Information-Theoretic Analysis of Input Strokes in Visual Object Cutout

Yadong Mu, Bingfeng Zhou and Shuicheng Yan

***Abstract*—Semantic object cutout serves as a basic unit in various image editing systems. In a typical scenario users are required to provide several strokes which indicate part of the pixels as image background or objects. However, most existing approaches are passive in the sense of accepting input strokes without checking the consistence with user's intention. Here we argue that an active strategy may potentially reduce the interaction burden. Before any real calculation for segmentation, the program can roughly estimate the uncertainty for each image element and actively provide useful suggestions to users. Such a pre-processing is particularly useful for beginners unaware of feeding the underlying cutout algorithms with optimal strokes.**

We develop such an active object cutout algorithm, named ActiveCut, which makes it possible to automatically detect ambiguity given current user-supplied strokes, and synthesize "suggestive strokes" as feedbacks. Generally suggestive strokes come from the ambiguous image parts and have the maximal potentials to reduce label uncertainty. Users can continuously refine their inputs following these suggestive strokes. In this way, the number of user-program interaction iterations can thus be greatly reduced. Specifically, the uncertainty is modeled by mutual information between user strokes and unlabeled image regions.

To ensure that ActiveCut works at a user-interactive rate, we adopt superpixel lattice based image representation, whose computation depends on scene complexity rather than original image resolution. Moreover, it retains the 2D-lattice topology and is thus more suitable for parallel computing. While for the most time-consuming calculation of probabilistic entropy, variational approximation is utilized for acceleration. Finally, based on submodular function theory, we provide a theoretic analysis for the performance lower bound of the proposed greedy algorithm. Various user studies are conducted on the MSRC image dataset to validate the effectiveness of our proposed algorithm.

I. Introduction

In recent years, visual object cutout from images [1], [2] or video clips [3], [4] has attracted increasing attention in both the computer graphics and computer vision communities. Unlike traditional image segmentation techniques such as Watershed [5] or Level Set [6] which mainly focus on structure information (line, edges, corners, etc.), object cutout aims to extract semantically consistent image (or video) regions, and are thus more suitable for object-level digital content editing, composition and retrieval.

Yadong Mu and Shuicheng Yan are with the Department of Electrical and Computer Engineering, National University of Singapore, 117576, Singapore. Part of work in this paper was done when Dr. Mu was a Ph.D. student at Peking University. E-mail: muyadong@gmail.com, eleyans@nus.edu.sg.

Bingfeng Zhou is with the Institute of Computer Science and Technology, Peking University, Beijing, 100871, China. Email: cczbf@pku.edu.cn

Acknowledgment: this work is supported by NSF from China under Grant number 60973054 and NUS Cross-faculty Grant of R-263-000-528-646 at Singapore.

Object cutout in images is notoriously difficult, mainly due to the gap between low-level image features (color, texture, intensity gradient etc.) and high-level semantic concepts, and the diversity of user intentions. Fully automated object cutout [7] is possible yet prone to errors. Usually this under-constrained problem can be solved by incorporating user interaction [8], [1], [2], multiple inter-related sources [9], [10], [11], [12], or a priori knowledge about object category. In the presence of blurred or mixed pixels pertaining to smoke, animal furs, hairs, etc., real-valued alpha values need be estimated for each pixel via *soft segmentation* or *image matting* [13]. Here we ignore this "soft" matting problem, focusing on "hard" segmentation.

Our investigation starts from a novel taxonomy based on the way to use the hint information. Roughly, existing approaches can be categorized as one of the *unsupervised*, *semi-supervised*, *supervised* or *active learning* case. Although having achieved promising results on many challenging images, current approaches still have their limitations. Particularly, most of them work in a passive style, without quality assessment about the user strokes. At runtime there are typically several iterations of the following loop: take user's input, calculate and display the cutout results, and then begin waiting for user's feedback. In these traditional interaction paradigms, users can only notice the inconsistency between their inputs and original intension until final results are shown, thus not efficient enough and having space to be improved.

II. Our Proposed Method: ActiveCut

An interactive cut-out procedure typically consists of several rounds of "refine the strokes $\rightarrow$ perform cut-out" loop. In many scenarios, it can be roughly divided into two stages. The first several loops aim to roughly locate the boundaries of target objects, and the following loops mainly focus on local adjustment. The goal of our proposed method (hereafter called ActiveCut) lies in reducing the efficacy of user interaction at the first stage. Unlike the passive mode, once user's strokes are available, the proposed algorithm first quickly evaluates the strokes' quality and returns several *suggestive strokes* (at an interactive rate even for large-size images) to indicate potential flaws in the original strokes, which guide users to lay down new strokes for refining. Here "flaws" are supposed to stem from the informational inadequateness of the initial strokes when they are used for confidently cutting out the desired objects. Importantly, an observation is that the major computing overhead comes from calling the underlying cutting-out algorithm whilst the burden of calculating suggestive strokes is comparably small. Note that in ActiveCut no real cutting-out computation is actually triggered, whereby with a high

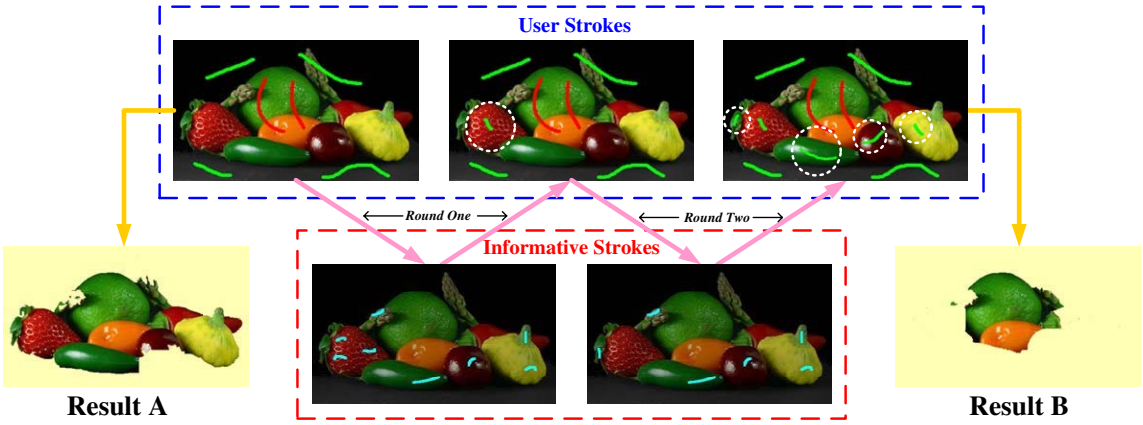


Fig. 1. Illustration of the flowchart of the proposed method. Red (or green) strokes indicate the seeds for objects (or background) respectively. Assume what the user want is something like Result B. Starting with initial user strokes which bring a segmentation in Result A, users progressively lay down additional strokes following the guidance of suggestive strokes that correspond to the most ambiguous parts in each interaction round. Newly-introduced user strokes are highlighted with dashed white circles. In this example, suggestive strokes are automatically generated using our proposed method ActiveCut, while the final cutout is obtained from a "GMM + GraphCut" algorithm, whose object function is approximately optimized by ActiveCut to ensure reasonable suggestions.

probability users retains fewer interaction round in the first stage and thus save the effort.

Such a technique is meaningful due to the following considerations: firstly, the computing burden of the first cutting-out stage is usually considerable for large-size images or long video clips. An active early-detect strategy provides the flexibility to refine user inputs without any real calculation for segmentation, thus greatly saving interaction effort. Secondly, noting the diversity of potential users, ActiveCut is especially helpful for start-up users who lack expertise in image processing or unaware of enough technique details of the underlying cutout algorithms. Thirdly, when handling complicated images, even experienced users can hardly accomplish the rough locating of desired objects in only several rounds. The initially given strokes usually ignore to specify some necessary pixel seeds, which are hardly detected by the users themselves in many cases. In this case, an assisting utility like ActiveCut is helpful.

Algorithm 1: The Proposed ActiveCut Algorithm

Step 1 (Initialization): user strokes for image background and desired objects are made available;

Step 2 (Evaluation-And-Feedback): based on information-theoretic analysis, generate n_k most informative strokes which are supposed to be best candidates to reduce image uncertainties;

Step 3 (Refinement): Users have two choices:

1) simply ignore suggestive strokes if satisfied with current user strokes, and perform segmentation using color-based algorithms like GrabCut.
2) otherwise add new strokes to the system following the guidance of suggestive strokes and go to Step 2.

The algorithm pipeline of ActiveCut is presented in Algorithm 1 and an illustrative example is found in Figure 1. Note that the functionality of ActiveCut relies on approximately optimizing the criterion of the underlying cutout algorithm. We would like to highlight that the aim here is not to provide an alternative cutout algorithm besides existing ones. Instead, ActiveCut can be seamlessly used as a plug-in for most popular systems (in this paper we focus on color-based systems like GrabCut or LazySnapping).

The rest of our presentation is organized as follows. Section III surveys related works. In Section IV, we elaborate the mathematical formulation of active object cutout, and show that the proposed entropy criterion can be decomposed into two data terms, i.e., *stroke correlation term* and *neighborhood correlation term*. In Section V and VI, we describe the details to obtain approximate solutions for these two data terms respectively. Evaluations on public image datasets are presented in Section VIII. Also, we discuss the lower bound of the algorithm performance based on submodular function theory, which is in Section VII.

III. Related Works

As stated above, we roughly cast the vast literature on visual object cutout into four categories:

Unsupervised Case (or Self-Cutout): in this case, no external hint information is supplied. The algorithm tries to find the segments by itself. A number of traditional segmenting techniques, including K-means, mean shift, Gaussian Mixture Model (GMM) + Expectation Maximization (EM), are able to group the image pixels into homogeneous regions, each of which consists of either a single connected component or several disjoint components (like in K-means). Previous works [7], [14] adopt the notation "superpixel" to name these extracted image patches. Recent progress along this direction further demonstrates enhanced performance given more reference images, either in the form of flash/no-flash images [9], or relevant image pair [10], [15], [11].

Semi-supervised Case (or Interactive Cutout) is a fruitful research direction in the past few years. For the targeted image, users manually label some pixels as "seeds", which are later

propagated over the whole image. One seminal work in this category may be the Graph Cuts framework [8] proposed by Yuri Boykov et al. in 2001. In the typical settings, it requires two kinds of strokes, one for the desired objects, the other for the image background. Later, in *GrabCut* [1] the stroke types are reduced to only one via introducing user-interaction primitives such as lasso or bounding box. However, although polynomial-time solvable, Graph Cuts is computational expensive for large-size images with pixels on the order of 10^6. For acceleration, another variant, *LazySnapping*, adopts superpixel as basic data structure. Other approaches outside the Graph cuts framework exist. For example, the seminal work in [16] is based on random walks for segmentation. In [17], the authors solve this problem by utilizing the Laplace-Beltrami operator defined on the image grid. While in both the work in [18] and [19], the authors propose an evolutional updating process with convergence guarantee.

The idea of semi-supervision can also be applied to multiple images. Given a well-segmented image or any user-supplied constraints, the cutout task on similar images can be easier and more accurate, as demonstrated in [20].

Supervised Case: If we know the object category in advance, or we target specific object kinds, cutting out an object from an image will be of less ambiguity. In practice, we can encode the prior knowledge about an object category via training on a large number of user-labeled samples, such as the "C"-like mug handles. the "Ω" shape formed by the head and shoulder for human.

Active-Learning Case: most approaches in Table I (certainly the list is incomplete) are "passive" methods, while no prior work on active object cutout has been developed (although we note that the work in [21] proposed similar technique, however, it is not for the active segmentation task), which motivates our work described in this paper.

TABLE I
TAXONOMY FOR OBJECT CUTOUT TECHNIQUES

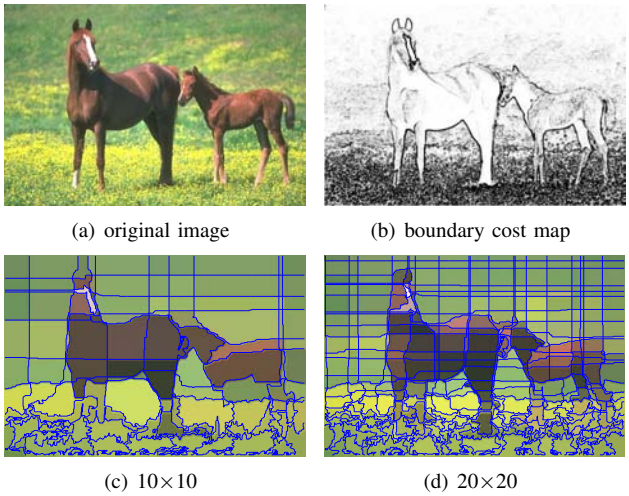
	Single Image	Multiple Images
Unsupervised learning	K-means, GMM+EM, Mean Shift, Normalized Cut	[10], [11], [9], [22], [15]
Semi-supervised learning	[8], [1], [2], [17], [18]	[20], [23]
Supervised learning	[24], [25], [26]	
Active learing	the proposed ActiveCut algorithm in this paper	

IV. Problem Formulation

A. Image model

Unsupervised over-segmentation of an image into superpixels is a crucial preprocessing step for many real-time image editing tasks. Superpixel-based image representation has several merits compared with pixel-based one. Intrinsically pixels are not natural entities to convey image meanings, since they primarily depend on the CCD resolution of cameras. In contrast, a superpixel is homogeneous, spatially-coherent, preserving most of the structure configuration information, stable over scales and image resolution, and has much smaller number (on the order of $10^3 \sim 10^4$) compared with that of pixels (typically 10^6).

Most superpixel algorithms are unable to keep the two-dimensional pixel-lattice structure of an image, outputting a general graph, as in [7], [27], [28]. In fact, a topology-keeping superpixel algorithm benefits in several aspects, especially for the consideration of efficacy. To generate a superpixel lattice, we adopt the similar idea as in [14], which is able to reduce a $m \times n$ image to $\tilde{m} \times \tilde{n}$ superpixel lattice ($\tilde{m} \ll m$ and $\tilde{n} \ll n$). The construction procedure is incremental: in each step, one more horizontal or vertical splitting line is greedily inserted along the low-density paths (i.e., the paths with high probabilities to be nearby object contours) in a pre-defined *boundary cost map*, until all $\tilde{m}-1$ horizontal and $\tilde{n}-1$ vertical lines are all added. To preserve the lattice topology, always we ensure that 1) each pair of horizontal and vertical path crosses only once, and 2) two vertical (or horizontal) paths never cross. Unlike [14], we adopt a local contrast based method to specify the *boundary cost map*, which can efficiently handle colorful images. An example is shown in Figure 2. At last, each superpixel can be represented by a six-dimensional vector, i.e. spLattice(i) = $\left(N_i, m_x^i, m_y^i, m_r^i, m_g^i, m_b^i\right)^T$, where N_i is the pixel number, while $M_s^i = \left(m_x^i, m_y^i\right)^T$ and $M_c^i = \left(m_r^i, m_g^i, m_b^i\right)^T$ denote the mean values of spatial coordinates and RGB colors respectively.



(a) original image (b) boundary cost map

(c) 10×10 (d) 20×20

Fig. 2. Illustration for "superpixel lattice". In (b), salient boundaries are rendered with lower intensity. (c)(d) are filtered images under different lattice resolution.

Our proposed object cutout approach utilizes above superpixel lattice as the basic data structure. Denote the valid index set for a $\tilde{m} \times \tilde{n}$ lattice as $\mathcal{V}$. Each lattice node is associated with a random variable $\mathcal{X}_i$. Here we focus on the binary case, since multi-class cutout problem can be reduced to a series of two-class sub-problems. Values of $\mathcal{X}_\mathcal{V}$ indicate the statuses for lattice nodes, typically 1 means "foreground" or "object", and -1 means "background" or "non-object". Most prior works such as GrabCut and Lazysnapping assume $\mathcal{X}_\mathcal{V}$ to be discrete, typically subject to the Ising model [29]. Here we assume

the random variables to be continuous and encourage the adjacent nodes to have similar values via utilizing a weighted quadratic loss function. Specifically, we presume that $\mathcal{X}_\mathcal{V}$ forms a Gaussian random field (GRF), with its *energy function* defined as:

$$E(\mathcal{X}_\mathcal{V}|W) = \frac{1}{2}\sum_i \sum_{j\in\mathcal{N}_i} w_{ij}\left(\mathcal{X}_i - \mathcal{X}_j\right)^2, \quad (1)$$

where $\mathcal{N}_i$ denotes the neighborhood system of $\mathcal{X}_i$. Traditional 4-connected or 8-connected neighbor system only keeps inadequate Markov locality. Based on the lattice representation, we can extend it to larger scale, i.e. $\Delta x \times \Delta y$ (in practice we set $\Delta x = \Delta y = 13$).

The weight coefficients in Equation (1) are designed to be symmetric and capture both the similarities in terms of both color and spatial distance, i.e.,

$$w_{ij} = (N_i + N_j)\exp\left(-\frac{\parallel M_c^i - M_c^j \parallel^2}{2\sigma_c^2} - \frac{\parallel M_s^i - M_s^j \parallel^2}{2\sigma_s^2}\right),$$

where σ_c and σ_s are parameters related to color and spatial variations respectively, which can be estimated from the original images. Note that although w_{ij} is defined mainly utilizing color information, the proposed method is actually general and supports information from other modality such as texture, shape prior [30], [31] or symmetries [32], by modifying the definition of w_{ij}.

According to the Hammersley-Clifford theorem [33], a MRF can be equivalently characterized by a Gibbs distribution. Thus, we have

$$P(\mathcal{X}_\mathcal{V}|W,\beta) = \frac{1}{Z_{W,\beta}}\exp\left(-\beta E(\mathcal{X}_\mathcal{V}|W)\right), \quad (2)$$

where β is the so-called "inverse temperature" parameter originated from statistical physics and $Z_{W,\beta}$ is a normalizing constant named "partition function" in physics to ensure the probability distribution a valid one, whose exact value can be obtained via integration over $\mathcal{X}_\mathcal{V}$, i.e.,

$$Z_{W,\beta} = \int \exp\left[-\beta E(\mathcal{X}_\mathcal{V}|W)\right] d\mathcal{X}_\mathcal{V}. \quad (3)$$

B. The Entropy Criterion

One of the major issues for active object cutout is to quantify the informativeness contained in current user strokes, and reliably estimate the information gain after adding one extra stroke. The best "suggestive stroke" is the one that reduces image uncertainty as much as possible in later pixel-classification phase. Intuitively, such a good criterion should reflect the following two considerations:

1) **Stroke Correlation Term (SC-term)**, i.e., how correlated the newly-added stroke is with existing strokes. Usually we prefer new strokes which have small correlation values with current strokes, since they potentially help estimate the labels of image regions which have very low confidence values under current strokes.
2) **Neighborhood Correlation Term (NC-term)**, i.e., the goal of an informative stroke is mainly to help reduce its neighbors' uncertainty. The elements covered by the informative stroke should have enough similar neighbors. Otherwise, if they are almost isolated from adjacent elements in the feature space, only tiny improvement can be expected.

In this paper, we formulate the active object cutout problem based on the above observations. Before proceeding, let us first clarify some notations in information theory (e.g., informational entropy) used here. Firstly, the entropy for a random vector $\mathcal{X}_\mathcal{V}$ with index set $\mathcal{V}$ can be expressed as follows:

$$H(\mathcal{X}_\mathcal{V}) = -\int p(\mathcal{X}_\mathcal{V})\log p(\mathcal{X}_\mathcal{V}) d\mathcal{X}_\mathcal{V}, \quad (4)$$

where $p(x) \geq 0$ and $\int p(x)\ dx = 1$. Suppose $\mathcal{V}$ is comprised of two exclusive subsets, i.e., $\mathcal{V} = \mathcal{S} \cup \mathcal{A}$. If the values of $\mathcal{X}_\mathcal{A}$ have already been observed, we can calculate the entropy of $\mathcal{X}_\mathcal{S}$ conditioned on $\mathcal{X}_\mathcal{A}$ as follows:

$$H(\mathcal{X}_\mathcal{S}|\mathcal{X}_\mathcal{A}) = -\int p(\mathcal{X}_\mathcal{S},\mathcal{X}_\mathcal{A})\log p(\mathcal{X}_\mathcal{S}|\mathcal{X}_\mathcal{A}) d\mathcal{X}_\mathcal{V}. \quad (5)$$

Some authors directly use this conditional entropy as their criteria in similar tasks like set covering problem. However, in these settings, the highest entropy set is usually characterized by stroke locations as far as possible from each other, which contradicts our expectation about reasonable solutions. A better candidate is the "mutual information". In [34], the authors utilize it to perform intelligent annotation of face images. And in [35], the authors apply it to the problem of sensor placement. Formally, mutual information of two subsets can be expressed as follows:

$$\mathrm{MI}(\mathcal{A}) = H(\mathcal{X}_\mathcal{S}) - H(\mathcal{X}_\mathcal{S}|\mathcal{X}_\mathcal{A}), \quad (6)$$

where $H(\mathcal{X}_\mathcal{S})$ and $H(\mathcal{X}_\mathcal{S}|\mathcal{X}_\mathcal{A})$ correspond to entropies before or after knowing $\mathcal{X}_\mathcal{A}$ respectively. The mutual information actually describes how the observations in $\mathcal{X}_\mathcal{A}$ decrease uncertainty (or entropy) in the rest random variables $\mathcal{X}_\mathcal{S}$.

Let us formally state the problem. Suppose elements in $\mathcal{A}$ are all covered by user strokes, and the status of each element in $\mathcal{S}$ need to be estimated from $\mathcal{A}$. For clarity, we assume there are totally k interaction rounds between users and cutout system before starting real computation for segmenting, and in each round users adopt the suggested strokes (in practice this restriction can be relaxed). The goal is to pursue optimal k strokes which are able to maximally reduce the amount of uncertainties in $\mathcal{S}$, i.e.,

$$\mathcal{A}^* = \arg\max_{\mathcal{A}\subseteq\mathcal{V};|\mathcal{A}|=k} \mathrm{MI}(\mathcal{A}). \quad (7)$$

C. Decomposition of MI($\mathcal{A}$)

Optimizing Equation (7) is difficult. It can be proved that, given rational M and the assumption that $\mathcal{V}$ is a Gaussian process (we will prove it later), deciding whether there exists a subset $\mathcal{A} \subseteq \mathcal{V}$ of cardinality k such that $\mathrm{MI}(\mathcal{A}) \geq M$ is NP-complete. Consequently, approximate optimization is indispensable. Here we adopt a greedy strategy. In each round we greedily pursue a single element $y \in \mathcal{V} \setminus \mathcal{A}$ to maximize $\Delta\mathrm{MI}_y = \mathrm{MI}(\mathcal{A} \cup y) - \mathrm{MI}(\mathcal{A})$.

Denote the random variable pertaining to y as $\mathcal{X}_y$ and $\bar{\mathcal{A}} = \mathcal{V} \setminus \mathcal{A}$. It is easily verified that:

$$\begin{aligned} & \text{MI}(\mathcal{A} \cup y) - \text{MI}(\mathcal{A}) \\ = \; & H(\mathcal{X}_{\mathcal{A}\cup y}) - H(\mathcal{X}_{\mathcal{A}\cup y}|\mathcal{X}_{\bar{\mathcal{A}}\setminus y}) - [H(\mathcal{X}_{\mathcal{A}}) - H(\mathcal{X}_{\mathcal{A}}|\mathcal{X}_{\bar{\mathcal{A}}})] \\ = \; & H(\mathcal{X}_{\mathcal{A}\cup y}) - H(\mathcal{X}_{\mathcal{V}}) + H(\mathcal{X}_{\bar{\mathcal{A}}\setminus y}) - [H(\mathcal{X}_{\mathcal{A}}) - H(\mathcal{X}_{\mathcal{V}}) \\ & + H(\mathcal{X}_{\bar{\mathcal{A}}})] \\ = \; & H(\mathcal{X}_y|\mathcal{X}_{\mathcal{A}}) - H(\mathcal{X}_y|\mathcal{X}_{\bar{\mathcal{A}}\setminus y}). \end{aligned} \tag{8}$$

Thus two terms $H(\mathcal{X}_y|\mathcal{X}_{\mathcal{A}})$ and $H(\mathcal{X}_y|\mathcal{X}_{\bar{\mathcal{A}}\setminus y})$ dominate ΔMI_y. The maximum of MI will be obtained at a large $H(\mathcal{X}_y|\mathcal{X}_{\mathcal{A}})$ together with a small $H(\mathcal{X}_y|\mathcal{X}_{\bar{\mathcal{A}}\setminus y})$. Furthermore, we can notice that $H(\mathcal{X}_y|\mathcal{X}_{\mathcal{A}})$ and $H(\mathcal{X}_y|\mathcal{X}_{\bar{\mathcal{A}}\setminus y})$ are elegantly corresponding to SC-term and NC-term respectively: $H(\mathcal{X}_y|\mathcal{X}_{\mathcal{A}})$ describes how it is correlated with existing strokes, and the latter term reflects the similarity between the image superpixel under consideration and its neighboring elements. Intuitively, large $H(\mathcal{X}_y|\mathcal{X}_{\mathcal{A}})$ indicates a higher potential of y to carry novel information missing in $\mathcal{A}$, and small $H(\mathcal{X}_y|\mathcal{X}_{\bar{\mathcal{A}}\setminus y})$ indicates that the labels of $\mathcal{X}_y$ and its neighbors are tightly related, thus uncovering $\mathcal{X}_y$ is able to reduce uncertainty.

V. Stroke Correlation Term Calculation

A. Variational Approximation

Exact calculation for conditional entropy is generally intractable. It is a common trick to approximate it either stochastically (Monte Carlo Markov Chain sampling) or deterministically (mean field, or expectation propagation [36]). Here we adopt a variational view, approximating the original problem by minimizing the value of a pre-specified functional.

In variational inference, typically we assume the targeted functions have a specific parametric form (such as a Gaussian) or the products of a series of factors. For binary object cutout, with the observation that we are mostly interested in only two discrete values 1 (object) or -1 (background), we discretize each random variable in $\mathcal{V}$ to $\{-1, 1\}$. In the standard variational method, the original joint probabilistic distribution can be factorized into the product of several simple functions. Particularly, for current problem we have $P(\mathcal{X}_{\mathcal{S}}|\mathcal{X}_{\mathcal{A}}) \approx Q(\mathcal{X}_{\mathcal{S}}) = \prod_{j\in\mathcal{S}} Q_j(\mathcal{X}_j)$, where $Q_j(\cdot)$ denotes a discrete function ranging over $\{-1, 1\}$. We further assume that $Q_j(\mathcal{X}_j) = \frac{1+m_j\mathcal{X}_j}{2}$ the same as in the Mean Field method [29], where $m_j = \langle\mathcal{X}_j\rangle_Q$ ($\langle\cdot\rangle_Q$ denotes the statistical expectation over distribution Q).

One step further, we can prove that, maximizing the lower bound of the likelihood of $\log p(\mathcal{X}_{\mathcal{A}})$ can be obtained via minimizing the following functional in a form of *Kullback-Leibler divergence* [37]:

$$\mathcal{KL}\left(Q(\mathcal{X}_{\mathcal{S}})\|P(\mathcal{X}_{\mathcal{S}}|\mathcal{X}_{\mathcal{A}})\right) = \int Q(\mathcal{X}_{\mathcal{S}}) \log\left\{\frac{Q(\mathcal{X}_{\mathcal{S}})}{P(\mathcal{X}_{\mathcal{S}}|\mathcal{X}_{\mathcal{A}})}\right\} d\mathcal{X}_{\mathcal{S}}.$$

For brevity, we define the following two notations:

$$V_Q = \int Q \log Q \; d\mathcal{X}_{\mathcal{S}}, \quad V_P = \int Q \log P \; d\mathcal{X}_{\mathcal{S}}.$$

Thus $\mathcal{KL}(Q||P) = V_Q - V_P$. Recall that here all random variables in $\mathcal{V}$ are discretized for tractability consideration, thus both V_Q and V_P can be expressed as function of $\{m_i,\ i \in \mathcal{S}\}$ after integrating over $\mathcal{X}_{\mathcal{S}}$. It can be verified that

$$V_Q = \sum_{i\in\mathcal{S}} \left(\frac{1+m_i}{2}\log\frac{1+m_i}{2} + \frac{1-m_i}{2}\log\frac{1-m_i}{2}\right),$$

$$V_P = \langle \log P(\mathcal{X}_{\mathcal{S}}|\mathcal{X}_{\mathcal{A}})\rangle_Q = \beta \sum_{i\in\mathcal{S}} \sum_{j\in\mathcal{N}_i} w_{ij}\ m_i m_j + \text{const.}$$

The objective $\mathcal{KL}(Q||P)$ is difficult to optimize due to its high non-linearity. However, we can derive an updating rule for m_i by investigating its first-order gradient $\frac{\partial\mathcal{KL}(Q||P)}{\partial m_i}$, whereby local optima are feasible. Taking partial derivative with respect to m_i, we get

$$\begin{aligned} \frac{\partial\mathcal{KL}(Q||P)}{\partial m_i} &= \frac{\partial(V_Q - V_P)}{\partial m_i} \\ &= \frac{1}{2}\log\frac{1+m_i}{1-m_i} - \beta\sum_{j\in\mathcal{N}_i} w_{ij}\ m_j \end{aligned} \tag{9}$$

For any locally optimal m_i^*, there is $\frac{\partial\mathcal{KL}(Q||P)}{\partial m_i} = 0$. However, the non-linear term $\log\frac{1+m_i}{1-m_i}$ in Equation (9) ruins the linear closed-form solution of m_i^*, which motivates further approximation. Note that $\left(\partial^2 V_Q/\partial m_i^2\right)_{m_i=0} = 1$, we can approximate the complicated logarithm function in $\partial V_Q/\partial m_i$ with a linear form, i.e.,

$$\frac{1}{2}\log\frac{1+m_i}{1-m_i} \approx m_i. \tag{10}$$

Set the partial derivative in Equation (9) to be zero, we can obtain the updating equation at iteration t:

$$m_i^{(t+1)} = \sum_j \beta\ w_{ij}\ m_j^{(t)} = \sum_j \tilde{w}_{ij}\ m_j^{(t)} \tag{11}$$

In practice, the value of the *inverse temperature parameter* β is carefully adjusted so that

$$\tilde{w}_{ij} = \frac{w_{ij}}{\max_{m\in\mathcal{V}}|\sum_n w_{mn}|}. \tag{12}$$

Obviously $0 \le \tilde{w}_{ij} \le 1$ for all valid indices $i, j \in \mathcal{V}$. Equation (11) provides an iterative method to obtain the optimal factorized posterior $Q_j(\mathcal{X}_j)$, which is presented in Algorithm 2. The calculation of stroke correlation terms can be greatly simplified once all $Q_j(\mathcal{X}_j)$ ($j \in \mathcal{S}$) are known. From Equation (5), we have:

$$\begin{aligned} & H(\mathcal{X}_y|\mathcal{X}_{\mathcal{A}}) \\ = \; & -\int \sum_{\mathcal{X}_y} P(\mathcal{X}_y, \mathcal{X}_{\mathcal{A}}) \log P(\mathcal{X}_y|\mathcal{X}_{\mathcal{A}}) \; d\mathcal{X}_{\mathcal{A}} \\ = \; & -\int P(\mathcal{X}_{\mathcal{A}}) \sum_{\mathcal{X}_y} P(\mathcal{X}_y|\mathcal{X}_{\mathcal{A}}) \log P(\mathcal{X}_y|\mathcal{X}_{\mathcal{A}}) \; d\mathcal{X}_{\mathcal{A}}. \end{aligned}$$

Note that each variable in set $\mathcal{A}$ is associated with a fixed label, i.e. $P(\mathcal{X}_{\mathcal{A}}) = \delta(\mathcal{X}_{\mathcal{A}} = \mathcal{L}_{\mathcal{A}})$, where $\mathcal{L}_{\mathcal{A}}$ is the user-specified labels (-1 or 1 in binary-classification case) and δ is the Dirac delta function, thus $H(\mathcal{X}_y|\mathcal{X}_{\mathcal{A}})$ can be simplified as:

$$\begin{aligned} H(\mathcal{X}_y|\mathcal{X}_{\mathcal{A}}) &= -\sum_{\mathcal{X}_y} P(\mathcal{X}_y|\mathcal{X}_{\mathcal{A}} = \mathcal{L}_{\mathcal{A}}) \log P(\mathcal{X}_y|\mathcal{X}_{\mathcal{A}} = \mathcal{L}_{\mathcal{A}}) \\ &= -\sum_{\mathcal{X}_y} Q_y(\mathcal{X}_y) \log Q_y(\mathcal{X}_y). \end{aligned} \tag{13}$$

Algorithm 2: $\mathcal{KL}$-minimizing factorized approximation

Input:
- $\mathcal{X}_{\mathcal{A}} = \mathcal{L}_{\mathcal{A}}$
- Learning rate $\gamma = 0.05$
- Error tolerance $e_{max} = 10^{-3}$
- max iteration T_{max}

Result: $Q_j^*(\mathcal{X}_j)$,$j \in \mathcal{X}_{\mathcal{S}}$

initialize $m_j^{(0)} = 0$ if $j \in \mathcal{X}_{\mathcal{S}}$, or user-specified value (-1 or 1) if $j \in \mathcal{X}_{\mathcal{A}}$.

for *t = 0 to* T_{max} **do**
  for $j \in \mathcal{X}_{\mathcal{S}}$ **do**
    compute m_j^{t+1} according to Equation 11;
    $m_j^{t+1} = m_j^t + \gamma\ (m_j^{t+1} - m_j^t)$;
  end
  if $\max_j |m_j^{t+1} - m_j^t| \leq e_{max}$ **then**
    Break;
  end
end

B. Convergence Analysis

In this section we will show that the iterative updating process in Algorithm 2 converges to a fixed point. Denote the coefficient matrix as $\widetilde{W}$, whose (i,j)-th entry is $\widetilde{w}_{ij}$ defined in Equation (12). It is easy to verify that

$$\mathcal{X}_{\mathcal{V}}^{t+1} = \left[(1-\gamma\ )I + \gamma\ \widetilde{W}\right] \mathcal{X}_{\mathcal{V}}^t, \tag{14}$$

where I is the identity matrix. Let $P = (1-\gamma\ )I + \gamma\ \widetilde{W}$. Recall that $\mathcal{V} = \mathcal{S} \cup \mathcal{A}$. Without loss of generality, suppose elements in $\mathcal{X}_{\mathcal{V}}$ are arranged in the form of $\left[\mathcal{X}_{\mathcal{S}}^T \ \ \mathcal{X}_{\mathcal{A}}^T\right]^T$, where the superscript T denotes matrix transposition. Accordingly, matrix P can be decomposed into four submatrices (similar to the trick used in [38]), i.e.,

$$P = \begin{bmatrix} P_{\mathcal{SS}} & P_{\mathcal{SA}} \\ P_{\mathcal{AS}} & P_{\mathcal{AA}} \end{bmatrix}. \tag{15}$$

From Equation (14), there is

$$\mathcal{X}_{\mathcal{S}}^{t+1} = P_{\mathcal{SS}}\mathcal{X}_{\mathcal{S}}^t + P_{\mathcal{SA}}\mathcal{X}_{\mathcal{A}}. \tag{16}$$

In each iteration, the values of $\mathcal{X}_{\mathcal{A}}$ are always clamped to be $\mathcal{L}_{\mathcal{A}}$, thus we can further simplify Equation (16):

$$\mathcal{X}_{\mathcal{S}}^{t+1} = P_{\mathcal{SS}}\mathcal{X}_{\mathcal{S}}^t + P_{\mathcal{SA}}\mathcal{L}_{\mathcal{A}}. \tag{17}$$

Based on the above matrix splitting operation, we are able to reach the main conclusion for convergence guarantee (proof is omitted):

Theorem V.1. *When $t \to \infty$, the sequence $\left\{\mathcal{X}_{\mathcal{S}}^t\right\}$ converges to a unique fixed point $\mathcal{X}_{\mathcal{S}}^* = (I - P_{\mathcal{SS}})^{-1}P_{\mathcal{SA}}\mathcal{L}_{\mathcal{A}}$, where I is the identity matrix.*

VI. Neighborhood Correlation Term Calculation

Unlike the calculation of the stroke correlation term, the neighborhood correlation term can be estimated in a more straightforward way. The main observation is that the joint probability defined in Equation (2) is intrinsically a Gaussian Process (GP), since (for clarity we omit the constant β)

$$P(\mathcal{X}_{\mathcal{V}}) \propto \exp\left(-\mathcal{X}_{\mathcal{V}}^T(D-W)\mathcal{X}_{\mathcal{V}}\right), \tag{18}$$

where D is a diagonal matrix with $D_{ii} = \sum_j w_{ij}$ and $\Sigma^{-1} = D - W$ is actually the so-called *combinatorial Laplacian*. For GPs, given the observation values $\mathcal{L}_{\mathcal{A}}$ for set $\mathcal{A}$, the conditional distribution of $\mathcal{X}_y$ is also a Gaussian with mean $\mu_{y|\mathcal{A}}$ and variance $\sigma_{y|\mathcal{A}}^2$ which can be estimated as below:

$$\mu_{y|\mathcal{A}} = \mu_y + \Sigma_{y\mathcal{A}}\Sigma_{\mathcal{AA}}^{-1}(\mathcal{L}_{\mathcal{A}} - \mu_{\mathcal{A}}), \tag{19}$$

$$\sigma_{y|\mathcal{A}}^2 = \Sigma_{yy} - \Sigma_{y\mathcal{A}}\Sigma_{\mathcal{AA}}^{-1}\Sigma_{\mathcal{A}y}, \tag{20}$$

where $\Sigma_{y\mathcal{A}}$ denotes the sub-matrix of Σ related to index sets y and $\mathcal{A}$, and $\Sigma_{\mathcal{A}y} = \Sigma_{y\mathcal{A}}^T$. Another useful theoretic result is that in Equation (5) the *differential entropy* of a Gaussian random variable $\mathcal{X}_y$ conditioned on some set of variables $\mathcal{X}_{\mathcal{A}}$ is actually a monotonic function of its variance:

$$H(\mathcal{X}_y|\mathcal{X}_{\mathcal{A}}) = \frac{1}{2}\log\left(2\pi e\sigma_{y|\mathcal{A}}^2\right) = \log\left(\sigma_{y|\mathcal{A}}\right) + \text{const.} \tag{21}$$

Fortunately, it can be directly computed in closed-form from Σ and its sub-matrices. However, in practice computing Σ via inverting $\Sigma^{-1} = D - W$ is numerically forbidden, since the Laplacian matrix $D - W$ is known to have at least one zero eigenvalue, thus it is deficient. A practical trick is to use the regularized Laplacian, i.e. $\Sigma = (D - W + \lambda I)^{-1}$, where λ is a small constant introduced for better numerical stability and I is the identity matrix.

VII. Further Analysis

In this section, we analyze the performance lower bound of the proposed greedy algorithm based on the fact that MI($\mathcal{A}$) belongs to the *submodular function* family, whose definition is given as below:

Definition Set function F on $\mathcal{V}$ is regarded as submodular if:
$\forall \mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$ and $s \in \mathcal{V} \setminus \mathcal{B}$, $F(\mathcal{A} \cup \{s\}) - F(\mathcal{A}) \geq F(\mathcal{B} \cup \{s\}) - F(\mathcal{B})$
or equivalently $\forall \mathcal{A}, \mathcal{B} \subseteq \mathcal{V}$: $F(\mathcal{A}) + F(\mathcal{B}) \geq F(\mathcal{A} \cup \mathcal{B}) + F(\mathcal{A} \cap \mathcal{B})$

Submodular functions are widely used for combinational optimization in economics, sensor placement, feature selection [35]. Intuitively they can be regarded as set functions exhibiting the *diminishing return* property, i.e., the benefit to include one more element will decrease when the selection set becomes larger. Many real-world phenomena show such a property, including the topic discussed in this paper. Based on the submodular function theory, we are able to quantitatively discuss how far the solution is away from the global optima. The following is our main observation:

Theorem VII.1. *The set function induced from mutual information $\mathcal{A} \mapsto$ MI($\mathcal{A}$) is submodular.*

Proof: First note that for $\mathcal{A} \subset \mathcal{V}$ and $y \in \mathcal{S} = \mathcal{V} \setminus \mathcal{A}$, according to Equation 8, MI($\mathcal{A} \cup y$) − MI($\mathcal{A}$) = $H(\mathcal{X}_y|\mathcal{X}_{\mathcal{A}}) - H(\mathcal{X}_y|\mathcal{X}_{\bar{\mathcal{A}}\setminus y})$. For two sets $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$, it is trivial to see that $H(\mathcal{X}_y|\mathcal{X}_{\mathcal{A}}) \geq H(\mathcal{X}_y|\mathcal{X}_{\mathcal{B}})$ and $H(\mathcal{X}_y|\mathcal{X}_{\bar{\mathcal{A}}\setminus y}) \leq H(\mathcal{X}_y|\mathcal{X}_{\bar{\mathcal{B}}\setminus y})$,

where we use the abbreviation $\bar{\mathcal{A}}$, $\bar{\mathcal{B}}$ to denote $\mathcal{V} \setminus \mathcal{A}$ and $\mathcal{V} \setminus \mathcal{B}$ respectively. The following fact holds by piling above observations together:

$$MI(\mathcal{B} \cup y) - MI(\mathcal{B}) \leq MI(\mathcal{A} \cup y) - MI(\mathcal{A}),$$

which demonstrates the diminishing return property of submodular functions, thus the conclusion follows. ■

Moreover, a submodular function is said to be monotonic if $F(y \cup \mathcal{A}) \geq F(\mathcal{A})$ for all $y \notin \mathcal{A}$. Nemhauser proved an important theorem for such functions:

Theorem VII.2 (Nemhauser et al., 1978)**.** *Let F be a monotone submodular set function over a finite ground set $\mathcal{V}$=0. Let $\mathcal{A}$ be the set of the first k elements chosen by the greedy algorithm, and let OPT = arg $\max_{\mathcal{A} \in \mathcal{V}, |\mathcal{A}|=k} F(\mathcal{A})$, then*

$$F(\mathcal{A}) \geq \left(1 - \left(\frac{k-1}{k}\right)^k\right) OPT \geq \left(1 - \frac{1}{e}\right) OPT.$$

Proof: See [39] for detailed proof. ■

Theorem VII.2 is potentially useful to analyze the worst performance, or lower bound of the proposed greedy algorithm. However, the mutual information criterion we adopted is not monotonic for all possible set $\mathcal{A}$. Note that $\text{MI}(0) = \text{MI}(\mathcal{V}) = 0$, the function value will first increase and then decrease. Fortunately, Theorem VII.2 needs not monotonicity over all sets. In fact, our proposed algorithm mainly works on the increasing phase. To see it, note that the increment of the MI value after adding a new element y to the existing labeled set $\mathcal{A}$ is the difference between the SC-term and the NC-term (see Equation (8)). The posteriors of random variables covered by potential suggestive strokes have nearly uniform distributions (i.e., mean value $m_i \approx 0$), which indicates high uncertainty (or large entropy) and thus results in a relatively larger SC-term value compared with NC-term value, thus making the MI value continually increasing. In practice, we can terminate the algorithm if the MI value begins to decrease. In a word, MI is approximately monotonic for our task, which indicates the lower bound of the greedy algorithm is roughly $1-1/e \approx 67\%$ of the global optimal solutions.

VIII. Experiment

We have implemented the proposed active cutout algorithm using Matlab, with optimized core functions written in C++ language. All experiments are conducted on a common PC equipped with Intel Xeon X5472 3GHZ CPU and 32 GB memory.

After users draw the indicating strokes, we calculate the entropy gain for each unlabeled superpixel and select n_k ($n_k = 4$ in most of our settings) optima in current round to construct suggestive strokes according to a *local suppression* strategy: after a superpixel is selected, all elements falling into its neighborhood will have their entropy gains lowered (typically by multiplying an attenuation rate smaller than 1, e.g., 0.95 in our implementation). By this means we reduce the possibility of overlapping suggestive strokes and yield a better spatial distribution.

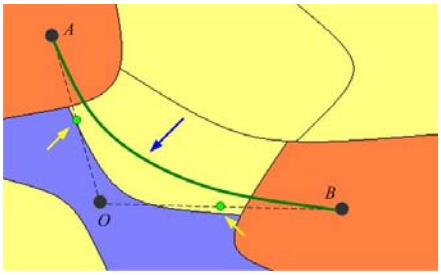


Fig. 3. Bezier curve generation. Yellow arrows point to the middle-points of straight lines connecting two geometrical centers. Blue arrow points to the fitted Bezier.

To make the suggestive strokes visually natural, a piece of Bezier curve is fitted from 4 points as the skeleton of each suggestive stroke. Suppose point O is the geometrical center for a selected superpixel, and A, B are centers for its two most-similar neighbors. A 3^{rd}-order Bezier curve that acts as stroke skeleton can be calculated, as Figure 3 illustrates.

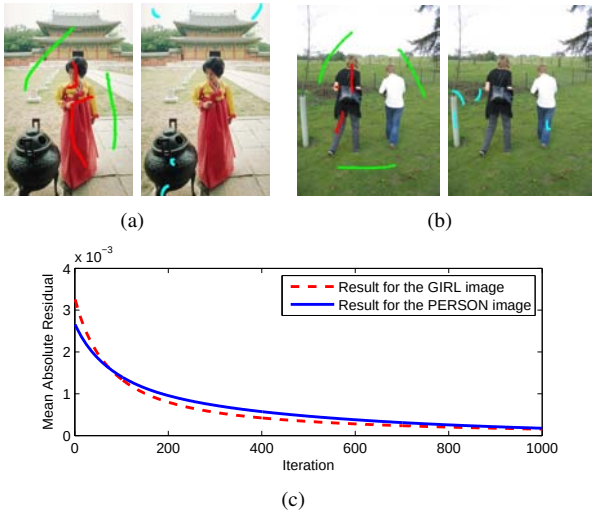


Fig. 4. ActiveCut results for images GIRL (top left) and PERSON (top right). User strokes are either in red or green, while the four most suggestive strokes are drawn in cyan, which capture many ambiguous image regions. In (c), we plot the evolution curves of Mean Absolute Residual (MAR) with respect to iteration counts in SC-term calculation, which consistently drop to zero and thus empirically validate the convergence analysis in Section VII.

With regard to computing speed, when utilizing a 1200-node superpixel lattice (note that it primarily depends on scene complexity, rather than original image resolution), the calculation of SC-term takes $0.1 \sim 0.4$ seconds, primarily proportional to the convergence speed of Algorithm 1 (see Figure 4(c)), while the calculation of NC-term typically takes less than 0.1 seconds. In multiple-round cutout procedure, later rounds always consume less time on SC-term calculation (typically less than 0.1 second) since they can use results of previous rounds for the initialization purpose. Although already able to work at user-interactive rate, however, the implementation of ActiveCut can be further optimized, since Algorithm 1 is highly parallel and thus can be accelerated on current general-purpose graphics hardware (GPGPU) or multi-core CPU.

In Figure 5 we present more results on several images taken

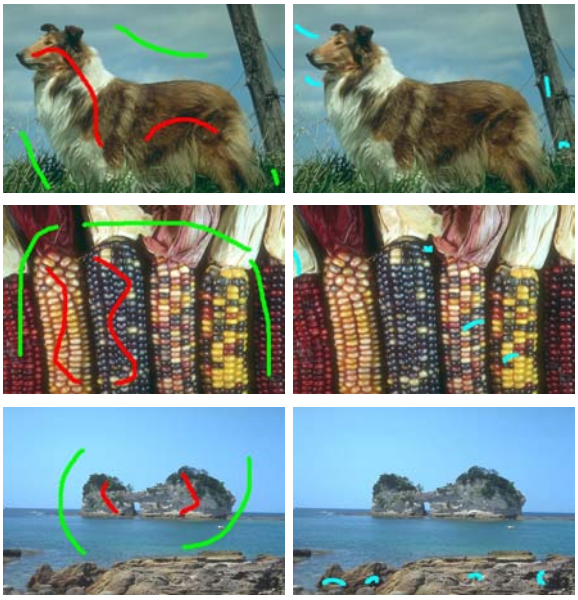


Fig. 5. More examples about suggestive strokes. The left column are images with user strokes, and the right column are corresponding suggestive strokes.

from the publicly available Berkeley image database [40]. In our implementation, the resolution for superpixel lattice is chosen to be either 30×40 (for $\frac{ImageHeight}{ImageWidth} < 1$) or 40×30 (for $\frac{ImageHeight}{ImageWidth} > 1$). Neighborhood parameters $(\Delta x, \Delta y)$ for SC-term adopt the value $(13, 13)$, while for similar neighborhood parameters in NC-term calculation, we use $(7, 7)$.

As stated above, ActiveCut is designated to be a plug-in (rather than an alternative) of existing cutout systems. Users are expected to make their next stroke following the guidance of suggestive strokes, or directly select one of the suggestive strokes. For final segmentation, one has to resort to other cutout algorithms with objective functionals similar to ActiveCut's. Note that in our formulation, ActiveCut works based on color-vector similarity and spatial smoothness, consistent with cutout algorithms like *GrabCut* or *LazySnapping*. In practice we adopt a *GrabCut*-style method, i.e., GMM assumption for underlying color distributions and Ising smoothness prior.

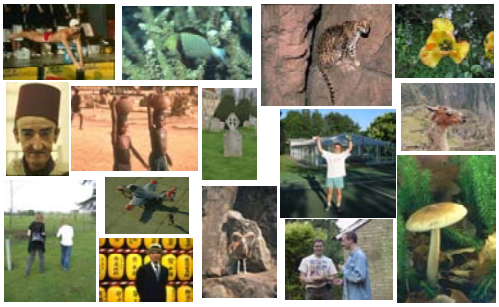


Fig. 6. Sample images used for user study.

To quantitatively investigate the benefits of ActiveCut, we build a data set consisting of 26 images, most of which are selected from the Microsoft Cambridge image segmentation database. Ground truth is obtained via manual labeling. Part of the image set are displayed in Figure 6.

As stated in Section II, the effectiveness of ActiveCut is mainly obvious in the stage of roughly locating the target objects. In the later local retouching stage, users can conveniently refer to intermediate cutout results, which makes the suggestive strokes less useful (thus we turn off its functionality in this stage). According to the above analysis, the experiments mainly focus on the improvement in terms of accuracy from the initial user strokes.

Eight volunteers are selected. During the experiment, subjects are given instructions to extract specific object from each test image. It is convenient for them to change the effect radius of the brush tool (from 3-pixel to 40-pixel) to flexibly handle object with varying widths. Moreover, subjects are told to add new strokes into the system according to the following two criteria: 1) User strokes should be adequate to cut out desired objects. 2) The set of user strokes should contain as little redundancy as possible.

Finally, both user-stroke information and corresponding n_k most salient suggestive strokes are stored for further analysis. The *GrabCut*-style cutout routine is called multiple times to testify the performance with the following three kinds of strokes:

1) Type-I: only user strokes.
2) Type-II: user strokes + suggestive stroke.
3) Type-III: user strokes + random stroke.

In object cutout errors are comprised of two parts: under-segmentation (i.e. classify object pixels as background) error and over-segmentation (i.e. classify background pixels as object) error. Formally, the overall error rate $e = e_u + e_o$, where e_u, e_o are error rates caused by under-segmentation (misclassifying object pixels as background) and over-segmentation (misclassifying background pixels as object) respectively.

Denote the error rate with Type-I strokes as e_1. Since there are n_k distinct suggestive strokes, we run for n_k times independently for Type-II stokes and set the error rate e_2 as the minimum value. For Type-III stokes, we generate 5 groups of random stokes, each of which consists of n_k randomly-generated strokes. For each group, the error rate is calculated in the same manner as Type-II. The final error rate e_3 is computed by averaging the error rate of each group. Since our major concern lies in the relative performance of the other two strokes with respect to Type-I, we further define two ratios:

$$r_2 = \frac{e_2}{e_1}, \quad \text{and} \quad r_3 = \frac{e_3}{e_1}$$

In Figure 7, we present the overall distributions, the statistics (including mean values and standard variations) of r_2, r_3 over image or subject. And some concrete examples can be found in Figure 8. Note that smaller error rate ratios indicate larger improvement. It can be observed that:

1) Both the averaged values of r_2 and r_3 over all images and subjects are below 1.0, while the results obtained by Type-II are better than that of Type-III (0.80 ± 0.24 vs. 0.88 ± 0.17). Moreover, $r_2 \leq 0.8$ in 38% cases (see Figure 7(a)), while only 20.7% of r_3 are below 0.8. It seems that both suggestive strokes and random strokes (with correct labels provided) improve the performance

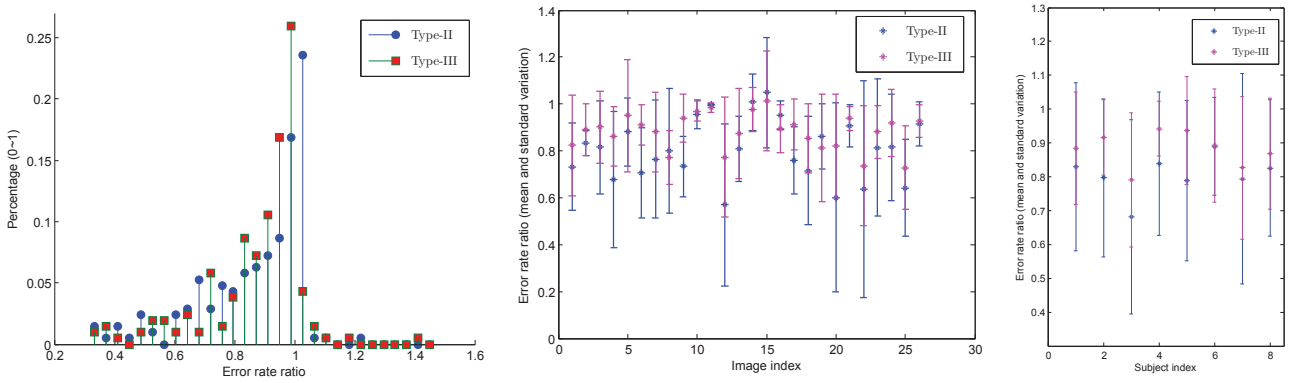


Fig. 7. User study for ActiveCut. **Left**: overall distribution of error rate ratio r_2, r_3 over all subjects and images. **Middle**: statistics of r_2, r_3 (mean and standard variation) over five subjects for each image (26 images in all). **Right**: statistics of r_2, r_3 for each subject.

Fig. 8. Selected cutout results during the user study. For each set of user strokes (top row), four most salient suggestive strokes are displayed (the 2^{nd} row). Cutout results with only user strokes and with user strokes + the best suggestive stroke are presented in the 3^{rd} and 4^{th} rows respectively. Note that the selected suggestive strokes are highlighted in magenta in the 2^{nd} row.

in terms of accuracy, and Type-II is statistically superior to Type-III.

2) The statistics of error rate ratios are highly image-dependent. For extreme difficult images (e.g., LLAMA and PUMA in Figure 6) or easy images (e.g. FLOWER), $r_2 \approx 1$. While for moderately complicated some images (e.g. images #13 and #18 in Figure 7(b)), r_2 have low mean values and small variations.

IX. Conclusion

In this paper we propose a novel interaction paradigm for image object cutout that actively generates suggestive strokes to guide users' further interaction. It can be easily incorporated into most of state-of-the-art image cutout systems as a pre-processing plug-in. The algorithm can perform in interactive rate. Moreover, theoretic analysis based on sub-modular function theory is presented and evaluation on various images demonstrates its effectiveness. For future work, we will work toward two directions: **1)** In this paper we focus on two-dimensional image. However, it also applies for spatio-temporal three-dimensional data. We will conduct experiments for video data in the future work. **2)** The proposed method is general, thus can also be applied to the active learning problem in machine learning. Validating its general performance is also meaningful.

References

[1] C. Rother, V. Kolmogorov, and A. Blake, ""grabcut": interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[2] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, "Lazy snapping." *ACM Trans. Graph.*, vol. 23, no. 3, pp. 303–308, 2004.

[3] J. Wang, P. Bhat, A. Colburn, M. Agrawala, and M. F. Cohen, "Interactive video cutout," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 585–594, 2005.

[4] Y. Li, J. Sun, and H.-Y. Shum, "Video object cut and paste," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 595–600, 2005.

[5] F. Meyer, "Topographic distance and watershed lines," *Signal Process.*, vol. 38, no. 1, pp. 113–125, 1994.

[6] L. A. Vese and T. F. Chan, "A multiphase level set framework for image segmentation using the mumford and shah model," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 271–293, December 2002.

[7] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

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

[9] J. Sun, S.-B. Kang, Z. Xu, X. Tang, and H.-Y. Shum, "Flash cut: Foreground extraction with flash/no-flash image pairs." in *CVPR*, 2007.

[10] C. Rother, T. P. Minka, A. Blake, and V. Kolmogorov, "Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs," in *CVPR*, 2006.

[11] Y. Mu and B. Zhou, "Co-segmentation of image pairs with quadratic global constraint in mrfs," in *ACCV*, 2007.

[12] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother, "Bi-layer segmentation of binocular stereo video." in *CVPR*, 2005.

[13] J. Wang and M. F. Cohen, "Image and video matting: A survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 2, pp. 97–175, 2007.

[14] A. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," in *CVPR*, 2008.

[15] A. Toshev, J. Shi, and K. Daniilidis, "Image matching via saliency region correspondences," in *CVPR*, 2007.

[16] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, 2006.

[17] O. Duchenne, J.-Y. Audibert, R. Keriven, J. Ponce, and F. Segonne, "Segmentation by transduction," in *CVPR*, 2008.

[18] F. Wang, X. Wang, and T. Li, "Efficient label propagation for interactive image segmentation," in *ICMLA*, 2007.

[19] J. Wang, F. Wang, C. Zhang, H. C. Shen, and L. Quan, "Linear neighborhood propagation and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1600–1615, 2009.

[20] J. Cui, Q. Yang, F. Wen, Q. Wu, C. Zhang, L. Van Gool, and X. Tang, "Transductive object cutout," in *CVPR*, 2008.

[21] P. Kohli and P. H. S. Torr, "Measuring uncertainty in graph cut solutions - efficiently computing min-marginal energies using dynamic graph cuts," in *ECCV*, 2006.

[22] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum, "Flash matting," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 772–778, 2006.

[23] S. Chen, L. Cao, J. Liu, and X. Tang, "Iterative map and ml estimations for image segmentation," in *CVPR*, 2007.

[24] X. Ren and J. Malik, "Learning a classification model for segmentation," in *ICCV*, 2003.

[25] V. Sharma and J. W. Davis, "Simultaneous detection and segmentation of pedestrians using top-down and bottom-up processing," in *CVPR*, 2007.

[26] J. Wang, Y. Ying, Y. Guo, and Q. Peng, "Automatic foreground extraction of head shoulder images," in *Computer Graphics International*, 2006.

[27] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[28] X. Ren, C. C. Fowlkes, and J. Malik, "Scale-invariant contour completion using conditional random fields," in *ICCV*, 2005.

[29] S. Z. Li, *Markov random field modeling in computer vision*. London, UK: Springer-Verlag, 1995.

[30] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp, "Image segmentation with a bounding box prior," in *ICCV*, 2009.

[31] O. Veksler, "Star shape prior for graph-cut image segmentation," in *ECCV*, 2008.

[32] S. Bagon, O. Boiman, and M. Irani, "What is a good image segment? a unified approach to segment extraction," in *ECCV*, 2008.

[33] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *Royal Stat*, vol. B-36, no. 2, pp. 192–236, 1974.

[34] Y. Tian, W. Liu, R. Xiao, F. Wen, and X. Tang, "A face annotation framework with partial clustering and interactive labeling," in *CVPR*, 2007.

[35] A. Krause and C. Guestrin, "Near-optimal observation selection using submodular functions," in *AAAI*, 2007.

[36] T. P. Minka, "Expectation propagation for approximate bayesian inference," in *UAI*, 2001.

[37] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[38] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," Carnegie Mellon University, Tech. Rep., 2002.

[39] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Inf. Process. Lett.*, vol. 70, no. 1, pp. 39–45, 1999.

[40] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *ICCV*, 2001.