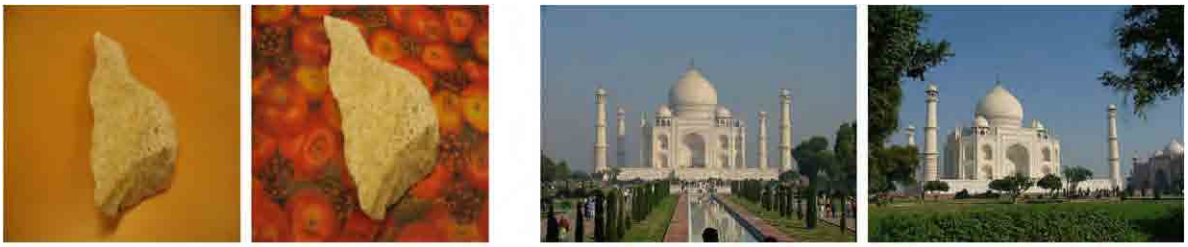
This paper deals with *Interactive* Co-segmentation of a group of (typically $\gg 2$) related images, and presents an algorithm that enables users to quickly guide the output of the co-segmentation algorithm towards the desired output via scribbles. Our approach uses successful ideas from the single-image interactive segmentation literature, where a user provides foreground/background scribbles (Boykov and Jolly 2001; Li et al. 2004) or a coarse bounding-box (Rother et al. 2004) of the object of interest in the image. The algorithm uses this information to not only learn foreground/background appearance models for this image, but also as hard constraints for some pixels, which typically makes the energy minimization problem easier. In our case, a user provides foreground/background scribbles on one (or more) images from a group and our algorithm uses these scribbles to produce cutouts from all images in this group.

In a single-image setup, a user visually inspects the produced cutout and gives more scribbles to correct mistakes made by the algorithm. However, this approach would not work for interactive co-segmentation because (1) as the number of images in the group increases, it becomes increasingly cumbersome for a user to iterate through all the images in the group to find the worst segmentation; and (2) even if the user were willing to identify an incorrect cutout, there might be multiple incorrect cutouts in the group, some more confusing to the segmentation algorithm than others. Observing labels on the most confusing ones first would help reduce the number of user annotations required. It is thus necessary for the algorithm to be able to suggest regions in images where scribbles would be the most informative.

Contributions The main contributions of this paper are:

- We present the first algorithm for intelligent Interactive Co-segmentation (iCoseg), that automatically suggests regions where the user should scribble next.
- We introduce and show results on the largest co-segmentation dataset yet, the CMU-Cornell iCoseg Dataset, containing 38 groups with 17 images/group on average (total 643 images) and pixelwise hand-annotated ground-truth. This dataset and annotations are now publicly available (Batra et al. 2009) to facilitate further work, and allow for easy comparisons. Figure 5 shows some prototypical images.
- We develop a publicly available interface (Batra et al. 2009) for interactive co-segmentation. We present results of simulated machine experiments as well as real user studies on our interface. We find that iCoseg can intelligently recommend regions to scribble on, and help users achieve good quality cutouts with significantly lower time and effort than having to examine all cutouts exhaustively.

Technique Our approach is composed of two main parts: (1) an energy minimization framework for interactive co-segmentation; and (2) a scribble guidance system that uses active learning and some intuitive cues to form a recommendation map for each image in the group. The system recommends a region with the highest recommendation score. See Fig. 1 for an overview.



(a) Stone-pair (Rother et al, 2006). (b) Taj-Mahal-pair from the CMU-Cornell iCoseg Dataset.

Fig. 2 What is foreground? The stone-pair (**a**) has significant variation in background with nearly identical foreground and thus unsupervised co-segmentation can easily extract the stone as foreground. The Taj-Mahal-pair is fairly consistent as a whole and thus the Taj Mahal cannot be cut out via unsupervised co-segmentation. Bringing a user in the loop is necessary for the problem of foreground extraction to be well defined

Organization The rest of this paper is organized as follows: Sect. 2 discusses related work; Sect. 3 presents our energy minimization approach to interactive co-segmentation of a group of related images; Sect. 4 presents our recommendation scheme for guiding user scribbles; Sect. 5 introduces our benchmark dataset and discusses some dataset statistics; Sect. 6 discusses the results of simulated machine experiments and a real user-study; Sect. 7 presents an application of interactive co-segmentation for interactive 3D modelling of an object of interest; Finally, Sect. 8 concludes the paper with discussions.

A preliminary version of this work appeared as a conference paper (Batra et al. 2010), and an expanded version of Sect. 7 is due to be presented at a workshop (Kowdle et al. 2010). This article differs both in presentation and experimental evaluation. We include a more thorough description of the problem statement and how it is different from other works in unsupervised co-segmentation. We perform a more thorough analysis of our introduced dataset (Sect. 5), including more dataset statistics (Sect. 5.1), experiments with scribbles restricted to a single image (Sect. 6.1.1) and effect of dense vs. sparse scribbles (Sect. 6.1.2). We compare our machine experiments with the user study, and analyze user scribble statistics and compare them with our synthetically generated scribbles (Sect. 6.3). We discuss limitations of the approach and failure cases (Sect. 6.4).

2 Related Work

Unsupervised Co-segmentation Rother et al. (2006) introduced the problem of (unsupervised) co-segmentation of image pairs. Their approach is to minimize an energy function that is a combination of the usual MRF smoothness prior *and* a histogram matching term that forces foreground histograms of images to be similar. Mu and Zhou (2007) extend this framework with quadratic global constraints. More recently, Mukherjee et al. (2009) proposed half-integrality algorithms, and Hochbaum and Singh (2009) modified the histogram matching term to propose max-flow based algorithms. Gallagher and Chen (2008) used co-segmentation of clothing to help in recognizing people. Lee and Grauman (2010) have recently proposed an iterative procedure to simultaneously discover object categories and co-segment them in image collections. The common theme here is *unsupervised* co-segmentation, which is achieved by forcing histogram consistency between foregrounds. As noted earlier, this would fail for pairs with related backgrounds (see Fig. 2), where the problem of identifying the foreground objects is ill-posed. This is where our work of *interactive* co-segmentation fits in, which allows a user to indicate the foreground objects through simple scribbles. Moreover, our technique is a natural extension of Boykov and Jolly (2001) that easily generalizes to multiple images (Sect. 3). We note that a recent work by Vicente et al. (2010) compares these different models for unsupervised co-segmentation and finds that a model similar to the one used in this work seems to perform best in practice.

Supervised Co-segmentation Schnitman et al. (2006) and Cui et al. (2008) learn to segment from a single fully segmented image, and then "induce" (Schnitman et al. 2006) or "transduce" (Cui et al. 2008) segmentations on a group of related images. We, on the other hand, utilize very sparse user interaction (in the form of scribles), which are not restricted to a single image and can be provided on multiple images in a group if desired.

Interactive Image Segmentation Boykov and Jolly (2001) posed interactive single-image segmentation given user scribbles as a discrete optimization problem. Li et al. (2004) and Rother et al. (2004) presented simplified user interactions. Bai and Sapiro (2007) and Criminisi et al. (2008) proposed techniques built on efficient geodesic distance computations. As will become clear in the next section, our approach to multiple-image interactive co-segmentation is a natural extension of Boykov and Jolly (2001).

Active Learning Related to our paper are works on active learning where algorithms are able to choose the data they learn from by querying the labelling oracle. This is a vast sub-field of machine learning and we refer the reader to Settles (2009) for a detailed survey. In computer vision, active learning has been used for object categorization (Kapoor et al. 2007), classifying videos (Yan et al. 2003), ranking images by informativeness (Vijayanarasimhan and Grauman 2009) and creating large datasets (Collins et al. 2008). Kohli and Torr (2008) showed how to compute uncertainties from graph-cut solutions and suggested that these may be helpful in interactive image segmentation applications. To the best of our knowledge, this is the first paper to use uncertainties to guide user scribbles.

3 iCoseg: Energy Minimization

Energy Minimization Our approach to multiple-image interactive co-segmentation is a natural extension of Boykov and Jolly (2001). Given user scribbles indicating foreground/background, we cast our labelling problem as minimization of Gibbs energies defined over graphs constructed over each image in a group. Specifically, consider a group of m image-scribble pairs $D = \{(\mathcal{X}^{(1)}, \mathcal{S}^{(1)}), \ldots, (\mathcal{X}^{(m)}, \mathcal{S}^{(m)})\}$, where the kth image is represented as a collection of n_k sites to be labelled, i.e. $\mathcal{X}^{(k)} = \{X_1^{(k)}, X_2^{(k)}, \ldots, X_{n_k}^{(k)}\}$, and scribbles for an image $\mathcal{S}^{(k)}$ are represented as the partial (potentially empty)[1] set of labels for these sites. For computational efficiency, we use superpixels as these labelling sites (instead of pixels).[2] For each image (k), we build a graph, $\mathcal{G}^{(k)} = (\mathcal{V}^{(k)}, \mathcal{E}^{(k)})$, over superpixels, with edges connecting adjacent superpixels.

At the start of our algorithm, we require at least one foreground and background scribble each. They can be in the same image, or in multiple images. Subsequent iterations can have a scribble just from foreground or background. Using these labelled sites, we learn a group appearance model $\mathcal{A} = \{A_1, A_2\}$, where A_1 is the first-order (unary) appearance model, and A_2 the second-order (pairwise) appearance model. This appearance model $(\mathcal{A})$ is described in detail in the following sections. We note that all images in the group share a common model, i.e. only one model is learnt. Using this appearance model, we define a collection of energies over each of the m images as follows:

$$E^{(k)}\big(\mathcal{X}^{(k)} : \mathcal{A}\big) = \sum_{i \in \mathcal{V}^{(k)}} E_i\big(X_i^{(k)} : A_1\big) + \lambda \sum_{(i,j) \in \mathcal{E}^{(k)}} E_{ij}\big(X_i^{(k)}, X_j^{(k)} : A_2\big), \quad (1)$$

where the first term is the data term indicating the cost of assigning a superpixel to foreground and background classes, while the second term is the smoothness term used for penalizing label disagreement between neighbours. Note that the (:) part in these terms indicates that both these terms are functions of the learnt appearance model. From now on, to simplify notation, we write these terms as $E_i(X_i)$ and $E_{ij}(X_i, X_j)$, and the dependence on the appearance model $\mathcal{A}$ and image (k) is implicit.

Data (Unary) Term Our unary appearance model consists of a foreground and background Gaussian Mixture Model, i.e., $A_1 = \{\text{GMM}_f, \text{GMM}_b\}$. Specifically, we extract colour features extracted from superpixels (as proposed by Hoiem et al. 2005). We use features from labelled sites in all images to fit foreground and background GMMs (where number of Gaussians was automatically learnt by minimizing an MDL criteria—Bouman 1997). We then use these learnt GMMs to compute the data terms for all sites, which is the negative log-likelihood of the features given the class model.

Smoothness (Pairwise) Term The most commonly used smoothness term in energy minimization based segmentation methods (Cui et al. 2008; Rother et al. 2004; Criminisi et al. 2008) is the contrast sensitive Potts model:

$$E(X_i, X_j) = \mathrm{I}(X_i \neq X_j) \exp(-\beta d_{ij}), \quad (2)$$

where I $(\cdot)$ is an indicator function that is 1(0) if the input argument is true (false), d_{ij} is the distance between features at superpixels i and j and β is a scale parameter. Intuitively, this smoothness term tries to penalize label discontinuities among neighbouring sites but modulates the penalty via a contrast-sensitive term. Thus, if two adjacent superpixels are far apart in the feature space, there would be a smaller cost for assigning them different labels than if they were close. However, as various authors have noted, this contrast sensitive modulation forces the segmentation to follow strong edges in the image, which might not necessarily correspond to object boundaries. For example, Cui et al. (2008) modulate the distance d_{ij} based on statistics of edge profile features learnt from a fully segmented training image.

In this work, we use a distance metric learning algorithm to *learn* these d_{ij} from user scribbles. The basic intuition is that when two features (which might be far apart in Euclidean distance) are both labelled as the same class

[1] Specifically, we require at least one labelled foreground and background site to train our models, but only one per *group*, not per image.

[2] We use mean-shift (Comaniciu and Meer 2002) to extract these superpixels, and typically break down 350×500 images into 400 superpixels per image.

by the user scribbles, we want the distance between them to be low. Similarly, when two features are labelled as different classes, we want the distance between them to be large, even if they happen to be close by in Euclidean space. Thus, this new distance metric captures the pairwise statistics of the data better than Euclidean distance. For example, if colours blue and white were both scribbled as foreground, then the new distance metric would learn a small distance between them, and thus, a blue-white edge in the image would be heavily penalized for label discontinuity, while the standard contrast sensitive model would not penalize this edge as much. The specific choice of this algorithm is not important, and any state-of-art technique may be used. We use the implementation of Batra et al. (2008).

We update both $A_1 = \{\text{GMM}_f, \text{GMM}_b\}$ and $A_2 = \{d_{ij}\}$ every time the user provides a new scribble. Finally, we note that contrast-sensitive Potts model leads to a submodular energy function. We use graph-cuts to efficiently compute the MAP labels for all images, using the publicly available implementations of Bagon (2006) and Boykov and Kolmogorov (2004); Boykov et al. (2001); Kolmogorov and Zabih (2004).

Comparing Energy Functions Our introduced energy functions (1) are different from those typically found in co-segmentation literature and we make the following observations. While previous works (Rother et al. 2006; Mu and Zhou 2007; Mukherjee et al. 2009; Hochbaum and Singh 2009) have formulated co-segmentation of image pairs with a single energy function, we assign to each image its own energy function. The reason we are able to do this is because we model the dependence between images implicitly via the common appearance model ($\mathcal{A}$), while previous works added an explicit histogram matching term to the common energy function. There are two distinct advantages of our approach. First, as several authors (Rother et al. 2006; Mu and Zhou 2007; Mukherjee et al. 2009; Hochbaum and Singh 2009) have pointed out, adding an explicit histogram matching term makes the energy function intractable. On the other hand, each one of our energy functions is submodular and can be solved with a single graph-cut. Second, this common energy function grows at least quadratically with the number of images in the group, making these approaches almost impossible to scale to dozens of images in a group. On the other hand, given the appearance models, our collection of energy functions are completely independent. Thus the size of our problem only grows linearly in the number of images in the group, which is critical for interactive applications. In fact, each one of our energy functions may be optimized in parallel, making our approach amenable to distributed systems and multi-core architectures. Videos embedded on our project website (Batra et al. 2009) show our (single-core) implementation co-segmenting ∼20 image in a matter of seconds.

To be fair, we should note that what allows us to set-up an efficiently solvable energy function is our incorporation of a user in the co-segmentation process, giving us partially labelled data (scribbles). While this user involvement is necessary because we work with globally related images, this involvement also means that the co-segmentation algorithm must be able to query/guide user scribbles, because users cannot be expected to examine all cutouts at each iteration. This is described next.

4 iCoseg: Guiding User Scribbles

In this section, we develop an intelligent recommendation algorithm to automatically seek user-scribbles and reduce the user effort. Given a set of initial scribbles from the user, we compute a recommendation map for each image in the group. The image (and region) with the highest recommendation score is presented to the user to receive more scribbles. Instead of committing to a single confusion measure as our recommendation score, which might be noisy, we use a number of "cues". These cues are then combined to form a final recommendation map, as seen in Fig. 3. The three categories of cues we use, and our approach to learning the weights of the combination are described next.

4.1 Uncertainty-Based Cues

Node Uncertainty (*NU*) Our first cue is the one most commonly used in uncertainty sampling, i.e., entropy of the node beliefs. Recall that each time scribbles are received, we fit $A_1 = \{\text{GMM}_f, \text{GMM}_b\}$ to the labelled superpixel features. Using this learnt A_1, for each superpixel we normalize the foreground and background likelihoods to get a 2-class distribution and then compute the entropy of this distribution. The intuition behind this cue is that the more uniform the class distribution for a site, the more we would like to observe its label.

Edge Uncertainty (*EU*) The Query by Committee (Seung et al. 1992) algorithm is a fundamental work that forms the basis for many selective sampling works. The simple but elegant idea is to feed unlabelled data-points to a committee/set of classifiers and request label for the data-point with maximal disagreement among classifier outcomes. We use this intuition to define our next cue. For each superpixel, we use our learnt distances (recall: these are used to define the edge smoothness terms in our energy function) to find K (=10) nearest neighbours from the labelled superpixels. We treat the proportion of each class in the returned list as the probability of assigning that class to this site, and use the entropy of this distribution as our cue. The intuition behind this cue is that the more uniform this distribution, the more disagreement there is among the returned neighbour labels, and the more we would like to observe the label of this site.

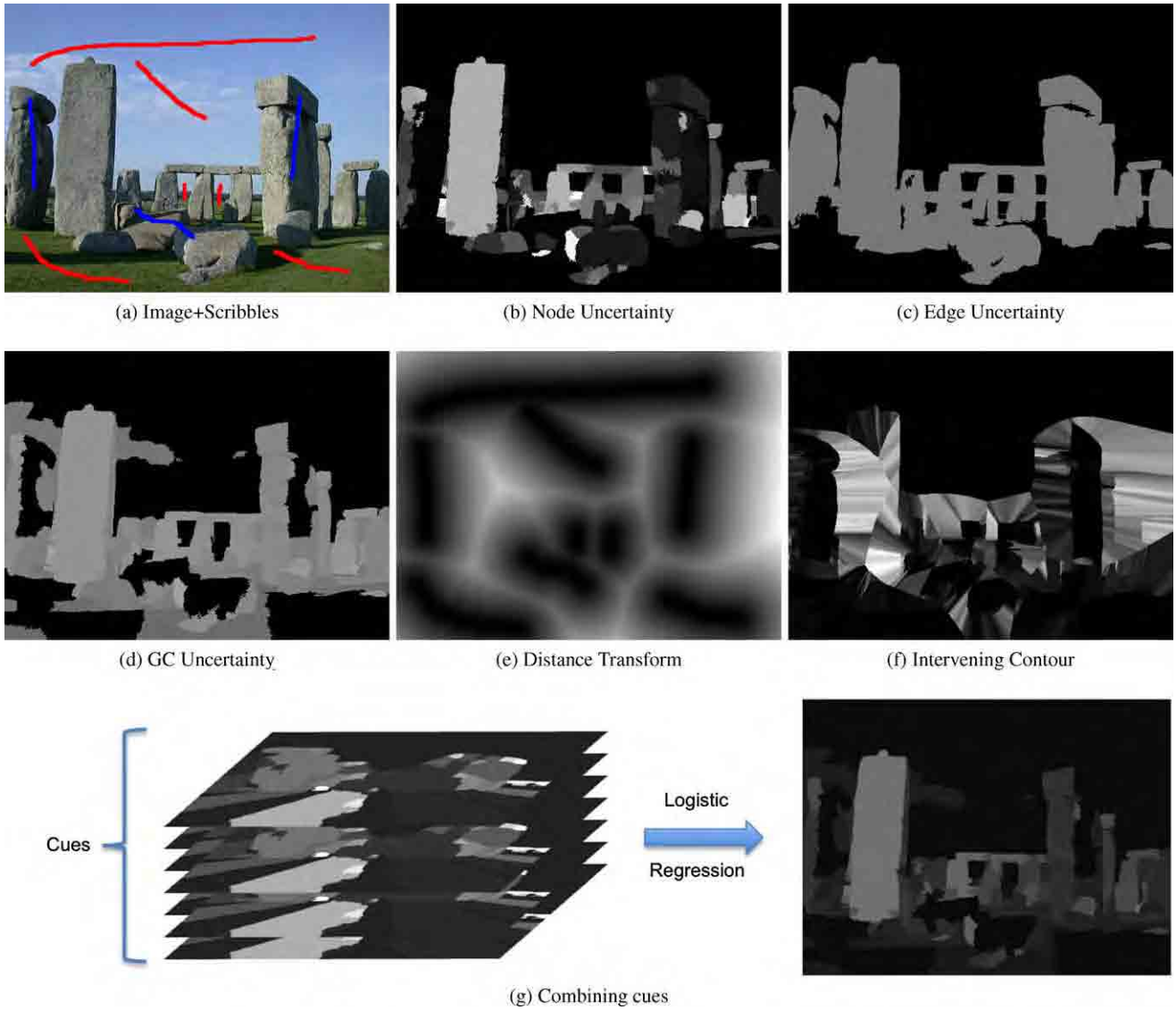


Fig. 3 Cues: (**a**) shows the image with provided scribbles; (**b**)–(**f**) show various cues; and (**g**) shows how these cues are combined to produce a final recommendation map

Graph-Cut Uncertainty (*GC*) This cue tries to capture the confidence in the energy minimizing state returned by graph-cuts. For each site, we compute the increase in energy by flipping the optimal assignment at that site. The intuition behind this cue is that the smaller the energy difference by flipping the optimal assignment at a site, the more uncertain the system is of its label. We note that min-marginals proposed by Kohli and Torr (2008) could also be used.

4.2 Scribble-Based Cues

Distance Transform over Scribbles (*DT*) For this cue, we compute the distance of every pixel to the nearest scribble location. The intuition behind this (weak) cue is that we would like to explore regions in the image away from the current scribble because they hold potentially different features than sites closer to the current scribbles.

Intervening Contours over Scribbles (*IC*) This cue uses the idea of intervening contours (Leung and Malik 1998). The value of this cue at each pixel is the maximum edge magnitude in the straight line to the closest scribble. This results in low confusions as we move away from a scribble until a strong edge is observed, and then higher confusions on the other side of the edge. The motivation behind this cue is that edges in images typically denote contrast change, and by observing scribble labels on both sides of an edge, we can learn whether or not to respect such edges for future segmentations.

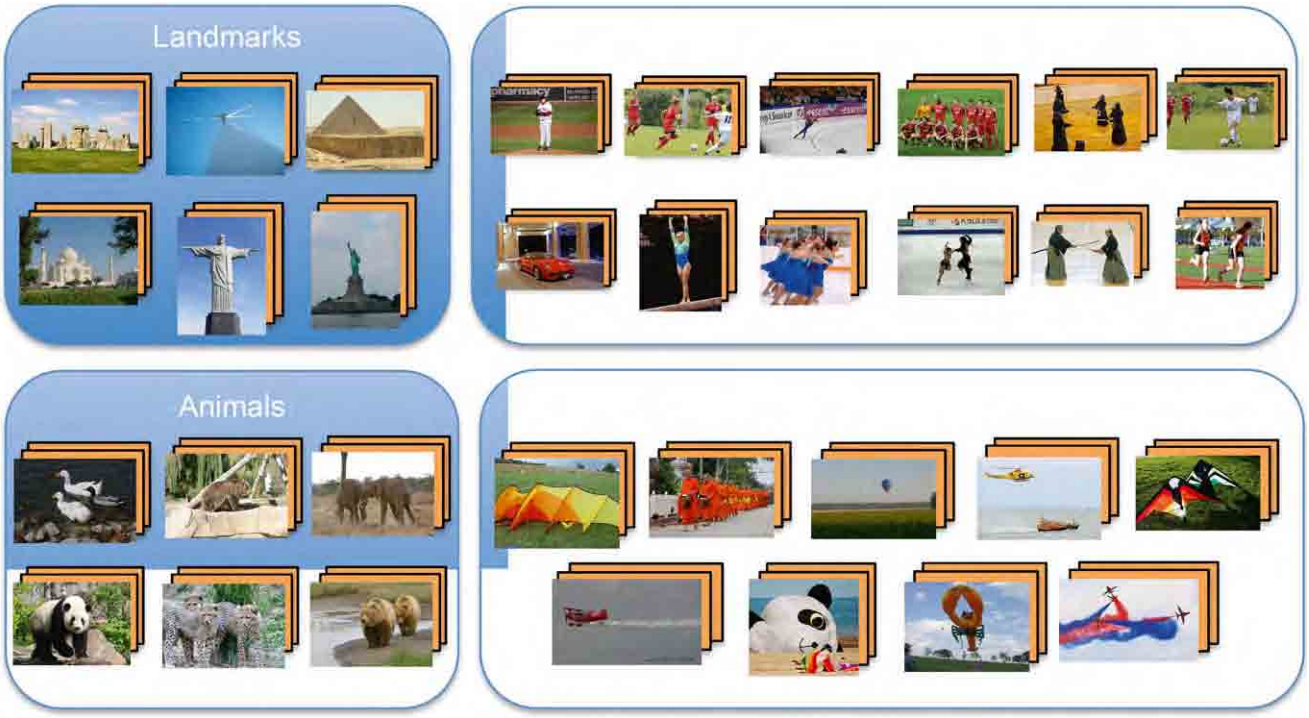


Fig. 4 CMU-Cornell iCoseg Dataset: figure shows prototypical images from the co-segmentation groups in our dataset. Note: *each image* shown above corresponds to a *group* of images

4.3 Image-Level Cues

The cues described so far, are local cues, that describe which region in an image should be scribbled on next. In addition to these, we also use some image-level cues (i.e., uniform over an image), that help predict *which* image to scribble next, not where.

Segment Size (*SS*) We observe that when very few scribbles are marked, energy minimization methods typically over-smooth and results in "whitewash" segmentations (entire image labelled as foreground or background). This cue incorporates a prior for balanced segmentations by assigning higher confusion scores to images with more skewed segmentations. We normalize the size of foreground and background regions to get class distributions for this image, and use the inverse of the entropy of this distribution as our cue.

Codeword Distribution over Images (*CD*) This image-level cue captures how diverse an image is, with the motivation being that scribbling on images containing more diversity among features would lead to better foreground/background models. To compute this cue, we cluster the features computed from all superpixels in the group to form a codebook, and the confusion score for each image is the entropy of the distribution over the codewords observed in the image. The intuition is that the more uniform the codeword distribution for an image the more diverse the appearances of different regions in the image.

4.4 Combined Recommendation Map

We now describe how we combine these various cues to produce a combined confusion map. Intuitively, the optimal combination scheme would be one that generates a recommendation map that assigns high values to regions that a user *would* scribble on, if they were to exhaustively examine all segmentations. Users typically scribble on regions that are incorrectly segmented. We cast the problem of learning the optimal set of weights for our cues as that of learning a mapping $\mathcal{F}:\phi_i \rightarrow \epsilon_i$, where ϕ_i is the 7-dimensional feature vector for superpixel i, corresponding to each of the 7 cues described above, and ϵ_i is the error indicator vector, which is 1 if the predicted segmentation at node i is incorrect, and 0 otherwise. We chose logistic regression as the form of this mapping. The ground-truth for training this logistic regression was generated by first scribbling on images,[3] co-segmenting based on these scribbles, and then using the mistakes (or the error-map) in these segmentations as the ground-truth. Our cue combination scheme is illustrated in Fig. 3.

[3] More precisely, by generating random automatic scribbles on images. See Sect. 6.1 for details.

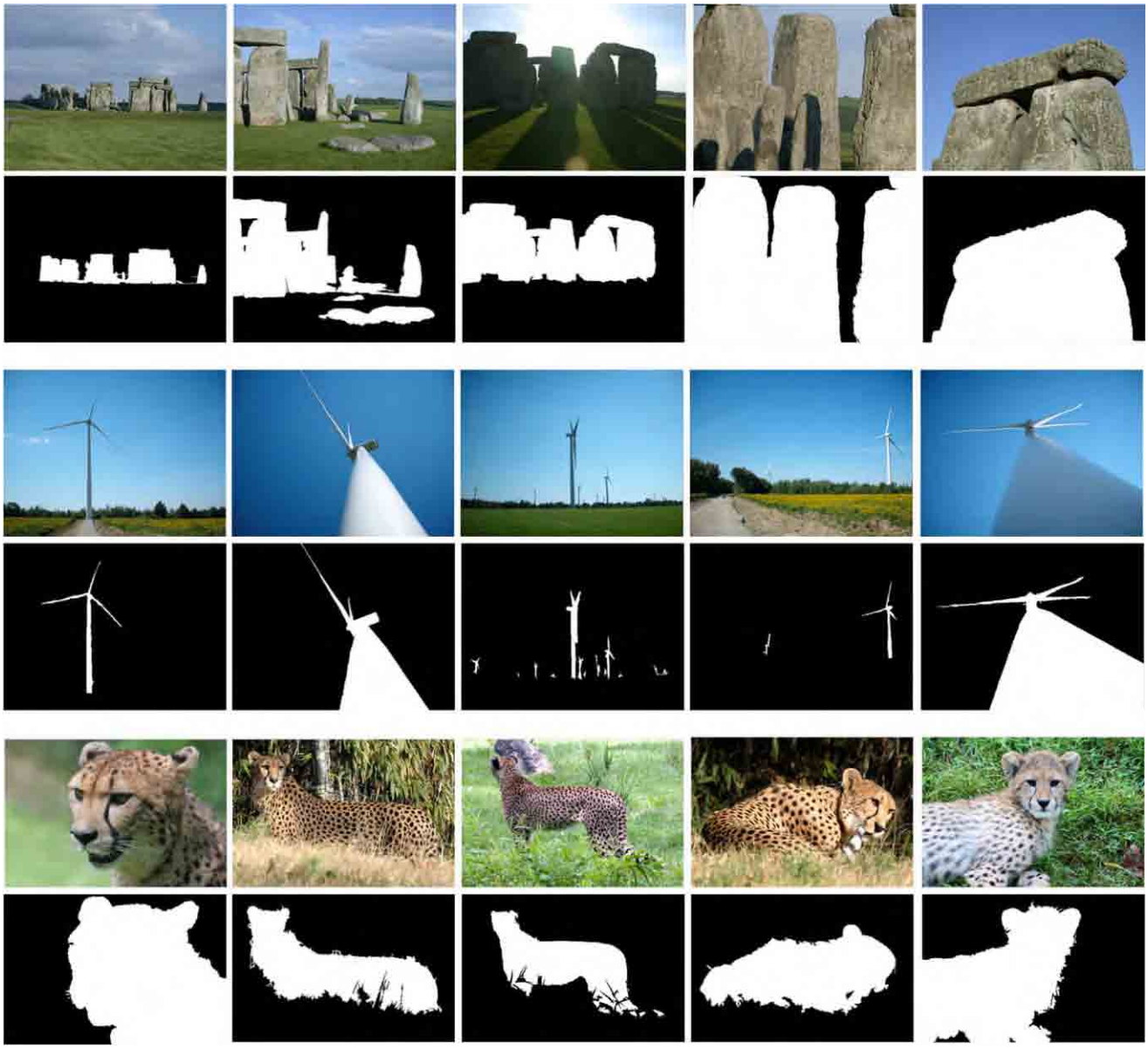


Fig. 5 CMU-Cornell iCoseg Dataset: figure shows the Stonehenge, windmill and cheetah groups. We can see the large variation in illumination, scale, pose and appearance. Masks show the detail in our hand-annotated groundtruth

5 The CMU-Cornell iCoseg Dataset

To evaluate our proposed approach and to establish a benchmark for future work, we introduce the largest co-segmentation dataset yet, the CMU-Cornell iCoseg Dataset. While previous works have experimented with a few pairs of images, our dataset contains 38 challenging groups with 643 total images (∼17 images per group), with associated pixel-level ground truth. We built this dataset from the Flickr® online photo collection, and hand-labelled pixel-level segmentations in all images. We used the "Group" feature in Flickr, where users form groups around popular themes, to search for images from this theme. Our dataset consists of animals in the wild (elephants, pandas, etc.), popular landmarks (Taj Mahal, Stonehenge, etc.), sports teams (Baseball, Football, etc.) and other groups that contain a common theme or common foreground object. For some (though not all) of the groups, we restricted the images to come from the same photographer's photo-stream, making this a more realistic scenario. Examples of these groups are shown in various figures in this paper, and Figs. 4 and 5 show some prototypical images. We have made this dataset (and annotations) publicly available (Batra et al. 2009) to facilitate further work, and allow for easy comparisons.

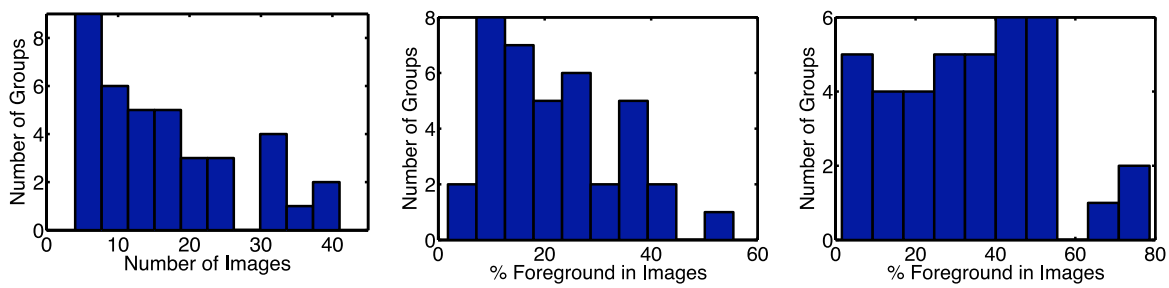
Dataset Annotation The ground-truth annotations for the dataset were manually generated by a single annotator using a labelling tool. The ground-truth was labelled on su-

(a) Girl-pair (Rother et al, 2006). (b) Stonehenge-pair from CMU-Cornell iCoseg Dataset.

	KL-Divergence		
	Foreground	Background	Ratio
Girl-pair (Rother et al, 2006)	1.06	45.78	43.03
Stonehenge-pair	8.49	17.66	2.08

(c) Foreground and background similarity statistics

Fig. 6 Appearance statistics: (**a**) shows a pair of images ("girl-pair") from (Rother et al. 2006); (**b**) shows a pair of images from the Stonehenge group in our CMU-Cornell iCoseg Dataset; (**c**) lists the KL-divergences between the two images for each pair. Images in our dataset are globally consistent, with comparable KL-divergence between foregrounds and backgrounds



(a) Group Sizes. (b) Hist. avg. foreground size (c) largest - smallest foreground

Fig. 7 Dataset statistics: (**a**) shows the histogram of the number of images in groups; (**b**) shows histogram of avg. foreground size in groups; (**c**) shows histogram of difference of largest and smallest foreground object within a group

Table 1 Dataset statistics

	# Groups	# Images	# Images/Group
Rother et al. (2006)	7	16	2.29
Hochbaum and Singh (2009)	23	46	2
CMU-Cornell iCoseg Dataset	38	643	16.92

perpixels. However, the labelling tool allowed the annotator to interactively obtain finer/coarser superpixels as desired. A useful strategy used by our annotator was to use coarse superpixels while labelling simple scenes, and to use finer superpixels when labelling complicated or cluttered scenes. This allowed for very high quality ground truth, without the labeling task being prohibitively tedious.[4]

[4]Since the superpixels were varied dynamically for each image, they were not the same as the ones used inside our co-segmentation algorithm (which only used a single setting of parameters for generating superpixels in all images).

5.1 Dataset Statistics

We now analyze some statistics (size, appearance variation, scale variation) of our introduced dataset.

Size Table 1 lists the number of groups, number of images and average number of images per group for our dataset. We note that this dataset is significantly larger than those used in previous works (Hochbaum and Singh 2009; Rother et al. 2006). Figure 7a shows a histogram of number of images in groups.

Appearance Recall that in Sect. 1 we argued that when consumers take multiple photographs of the same event or object, the images are usually globally consistent. To quantify that images in our dataset do capture this property, we perform the following experiment. Figure 6 shows a pair of images ("girl-pair") from Rother et al. (2006), and another pair from our CMU-Cornell iCoseg Dataset ("Stonehenge-pair"). All the foreground pixels from the first pair of images were clustered into 64 color codewords using k-means

(a) Largest scale-change (b) Smallest scale-change

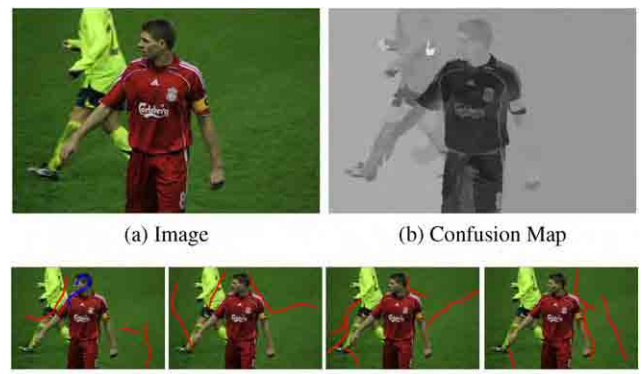
Fig. 8 Scale change: (**a**) shows the largest scale change, and (**b**) shows the smallest scale change in our dataset

clustering. A foreground histogram was built over this dictionary for each of the images, and the KL distance was computed between these normalized histograms. Similarly a color dictionary was built using the background pixels and the KL distance between the background distances was computed. This process was repeated for the second pair. We can see that for the "girl-pair" the appearance variation in foreground is considerably smaller than the variation in background. The "Stonehenge-pair", on the other hand, shows a more comparable variation. This is not to say that the CMU-Cornell iCoseg Dataset is inherently harder than previous datasets. Our intent is to point out that our dataset contains images where the previous works on co-segmentation would fail, because they have been designed for a different scenario. We hope the introduction of our dataset motivates further research in the problem we consider in this paper.

Scale To quantify the amount of scale change in our dataset, we show the histogram of average foreground size in groups in Fig. 7b. We can see that some groups contain very small foreground objects (on avg. $\leq$5% of the image) while some groups contain very large foreground objects (on avg $\geq$40% of image). In addition, the histogram (Fig. 7c) of difference between largest and smallest foreground object in a group shows that even within a group there is significant scale change. Figure 8 shows images with the largest and smallest scale change in our dataset. We can imagine that the large scale changes can be quite challenging for co-segmentation algorithms. In the case of our algorithm however, scale changes should not matter too much.

6 Experiments

For experimental evaluation, we performed machine experiments (Sect. 6.1) by generating synthetic scribbles, and also performed user-study (Sect. 6.2). In all experiments in this paper, we quantify the accuracy of an image segmentation as the percentage of pixels whose labels are correctly predicted. Co-segmentation accuracy for a group is the average segmentation accuracy over all images in this group.



(a) Image (b) Confusion Map

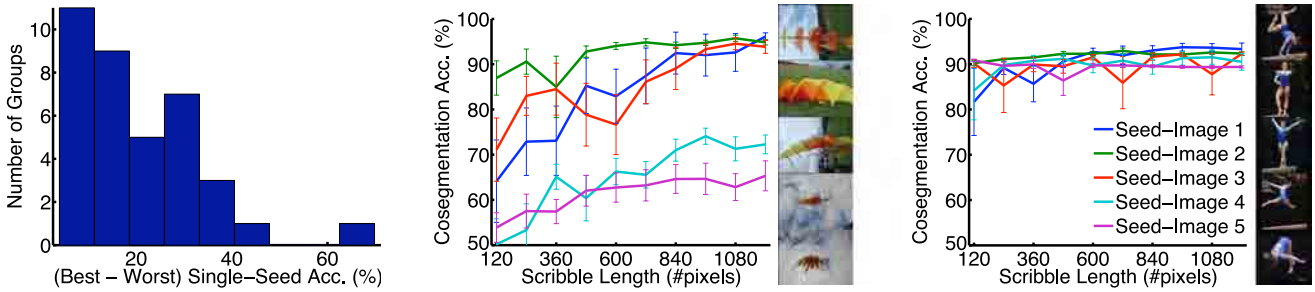
(c) Sampled Scribbles

Fig. 9 Example simulated scribbles: note that these scribbles never cross foreground boundaries

6.1 Machine Experiments

To conduct a thorough set of experiments and evaluate various design choices, it is important to be able to perform multiple iterations without explicitly polling a human for scribbles. Thus, we develop a mechanism to generate automatic scribbles, that mimic human scribbles. We model the scribbles as (smooth) random walks that do not cross foreground-background boundaries. Our scribble generation technique consists of sampling a starting point in the image uniformly at random. A direction angle is then randomly sampled such that it is highly correlated with the previous direction sample (for smoothness) for the scribble,[5] and a fixed-size (=30 pixels) step is taken along this direction to extend the scribble (as long as it does not cross object boundaries, as indicated by the groundtruth segmentation of the image). To mimic user-scribbles given a recommendation map, the initial as well as subsequent points on the scribble are picked by considering the recommendation map to be a distribution. Using synthetic scribbles allows us to control the length of scribbles and observe the behavior of the algorithm with increasing information. Example synthetic scribbles are shown in Fig. 9. For all experiments in

[5]For the first two sampled points, there is no previous direction and this direction is sampled uniformly at random.



(a) All groups (b) Group with a large diff. (c) Group with a small diff.

Fig. 10 Diversity within a group: (**a**) histogram of the difference in accuracy between the best and worst seed-images for all the groups in our dataset. A large difference indicates diversity in appearance. Some groups such as (**b**) "Kite" have images with varied appearances (two images providing worst segmentation accuracies don't contain any grass), while other groups such as (**c**) "Gymnast" are more homogenous

this paper, the length of each individual scribble was capped at 120 pixels.

6.1.1 Baseline 1: Scribbles Restricted to a Single Image

To establish the simplest baseline, we ask the following question: "how well would interactive co-segmentation work if we were restricted to scribbling on a single image?" If a group consisted of successive frames from a video sequence, the choice of this chosen image (seed-image) would not matter much. The higher the diversity in the images among a group, the more variation we would observe in the group segmentation accuracies achieved by various seed-images, because not all seed-images would provide useful statistics for the group as a whole. We use the synthetically generated scribbles (described above) to test this.

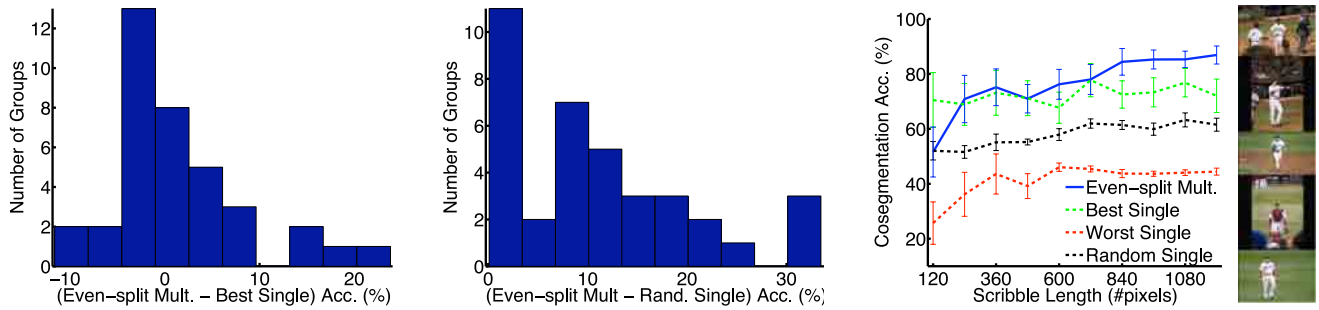
In Fig. 10a we show the histogram of the difference in accuracy between the best and worst seed-images for all the groups in our dataset.[6] We can see that the histogram has a heavy tail, with 28 (of 37) groups having greater than 10% difference (and one as high as 69%), indicating that most groups have a lot of variation. Figures 10b and 10d show the co-segmentation accuracies (Y-axis) for the "kite" and "gymnast" group respectively, as a function of the total length of scribbles (X-axis) on different seed-images in these groups (shown next to them). Images are ordered in decreasing accuracy. The "Kite" groups is an example of the groups with a lot of variation while the images in the "Gymnast" group have similar performances. Notice that in the "Kite" group, the two "bad" images do not contain any grass, and thus irrespective of the length of the scribbles on them, the algorithm will not know whether grass is considered foreground or background, and thus the group co-segmentation accuracy does not rise above 75%.

[6]In order to keep statistics comparable across groups, we select a random subset of 5 images from all groups in our dataset. One of our groups consisted of 4 images only, so all our results are reported on 37 groups.

6.1.2 Baseline 2: Uniform Recommendation Maps

In the previous baseline we were restricted to scribbling on a single image. As we saw, the performance between the best and the worst seed-images could be significantly different. However, a priori we have no way of knowing the best image to scribble on. In this section, we consider user scribbles to be a limited resource and evaluate whether it is better to seek sparse scribbles on multiple images or dense scribbles on a single image. We follow a similar setup as in the last section, only now the scribbles are evenly split across all images in the group, which corresponds to uniform recommendation maps on all images. This way we can compare 1200-pixel scribbles (which would be dense) in a single image with five 240-pixel scribbles (which would be sparse).[7] In practice, instead of making one long 1200-pixel scribble, we sample scribbles of length at most 120 pixels, and evenly split scribbles between foreground and background. As before, we perform 10 random runs. Figure 11c shows the average co-segmentation accuracies in the group (Y-axis) for the worst (single) seed-image, the best (single) seed-image, a random (single) seed-image, and the accuracy achieved by evenly splitting scribbles across all images (called even-split) as a function of the total length of scribbles (X-axis). We can see that for the same length of scribbles, evenly splitting them across all images in the group and getting sparse scribbles performs better than dense scribbles on any image in this group. Figures 11a and 11b show the histogram of accuracy gains of even-split over the best (single) seed-image and the random (single) seed-image experiments over all of the groups. The accuracies for Figs. 11a and 11b were computed using scribbles of total length of 1200 pixels, i.e., they correspond to the rightmost datapoint in Fig. 11c. We can

[7]This is one of the reasons for keeping a constant number of images per group. If each group had different images, even-split performance would no longer be comparable across groups.



(a) All groups (b) All groups (c) Example group

Fig. 11 Sparse vs. dense scribbles: (**a**) and (**b**) show the histograms of accuracy gains of even-split over the best (single) seed-image and a random (single) seed-image, respectively; (**c**) compares the accuracies achieved by dense scribbles on a single image, and sparse scribbles on multiple images for the group shown. Again, images are ordered in decreasing accuracy. The second histogram (**b**) shows providing scribbles on multiple images is a better strategy than committing to a single one

see while even-split performs better than the best (single) seed-image for most groups, it is strictly better than a random (single) seed-image for *all* of the groups.

6.1.3 iCoseg

We first analyze the informativeness of each of our 7 cues. We start by generating a foreground and background scribble each of length at most 120 pixels on a random image in a group. We then compute each of our cues, and treat each individual cue as a recommendation map. We generate the next synthetic scribble (again of at most 120 pixels) as guided by this recommendation map, meaning that points are sampled by treating this recommendation map as a probability distribution (instead of sampling them randomly). We repeat this till we have scribbled about 1200 pixels across the group, and compute the average segmentation accuracy across the images of a group. We rank the 7 cues by this accuracy. Figure 12 shows the mean ranks (across groups, average of 10 random runs) achieved by these cues. Out of our cues, the graph-cut cue (GC) performs the best, while both distance transform (DT) and intervening contour (IC) are the weakest. GC cue quantifies the uncertainty of the entire model (including node and edge potentials) and thus is expected to provide the best indication of where more information is required (from an active learning perspective). Thus, it is not surprising that this cue performs the best. DT and IC on the other hand completely ignore the learnt model, and only consider low-level cues like where (in x, y co-ordinates) we have scribbled in the image so far and the gradients in the image which often do not coincide with object boundaries. Thus, it is not surprising that they provide the least information to recommend meaningful regions to scribble further on.

We now evaluate iCoseg, our recommendation system, as a whole. The experimental set up is the same as that

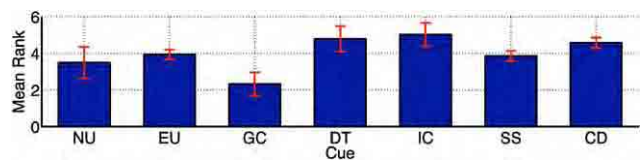


Fig. 12 Mean ranks achieved by individual cues (see Sect. 6.1)

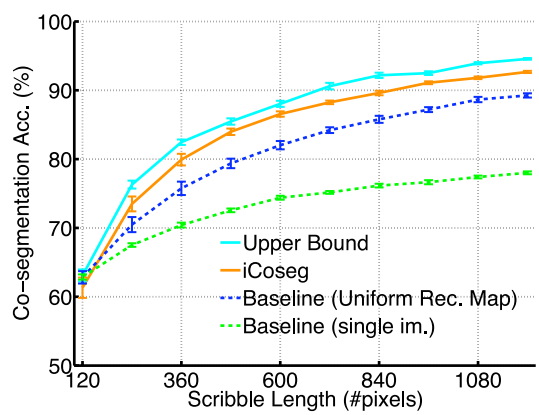


Fig. 13 Machine experiments: figure shows average co-segmentation accuracy as a function of the number of scribbles (each scribble is 120 pixels). iCoseg significantly outperforms baselines and is close to a natural upper-bound (see Sect. 6.1 for details)

described above, except now we use the combined recommendation map to guide subsequent scribbles (and not individual cues). The cue combination weights are learnt from all groups except one that we test on (leave-one-out cross validation). We compare to two baselines described above. One is that of using a uniform recommendation map on all images in the group, which essentially means randomly scribbling on the images (respecting object boundaries of course). And the other (even weaker) baseline is that of selecting only one image (randomly) in a group to scribble on (with a uniform recommendation map on this image).

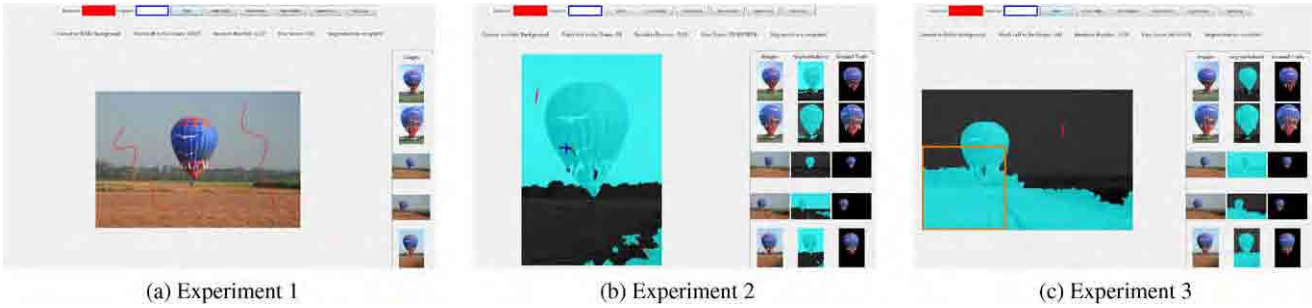


Fig. 14 User study screenshots: (**a**) Exp. 1: subjects were not shown cutouts and were free to scribble on any image/region while respecting the foreground/background boundaries; (**b**) Exp. 2: subjects exhaustively examine all segmentations and scribble on mistakes (*highlighted regions* indicate foreground); (**c**) Exp. 3: users were instructed to scribble in the region recommended by iCoseg

Figure 13 shows the performance of our combined recommendation map (iCoseg) with increasing scribble length, as compared to the baselines. We see that our proposed recommendation scheme does in fact provide meaningful guidance for regions to be scribbled on next (as compared to the two baselines). A meaningful upper-bound would be the segmentation accuracy that could be achieved if an oracle told us where the segmentations were incorrect, and subsequent scribbles were provided only in these erroneous regions. As seen in Fig. 13, iCoseg performs very close to this upper bound, which means that users following our recommendations can achieve cutout performances comparable to those achieved by analyzing mistakes in all cutouts with significantly less effort *without* ever having to examine all cutouts explicitly.

6.2 User Study

In order to further test iCoseg, we developed a java-based user-interface for interactive co-segmentation.[8] We conducted a user study to verify our hypothesis that our proposed approach can help *real* users produce good quality cutouts from a group of images, without needing to exhaustively examine mistakes in all images at each iteration. Our study involved 15 participants performing 3 experiments (each involving 5 groups of 5 related images). Figure 14 shows screen-shots from the three experiments. The subjects were informed that the first experiment was to acclimatize them to the system. They could scribble anywhere on any image, as long as they used blue scribbles on foreground and red scribbles on background. The system computed cutouts based on their scribbles, but the subjects were never shown these cutouts. We call this experiment "freeform-scribbling". In the second experiment, the subjects were shown the cutouts produced on all images in the group from their scribbles. Their goal was to achieve 95% co-segmentation accuracy in as few interactions as possible, and they could scribble on any image. We observed that a typical strategy used by subjects was to find the worst cutout at every iteration, and then add scribbles to correct it. In the third experiment, they had the same goal, but this time, while they were shown all cutouts, they were constrained to scribble within a window recommended by our algorithm, iCoseg. This window position was chosen by finding the location with the highest average recommendation value (in the combined recommendation map) in a neighbourhood of 201×201 pixels. The use of a window was merely to make the user-interface intuitive, and other choices could be explored. In all three experiments, users were restricted to use only 120 pixels of scribbles per iteration. Our UI displayed a counter that showed how many pixels they had left. Once their quota of pixels was over, they had no choice but to ask the system to co-segment using these scribbles, after which they were given a new quota of 120 pixels to scribble with. They did not have to use the entire quota before co-segmenting.

Figure 15 shows the average segmentation accuracies achieved by the subjects in the three experiments (Y-axis) as a function of the length of their scribbles (X-axis). We can see that, as with the machine experiments, iCoseg helps the users perform better than freeform scribbling, in that the same segmentation accuracy (83%) can be achieved with about 75% the effort. In addition, the average time taken by the users for one iteration of scribbling reduced from 20.2 seconds (exhaustively examining all cutouts) to 14.2 seconds (iCoseg), an average saving of 60 seconds per group. Thus, our approach enables users to achieve cutout accuracies comparable to those achieved by analyzing mistakes in all cutouts, in significantly less time. This fact is further shown in Fig. 16 where the co-segmentation accuracy achieved (Y-axis) is plotted as a function of time taken (X-axis) for each of the three expirments, averaged across users and groups. We can see that our approach allows users to reach highest accuracies given the same time budget.

[8] We believe this interface may be useful to other researchers working on interactive applications and we have made it publicly available (Batra et al. 2009).

Table 2 Comparison between user study and machine experiments

Machine experiments	Mean number of fg./bg. scribbles per iteration	Mean length of scribbles (pixels)	Mean %-age of fg. in scribbles	Amount of uncertainty under the scribbles
	1.00/1.00	120	23%	–
User studies				
Exp. 1	1.04/1.05	106	46%	0.49 ± 0.04
Exp. 2	1.07/1.07	110	–	0.49 ± 0.04
Exp. 3	1.05/1.06	99	–	0.53 ± 0.04

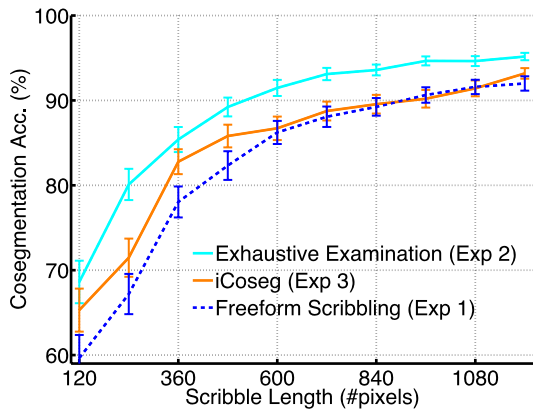


Fig. 15 User study: average performance of subjects in the three conducted experiments (see Sect. 6.2). iCoseg (Exp. 3) requires significantly less effort for users, and allows them to reach 80% co-seg accuracy with 75% the effort of Exp. 1

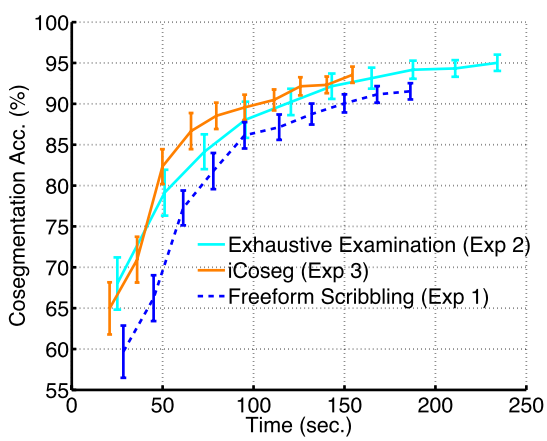


Fig. 16 User study: average performance of subjects in the three conducted experiments as a function of time (see Sect. 6.2). We can see that for a fixed amount of time, iCoseg (Exp. 3) typically achieves highest co-segmentation accuracy

6.3 Comparing Machine Experiments and User Study

In order to understand how users scribbled in our user-study and study how well our automatic scribbles (machine experiments) emulate this, we analyze similarities between the user and synthetic scribbles. Table 2 compares some statistics.

Our automatic scribbles were generated for foreground and background with a fixed length of 120 pixels, and we see that they are comparable to the user scribbles in both the length and the average number of scribbles.

Interestingly, we also found that while our subjects were not given an explicit goal in Exp. 1 (freeform scribbling experiment) and were not shown the groundtruth, they were implicitly aware of the common foreground and their scribbles reflected that knowledge. The proportion of foreground pixels in all scribbles given by our subjects (for Exp. 1) was 46%, while the groups they viewed only contained 23% foreground. Clearly, they weren't scribbling uniformly randomly over an image, but were dividing their scribbles somewhat evenly over the foreground and background. Thus, even though Fig. 15 seems to suggest that iCoseg has fallen to the performance level of the freeform-scribble baseline (Exp. 1), in reality, "freeform scribbling" has become a smart human-attention based algorithm. The truly random baseline can be seen in Fig. 13 where we force the random scribbles to be truly uniformly random, which our algorithm easily outperforms.

We also measured whether users were scribbling on confusing areas, as measured by our combined uncertainty map (which is normalized to be a spatial probability distribution, and thus between 0 and 1). We notice that the amount of uncertainty under user-scribbles is 0.49 ± 0.04 for both Exp. 1 (freeform scribbling) and Exp. 2 (exhaustive examination), again indicating that the users were implicitly aware about the common foreground and scribbled over incorrect segmentations which are typically regions with high uncertainty. We note that the uncertainty under user scribbles increased to 0.53 ± 0.04 for Exp. 3 (iCoseg), which is understandable, because the users were guided to scribble within the indicated regions of high uncertainty.

6.4 Limitations and Failure Cases

An assumption of our approach is that the foreground and background models are different enough in the chosen feature space (i.e. colour for our experiments) to allow for reliable labelling of both classes. The interactive nature of our

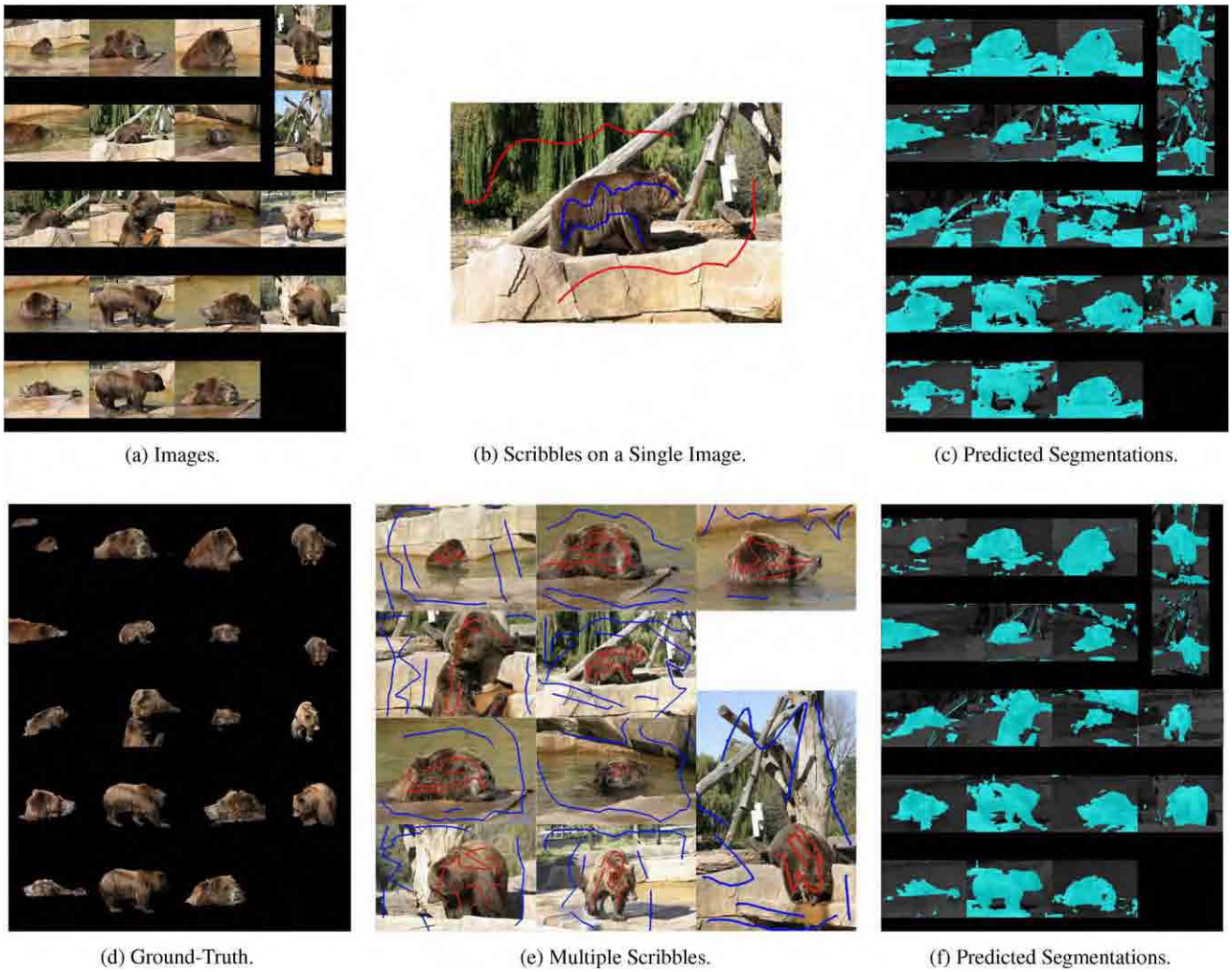


Fig. 17 Common failure case: (**a**) shows the group of images; (**c**) shows the segmentations achieved by scribbling on a single image, shown in (**b**). *Highlighted regions* indicate foreground. (**f**) Shows the segmentations achieved by scribbles on multiple images, shown in (**e**). When foreground and background have a lot of overlap in colour distributions, our interactive segmentation method faces difficulty in producing accurate segmentations (compare segmentations in (**c**), (**f**) with ground-truth in (**d**)). However, our algorithms allows for straightforward incorporation of more sophisticated features (e.g. colour-pallet of Cui et al. 2008), which should result in better performance

system makes the choice of features and appearance models seem less critical. However, they play an important role, and it is important to analyze the limitations and failure cases of our approach.

Non-discriminative Features The most common failure case for our method results from the choice of colour features. Figure 17 shows a difficult group to segment because the foreground colour distribution is very similar to the background colour distribution. Thus, even though the scribble guidance leads users to useful locations, the co-segmentation quality does not significantly improve despite multiple rounds of scribbles. Figures 17b and 17c show the co-segmentations after scribbling on a single image, and Figs. 17e and 17f show the co-segmentations after scribbling on multiple images. We note that the choice of colour features is not inherent to the system, and more sophisticated features can be seamlessly incorporated. One choice of better features would be color-pattern features of Cui et al. (2008) that capture the spatial distribution of colors in a neighborhood. These would provide more discriminative power (which should result in improved performance), as well as help overcome the local nature of features extracted at superpixels.

Superpixel Leaks We use superpixels as the labelling sites in our framework. This speeds up our implementation because the graph constructed on superpixels is significantly smaller than the grid-graph on pixels. However, because we use a single parameter setting to generate superpixels for all

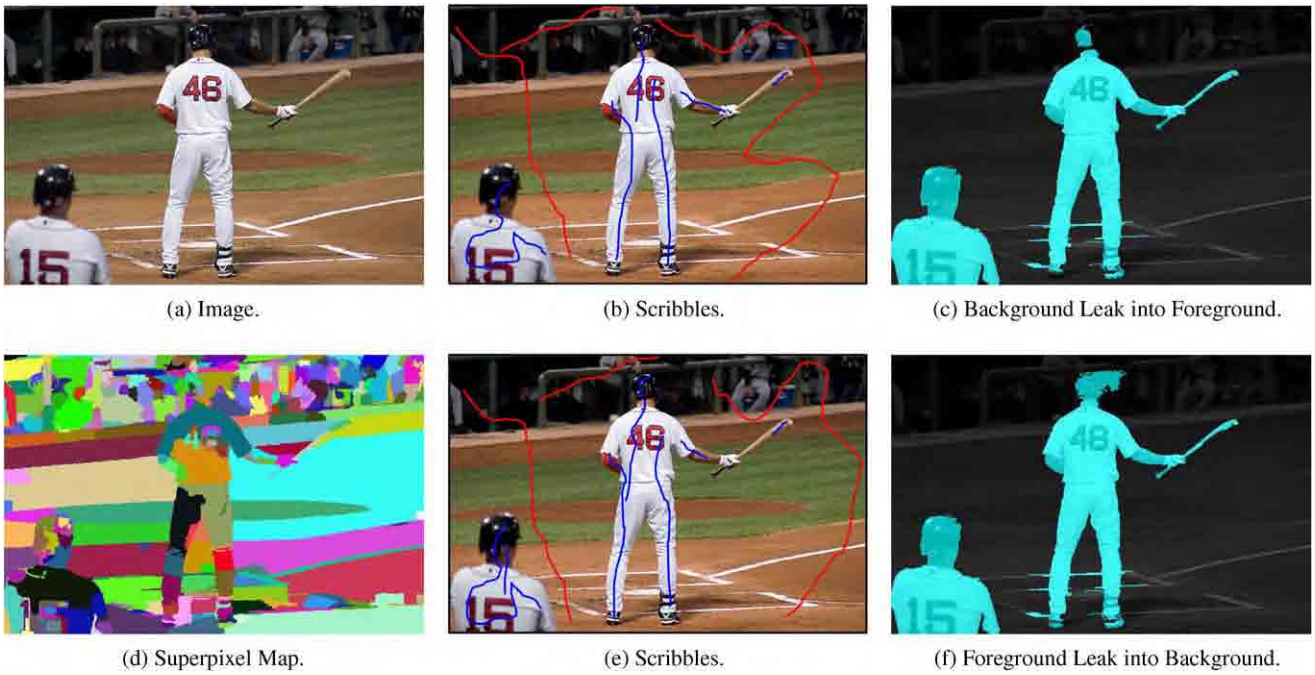


Fig. 18 Superpixel leaks: we use a single parameter setting to generate superpixels for all images in our dataset, and thus some images show superpixel leaks across foreground objects. For example, in (**d**), the superpixel on the head of the baseball player leaks into the background. As a result, the segmentation also tends to either leak from foreground to background (**f**), or from background to foreground (**c**), depending on the choice of scribbles, (**e**) and (**b**) respectively

images in our dataset, some images show superpixel leaks across foreground objects. Figure 18 shows an example image. Notice that some superpixels leak across object boundaries, e.g., the one on the head of the baseball player. As a result of this superpixel leak, the segmentation also tends to either leak from foreground to background or from background to foreground, depending on the choice of scribbles. Having said this, our approach can be trivially extended to work with pixels, for applications that require highly accurate segmentations.

Single Background Model It is conceivable that the use of multiple background models within a group could be beneficial. However, the more models we wish to build, the more scribbles we are likely to need from users for the models to be informative. In the extreme, in order to have one background model for every image in the group, we would need sufficient scribbles in all images in the group. This, to some extent, would defeat the purpose of having a co-segmentation system, where the goal is to leverage the fact that topically-related images share foreground and background statistics, and hence can be co-segmented, and need not be segmented individually. As seen in our examples, a large proportion of the images within a group do share similar backgrounds, a property that should be exploited when possible, but these properties are of course, application dependent.

In order to quantify the above intuition, we performed the following experiment. We performed co-segmentation with synthetic scribbles for the following three cases:

- Multiple Models, Independent Segmentation (MMIS). In this case, the synthetic user scribbles on each of the five images in a group, and each image is independently segmented. Intuitively, this is equivalent to running a standard Grab-cut-like method on each image in the group, thus forcing the user to scribble on all images. The appearance models are not shared and the user is forced to scribble on all images to be segmented.
- Multiple Models, Co-Segmentation (MMCS). In this case, the synthetic user again scribbles on a single image in the group, however now all images in the group are segmented by sharing the appearance model learnt from the single scribbled image. This is repeated by scribbling on all images in the group one at a time. As we have already observed in Sect. 6.1.2 and Fig. 11, we do not expect this combination to perform well.
- Single Model, Co-Segmentation (SMCS). This is the case described in our machine experiments (Sect. 6.1 and Fig. 13), where the synthetic user scribbles on all images in the group. All images in the group are segmented by sharing the appearance model learnt from all the scribbled images.

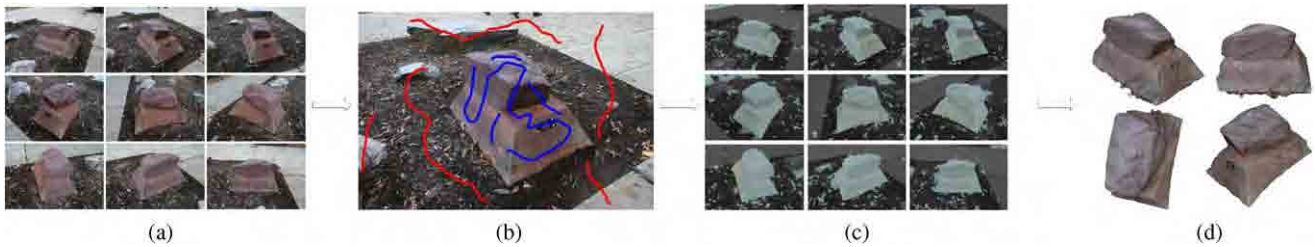


Fig. 19 (Color online) Interactive 3D modelling: (**a**) stone dataset (24 images)—subset of images given to the system shown; (**b**) user interactions to indicate the object of interest (*blue scribbles* = object of interest, *red scribbles* = background); (**c**) resulting silhouettes after co-segmentation (object of interest *highlighted*); (**d**) some sample novel views of the rendered 3D model

Table 3 Segmentation accuracies for various setups averaged across groups

	MMIS	MMCS	SMCS
Segmentation accuracy	97.07%	79.71%	92.67%

As we can see in Table 3, MMIS is the best thing to do, i.e. to scribble on every image and segment all images independently. However, this requires users to scribble on all images, which is not feasible for scenarios where the group contains many images. On the other hand, SMCS relieves the user from this constraint of scribbling on all images and as our machine experiments and user study show, the savings provided by iCoseg are crucial.

7 Interactive Co-segmentation for Object-of-Interest 3D Modeling

In this section, we present an application of iCoseg to interactively create 3D models of objects of interest from a collection of images of these objects in their natural potentially-cluttered environments. To give a concrete example of the application, consider a gamer who wants to "scan" his own skateboard and use it as a virtual skateboard in a video game. The computer-vision task then is to create a 3D model of this object of interest from a collection of images of this object.

One approach to achieve this, would be to haul an expensive laser scanner to get precise depth estimates in a controlled setup, and reconstruct the object (Levoy et al. 2000). However, this might be not be a feasible solution for average users. Another typical approach is to capture images of the object in a controlled environment like a multi-camera studio with mono-color screen (Franco and Boyer 2003; Starck and Hilton 2007; Vlasic et al. 2008; Curless and Levoy 1996; Chen and Medioni 1992; Fitzgibbon et al. 1998) or structured lighting (Zhang et al. 2002), and then use something like a shape-from-silhouette algorithm (Szeliski 1993; Fang et al. 2003; Chen et al. 2008; Forbes et al. 2006) to render the 3D model. Although these techniques have produced promising results in these constrained settings, this is a tedious process, and in some cases not an option (for example, immovable objects like a statue, historically or culturally-significant artifacts). Moreover, general scene reconstruction algorithms are not designed to focus on the object of interest to the user.

In our approach, we begin by first interactively co-segmenting the object of interest in the group of images using iCoseg. We then use the structure-from-motion implementation by Snavely et al. (2006) called 'Bundler' to recover camera parameters for each image in this group. Using the silhouettes from iCoseg, and camera parameters obtained from structure-from-motion (Snavely et al. 2006), in conjunction with a octree-reconstruction-based shape-from-silhouette algorithm (Szeliski 1993; Chen et al. 2008) we generate a texture mapped 3D model of the object of interest. Figure 19 shows an overview. For more details the reader is referred to Kowdle et al. (2010).

Results Figures 20, 21, 22, 23, 24, 25 show results on a number of datasets ranging from a simple collection taken in a controlled setup to a community photo collection and a video captured in cluttered scenes. In each figure, we show the rendered 3D model for each dataset, captured from novel view-points. For all datasets except the dino dataset, we texture map the model by back-projecting the faces of the mesh onto a single image.

8 Conclusions

We present an algorithm for interactive co-segmentation of a group of realistic related images. We propose iCoseg, an approach that co-segments all images in the group using an energy minimization framework, and an automatic recommendation system that intelligently recommends a region among all images in the group where the user should scribble next. We introduce and make publicly available the largest co-segmentation dataset yet, the CMU-Cornell iCoseg Dataset, containing 38 groups (643 images), along

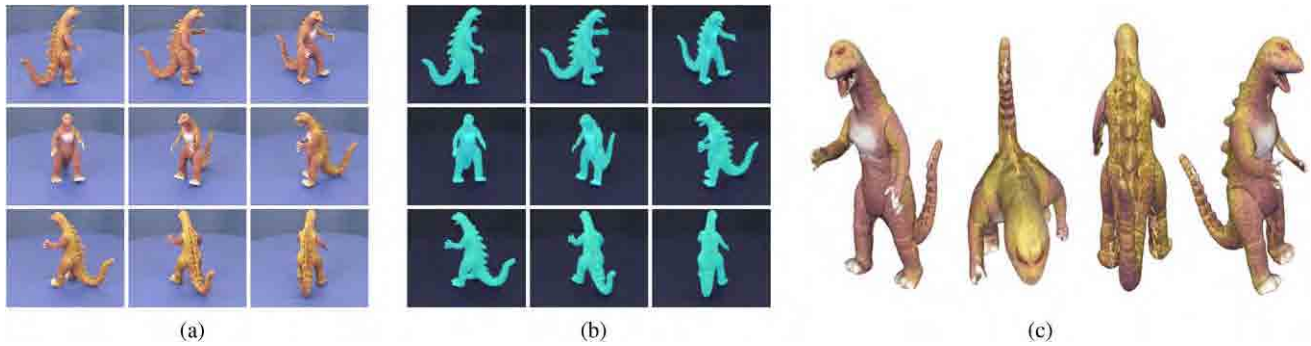


Fig. 20 Dino dataset (36 images): (**a**) subset of the collection of images given to the system where the dino was marked the object of interest; (**b**) resulting silhouettes after co-segmentation; (**c**) some sample novel views of the 3D model

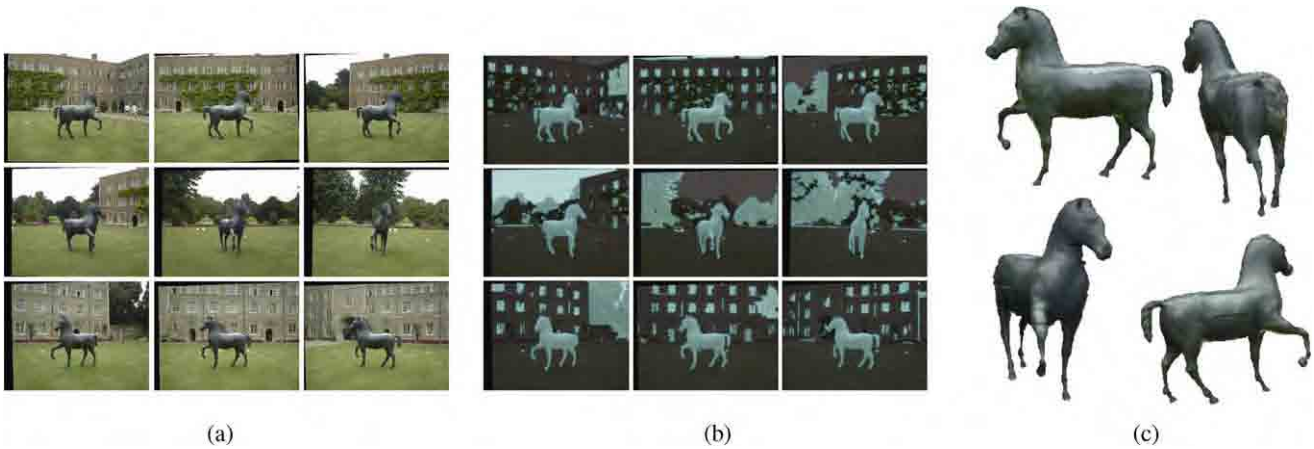


Fig. 21 Cambridge unicorn dataset (14 images): (**a**) subset of the collection of images given to the system where the unicorn statue was marked as the object of interest; (**b**) resulting silhouettes after co-segmentation; (**c**) some sample novel views of the 3D model

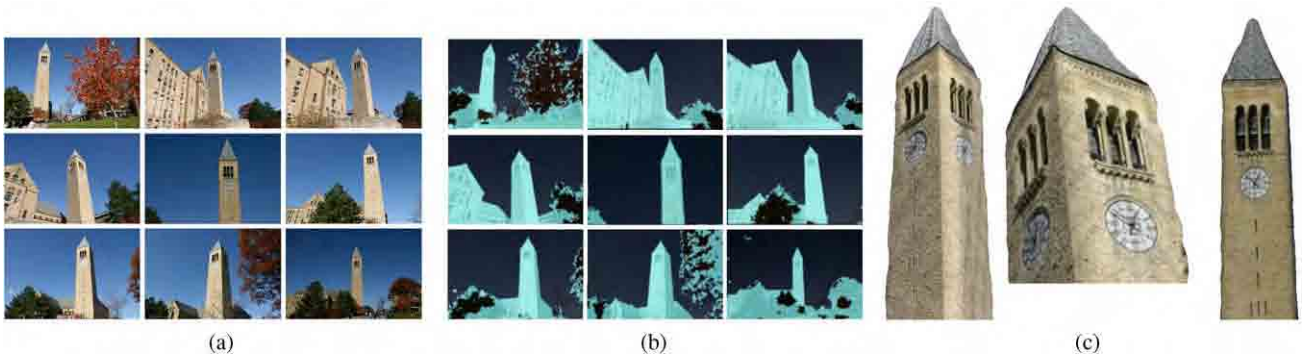


Fig. 22 Clock tower dataset (32 images): (**a**) subset of the collection of images given to the system where the clock tower was marked as the object of interest; (**b**) resulting silhouettes after co-segmentation; (**c**) some sample novel views of the 3D model

with pixel groundtruth hand annotations. In addition to machine experiments with synthetic scribbles, we perform a user-study on our developed interactive co-segmentation interface (also available online), both of which demonstrate that using iCoseg, users can achieve good quality segmentations with significantly lower time and effort than exhaustively examining all cutouts.

Acknowledgements The authors would like to thank Yu-Wei Chao for data collection and annotation, and Kevin Tang for developing the java-based GUI (iScrrible) (Batra et al. 2009) used in our user-studies.

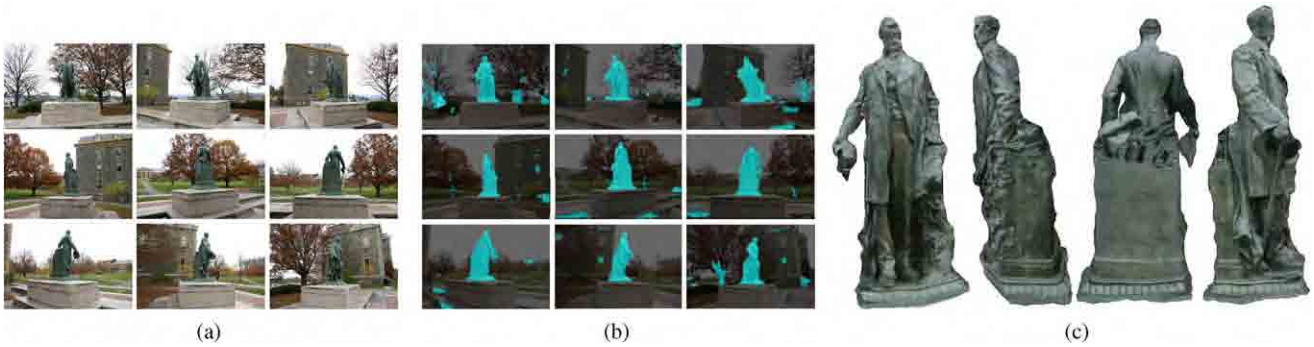


Fig. 23 Statue dataset (38 images): (**a**) subset of the collection of images given to the system where the statue was marked as the object of interest; (**b**) resulting silhouettes after co-segmentation; (**c**) some sample novel views of the 3D model

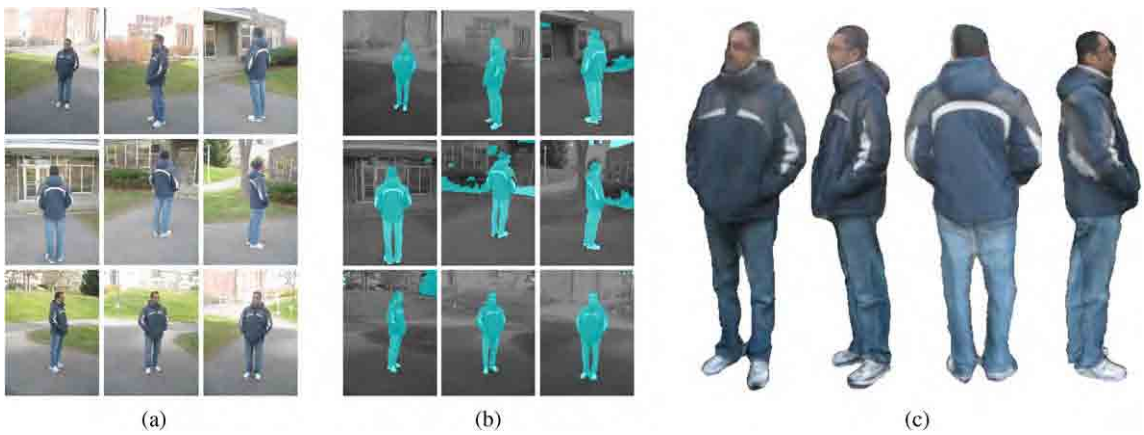


Fig. 24 Video dataset (17 images obtained by sampling the video): (**a**) subset of the collection of images given to the system where the person was considered the object of interest; (**b**) resulting silhouettes after co-segmentation; (**c**) some sample novel views of the 3D model

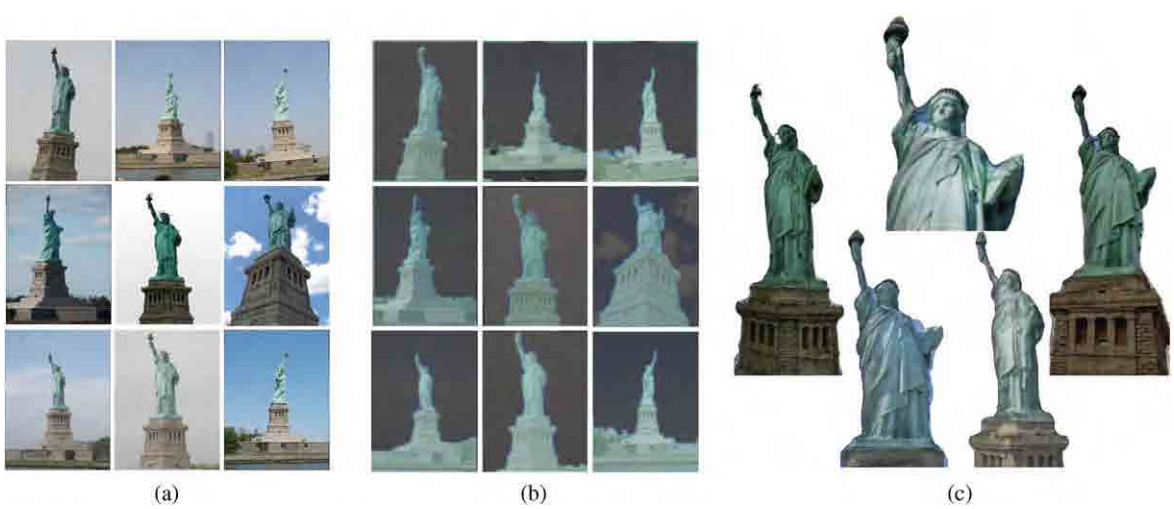


Fig. 25 Community photo collection—Statue of Liberty dataset: (**a**) subset of the collection of images given to the system—for our co-segmentation algorithm we use a subset of 15 images spanning a large field of view from a collection of 1600 images; (**b**) resulting silhouettes after co-segmentation; (**c**) some sample novel views of the 3D model

References

Bagon, S. (2006). Matlab wrapper for graph cut. http://www.wisdom.weizmann.ac.il/~bagon.

Bai, X., & Sapiro, G. (2007). A geodesic framework for fast interactive image and video segmentation and matting. In *ICCV*.

Batra, D., Sukthankar, R., & Chen, T. (2008). Semi-supervised clustering via learnt codeword distances. In *BMVC*.

Batra, D., Kowdle, A., Parikh, D., Tang, K., & Chen, T. (2009). http://amp.ece.cornell.edu/projects/touch-coseg/. Interactive Co-segmentation by Touch.

Batra, D., Kowdle, A., Parikh, D., Luo, J., & Chen, T. (2010). icoseg: interactive co-segmentation with intelligent scribble guidance. In *CVPR*.

Bouman, C. A. (1997). Cluster: an unsupervised algorithm for modeling Gaussian mixtures. Available from http://www.ece.purdue.edu/~bouman.

Boykov, Y., & Jolly, M. P. (2001). Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV*.

Boykov, Y., & Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(9), 1124–1137.

Boykov, Y., Veksler, O., & Zabih, R. (2001). Efficient approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(12), 1222–1239.

Chen, Y., & Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and Vision Computing*, *10*(3), 145–155.

Chen, Z., Chou, H. L., & Chen, W. C. (2008). A performance controllable octree construction method. In *ICPR* (pp. 1–4).

Collins, B., Deng, J., Li, K., & Fei-Fei, L. (2008). Towards scalable dataset construction: an active learning approach. In *ECCV*.

Comaniciu, D., & Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(5), 603–619.

Criminisi, A., Sharp, T., & Blake, A. (2008). Geos: geodesic image segmentation. In *ECCV* .

Cui, J., Yang, Q., Wen, F., Wu, Q., Zhang, C. Gool, L. V., & Tang, X. (2008). Transductive object cutout. In *CVPR*.

Curless, B., & Levoy, M. (1996). A volumetric method for building complex models from range images. In *SIGGRAPH '96: proceedings of the 23rd annual conference on computer graphics and interactive techniques* (pp. 303–312). New York: ACM. doi:http://doi.acm.org/10.1145/237170.237269.

Fang, Y. H., Chou, H. L., & Chen, Z. (2003). 3d shape recovery of complex objects from multiple silhouette images. *Pattern Recognition Letters*, *24*(9–10), 1279–1293.

Fitzgibbon, A. W., Cross, G., & Zisserman, A. (1998). Automatic 3d model construction for turn-table sequences. In *Proceedings of SMILE workshop on structure from multiple images in large scale environments* (Vol. 1560, pp. 154–170).

Forbes, K., Nicolls, F., de Jager, G., & Voigt, A. (2006). Shape-from-silhouette with two mirrors and an uncalibrated camera. In *ECCV* (pp. 165–178).

Franco, J. S., & Boyer, E. (2003). Exact polyhedral visual hulls. In *BMVC* (Vol. 1, pp. 329–338). http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.4.5634.

Gallagher, A., & Chen, T. (2008). Estimating age, gender and identity using first name priors. In *CVPR*.

Hochbaum, D. S., & Singh, V. (2009). An efficient algorithm for co-segmentation. In *ICCV*.

Hoiem, D., Efros, A. A., & Hebert, M. (2005). Geometric context from a single image. In *ICCV*.

Kapoor, A., Grauman, K., Urtasun, R., & Darrell, T. (2007). Active learning with Gaussian processes for object categorization. In *ICCV*.

Kohli, P., & Torr, P. H. S. (2008). Measuring uncertainty in graph cut solutions. *Computer Vision and Image Understanding*, *112*(1), 30–38.

Kolmogorov, V., & Zabih, R. (2004). What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(2), 147–159.

Kowdle, A., Batra, D., Chen, W. C., & Chen, T. (2010). imodel: interactive co-segmentation for object of interest 3d modeling. In *Workshop on reconstruction and modeling of large-scale 3D virtual environments at European conference on computer vision*.

Lee, Y. J., & Grauman, K. (2010). Collect-cut: segmentation with top-down cues discovered in multi-object images. In *CVPR*.

Leung, T., & Malik, J. (1998). Contour continuity in region based image segmentation. In *ECCV*.

Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J., & Fulk, D. (2000). The digital Michelangelo project: 3d scanning of large statues. In *SIGGRAPH* (pp. 131–144).

Li, Y., Sun, J., Tang, C. K., & Shum, H. Y. (2004). Lazy snapping. In *SIGGRAPH*.

Mu, Y., & Zhou, B. (2007). Co-segmentation of image pairs with quadratic global constraint in MRFs. In *ACCV*.

Mukherjee, L., Singh, V., & Dyer, C. R. (2009). Half-integrality based algorithms for co-segmentation of images. In *CVPR*.

Rother, C., Kolmogorov, V., & Blake, A. (2004). "Grabcut": interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*.

Rother, C., Minka, T., Blake, A., & Kolmogorov, V. (2006). Cosegmentation of image pairs by histogram matching—incorporating a global constraint into MRFs. In *CVPR*.

Schnitman, Y., Caspi, Y., Cohen Or, D., & Lischinski, D. (2006). Inducing semantic segmentation from an example. In *ACCV*.

Settles, B. (2009). *Active learning literature survey* (Computer Sciences Technical Report 1648). Madison: University of Wisconsin.

Seung, H. S., Opper, M., & Sompolinsky, H. (1992). Query by committee. In *COLT*.

Snavely, N., Seitz, S., & Szeliski, R. (2006). Photo tourism: exploring photo collections in 3d. In *SIGGRAPH* (pp. 835–846).

Starck, J., & Hilton, A. (2007). Surface capture for performance-based animation. *IEEE Computer Graphics and Applications*, *27*(3), 21–31.

Szeliski, R. (1993). Rapid octree construction from image sequences. *CVGIP. Image Understanding*, *58*(1), 23–32.

Vicente, S., Kolmogorov, V., & Rother, C. (2010). Cosegmentation revisited: models and optimization. In *ECCV*.

Vijayanarasimhan, S., & Grauman, K. (2009). What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *CVPR*.

Vlasic, D., Baran, I., Matusik, W., & Popović, J. (2008). Articulated mesh animation from multi-view silhouettes. In *SIGGRAPH* (pp. 1–9). New York: ACM.

Yan, R., Yang, J., & Hauptmann, A. (2003). Automatically labeling video data using multi-class active learning. In *ICCV*.

Zhang, L., Curless, B., & Seitz, S. M. (2002). Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *3DPVT* (p. 24).