https://doi.org/10.2352/ISSN.2470-1173.2017.10.IMAWM-167

Interactive Segmentation For Indoor Scenes

Chun-Jung Tai [a]***; Tongyang Liu*** [a]***; Judy Bagchi*** [b]***; Fengqing M. Zhu*** [a]***; Jan P. Allebach*** [a]***;***
[a] ***Purdue University, West Lafayette, IN, U.S.A;*** [b] ***Dzine Steps Inc., Eagle, Idaho, U.S.A***

Abstract

We present a click-based interactive segmentation for indoor scenes, which allows the user to select an object or region within the scene in a few clicks. The goal for the click-based approach is to provide the user with a simple method to reduce the amount of input required for segmentation. We first present an effective global segmentation strategy, which provides a rough separation of different textures. The user, then, places a few clicks to segment the target. A novel Trimap assignment strategy is proposed to utilize the click information. To study the performance of our method, psychophysical experiments were conducted to compare our click-based approach with other existing methods.

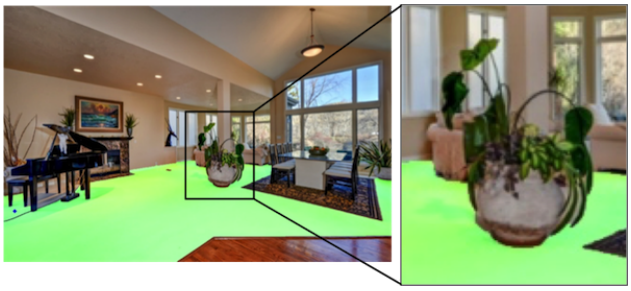
1. Introduction

Interactive segmentation has been widely implemented in the image editing tools. It allows the user to edit a segmentation through some user inputs. An ideal interactive segmentation method allows a user to alter and edit the result as close as possible to their needs.

The early approaches of the interactive segmentation utilize contour tracking in a constraint region [1][2][3]. These algorithms choose a path with the strongest boundary. The users interact with the system by indicating a rough path. This approach, usually, only considers the intensity of contours. To incorporate the color cues, most of the interactive segmentation algorithms utilize the Graph-Cut approach which was proposed by Boykov[4]. In this approaches, the algorithm requires the user to place a hard constraint on the foreground and the background areas, and it treats the user inputs as the empirical cues for segmentation. The cost of cut is then defined as a soft constraint that considers both color cues as well as boundary cues discussed in [5]. Among these kind of strategies, [6] proposes to simplify user inputs by putting a box around the object. A paint-based approach is proposed in [7], in which a user paints both the foreground and the background. An improvement similar to Adobe Photoshop Quick Selection Tool is presented by Liu [8]. It only requires a user to paint the foreground area. Veksler [9] proposed to add a star shape prior to Graph-Cut implementation. Gulshan[10] extended the use of star shape prior to using shape constraint. Grady[11] proposed using Random Walk to solve the graph partition with some known foreground and background information. Bagon [12] proposed to use only one seed point to find the visually meaningful object for nature images. Arvekaez [13] proposed the use of multiple points to label different groups in an image. After this, the segmentation is performed by finding the strongest boundary between two groups using the Ultrametric Contour Map.

Most of these interactive segmentation methods employ different ways to interact with the user. Some commonly seen user operations are placing points along the boundary, placing a box around the objects, clicking on the target, and drawing strokes on the targets and the background. Because of the wide varieties of interactive methods, a new user is expected to go through some tutorials to learn how to use the tool.

(a) Example of clicking based interactive segmentation.



(b) Segmetation result after two clicks are made.

Figure 1. *An Example of Click-based Segmentation. (a) The user places two clicks marked as the plus signs. (b) The highlighted green color shows the segmentation result with our proposed method.*

When we move the image segmentation tool to an online service and switch our targeted users to general users, the users have very low patience to learn how to use the segmentation tool. When we deliver a software-as-a-service (Saas) model [14], the interactive platform must deliver an effective and intuitive segmentation tool. In the internet advertising and feeds ranking community, the time spent by a user on an advertisement is commonly evaluated by a click-through rate[15] and dwell time[16]. We argue that for an online segmentation tool, it is also crucial to attract the users' interests by providing them a good user experience, in order to keep them on the web service. In the other words, the evaluation factors that being used mostly for web advertisement are also relevant to an online interactive segmentation tool. We argue that it is important for an interactive segmentation tool to provide a simple and effective segmentation method with accurate results.

In this paper, we propose to use only clicks to complete the segmentation. This provides the users a very simple instruction of what they are expected to do. The users are allowed to use two types of click, add or remove. A possible operation on a computer is that the users use left clicks to add more selections and use the right clicks to remove the selected regions. To make them to be more patient on the segmentation process, we propose a novel segmentation algorithm that gives a very quick selection for an indoor picture with a specific finishing within a few clicks.

2. Indoor Scene Segmentation

This section proposes a method to quickly make the global segmentation as the starting point of the interactive segmentation. The objective of global segmentation is to replace the clicking and drawing that were required by other interactive segmentation methods. Moreover, an accurate indication of the global segmentation can significantly reduce the number of clicks required from the users. The precise boarder indication is left to the later steps. The goal for the first step is to compose a rough indication of the area with the same material. An effective block-based separation method is proposed in [17]. In the rest of the paper, we will call this method as the global block-based segmentation and shorten it as gBb.

To perform gBb, we divide an image into N blocks. If less blocks are picked, the segmentation would be less accurate. On the other hand, if we use too many blocks, the overall computation would be slow. We test different selection of the block number from 40 images and pick N = 20000 blocks. Figure 2 shows an example of segmentation with different block sizes by using the method we proposed in this section. To measure the similarity between two blocks, we used Kullback-Leibler divergence [18] as the distance measurement of the similarity between the two blocks. The Kullback-Leibler divergence measurement is known as the relative entropy in the information theory, which assumes one of the probability is the "true" distribution, and the other probability is the approximated distribution of the true probability. It is also used as a type of texture cue [19].

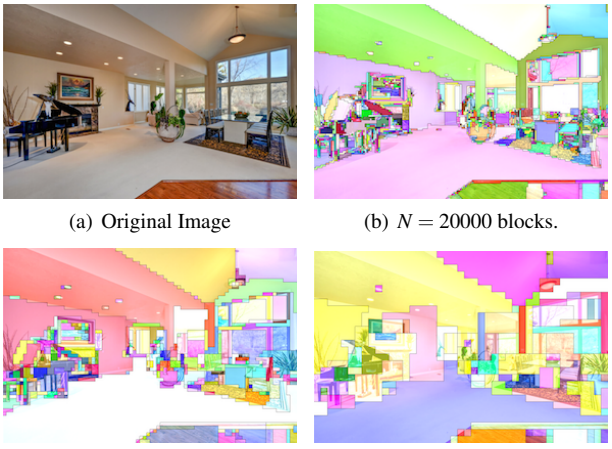
$$KL(B_i||B_j) = \sum_{x\in\mathscr{X}} B_i(x) log \frac{B_i(x)}{B_j(x)}, \tag{1}$$

In Equation 1, B_i and B_j are the probability mass function (pmf) of the two regions. In general, Kullback-Leibler divergence is not symmetric, so $KL(x,y) \neq KL(y,x)$. We define the distance metric as,

$$D(B_i||B_j) = \frac{1}{2}(KL(B_i||B_j) + KL(B_j||B_i)) \tag{2}$$

. To enforce the symmetric, we assume each block is a Gaussian distribution on CIE L*a*b color space with $\theta = (\mu, \Sigma)$, and we also assume each channel is independent. The estimation of $\theta = (\mu, \Sigma)$ for each block is,

$$\begin{aligned} &\hat{\mu} = (\hat{\mu_L}, \hat{\mu_a}, \hat{\mu_b}), \\ &\hat{\Sigma} = diag(\hat{\sigma_L}^2, \hat{\sigma_a}^2, \hat{\sigma_b}^2). \end{aligned} \tag{3}$$



(a) Original Image (b) $N = 20000$ blocks.

(c) $N = 5000$ blocks. (d) $N = 1250$ blocks.

Figure 2. *An Example of Different Selections of the Block Number. (a) is the original image which is an indoor scene. (b) shows the the result of gBb initialized with 20000 blocks. The result is able to separate the carpet, the wall, the ceiling, and the wooden floor. However, the wooden floor tends to break into pieces. (c) shows the the result of gBb using 5000 blocks. (d) shows the the result of gBb using 1250 blocks.*

Equation 1 can be reduced to

$$\begin{aligned} &KL(B_i, B_j) \\ &= \frac{1}{2}(ln\frac{|\hat{\Sigma_i}|}{\hat{\Sigma_j}} + tr(\hat{\Sigma_j}^{-1}\hat{\Sigma_i}) + (\hat{\mu_j} - \hat{\mu_i})^T \hat{\Sigma_j}^{-1}(\hat{\mu_i} - \hat{\mu_j}) - 3) \\ &= \frac{1}{2}\sum_{c=L,a,b}(ln\frac{\hat{\sigma}^2_{c,j}}{\hat{\sigma}^2_{c,i}} + \frac{\hat{\sigma}^2_{c,i}}{\hat{\sigma}^2_{c,j}} + \frac{(\hat{\mu}_{c,i} - \hat{\mu}_{c,j})^2}{\hat{\sigma}^2_{c,j}} - 1). \end{aligned} \tag{4}$$

We compute all the distances between two blocks by using equation 2 with the simplified KL distance by Equation 4.

The next step is to merge the blocks according to the between blocks distance. We first sort all the between blocks distances. The merging procedure gradually combines regions with the smallest distance. After merging two blocks, the distance will be removed from the sorted list. Then, we take the next smallest distance and merge the neighboring block again. The process stops when the smallest distance exceeds a threshold T. The threshold selection should provide enough separation. We set the threshold as $T = max(t_0, p)$, where p is the percentile of all the between block distances, and t_0 is a constant.

Figure 2 is an example of the block-based segmentation with different amount of blocks. The number of the blocks decides how much detail to retain after the separation. In Figure 2, we observe that when choosing $N = 20000$ and $N = 5000$, the amount of details of the image is retained. It is especially obvious at the windows area. We can also observe that the blocks do not connect very well at the piano and the tree area. Most regions which are located near the piano area and the plants area are separated into small pieces. On the other hand, the floor and the ceiling are mostly grouped into one large region. Even though the edge of each region are blocky, the separation between different indoor furnishing materials is reasonable.

3. Main Algorithm

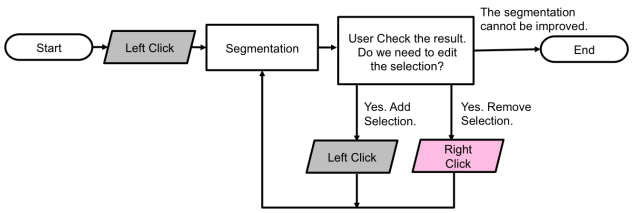


Figure 3. *Flowchart of Clicked-based Interactive Segmentation. The user has two types of operations – the left click to select and the right click to remove.*

In Section 2, we have proposed a global segmentation method to separate an image into a finite number of regions. Without user interaction, the automatic global segmentation cannot be modified. The potential problem for the global segmentation without further editing is that a meaningful target is hard to define. The algorithm tends to (1) over-segment the image into too many pieces, and each region is not a visually meaningful target, or (2) under-segment the image into less regions than desired. The exact definition of what is a correct segmentation is a challenge, and sometimes varies between individuals and applications. We propose an interactive segmentation algorithm to allow users to make further editing. In particular, we propose to reduce the user inputs from drawing strokes, placing points, and putting bounding boxes to only use clicks. Hence, we call this method Click-Cut, shortened as CC.

The steps of system flow are shown in Figure 3. We allow two types of clicks – a left click or a right click. A left click indicates the region that the user wants to select. A right click indicates the region that the user wants to remove from the selection. The first click, by default, is always a left click, from which the user indicates a seed point to select. If one click is not enough, user may place more clicks. In this paper, we will call the left click the foreground click, and the right click the background click. The foreground area represents the place the user want to select, and the background area represents the place to discard.

We organize the rest of sections as follows. First, we initialize the segmentation with a three layer image hierarchy described in Section 3.1. A sequential Trimap assignment is proposed in Section 3.2. Finally, an iterative optimization is applied to search for possible neighborhood at Section 3.3. Algorithm 1 summarizes the process of CC at the end of this section.

3.1 Initialization

This section presents a three layer image hierarchy which preserves different levels of segmentation. The three layers are arranged from fine to coarse. The first layer denoted as L^O is the original image. The second layer denoted as L^S is a super pixel segmentation which we use SLIC[20] with 1000 regions. The third layer L^{gBb} is a global segmentation map that we have introduced in Section 2. Figure 4 shows an example of the three layers hierarchy of an indoor scene.

To represent the three layer of image hierarchy mathematically, We represent second layer as a set of regions denoted as R_i^S, where $i = 1,2,...,1000$, and the third layer as another set of regions denoted as $R_i^{gBb}, i = 1,2,...,N$, where N varied between

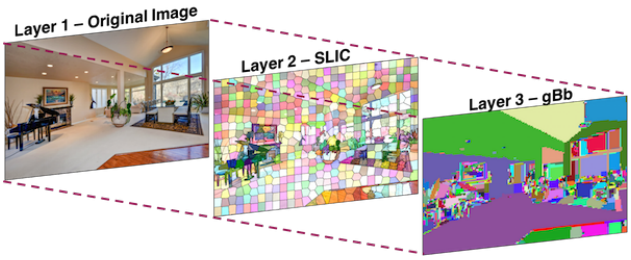


Figure 4. *Three Layer Image Hiearchy. The bottom layer is the original image. The second layer is the superpixel segmentation using SLIC. The third layer is the global block-based segmentation.*

different images. We use $|R|$ to represent the area of a region R, where $|R| = \sum_{x \in R} 1$. To speed up the implementation, the set of regions can be stored as an undirected graph, where each region represents a node and the connected nodes are defined as the regions in its neighborhood.

3.2 Trimap Assignment

This section proposes an algorithm to assign pixels into three categories, the known foreground, the known background and the unknown. We call the output the Trimap T. We represent Trimap T as follows.

$$T = \{F \cup B \cup U\}, \tag{5}$$

where F, B, and U are three mutually exclusive sets which represent the foreground, the background and the unknown set respectively. This section first presents the Trimap assignment for the first click, and extends it to the general case for any clicks.

The First Click

In our application, the first click is default as the selection of foreground. Every pixel in the initial Trimap is set to the background. When the user places the first click, we set the point as a seed point x_s, and take the region contains x_s from layer L^{gBb}. We assume this region is R_s^{gBb}. To make sure the first foreground assignment always includes a good initial size, we also take the region contains x_s from layer L^S into consideration, where x_s guarantees a uniform area size. We define the seed region (R_s) as,

$$R_s = max(|R_s^{gBb}|, |R_s^S|). \tag{6}$$

Equation 6 selects the larger area region from the two layers. The foreground F of Trimap is

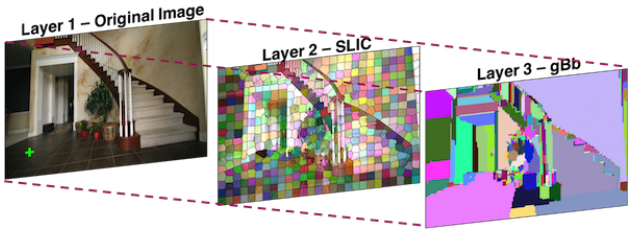
$$F = \{x : x \in \{R_s \ominus K_s\}\}, \tag{7}$$

where K_s is an erosion kernel. Equation 7 shows the mathematical form of the foreground set. The foreground set is basically the eroded seed region. We use two different erosion kernels. If $R_s = R_s^{gBb}$, we use the kernel K_{blk}

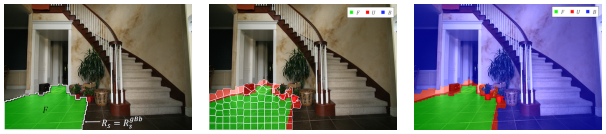
$$K_{blk}[m,n] = \begin{cases} 1 & \text{,where } 1 \le m,n \le 6 \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

If $R_s = R_s^S$, we use the kernel K_{pxl}

$$K_{pxl}[m,n] = \begin{cases} 1 & \text{, if } n = 0 \text{ or } m = 0 \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$



(a) Three Layers Hierarchy



(b) F (foreground) (c) Neighborhood of Seed Region (d) Trimap Assignment

Figure 5. *Demonstration of Initial Trimap Assignment. (a) shows the original image, where the plus sign is the clicked location. (b) applies gBb to form layer L^{gBb}. (c) shows the seed region R_s and the foreground F. (d) shows the dilated seed region highlighted in green and the grid boundary is the neighborhood of seed region selected from layer L^S. Finally, (e) shows the final Trimap assignment for this clicked location.*

where $1 \leq m,n \leq 3$. The erosion is required because the gBb is a blocky segmentation. The edge of the segmentation is not accurate. The erosion allows the Graph-Cut to optimize the segmentation boundary.

$$U = \{x : x \in \cup R_i\}, \text{where } R_i \cap R_s \neq \emptyset\} \cap F^c \tag{10}$$

$$B = \{F \cup U\}^c \tag{11}$$

Equation 10 shows the definition of the unknown set. The underlying idea is, we want to set the neighborhood of seed region to be in the unknown set. However, the neighborhood of L^{gBb} tends to break into small regions at the boundary area which makes the neighborhood ineffective to use. Instead of selecting the regions from L^{gBb}, We take the neighborhood of the foreground set from L^S. The neighborhood is defined as the regions that overlap with F. The rest of the Trimap is set to the background as shown in Equation 11.

Sequential Clicks

For sequential user clicks, the algorithm keeps tracking where has been clicked and what seed regions have been used. We assume the current click as click n and a past click as click j. For the click j, we record (1) the seed locations as ${x_s}^{(j)}$ associated with, (2) the user operation, and (3) the seed region being used. The $c^{(j)}$ represents the click types, where we define a click to select as 0 and a click to remove as 1. We write the sequential inputs as a series of user operations $({x_s}^{(1)}, c^{(1)}, R_s^{(1)}), ({x_s}^{(2)}, c^{(2)}, R_s^{(2)}), ..., ({x_s}^{(n)}, c^{(n)}, R_s^{(n)})$. In addition, for a past click j, the final segmentation separates the unknown set $U^{(j)}$ to a predicted foreground set $\hat{F_U}^{(j)}$ and a predicted background set $\hat{B_U}^{(j)}$. The segmentation result is denoted as $\hat{T}^{(j)} = \{\hat{F}^{(j)} \cup \hat{B}^{(j)}\}$, where $\hat{F}^{(j)} = F^{(j)} \cup \hat{F_U}^{(j)}$ and $\hat{B}^{(j)} = B^{(j)} \cup \hat{B_U}^{(j)}$.

We consider the following five cases:
Case 1 – $R_s^{(n)} \neq R_s^{(j)}$ where $1 \leq j < n$ and $c^{(n)} = 0$,
Case 2 – $R_s^{(n)} \neq R_s^{(j)}$ where $1 \leq j < n$ and $c^{(n)} = 1$,
Case 3 – $R_s^{(n)} = R_s^{(j)}$ where $1 \leq j < n$ and $c^{(n)} = c^{(j)}$,
Case 4 – $R_s^{(n)} = R_s^{(j)}$ where $1 \leq j < n$ and $c^{(n)} \neq c^{(j)}, {x_s}^{(n)} = {x_s}^{(n-1)}$, and
Case 5 – $R_s^{(n)} = R_s^{(j)}$ where $1 \leq j < n$ and $c^{(n)} \neq c^{(j)}, {x_s}^{(n)} \neq {x_s}^{(n-1)}$.
The five cases include all the possible scenarios given the click history.

For Case 1, the user selects a seed region that has not yet been used. The Trimap assignment is straightforward and similar to Equation 7. Let the foreground F be

$$F = \{x : x \in \{R_s^{(n)} \ominus K_s\} \cup \hat{F}^{(n-1)}\} \tag{12}$$

. The foreground is the union of estimated foreground from the $(n-1)^{th}$ click and current seed region. The unknown set is the neighborhood of the foreground region, where

$$U = \{x : x \in \cup R_n\}$$
$$\text{, where } R_n \cap \{\{R_s^{(n)} \cup \hat{F}^{(n-1)}\} \oplus K_{pxl}\} \neq \emptyset\} \cap F^c. \tag{13}$$

The background is set to the rest of the pixels.

Case 1 is similar to the first click. In fact, if we set the the foreground at click zeros as a empty set, namely $\hat{F}^{(0)} = \emptyset$. The Trimap assignment is the same as Case 1. The first click can be combined with Case 1 when we initialize the foreground set as empty.

For Case 2, the user selects a seed region to remove. In this case, we flip this seed region from the foreground to the background and turn the neighborhood of background region to unknown. That is we flip some of the predicted foreground to unknown. The Trimap assignment becomes

$$B = \{x : x \in \{R_s^{(n)} \ominus K_s\} \cup \hat{B}^{(n-1)}\}. \tag{14}$$

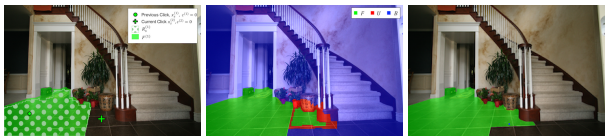
$$U = \{x : x \in \cup R_n\}$$
$$\text{, where } R_n \cap \{\{R_s^{(n)} \cup \hat{B}^{(n-1)}\} \oplus K_{pxl}\} \neq \emptyset\} \cap B^c \tag{15}$$

. The rest of pixels are set as F.

For Case 3, it occurs when the users click the same seed region which is clicked before with the same user operation. Because the Trimap assignment always preserves the seed region to the known foreground or the known background, clicking the same seed region with the same operation cannot change the Graph-Cut result. In this case, the Trimap will not be updated, so the segmentation remains the same. Hence, $\hat{T}^{(n)} = \hat{T}^{(n-1)}$.

For Case 4, it indicates there exists two clicks with different user operation at exactly the same location. In this case, we change the Trimap assignment for this seed region. If user places a foreground click, the Trimap assignment follows Case 1. If the user places a background click, the Trimap changes according to Case 2.

(a) Segmentation Result of the First Click. From left to right, the images shows the location of the first click, the trimap assignment, and the segmentation result of the first click highlighted in green color.



(b) Case 1: The user places a foreground click on a new seed region.

(c) Case 2: The user places a background click on a new seed region.

(d) Case 3: The user places a foreground click at the same place.

(e) Case 4: The user places a background click at the same place.

(f) Case 5: The user places a background click on a seed region that has been selected as left.

Figure 6. *An Example of Trimap Assignment for the Second Click. The example shows the second click followed by the first click showed at (a). Each case from the left to right shows the new seed region, the Trimap assignemnt, and the segmentation result. (a) The first left click selects the tiled floor. (b) A foreground click is placed at a region that doesn't belong to previous seed region highlighted in dot pattern. (c) A background click is placed at a region that doesn't belong to previous seed region highlighted in dot pattern. (d) A foreground click is placed at the same location as a previous background click. (e) A background key is placed at the same location as a previous left click. (f)A background key is placed at a selected seed region which was selected by a left click before.*

For Case 5, it indicates there exists two clicks that are located inside the same seed region with different types of operation. The two clicks contradict each other. In this case, it indicates this seed region may contain both the foreground and the background. Therefore, an in-region partition is required to separate the two seed points into two seed regions. Case 5 occurs when the boundary between two seed regions is weak, or when the colors of the two regions are similar. However, under this case, the region is generally hard to separate by Graph-Cut with few user inputs. In this case, we use the seed region from the superpixel layer, R_s^S. That is, $R_s = |R_s^S|$. Then, the Trimap assignment is made based on the user operation. If the foreground click is placed, Case 1 is applied, and if the background click is placed, Case 2 is applied.

Figure 6 shows 5 different places to click after the first click. We first show the first click at (a). The user can click anywhere for the second click. The place the user clicks may make the segmentation fall into the 5 different cases shown in (b) to (f). We show three images for each case. They are the place of the new click, the Trimap assignment, and the final segmentation result highlighted in green color. The first example at 6(b) shows the case when the user places the second click on a new location which has not been selected. The seed region associates with this location has not yet been used in the past. Similar to Case 1, (b) shows an example when the user places the second click on a new location, where the seed region has not yet been used in the past. However, unlike Case 1, the user places a background click to remove the area. For Case 3 to 5, the new point is placed inside $R_s^{(1)}$. At case 3, the user places the click at the same location as the first click with the same operation. The Trimap assignment doesn't change, and the segmentation result remains the same as the first click. For Case 4, the user clicks at the same location, but with a different user operation. The entire $R_s^{(1)}$ changes to background. For Case 5, the user clicks inside $R_s^{(1)}$ with a different user operation, In this case, we apply the seed region from Layer $2-$ SLIC.

3.3 Graph-Cut

Based on the Trimap, we apply the cost of cut which is proposed in [6]. We define vector A as the segmentation, and A divides $\forall p \in U$ to $\{\hat{F}, \hat{B}\}$. The $C(A)$ is the cost of cut.

$$C(A) = \lambda \cdot R(A) + E(A), \tag{16}$$

where $R(A)$ is the Gaussian mixture model for F and B. We use CIE L*a*b color space and represent each pixel as a three dimensional vector (L,a,b). $E(A)$ is the cost to assign neighboring pixel to different groups. We write the $R(A)$ as

$$R(A) = \sum_{p \in \{\hat{F}, \hat{B}\}} R_p(A_p), \tag{17}$$

where $R_p(A_p)$ is the GMM model built from F and G respectively.

$$R_{\hat{F}}(p \in \hat{F}) = \sum_{k=1}^{5} (-log\ g(p|\theta_{F,k}) - log\ \pi_{F,k}), \tag{18}$$

where $g(p|\theta_{F,k})$ is the k^{th} Gaussian distribution with parameters $\theta_{F,k}$) and $\pi_{F,k}$ is the weight of mixture k for the foreground. Similarly for $p \in \hat{B}$,

$$R_{\hat{B}}(p \in \hat{B}) = \sum_{k=1}^{5} (-log\ g(p|\theta_{B,k}) - log\ \pi_{B,k}). \tag{19}$$

The $E(A)$ represents the potential for being a edge. We apply the edge energy proposed by [6]

$$E(A) = \sum_{(A_i,A_j)\in \mathrm{N}} I(A_i,A_j)\cdot exp(\frac{-||A_i-A_j||^2}{2\cdot E(||A_i-A_j||^2)}), \quad (20)$$

where $(A_i,A_j) \in \mathrm{N}$ is denoted as the set of neighboring pixels and $E(||A_i-A_j||^2)$ is the expectation of difference between neighboring pixels.

3.4 Iterative Optimization

After the first Graph-Cut has been made, the potential foreground may still exist in the background region and vise versa. Hence, we propose an iterative Trimap optimization procedure. The idea is when the user places a foreground click, the algorithm should keep searching for the neighborhood of the foreground until no more foreground can be found. Similarly, this also applies to the case when the user places a background click. We will first discuss the Trimap refinement for the foreground click and the Trimap refinement for the background click follows similar logic.

After Graph-Cut, the unknown group U is separated into the predicted foreground $\hat{F}_U$ and the predicted background $\hat{B}_U$. By hypothesis, this refinement is associated with the foreground click, which indicates that the users want to increase the foreground. We define the converge criterion as every region in layer L^S does not contain more than p percentile of the estimated foreground. That is, the segmentation converges when

$$\forall R_i^S \in L^S, \frac{|\hat{F_U}^{(i)} \cap R_i^S|}{|R_i^S|} < p \quad (21)$$

If the segmentation is not yet converged, the potential foreground may still exist in the current background area. Therefore, we expend our unknown area toward the background. We update the Trimap as,

$$F^{(i)} \leftarrow F^{(i)} \cup \hat{F_U^{(i)}}, \quad (22)$$

$$\begin{aligned} U^{(i)} \leftarrow & \{x : x \in \cup R_i\} \\ & \text{, where } R_i \cap \{F^{(i)} \oplus K_{pxl}\} \neq \emptyset\} \cap (F^{(i)})^c, \end{aligned} \quad (23)$$

and

$$B^{(i)} \leftarrow \{F^{(i)} \cup U^{(i)}\}^c. \quad (24)$$

The rest of the area belongs to $B^{(i)}$. Equation 23 is very similar to Equation 10. A subtle difference is the foreground from Equation 7 is eroded from the seed region. However, the foreground from Equation 22 does not require erosion. Hence, Equation 23 directly takes the neighborhood from the foreground.

Similarly, we can apply this method to the background click. We shrink the foreground and expand the background. The convergence criteria becomes

$$\forall R_i^S \in L^S, \frac{|\hat{B}_U^{(i)} \cap R_i^S|}{|R_i^S|} < p. \quad (25)$$

If the segmentation has not converged, we update the Trimap as

$$B^{(i)} \leftarrow B^{(i)} \cup \hat{B}_U^{(i)}, \quad (26)$$

$$\begin{aligned} U^{(i)} \leftarrow & \{x : x \in \cup R_i\} \\ & \text{, where } R_i \cap \{B^{(i)} \oplus K_{pxl}\} \neq \emptyset\} \cap (B^{(i)})^c, \end{aligned} \quad (27)$$

and

$$F^{(i)} \leftarrow \{B^{(i)} \cup U^{(i)}\}^c. \quad (28)$$

The algorithm continues to refine the segmentation until the segmentation result meets the convergence requirement. Algorithm 1 summarizes the Click-Based Interactive Segmentation Algorithm described in this section.

Algorithm 1 Click-Cut

```
/*Initialization*/
Initialize three layers image hierarchy with L^S, L^gBb, and original image.
Clear the clicked history. Set T̂^(0) ← B
/*Start sequential clicks*/
while true do
    T^(i) ← T̂^(i−1)
    Read the i^th click. Get click point x_s^(i)
    if i = 1 then
        Set c^(1) = 0
    Get seed region R_s^(i).
    Assign Trimap T^(i) based on Case 1 to 5.
    Build GMM model and construct cost function.
    T^(i) ← Graph-Cut output.
    if T^(i) has not converged then
        T^(i) ← updated T̂^(i)
        T̂^(i) ← update GMM and perform Graph-Cut.
    if User satisfies with the result then
        End segmentation
```

4. Experiment Result

We conducted a psychophysical experiment to evaluate the algorithm. We compared our method to four other existing methods shown in Table 1 — the Post-Processing output of Graph-Cut to remove the disconnected foreground[8] (GC); Random Walker[11] (RW); Euclidean Star Convexity (ESC); and Geodesic Star Convexity[10] (GSC). All of these methods allow users to place clicks to complete segmentation. The goal of the experiment was to study the segmentation processes when clicks were used as segmentation inputs. The segmentation objects were selected from 25 different indoor scenes. The scenes included the bedrooms, the living rooms and the kitchens. We selected 50 objects from these images. The objects selections were made in 5 different categories in which a user might be interested in replacing the surface. The 5 categories were walls or ceilings (11 objects); floors (10 objects); cabinets (10 objects); table tops (10 objects); and furniture (9 objects). The procedure for the experiment is detailed below.

The subjects were provided with an explanation on how to use the different segmentation tools and were given time to get familiar with each tool. After this, they were presented with the goal for their segmentation image, which was a binary mask representing the desired segmentation. The subjects were then assigned a segmentation method to segment the image. Their objective was

Table 1: List of Compared Interactive Segmentation Algorithms

Abbreviation	Methods
CC	Click-Cut: The Proposed Algorithm
GC	Post-Processing Output of Graph-Cut
RW	Random Walker
ESC	Euclidean Star Convexity
GSC	Geodesic Star Convexity

to click on the image to get a result similar to the desired segmentation. They were allowed to place sequential clicks in case one click was not adequate. The segmentation process ended when the participant report being satisfied with their segmentation result, or when they had placed more than 20 clicks. Each subject participated in 15 trials that includes using all 5 segmentation methods on 15 different objects. In all, 17 subjects participated in a total of 250 trials that used 50 different objects and 5 different segmentation methods. The entire experiment took every subjects 40 to 60 minutes.

During the experiment, we observed two types of clicking habits for different subjects. Most subjects showed a tendency to click at the center of the desired object. Some tended to click on the corner of the segmentation just like using the polygon tool. Most participants had consistent clicking habits and patterns throughout the experiment. Because of the algorithm constraints, all the algorithms except CC required making the first click on the foreground and the second click on the background in order to produce the initial segmentation. For these algorithms, the participants could click on the background or the foreground third click onwards. The algorithm, CC, only required the participants to make their first click on the foreground, after which they could make any other clicks. The results for the experiment are presented in terms of segmentation accuracy. We compare the final segmentation with its F1 Score.

$$\mathrm{F1} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \tag{29}$$

We performed 50 trials with 5 different segmentation methods. Figure 7 shows how many trials are finished before a certain number of clicks. The result shows that CC requires fewer clicks than any other segmentation method. 31 out of 50 trials for CC utilized 6 or fewer clicks.

Figure 8 shows the convergence of segmentation. The segmentation result is expected to gradually improved when the subject places more clicks. Each line in the plot represents the average segmentation accuracy over 50 trials. Because most trials took 20 clicks or less, if a trial ends before 20 clicks, the segmentation accuracy remains the same from the last click to the twentieth click. The error bar shows the variance of the 50 trials. Our experiment result shows that all the segmentation methods are able to gradually converge, but CC converges the fastest out of the 5 methods. Our method also shows a smoother trend comparing to other methods.

In previous work on interactive segmentation, only the final segmentation result were presented and discussed. However, this does not take the user's experience into consideration. The

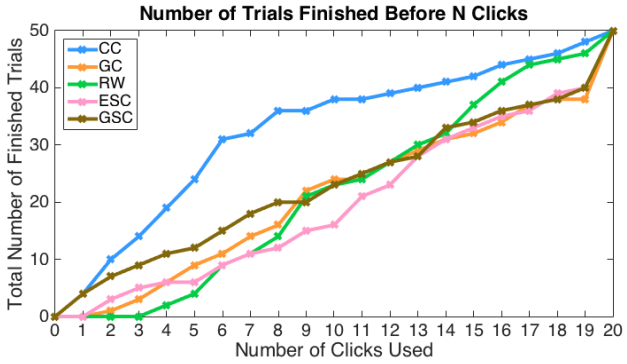


Figure 7. *Count of Trials Finished Before a Number of Click. Each method performs 50 trials. The x-axis shows how many clicks has been performed, and the y-axis counts the number of trials that are able to finished before a certain click number. Our method, CC requires fewer clicks than any other segmentation method. For our methods, 31 out of 50 trials are finished in 6 clicks.*

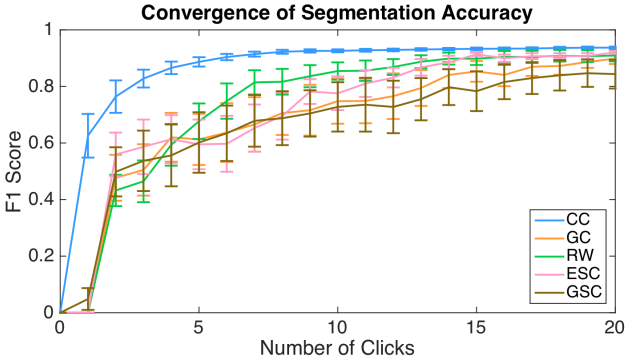
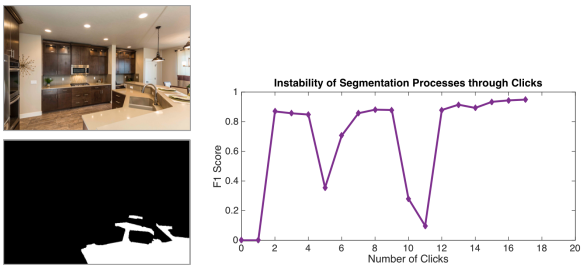
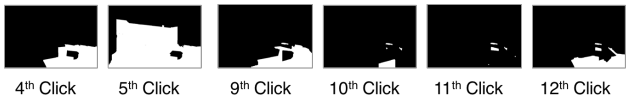


Figure 8. *Convergence of Segmentation Accuracy through Clicks. The plot shows the average segmentation accuracy of 50 trials. The error bar shows the variance of the 50 trials. CC converges the fastest out of the 5 methods.*

user tends to get frustrated when the segmentation result can not be consistently improved. This phenomenon is especially serious when using only click as the segmentation inputs. Most of the algorithms show an instability trend through clicks. Figure 9 shows an example of the segmentation accuracy using the method GSC. The original image and the segmentation goal are represented in (a). The white color represents the desired segmentation, and the black color represents the unwanted region. The segmentation accuracy at each click is reported in (b). A significant decrease at the fifth click, the tenth click, and the eleventh click were spotted. Figure 9 (c) shows the intermediate segmentation result at these clicks. Based on our experiment result, we found the instability is commonly seen at click-based interactive inputs. Figure 10 shows the trials from the object category - Furniture. Each method contains 9 trials. For each plot, every dotted line represents one different trial, and the solid line represents the average segmentation accuracy of the 9 trials. The average smooths out the instability. We propose to measure the instability as accumulated decrease in which we kept tracking the sum of accuracy decrease through clicks. The equation of accumulative decrease ($AccD$) at click i is



(a) Segmentation Goal (b) An Example of Segmentation Accuracy



(c) Examples of Segmentation Result through Clicks

Figure 9. *An Example of the Instability of Segmentation Accuracy. (a) The top image shows the original image, and the bottom image shows the desired segmentation. The white color represents the desired segmentation, and the black color represents the unwanted region. (b) The segmentation accuracy at each click. A significant decrease at the fifth click, the tenth click, and the eleventh click can be spotted. (c) shows the intermediate segmentation result.*

formatted as follows.

$$AccD(i) = \begin{cases} 0 & \text{, if } i \leq 1 \\ \sum_{x=2}^{i} \Delta F1(x) \cdot \mathbb{1}_{\Delta F1(x)<0} & \text{, otherwise} \end{cases} \quad (30)$$

where $\Delta F1(x) = F1(x) - F1(x-1)$, and $\mathbb{1}_{\Delta F1(x)<0}$ is an indicator function.

Figure 11 shows the measurement of accumulative decrease at different segmentation methods. Our algorithm CC shows the segmentation has very small instability comparing to all the other methods. The segmentation accuracy shows the trend of consistent increasing.

5. Conclusion

This paper presents a novel interactive segmentation method that uses only clicks to perform the segmentation for the indoor scenes. We first present an effective global segmentation. Then, a sequential Trimap assignment is proposed to segment the selection. We conducted the psychophysical experiment to study the effectiveness of the segmentation as well as the segmentation accuracy. We propose a novel metric to measure the instability of segmentation output through sequential clicks that causes a bad user experience.

References

[1] Michael Kass, Andrew Witkin, and Demetri Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[2] Eric N. Mortensen and William A. Barrett, "Intelligent scissors for image composition," in *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1995, SIGGRAPH '95, pp. 191–198, ACM.

[3] William A. Barrett and Eric N. Mortensen, "Interactive live-wire boundary extraction," *Medical Image Analysis*, vol. 1, no. 4, pp. 331 – 341, 1997.

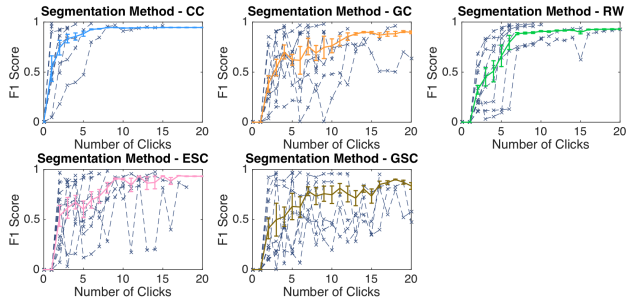


Figure 10. *Trials Performs on Objects with Category - Furniture. Each line represents a segmentation method. The method CC shows very little tendency of accuracy decrease through clicks. The first row from left to right represents the segmentation accuracy using method CC, GC and RW. The second row from left to right represents the method ESC and GSC. Each method includes 9 dotted lines representing 9 trials, and one solid line shows the average of the 9 trials.*

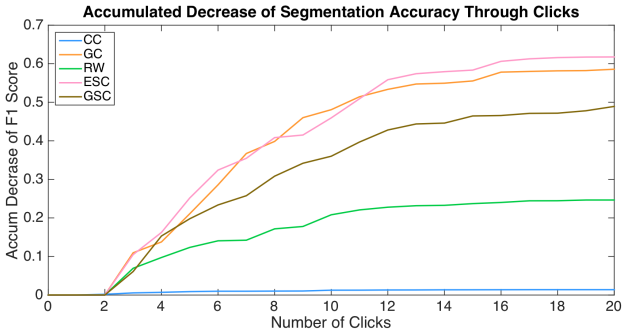


Figure 11. *Average Accumulated Decrease on 50 Objects. Each line represents a segmentation method. The algorithm CC outperforms other algorithms by having very small decrease of accuracy.*

[4] Y. Y. Boykov and M. P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images," in *Proceedings Eighth IEEE International Conference on Computer Vision.*, 2001, vol. 1, pp. 105–112 vol.1.

[5] Vladimir Kolmogorov and Ramin Zabih, "What energy functions can be minimized via graph cuts?," in *Proceedings of the 7th European Conference on Computer Vision-Part III*, London, UK, UK, 2002, ECCV '02, pp. 65–81, Springer-Verlag.

[6] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM SIGGRAPH 2004 Papers*, New York, NY, USA, 2004, SIGGRAPH '04, pp. 309–314, ACM.

[7] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum, "Lazy snapping," in *ACM SIGGRAPH 2004 Papers*, New York, NY, USA, 2004, SIGGRAPH '04, pp. 303–308, ACM.

[8] Jiangyu Liu, Jian Sun, and Heung-Yeung Shum, "Paint selection," in *ACM SIGGRAPH 2009 Papers*, New York, NY, USA, 2009, SIGGRAPH '09, pp. 69:1–69:7, ACM.

[9] Olga Veksler, *Star Shape Prior for Graph-Cut Image Segmentation*, pp. 454–467, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[10] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman, "Geodesic star convexity for interactive image segmentation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 3129–3136.

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

[12] Shai Bagon, Oren Boiman, and Michal Irani, *What Is a Good Image Segment? A Unified Approach to Segment Extraction*, pp. 30–44, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[13] P. Arbelaez and L. Cohen, "Constrained image segmentation from hierarchical boundaries," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1–8.

[14] Mark Turner, David Budgen, and Pearl Brereton, "Turning software into a service," *Computer*, vol. 36, no. 10, pp. 38–44, Oct. 2003.

[15] Rex Briggs and Nigel Hollis, "Advertising on the web: Is there response before click-through?," *Journal of Advertising Research*, vol. 37, no. 2, pp. 33–46, 1997.

[16] Xing Yi, Liangjie Hong, Erheng Zhong, Nanthan Nan Liu, and Suju Rajan, "Beyond clicks: Dwell time for personalization," in *Proceedings of the 8th ACM Conference on Recommender Systems*, New York, NY, USA, 2014, RecSys '14, pp. 113–120, ACM.

[17] Hengzhou Ding, *A semi-automatic framework for text insertion and replacement in natural images*, Ph.D. thesis, PURDUE UNIVERSITY, 2011.

[18] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 03 1951.

[19] Timo Ojala, Matti Pietikinen, and David Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51 – 59, 1996.

[20] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

Author Biography

Chun-Jung Tai received her BS in Electrical and Computer Engineering from the National Chiao Tung University, Hsinchiu, Taiwan (2011). She is currently pursuing her PhD degree in Electrical and Computer Engineering at Purdue University, West Lafayette, IN, USA. She works with DzineSteps on her current research, where she completed the work reported in this paper.

Tongyang Liu is currently a PhD student working under Professor Jan P. Allebach in the School of Electrical and Computer Engineering at Purdue University. His research is focused on color image processing, imaging and printing. He obtained his bachelor's degree from University of Science and Technology of China.

Judy Bagchi is founder and CEO of Dzine Steps, a cloud software provider to home builders and independent design centers nationwide. Prior to this Judy held Research & Development Management roles at Hewlett-Packard and Nortel. She is an industry veteran with more than 20 years of experience in the entire business value chain. She has a keen interest in and has led and participated in various activities supporting Women in Technology.

Fengqing Zhu is an Assistant Professor of Electrical and Computer Engineering at Purdue University, West Lafayette, IN. Dr. Zhu received her Ph.D. in Electrical and Computer Engineering from Purdue University in 2011. Prior to joining Purdue in 2015, she was a Staff Researcher at Huawei Technologies (USA), where she received a Huawei Certification of Recognition for Core Technology Contribution in 2012. Her research interests include Image processing and analysis, video compression, computer vision and computational photography.

Jan P. Allebach is Hewlett-Packard Distinguished Professor of Electrical and Computer Engineering at Purdue University. Allebach is a Fellow of the IEEE, the National Academy of Inventors, the Society for Imaging Science and Technology (IS&T), and SPIE. He was named Electronic Imaging Scientist of the Year by IS&T and SPIE, and was named Honorary Member of IS&T, the highest award that IS&T bestows. He has received the IEEE Daniel E. Noble Award, and is a member of the National Academy of Engineering. He currently serves as an IEEE Signal Processing Society Distinguished Lecturer (2016-2017).