A Comparative Evaluation of Interactive Segmentation Algorithms

Kevin McGuinness, Noel E. O'Connor

Center for Digital Video Processing, CLARITY: Centre for Sensor Web Technologies, Dublin City University, Glasnevin, Dublin 9, Ireland

Abstract

In this paper we present a comparative evaluation of four popular interactive segmentation algorithms. The evaluation was carried out as a series of user-experiments, in which participants were tasked with extracting one hundred objects from a common dataset: twenty-five with each algorithm, constrained within a time limit of two minutes for each object. To facilitate the experiments, a "scribble-driven" segmentation tool was developed to enable interactive image segmentation by simply marking areas of foreground and background with the mouse. As the participants refined and improved their respective segmentations, the corresponding updated segmentation mask was stored along with the elapsed time. We then collected and evaluated each recorded mask against a manually segmented ground-truth, thus allowing us to gauge segmentation accuracy over time. Two benchmarks were used for the evaluation: the well-known Jaccard index for measuring object accuracy, and a new fuzzy metric, proposed in this paper, designed for measuring boundary accuracy. Analysis of the experimental results demonstrates the effectiveness of the suggested measures and provides valuable insights into the performance and characteristics of the evaluated algorithms.

Key words: Image Segmentation, Interactive Segmentation, Objective Evaluation, Subjective Evaluation, Fuzzy Sets, User Experiments

1 Introduction

Image segmentation is a critical component in many machine vision and information retrieval systems. It is typically used to partition images into regions

Email addresses: kevin.mcguinness@eeng.dcu.ie (Kevin McGuinness), oconnorn@eeng.dcu.ie (Noel E. O'Connor).

that are in some sense homogeneous, or have some semantic significance, thus providing subsequent processing stages high-level information about scene structure. The diverse requirements of systems that use segmentation have led to the development of segmentation algorithms that vary widely in both algorithmic approach, and the quality and nature of the segmentation produced. Some applications simply require the image to be divided into coarse homogeneous regions, others require rich semantic objects. For some applications precision is paramount, for others speed and automation.

The importance and utility of image segmentation has resulted in extensive research and numerous proposed approaches, both automatic and interactive. Indeed, so numerous are the proposed techniques that selecting an optimal algorithm for a particular application has become an arduous and time consuming task. Consequently, research into methods for evaluating the quality of image segmentation algorithms has recently been recognized as an important topic. Several such segmentation evaluation benchmarks have been proposed: Ge [1], McGuinness [2], and Jiang [3] (et al.) all discuss systems and benchmarks for supervised evaluation, that is, evaluation with respect to a ground-truth reference, and Zhang et al. [4] review unsupervised evaluation methods. All of these methods, and indeed the vast majority of segmentation evaluation research to date, have however, been focused exclusively on automatic segmentation; comparatively little attention has been dedicated to evaluating interactive segmentation.

Automatic segmentation algorithms are effective solution for applications, such as multimedia indexing and retrieval, that require quick, coarse, region-based segmentation. Some applications, however, require accurate semantic objects. When such objects are necessary fully-automatic segmentation is typically impossible; some high-level information is needed to traverse the "semantic-gap" between homogeneous regions and perceived objects.

Interactive segmentation algorithms [1] provide a solution to this by invoking the aid of a human operator to supply the high-level information needed to detect and extract semantic objects through a series of interactions. Typically, operators mark areas of the image as object or background, and the algorithm updates the segmentation using the new information. By iteratively providing more interactions the user can refine the segmentation. The goal of interactive segmentation is thus to provide a means of accurately extracting semantic objects from an image quickly and accurately.

In this paper we focus on evaluating the performance of interactive segmentation algorithms via extensive user experiments. To this end, we have developed a complete framework for performance evaluation and benchmarking of inter-

[1] Also referred to as semi-supervised or semi-automatic segmentation algorithms

active segmentation algorithms on natural images. The main contributions of the paper are as follows: First, a software platform designed for hosting and evaluating different segmentation algorithms in a uniform environment. The platform currently includes four state-of-the-art interactive segmentation algorithms, and has been made available for public download from our website [2]. Second, a ground-truth dataset created specifically for evaluating interactive segmentation. The dataset consists of 100 objects from natural images with accompanying descriptions, and has also been made available on-line. Third, we propose and investigate two measures appropriate for evaluating interactive segmentation, including a new benchmark specifically designed to measure boundary accuracy against a ground-truth. We compare the suggested measures with other potential measures and demonstrate their effectiveness. Finally, we evaluate and compare four popular interactive segmentation algorithms and demonstrate their performance and characteristics.

The remainder of the paper is organized as follows. In section 2 we review each of the algorithms that were chosen for the evaluation. In section 3 we discuss the measures selected for evaluation, and formulate a new benchmark for measuring object boundary accuracy. In section 4 we discuss the user experiments, including details of the participants involved and the software and tools used. We present the interactive segmentation tool we developed to host the various algorithms, the dataset and ground-truth we used for the experiment, and our experiment setup and deployment strategy. In section 5 we analyze the results of the experiment, validate the selected evaluation measures, and demonstrate which algorithms performed best. Finally, in section 6 we present our conclusions and potential future work.

2 Algorithms

Different segmentation algorithms are often created with different application domains in mind, and thus suited to different tasks. For example, some algorithms, such as active contours [5] and other similar approaches [6], are most effective at extracting regions of interest from medical images. Other algorithms, such as GrabCut [7], are designed for photo-editing applications and extracting objects from photographs of natural scenes. Due to the disparity of intended application, one cannot expect an algorithm designed for, say, biomedical image analysis to be equally effective when applied to a different domain, such as photo-manipulation.

Our evaluation focuses on interactive segmentation techniques appropriate for object extraction from photographs and natural scenes. Specifically, we only

[2] `http://kspace.cdvp.dcu.ie/public/interactive-segmentation/`

Table 1
Algorithmic approaches to interactive segmentation

Method	**Example Algorithm**
Thresholding	Simple gray-scale thresholding
Region Growing	Seeded Region Growing* [8]
Classifiers	Simple Interactive Object Extraction* [9]
Graph and MRF Models	Interactive Graph Cuts* [10]
Hierarchical / Split & Merge	Interactive Segmentation using Binary Partition Trees* [11,12]
Deformable Models	Active Contours (Snakes) [5]

*Algorithms selected for the evaluation

evaluate algorithms whose interactions can be modelled by pictorial input on an image grid [21]; we do not consider interactive segmentation algorithms based on parameter tuning or other forms of interaction. By narrowing our focus thus, we evaluate algorithms that are more directly comparable; the intention being a consistent and fair evaluation, albeit on a smaller subset of the available algorithms.

We chose four algorithms for the evaluation. The algorithms we selected provide good coverage of the various underlying algorithmic approaches used by current methods in the literature for object extraction from natural scenes. Table 1 gives a broad classification of interactive segmentation methods reported in the literature, along with sample algorithms that implement these methods. The asterisked algorithms are the ones we selected for evaluation. Note that we do not consider algorithms based on thresholding or deformable models, as the former cannot be adapted in a straightforward way to pictorial input, and the latter tends to perform better on medical images than on natural scenes. It is also worth noting that there are two other algorithmic approaches to interactive segmentation *not* listed in Table 1: artificial neural nets and atlas guided approaches (see Pham et al. [13]). Their adoption has, however, principally been confined to biomedical image analysis, so we do not consider them further.

In the following subsections, each selected algorithm is reviewed briefly. For further details, readers are directed to the cited publications.

2.1 Seeded Region Growing

The seeded region growing algorithm, proposed by Adams and Bischof [8] is a simple and computationally inexpensive technique for interactive segmen-

tation of images in which the relevant regions are characterized by connected pixels with similar color values. Although it does not have any statistical, optimizational or probabilistic mathematical foundation, and suffers from certain limitations, it has gained popularity due to its speed and simplicity of implementation.

The seeded region growing technique requires as input a set of seed points that have been grouped into n disjoint sets $S = \{S_j : 0 < j \leq n\}$, where n is the number of desired regions in the segmentation. For simple object-background segmentation, as in our case, $n = 2$, giving two sets of seed pixels: S_1 for the object seeds and S_2 for the background seeds. At each step, a single pixel adjacent to the object or background seeds is selected and is added to the corresponding seed set. The pixel is chosen to be the one with minimum distance to the average color of the pixels in S_1 or S_2. The algorithm iterates thus until all pixels have been grouped.

In their paper, Adams and Bischof use simple gray-level differences and averages for computing color distances and means. In our implementation we used the more perceptually uniform CIELUV color space [14] (assuming the D65 reference white as the illuminant), as it was observed to improve performance in our experiments.

2.2 Interactive Graph Cuts

The interactive graph cut algorithm, proposed by Boykov and Jolly in [10], formulates the interactive segmentation problem within a MAP-MRF framework [15], subsequently determining a globally optimal solution using a fast min-cut/max-flow algorithm. Due to the algorithm's speed, stability and strong mathematical foundation, it has become popular and several variants and extensions have been proposed. The "GrabCut" algorithm [7] and the "Lazy Snapping" algorithm [16] are two such variants developed by Microsoft. We used the original algorithm in our experiments.

The algorithm operates by minimizing a cost function that captures both the hard constraints provided by user interactions, and the soft constraints expressing the relationships between pixels in the spatial and range domains of an image. If $L = \{L_p \,|\, p \in \mathcal{P}\}$ is an object-background labelling of an image $\mathcal{P}$ (i.e. a segmentation), the energy of the labelling can be expressed as the cost function:

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

where $D_p()$ is a data penalty function, $V_{p,q}()$ is an interaction potential and $\mathcal{N}$ is the set of all pairs of neighboring pixels. The data penalty function represents a set of hard constraints that control which pixels are required belong

to the object or background, and is derived from the user interactions. The interaction potential is used to encourage spatial coherence between similar neighboring pixels.

To minimize (1), the image and user interactions are combined into a weighted undirected graph. The graph is constructed by adding a node for each pixel in the image, then connecting each node to its neighbor with a weighted edge reflecting the similarity between the pixels (the interaction potential). Each node in the graph is also connected to two special terminal nodes using a weighted edge reflecting the user interactions (the data penalty function). Using this graph, an optimal minimization of (1) can be found efficiently using min-cut/max-flow algorithm described in [17].

Boykov and Jolly also describe a mechanism to encourage disconnected regions with similar grey-level histograms to be automatically combined. We did not use this mechanism in our implementation.

2.3 Simple Interactive Object Extraction

The simple interactive object extraction algorithm, described in [9], uses the pixels marked by the user to build a color model of the object and background regions. It then classifies the pixels in the image as either object or background based on their distance from this model. The algorithm has recently been integrated into the popular imaging program GIMP as the "Foreground Select Tool".

The algorithm assumes a feature space that correlates well with human perception of color distances with respect to the Euclidean metric. As such, the first step in the method is to transform the image color into the CIE-Lab space [14].

Once the image has been transformed into an appropriate color space, the next step is to generate a color signature [18] for the known object and background pixels indicated by the user markup. Using the generated color signatures, represented as a weighted set of cluster centres, the unknown image pixels are then classified as foreground or background according to the minimum distance to any mode in the foreground or background color signatures. The result is a confidence matrix, consisting of a value between zero and one, zero denoting background, one denoting foreground. In the final stage of the algorithm, the confidence matrix is smoothed and regions disconnected from the largest object are removed.

Table 2
The evaluated algorithms and their abbreviated names

Abbreviated Name	**Algorithm**
SRG	Seeded Region Growing
IGC	Interactive Graph Cuts
SIOX	Simple Interactive Object Extraction
BPT	Interactive Segmentation using Binary Partition Trees

2.4 Interactive Segmentation using Binary Partition Trees

The binary partition tree algorithm, initially proposed in [11], and later expanded and improved in [12], transforms a hierarchical region segmentation into an object-background segmentation by using the user interactions to split and merge regions in the tree. The algorithm can be adapted to use any automatic segmentation technique that can be tailored to produce hierarchical output in the form of a binary partition tree, in which the root node represents the entire image, and nodes lower down the tree represent regions at increasing levels of detail, with the leaf nodes being the individual image pixels. In our implementation we use the RSST [19] based algorithm suggested by Adamek et al. in [20].

To transform the tree into an object-background segmentation, the algorithm proceeds as follows. In the first stage, the leaf nodes of the tree are assigned labels according to the pixels marked by the user as object and background. The second stage involves propagating the labels upward toward the root of the tree. Each marked leaf node is propagated toward the root node, labelling each intermediate node with the same label, until a conflict occurs when a parent node has already been labelled differently by the current node's sibling during a previous propagation stage. In this situation, the parent node is marked as conflicting and the algorithm proceeds to the next leaf node. This is repeated for every marked leaf in the tree. In the third stage of the algorithm, each non-conflicting labelled node is visited, and its label propagated to any unlabeled child nodes in the subtree.

At this stage in the algorithm, certain subtrees may yet remain unlabeled, being judged "too different" with respect to the regions defined by the user markup. The original technique for filling these unlabeled regions, proposed in [11], contains a flaw [12]. As an alternative approach [12] proposes labelling each unclassified region with the label of an adjacent but previously classified region. If there are several such regions, the one with the shortest distance is chosen. Adamek suggests using the Euclidean distance between the average colors of the regions in CIELUV space to compute this distance.

3 Evaluation

In this section we will discuss the methods and measures used for the evaluation. To effectively evaluate interactive segmentation, we need to consider three criteria [21]:

Accuracy: the degree to which the delineation of the object corresponds to the truth,
Efficiency: the amount of time or effort required to perform the segmentation, and
Repeatability: the extent to which the same result would be produced over different segmentation sessions when the user has the same intention.

This section is concerned with measuring accuracy; efficiency and repeatability are considered in section 4 and 5. Nevertheless, it is important to note that for interactive segmentation the criteria are highly related. In particular, accuracy and efficiency are interdependent: given more time users can usually produce more accurate segmentations.

Table 2 contains abbreviated names for each of the previously described algorithms that, for brevity, will be used in the subsequent sections.

3.1 Human Factors

Interactive segmentation is sufficiently different from automatic segmentation to warrant a distinct approach to its evaluation. The most important difference between automatic and interactive segmentation algorithms is, of course, that interactive segmentation algorithms require a human operator. The interactions provided by this operator usually have a pronounced affect on the resulting segmentation: good markup is usually needed to find a good segmentation. Clearly this is to be expected - if the interactions did not have such a profound affect on the result, they could be provided automatically, thus eliminating the need for human supervision.

The introduction of this human operator in the segmentation procedure requires several considerations. The nature of the image regions that human operators typically extract is different from those extracted by automatic methods. Humans typically desire more complex and meaningful semantic objects: a tree, a car, a person or a face. Fully automatic algorithms, however, typically only parse images into regions of homogeneous color or texture, which may or may not correspond to a semantic object. Also, for many applications, such as photo-editing, people require very precise objects. For instance, if we wish to replace the background in an image, the segmented boundary of the

object of interest needs to be highly accurate for the effect to be convincing. The required accuracy for this kind of application is higher than that usually required by applications that use automatic segmentation, such as multimedia indexing and retrieval.

The necessity for accurate semantic objects has direct consequences for evaluation. The accuracy requirement means that the measures we use to gauge performance must be sufficiently sensitive to any noticeable variation in object boundary precision. The need for semantic objects means unsupervised evaluation techniques [4] and measures of empirical goodness are inappropriate: the features that characterize good semantic objects are decidedly more difficult to measure without ground-truth than those that characterize good homogeneous regions.

3.2 Evaluation Measures

As unsupervised evaluation techniques are inappropriate for interactive segmentation, we will use supervised evaluation. This necessitates the creation of a ground-truth dataset for the evaluation. Further details of the ground-truth dataset we developed are discussed in section 4.2, however, we will discuss one aspect in this section, as it is pertinent to the evaluation measures we develop. The creation of a pixel accurate ground truth is, in general, impossible for natural images; alpha blending of pixels along the edges of objects make the true border position unattainable with absolute certainty. Our performance measures therefore need to balance the need for sensitivity to border variation with the inherent uncertainty in the boundary pixels of objects in the ground-truth.

Aside from the imprecise nature of the object border pixels, it is also intuitively desirable for any measure we use to penalize a small imprecision near the object border less than, say, a large hole or missing piece of the object. Furthermore, as the objective of interactive segmentation is typically to extract some perceived object from a scene, our evaluation measures should reflect in some sense, the perceived accuracy of the segmentation i.e., there should be a correlation between measured accuracy and perceived accuracy.

Additionally, it is also valuable to have an evaluation measure that is easy to interpret and compare. As such, it is desirable for any measure we use to be appropriately normalized in the interval [0..1]. For consistency, we define all employed measures as similarity functions (performance indicators): values closer to 1 indicate a better segmentation.

(a) (b) (c)

Fig. 1. The internal border pixels of two similar objects (a), (b), and the pixels they have in common (c). The binary Jaccard accuracy measure $\mathcal{A}_B$ is only 0.1. The fuzzy Jaccard measure for the same objects $\tilde{\mathcal{A}}_B$ is 0.85 when the bandwidth parameter $\sigma = 4$.

3.3 Boundary Accuracy

We now develop a means of measuring object boundary accuracy against a ground-truth. Let $\mathbf{v} \in \mathbb{Z}^2$ be any pixel inside the ground-truth object, and $G_{\mathcal{O}} = \{\mathbf{v}\}$ be the set of all of these pixels. Similarly, define $M_{\mathcal{O}}$ to be the set of all pixels in the machine-segmented object. $G_{\mathcal{B}}$ and $M_{\mathcal{B}}$ denote the complements of these sets. Let $\mathcal{N}_{\mathbf{x}}$ be the standard set of 8-neighbors of any $\mathbf{x} \in \mathbb{Z}^2$. The internal border pixels for the ground-truth object are defined as the set B_G, and for the machine-segmentation, the set B_M, as follows:

$$B_G = \{\mathbf{x} : \mathbf{x} \in G_{\mathcal{O}} \wedge \mathcal{N}_{\mathbf{x}} \cap G_{\mathcal{B}} \neq \emptyset\} \tag{2}$$

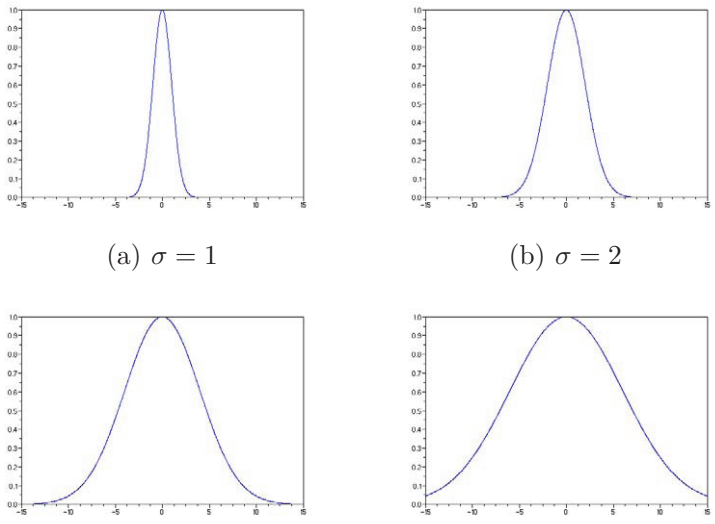
$$B_M = \{\mathbf{x} : \mathbf{x} \in M_{\mathcal{O}} \wedge \mathcal{N}_{\mathbf{x}} \cap M_{\mathcal{B}} \neq \emptyset\} \tag{3}$$

Given the above definition of the border pixels, we could compute a measure of the accuracy of the border pixels as follows:

$$\mathcal{A}_B = \frac{|B_G \cap B_M|}{|B_G \cup B_M|} \tag{4}$$

Note that the value $\mathcal{A}_B$ is equivalent to the well known Jaccard index [1]. Unfortunately, due to the previously discussed ambiguity in the positions of the boundary pixels in the ground-truth, the value of $\mathcal{A}_B$ will typically be excessively low. This is demonstrated in Figure 1. The object borders in 1(a) and 1(b) seem to be reasonably similar. Nevertheless, Figure 1(c) shows that the binary overlap between the pixels is quite small, resulting in a Jaccard index $\mathcal{A}_B = 0.1$. An additional problem is that small imprecisions near the object borders are penalized in equal measure to holes or missing pieces of the object.

To adapt the Jaccard index so that it is more appropriate for our purposes, we need to introduce some tolerance to error near the border pixels. A natural



(a) $\sigma = 1$ (b) $\sigma = 2$

(c) $\sigma = 4$ (d) $\sigma = 6$

Fig. 2. Representation of the fuzzy membership function for different bandwidth parameters $\sigma = \{1, 2, 4, 6\}$

way of accomplishing this is to extend the definition of our sets of border pixels B_G and B_M using fuzzy-set theory [22] so as to capture the intrinsic uncertainty in the edge positions.

Of course, the degree of uncertainty, or tolerance, needs to be specified. Hence, it is necessary to introduce a parameter that quantifies the uncertainty, which we will denote σ. Using this bandwidth parameter, we propose to "fuzzify" the border pixel sets using the following Gaussian form:

$$\tilde{B}_G(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \hat{\mathbf{x}}\|^2}{2\sigma^2}\right) \tag{5}$$

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{y} \in B_G} \|\mathbf{x} - \mathbf{y}\| \tag{6}$$

The fuzzy set for the border of the machine segmentation $\tilde{B}_M$ is similarly defined. The above definition effectively sets $\tilde{B}_G(\mathbf{x}) = 1$ for all $\mathbf{x} \in B_G$ with values decreasing with the Euclidean distance of $\mathbf{x}$ from B_G at a rate controlled by the bandwidth parameter σ. Moreover, the exponential function causes the value of $\tilde{B}_G(\mathbf{x})$ to approach zero for pixels that are a large distance from the border. This effect can be interpreted to mirror the saturation that has been observed in the human visual system: it is easier for us to quantify small errors, but more difficult for larger ones. A representation of the function for different bandwidth parameters is shown in Figure 2.

Given the above fuzzy sets of border pixels $\tilde{B}_G$ and $\tilde{B}_M$, we can reformulate the Jaccard index using fuzzy set theory as follows:

$$\tilde{\mathcal{A}}_B = \frac{\sum_{\mathbf{x}} \min(\tilde{B}_G(\mathbf{x}), \tilde{B}_M(\mathbf{x}))}{\sum_{\mathbf{x}} \max(\tilde{B}_G(\mathbf{x}), \tilde{B}_M(\mathbf{x}))} \tag{7}$$

The above formulation is already normalized in the desired range [0..1], and takes the value 1 only for an exact match. Like the binary Jaccard index, the measure is symmetric, however, in contrast to the binary set formulation, close matches are now penalized proportional to the bandwidth parameter σ. Also as sigma approaches zero, $\tilde{\mathcal{A}}_B$ approaches the binary Jaccard index.

3.4 Object Accuracy

When considering the entire region accuracy, as opposed to the accuracy of the border, it is less important to "fuzzify" the evaluated sets. For regions, small inaccuracies around the border tend to be offset by larger overlapping areas, whereas for borders, the sets, even those very nearby spatially, may not strictly overlap at all. As such, we employ the previously described binary Jaccard index to measure the object accuracy. This is consistent with our border accuracy measure, and also has the advantage of allowing the results presented herein to be directly comparable with previous work in segmentation evaluation, such as [1,3,24]. The object accuracy measure is given by:

$$\mathcal{A}_O = \frac{|G_O \cap M_O|}{|G_O \cup M_O|} \tag{8}$$

3.5 Choosing Sigma

The fuzzy boundary accuracy measure requires appropriate selection of a bandwidth parameter σ to regulate its sensitivity to error. The parameter should be chosen to reflect the degree of uncertainty of the object border pixels in the ground-truth. If the bandwidth parameter is too small, the measure becomes over-sensitive to inaccuracies in the ground-truth, and will not reflect the perceived border accuracy. If the parameter is too large, the measure will not be sensitive enough to capture noticeable differences in precision.

For our experiments we chose a bandwidth parameter $\sigma = 4$. Using this value, pixels with a Euclidean distance less than 3 from the boundary are considered over 75% inside the boundary set, and pixels with a distance greater than 8

Table 3
Evaluation measures and their symbols

$\mathcal{A}_O$	Object accuracy (Jaccard index)
$\tilde{\mathcal{A}}_B$	Boundary accuracy (Fuzzy Jaccard index on border pixels)
$\mathcal{A}_B$	Binary boundary accuracy (Binary Jaccard index on border pixels)
$\mathcal{P}r$	Precision
$\mathcal{R}e$	Recall
$\mathcal{RI}$	Rand Index

are less than 15% inside. The value was chosen empirically, based on a simple experiment. In the experiment, two different segmentations of the same object were chosen, one with a higher perceived accuracy than the other. For the two segmentations, the fuzzy boundary accuracy measure was computed using increasing values of sigma. From the resulting series, σ was chosen such that the difference between the computed values was consistent with the perceived difference in accuracy. The experiment was repeated several times with the value 4 giving the most consistent result.

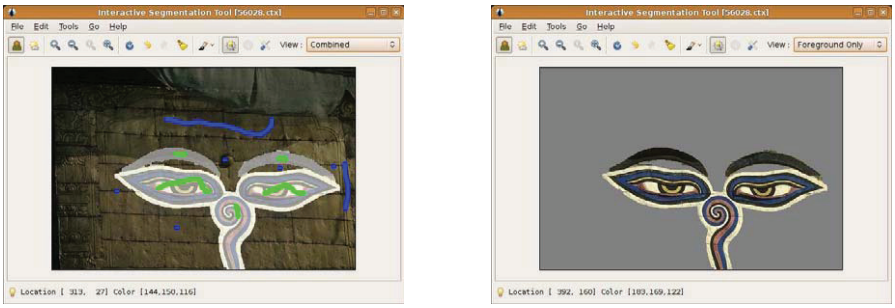
3.6 Other Measures

To validate the effectiveness of the selected measures, we also computed some other popular measures for comparison, including precision, recall and the Rand index [25]. The computed evaluation measures and the corresponding symbols that will be used in the remainder of the text are shown in Table 3.

4 Experiment

In this section we discuss the evaluation experiment, detailing information about the participants involved, the software and ground-truth used, and the experiment setup and deployment strategy. To create an effective experiment plan, we will refer again to the three evaluation criteria from the beginning of section 3: accuracy, efficiency, and repeatability; all three have repercussions for the experiment setup.

To effectively measure accuracy, the ground-truth must be as precise as possible; errors in the ground-truth directly affect the accuracy benchmarks. To effectively measure efficiency, changes to the segmentation need to be recorded as new refinements are added by the user over time. Accuracy and time are dependent; accuracy needs to be viewed as a function of time. Furthermore,



(a) A combined view of the image, markup, and the segmented object. The segmentation mask is overlaid semi-transparently.

(b) A view displaying the segmented object only; the background region (gray) is suppressed.

Fig. 3. Screenshots of the Interactive Segmentation Tool (on the Linux platform)

it is prudent to prevent users spending *too* much time refining a segmentation. This is justified: the primary purpose of interactive segmentation is to provide an accurate segmentation faster than it would take to produce it by hand. To effectively measure repeatability, we need to ensure we have a sufficient number of participants; if we use enough participants to segment each image several times, then on average algorithms with good repeatability will benchmark higher than will algorithms with poor repeatability.

4.1 Software

It is important to provide a single user interface with consistent capabilities for the experiment, allowing participants to segment the relevant objects in a uniform way using different algorithms. To this end, we developed a standalone scribble-based interactive segmentation application. The tool supports any segmentation technique that can be adapted to use a scribble driven interaction paradigm for providing iterative updates. All four algorithms discussed in section 2 are fully integrated. Screenshots of the tool are shown in Figure 3.

To extract and object from an image, users mark foreground pixels using the left mouse-button, and background pixels using the right mouse button, or by using the left-button while depressing the *Ctrl* key. As each interaction is provided the corresponding segmentation mask is updated. The segmentation can be visualized within the tool in various ways, including: a hybrid-view showing the segmentation mask transparently overlaid on the image, a view showing the object borders, and a view of the object with background elements removed.

The tool itself was developed as a general purpose application – we envisioned its utility would go beyond the experiment described in this paper. To sup-

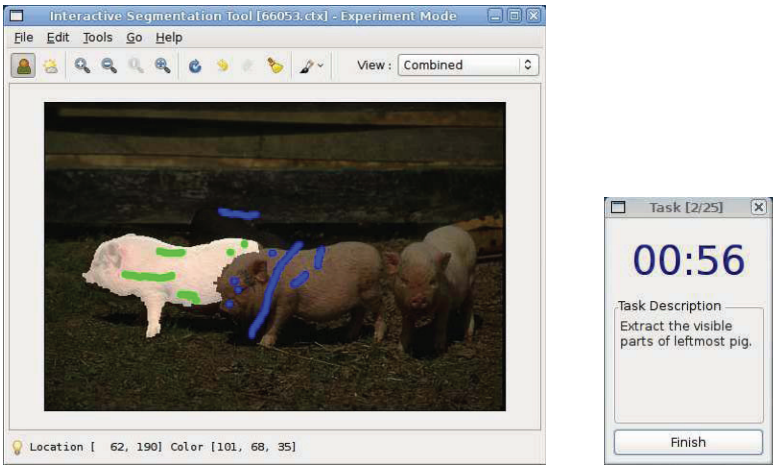


Fig. 4. Screenshot of the interactive segmentation tool in experiment mode

port the constraints of the experiment an "experiment mode" was, however, included. In this mode the relevant algorithm is selected and locked automatically for the participant. The participant is shown an image and a short description of the object they are required to extract. When the participant clicks on *Start*, a timer begins a countdown, giving the user a finite period to extract the object as best they can using the current algorithm. The tool stores each segmentation mask and a corresponding timestamp as new refinements are added, forming a progressive collection of segmentations over time. When the user finishes, or the time elapses, the next image and description are displayed. The process repeats until the experiment is completed. Figure 4 shows an example of the application in experiment mode.

In addition to the base functionality required for the experiment, we also considered it important in a realistic evaluation to provide features that are typically found in other modern graphics packages. As such, several other features were included, including zooming, undo/redo support, and altering the markup brush size.

The interactive segmentation tool, complete with the four algorithms described in the paper, is available for public download from our website. It is compatible with Linux, Windows and Mac OS X.

4.2 Ground-Truth

The images we used to compile the dataset for the experiments were taken from the publicly available Berkeley Segmentation Dataset [23]. The compiled dataset consists of 100 distinct objects selected from 96 of the 300 images in the Berkeley set. These images were chosen so that each image had one

Table 4
Experiment variants with ground-truth set and algorithm assignments.

Variant	*Ground-Truth Set*			
A	S_1	S_2	S_3	S_4
B	S_2	S_3	S_4	S_1
C	S_3	S_4	S_1	S_2
D	S_4	S_1	S_2	S_3
Algorithm	A_1	A_2	A_3	A_4

i.e., variant B uses ground-truth set S_2 with algorithm A_1, ground-truth set S_3 with algorithm A_2, S_4 with A_3, and S_1 with A_4.

or more objects that could be unambiguously described to participants for extraction. Care was also taken to select images that were representative of a large variety of segmentation challenges, such as texture, camouflage and various lighting conditions.

To ensure the highest possible accuracy, the ground-truth was created entirely by hand; no semi-automatic technique was used. This was also important to avoid potential bias to any algorithmic facet of the procedure used to create it. The object extraction was performed by marking pixels on the object border using a graphics tablet, and subsequently filling the object interior. The result is a series of binary masks, one for each object in the dataset, where zero valued pixels denote the background and non-zero valued pixels denote the object.

As noted in section 3, creating a 100% pixel accurate ground-truth is, in general, impossible, due to the ambiguity in the true positions of the border pixels. It is necessary, however, when creating a binary ground-truth to decide which pixels belong to the object and which pixels belong to the background. To handle this ambiguity in the object border pixels, a simple heuristic was applied: retain pixels that appear to contain some of the objects color along the object border, and that do not appear to be image compression artifacts. This heuristic was chosen so that each pixel along the border would be, on average half-inside and half-outside the the true form of the foreground object.

Each object mask was annotated with a description of the object in the image to which it relates. The full ground-truth dataset, including object masks and descriptions, is publicly available for download from our website.

(a) Task Image (b) Expected object

Fig. 5. Sample task image and the expected object. The task description is *"Extract the person, hat and bucket from the background."*

4.3 Setup

A total of 20 volunteers participated in the experiment. Most of the participants were computer science or engineering graduates. Some of the participants were familiar with image processing and information retrieval techniques, however, none had any particular expertise in interactive segmentation. Each participant was given a user guide and sufficient time to familiarize themselves and become proficient with the software that would be used for the experiment. Sample images were provided for training, however, participants were not given access to the experiment dataset.

We considered it overly demanding to ask each participant to extract the entire set of 100 objects using all 4 segmentation algorithms. We therefore divided the ground-truth randomly into 4 equally sized sets $\{S_1, S_2, S_3, S_4\}$ each containing 25 tasks. Each participant was given the task of segmenting the sets using a different algorithm for each set, resulting in a total of 100 tasks (as opposed to 400). Denoting the algorithms $\{A_1, A_2, A_3, A_4\}$, this gives 4 experiment variants, as shown in Table 4.

By distributing experiment variants to participants equally, we ensure that every image is segmented at least 5 times by each algorithm. Thus, we can minimize the affect of an individual's markup skills and other human influenced variation by computing the average of the resulting benchmarks across segmentations of the same image with the same algorithm by different users. Repeatability is therefore implicitly evaluated: if a good segmentation is not repeatable by multiple users, the average evaluation measure will be lower.

The experiment proceeds as follows. Each task is presented to the user in the form of an image and task description. The image, of course, contains the relevant object, and the task description expresses as unambiguously as possible the part of the image to be extracted. Figure 5 presents a typical task, the corresponding description, and the expected object. Users are required to study the image and description and when ready, click on a *Start* button, and begin

to extract the object as accurately as they can by marking areas of the image as foreground or background with the mouse. Since it is possible to achieve near perfect accuracy by manually segmenting an object (i.e. without the aid of interactive segmentation algorithms) when given an arbitrary amount of time, the usefulness of an interactive segmentation algorithm is in its ability to create a reasonably accurate segmentation in a significantly shorter timespan. For this reason, and to prevent some participants expending much more effort in improving their final segmentation than others, it is important to impose a reasonable time limit. We therefore restrict users to a maximum of 2 minutes per object. They may, however, proceed to the next task earlier if satisfied with their segmentation.

After the participant has finished extracting an object, they are asked to fill out a short questionnaire. The questionnaire was designed to coarsely assess, in subjective terms, how difficult the users found the segmentation, what they considered to be the primary causes of any difficulties, and how accurate they perceived their final segmentation to be. Users are asked to rate how difficult they considered the task on a scale of 1 to 5, rate how accurate they considered their segmentation on a scale of 1 to 5, and to check a series of boxes indicating what they perceived to be the primary causes of any difficulty encountered. These checkboxes corresponded to low-level image features, such as color, texture, and object size.

When the entire set of 25 objects are extracted, participants are requested to rate the segmentation algorithm that they just used, again on a scale of 1 to 5. Once completed, the software automatically selects the next algorithm and participants proceed to extracting the next 25 objects. The experiment continues thus until all objects are extracted.

4.4 Deployment

The experiments were carried out by each user independently, and in their own time. Experiments took about 3 hours each to complete. Participants were permitted to take breaks between tasks: a continuous sitting was not required.

To ease deployment of the experiment, and efficiently collect the results, a deployment tool was created. When executed the tool prepares the user's system for the experiment as follows:

(1) Information identifying the participant is collected.
(2) The image and ground truth data files are automatically downloaded from a central server. These are placed in a known location on the participant's machine.

Table 5
Overall average boundary accuracy and object accuracy

	Boundary Accuracy $\tilde{\mathcal{A}}_B$		Object Accuracy $\mathcal{A}_O$	
Algorithm	**Best**	**Final**	**Best**	**Final**
BPT	0.78	0.78	0.93	0.92
IGC	0.78	0.77	0.93	0.92
SRG	0.70	0.70	0.88	0.88
SIOX	0.64	0.64	0.85	0.85

(3) The deployment tool contacts a web-service running on the server, which assigns a particular experiment variant to the user. The web service maintains a database of participant-variant pairs, and assigns the variant using a round-robin system, to ensure equal coverage of each of the 4 task sets with all corresponding algorithms.
(4) Experiment files compatible with the segmentation tool are generated and the user is instructed to begin the experiment.

The deployment tool also displays the relevant questionnaire pages to the user at each stage of the experiment, and stores the answers. When the experiment is complete, all data generated by the segmentation tool and the deployment tool is automatically compressed and uploaded to the server for analysis.

5 Results & Analysis

All 20 participants completed the experiment in full, resulting in over 40 000 segmentation masks being collected for evaluation. In this section we present the results of the evaluation, and discuss their implications. To give a high-level idea of the accuracy and efficiency of the algorithms, we first describe the overall average accuracy (with respect to the measures discussed in section 3) and the overall average time required to perform the segmentation with each algorithm. We then present average accuracy as a function of time, to attain a better understanding of the characteristics of each algorithm. Next we discuss perceived accuracy, as specified by participants in the questionnaires, and its significance. Finally, we show the correlations between the computed evaluation benchmarks and perceived accuracy.

Table 6
Average time required for users to achieve their best accuracies and average total time used to complete a task (seconds).

Algorithm	**Best $\tilde{\mathcal{A}}_B$**	**Best $\mathcal{A}_O$**	**Final/Total**
BPT	59.76	59.09	64.25
IGC	62.93	62.53	66.43
SIOX	69.88	68.90	73.08
SRG	80.77	80.73	85.32

5.1 Object & Border Accuracy

Using the object and boundary accuracy measures discussed in section 3, for each algorithm evaluated we measured:

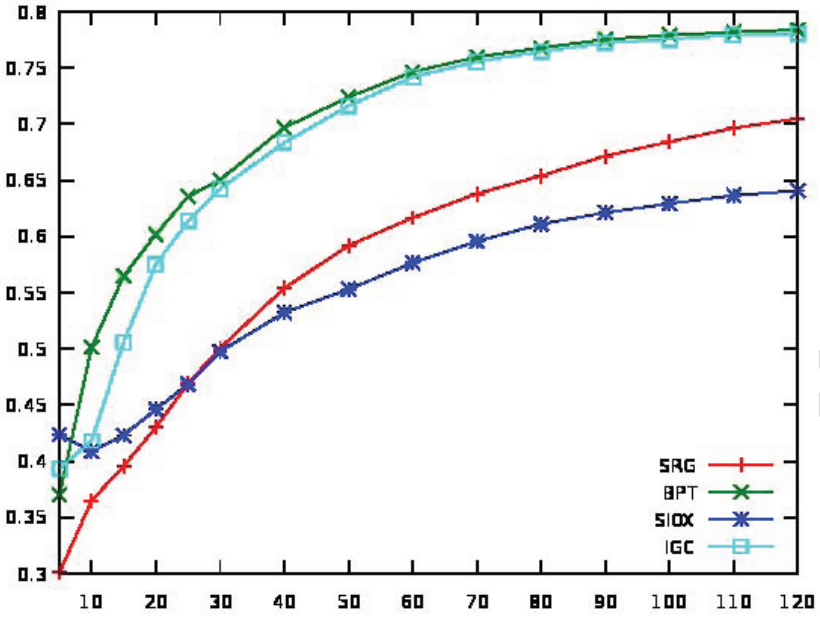
- The average final segmentation accuracy: the object and boundary accuracy measured when the participant was finished the segmentation or the allocated time elapsed, averaged over all objects from the same segmentation algorithm.
- The average best segmentation accuracy: the best object and boundary accuracy achieved per object, averaged over all objects from the same segmentation algorithm.

The resulting values are shown in Table 5. From the Table it is clear that the best performing algorithms, in terms of measured accuracy, are the BPT and IGC algorithms, which perform equally well on average. The SIOX algorithm is the poorest; this is perhaps due to the difficulty, noted by some participants in the questionnaires, of producing any reasonably accurate segmentation for some images in the dataset.

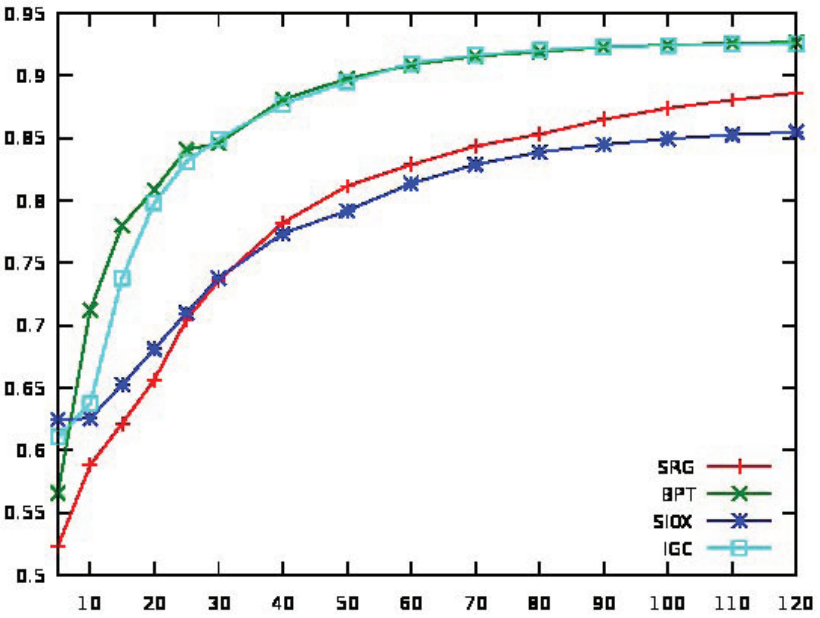
In addition to accuracy, it is also critical to measure time when evaluating interactive segmentation: given enough time arbitrary precision can be achieved manually. Table 6 shows, for each algorithm, the average time required until a user attains their best object and boundary accuracy for an image, and the average total time spent per image. From this, we can see that users spent the least amount of time with the BPT algorithm, and the most with the SRG algorithm.

The times given in Table 6 are, however, likely achieved at varying accuracies for each individual algorithm. Thus, the table only gives an overview of the typical time required to achieve the best possible result with each algorithm. By observing the average measured accuracy across time for each algorithm, we can get a better idea of the time-accuracy characteristics of each of the algorithms. Since time and precision are dependent, this provides the most

useful illustration of an algorithms performance. Figure 6 demonstrates the relationship for the each of the four algorithms.

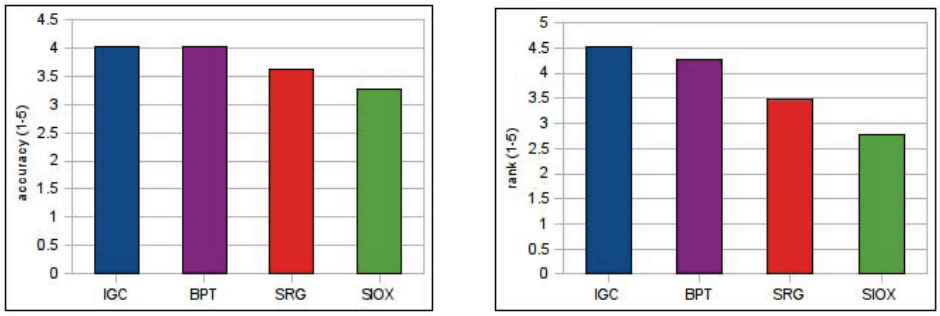


(a) boundary accuracy / time



(b) object accuracy / time

Fig. 6. Average accuracy displayed over time



(a) Average user perceived accuracy

(b) Average algorithm rank given by participants

Fig. 7. Results of user feedback

Several observations can be drawn from the figure. We see that the SIOX algorithm gives the best initial segmentation. After 5-10 seconds however, the BPT and IGC algorithms surpass the performance of SIOX, obtaining a wide margin of improved accuracy after 30 seconds. The SRG algorithm performs poorest initially, but also obtains better accuracy than SIOX after approximately 40 seconds. This implies that although SIOX gives a superior initial guess, it is one of the least responsive algorithms: it tends to inhibit iterative improvement.

The BPT and IGC algorithms have comparable performance throughout. The IGC algorithm gives a better initial guess, but the BPT algorithm outperforms it marginally thereafter for a term; after approximately 50 seconds the difference in average precision between the two is negligible.

It is also worth noting that the two accuracy measures are well correlated. The Pearson correlation coefficient for the two measures, computed over all recorded measurements, is 0.834. The measures also demonstrate high rank correlation: Spearman's ρ coefficient over all recorded measurements is 0.823.

5.2 Perceived Accuracy

To measure perceived accuracy, participants were asked to rank how accurate they perceived their final segmentation on a scale of 1 to 5: 5 meaning highly accurate, and 1 meaning highly inaccurate. Figure 7(a) shows the results of the survey. The results agree with the average measured accuracies from Table 5. We also asked users to rank each of the algorithms on a scale of 1 to 5, again higher ranks indicating better performance. The resulting average ranking is shown in Figure 7(b).

Interestingly, more users preferred the IGC algorithm to the BPT algorithm, despite the observation that the IGC algorithm slightly underperforms the

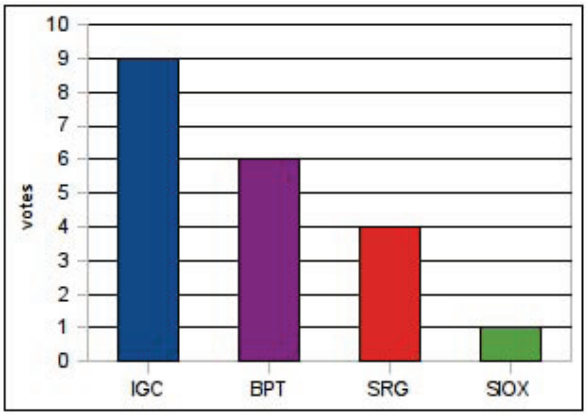


Fig. 8. Preferred algorithm as voted by users

BPT algorithm for the time period shown in Figure 6. Potential reasons for this are explored in the next section.

5.3 User Feedback

In addition to asking participants to rank the the accuracy of their final segmentations in the questionnaires, we also asked participants to comment on each of the evaluated algorithms. This included asking participants: which algorithm they preferred, what they believed were the strengths and weaknesses of each, and if they had any other general remarks or comments. Figure 8 shows the distribution of votes given by users for their preferred algorithm.

By observing Figure 7(b) and Figure 8, it is clear that most users preferred the IGC algorithm, despite their comparable performance in terms of their time accuracy profiles (Figure 6). Analysis of the user comments revealed an interesting explanation for the discrepancy – the algorithms *behavioral predictability*. The IGC algorithm tends to behave more conservatively than BPT: additional interactions tend to produce small predictable changes, whereas larger more unpredictable changes can sometimes occur with BPT. This gives the BPT algorithm the potential to improve its segmentation faster than IGC, but may also induce the perception of erratic behavior.

From the comments it was clear that participants strongly preferred more conservative algorithms. For the IGC algorithm, users remarked that the algorithm "reacted well to local changes, without causing too much global deformation." They liked that "small localized scribbles only have a local effect." Conversely, users disliked algorithms in which small additions to the markup could cause large differences to the segmentation. Commenting on the SRG algorithm one user complained that "adding one scribble can completely change the segmentation." This apparent erratic behavior was also noted by partic-

ipants with regard to the BPT and SIOX algorithm, and is likely the reason why many participants preferred the IGC algorithm to the BPT algorithm.

Another issue commonly indicated as important in the feedback was *algorithm responsiveness*. Participants, in general, disliked algorithms that made it difficult for them to refine their segmentation. This was the most common reason that users cited for disliking the SIOX algorithm: although it was sometimes "very quick to capture initial object," "if it doesn't find the correct boundary in the beginning, then it is simply impossible to refine." The comments revealed that many users become quickly frustrated with algorithms that make it difficult to add iterative refinements to their segmentation. This is further reinforced by comparing the time spent on each task with the rankings given: users prefer using algorithms that require longer to segment an object, but allow iterative improvements (SRG), than using algorithms that make a better initial guess, but make improvement more difficult (SIOX).

Similar observations have also been made when evaluating other interactive systems. Koenemann and Belkin [26] showed that users perform better when using information retrieval systems if they understand the underlying relevance-feedback mechanism. They also point out that users subjectively preferred more transparent systems. This is related to behavioral predictability - systems that are easier to understand are easier to predict. As a design principal for creating semi-automatic annotation interfaces, Suh and Bederson [27] propose that users should be in control at all times, and that systems should not hamper a users freedom to make manual annotations. This proposition is supported by the comments made by our users when the algorithms provided inadequate response to their attempted refinements.

In the feedback, participants not only identified properties of interactive segmentation algorithms that they felt were important, but also identified specific image features that appeared to cause difficulties. Two image features in particular were recognized as a source of difficulty for all of the algorithms evaluated: *texture* and *object-detail*. Users commented that the algorithms were often "confused by texture," and had "difficulty with very fine details." The problems with texture are expected: none of the algorithms explicitly use texture features. The problems with segmenting edge detail are often related to contour smoothing performed by algorithms to prevent jagged object boundaries, which tend to be visually disturbing. Interestingly, one participant recognized this link between object detail and boundary jaggedness, suggesting that "there could be a boundary smoothness tool to control the jaggedness of a region."

Table 7
Correlation of measured and perceived accuracy

	Correlation Coefficient	**Kendall's τ**
$\tilde{\mathcal{A}}_B$	0.679	0.494
$\mathcal{A}_O$	0.669	0.516
$\mathcal{A}_B$ (binary)	0.606	0.445
$\mathcal{P}r$	0.564	0.448
$\mathcal{R}e$	0.469	0.382
$\mathcal{RI}$	0.375	0.350

5.4 Validation

To demonstrate the benefits of the proposed measures we compare our suggested benchmarks, and several other popular measures, with perceived accuracy as indicated by the participants. For the comparison, we computed two measures of correlation between measured accuracy and perceived accuracy, specifically: the (Pearson product-moment) correlation coefficient, and Kendall's tau [28] rank correlation coefficient. Kendall's tau is a measure of the strength of association of cross tabulations, and has values in the interval $[-1, 1]$, where 1 indicates perfect agreement and -1 perfect disagreement.

Instead of computing the correlation coefficients directly against all the perceived and measured accuracies for all final segmentations, we first average the values for the same segmentation of the same image (with the same algorithm) for different users. This pre-averaging helps to mitigate outliers, and was motivated by the fact that participants expressed that they had either made some errors in the questionnaires, or had mis-read some of the task descriptions.

The resulting correlation values are shown in Table 7. The values show that the suggested object and boundary accuracy measures are more closely correlated with human perception than are the other tested measures, with boundary accuracy $\tilde{\mathcal{A}}_B$ having a higher correlation coefficient and object accuracy $\mathcal{A}_O$ having a higher value of Kendall's tau. Furthermore, the proposed fuzzy version of boundary accuracy is also better correlated with perceived accuracy than the binary case $\mathcal{A}_B$ for both coefficients.

6 Conclusion & Future Work

In this paper we have presented a comparative evaluation of four interactive segmentation techniques. This evaluation was carried out in the form of a

user experiment in which 20 participants were asked to segment objects using different interactive segmentation algorithms. To support the experiment, we developed a consistent user interface for hosting scribble driven interactive segmentation algorithms, that also supports the most important features of other image editing tools. We selected a set of 100 objects from a publicly available dataset, containing a good cross-section of segmentation challenges. These images were then manually segmented, and annotated with unambiguous descriptions of the desired objects.

We selected two measures for evaluation: the Jaccard index to measure object accuracy, and a new fuzzy Jaccard index to evaluate boundary accuracy. Object segmentation masks were stored after each participant performed a new interaction, and the accuracy benchmarks were computed against each stored mask. The resulting plots of average accuracy over time demonstrated that the two most effective techniques were the interactive graph cuts algorithm and the binary partition tree algorithm.

In addition to measuring accuracy against a ground-truth, participants were asked to rank the accuracy of each final segmentation. The results of this ranking was observed to correspond well with the average measured accuracy. Furthermore, the correlation between perceived accuracy and measured accuracy was shown to be higher for the proposed measures than for other commonly used measures, including precision, recall and the Rand index.

The interactive segmentation tool, complete with the four algorithms from this paper, and the ground-truth dataset have been made available online for download. We hope that others will find these resources valuable, and allow authors to more easily evaluate their interactive segmentation algorithms in the future. The website URL is given in the appendix.

In the future, we would like to expand on the work presented here by developing a new interactive segmentation technique. Specifically, one that explicitly considers texture, a feature neglected by the algorithms tested, and to determine if this new algorithm is an improvement on the state-of-the-art by evaluating it with the described framework.

Additionally we would like to further explore the influence of the bandwidth parameter on the fuzzy boundary accuracy measure. In particular, it may be useful make the parameter a function of object size.

Acknowledgment

This material is based upon work supported by by the European Commission under contract FP6-027026, K-Space: Knowledge Space of semantic inference for automatic annotation and retrieval of multimedia content. This work is supported by Science Foundation Ireland under grant 07/CE/I1147.

The authors also wish to acknowledge all the volunteers who participated in the experiment.

Appendix

To download the interactive segmentation tool or the ground-truth dataset, please visit:
`http://kspace.cdvp.dcu.ie/public/interactive-segmentation`

References

[1] F. Ge, S. Wang, T. Liu, New benchmark for image segmentation evaluation, Journal of Electronic Imaging 16 (3) (2007), 033011.

[2] K. McGuinness, G. Keenan, T. Adamek, N. OConnor, Image segmentation evaluation using an integrated region based segmentation framework, in: VIE'07 – Proceedings of the 4th IET Visual Information Engineering Conference, Royal Statistical Society, London, UK, July 2007.

[3] X. Jiang, C. Marti, C. Irniger, H. Bunke, Distance measures for image segmentation evaluation, EURASIP Journal on Applied Signal Processing 2006 (1), 209–209.

[4] H. Zhang, J. E. Frittsm, S A. Goldman, Image segmentation evaluation: A survey of unsupervised methods, Computer Vision and Image Understanding 110 (2) (2008), 260–280.

[5] M. Kass, A. Witkin, D. Terzolpoulos, Snakes: Active contour models, International Journal of Computer Vision 1 (4) (1988), 321–331.

[6] J. Liang, T. McInerney, D. Terzopoulos, Interactive medical image segmentation with united snakes, Medical Image Computing and Computer-Assisted Intervention 1679 (1999), 116–127.

[7] C. Rother, V. Kolmogorov, A. Blake, Grabcut: interactive foreground extraction using iterated graph cuts, ACM Trans. Graphics 23 (3) (2004), 309–314.

[8] R. Adams, L. Bischof, Seeded region growing, IEEE Trans. on Pattern Anal. and Mach. Intell. 16 (6) (1994), 641–647.

[9] G. Friedland, K. Jantz, R. Rojas, SIOX: Simple interactive object extraction in still images, in: Proceedings of the IEEE International Symposium on Multimedia, Irvine, California, USA, December 2005, pp. 253–260.

[10] Y. Boykov, M. Jolly, Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images, in: ICCV'01 – Proceedings of the International Conference on Computer Vision 2001, Vancover, Canada, July 2001, pp. 105–112.

[11] P. Salembier, L. Garrido, Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval, IEEE Trans. Image Processing 9 (2000), 561–576.

[12] T. Adamek, Using contour information and segmentation for object registration, modeling and retrieval, Ph.D. dissertation, Dublin City University, June 2006.

[13] D. L. Pham, C. Xu, J. L. Prince, A survey of current methods in medical image segmentation, Annual Review of Biomedical Engineering 2 (2000), 315–337.

[14] G. Wyszecki, W. S. Stiles, Color Science: Concepts and Methods, Quantitative Data and Formulae. John Wiley and Sons, New York, 1982.

[15] D. Greig, B. Porteous, A. Seheult, Exact maximum a posteriori estimation for binary images, Journal of the Royal Statistical Society. Series B (Methodological) 51 (2) (1989), 271–279.

[16] Y. Li, J. Sun, C. K. Tang, and H. Y. Shum, Lazy snapping, ACM Transactions on Graphics 23 (3) (2004), 303–308.

[17] Y. Boykov, V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, IEEE Trans. on Pattern Anal. and Mach. Intell. 26 (9) (2004), 1124–1137.

[18] Y. Rubner, C. Tomasi, L. Guibas, The earth mover's distance as a metric for image retrieval, International Journal of Computer Vision 40 (2) (2000), 99–121.

[19] O. Morris, M. Lee, A. Constantinides, Graph theory for image analysis: an approach based on the shortest spanning tree, IEE Proceedings. Part F. Communications, Radar and Signal Processing 133 (2) (1986), 146–152.

[20] T. Adamek, N. E. O'Connor, N. Murphy, Region-based segmentation of images using syntactic visual features, in: WIAMIS'05 – Proceedings of the 6th International Workshop on Image Analysis for Multimedia Interactive Services, Montreux, Switzerland, April 2005.

[21] S. D. Olabarriaga, A. W. M. Smeulders, Interaction in the segmentation of medical images: A survey, Medical Image Analysis 5 (2) (2001), 127–142.

[22] L. Zadeh, Fuzzy sets and systems, Information and Control, 8 (3) (1965), 338–353.

[23] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: ICCV'01 – Proceedings of the 8th International Conference Computer Vision, Vancover, Canada, July 2001, vol. 2. pp. 416–423.

[24] F. Ge, S. Wang, T. Liu, Image-segmentation evaluation from the perspective of salient object extraction, in: CVPR'06 – Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, New York, USA, June 2006, vol. 1, pp. 1146–1153.

[25] W. Rand, Objective criteria for the evaluation of clustering methods, Journal of the American Statistical Association 66 (336) (1971), 846–850.

[26] J. Koenemann, N. J. Belkin, A case for interaction, a study of interactive information retrieval behavior and effectiveness, in: CHI'96 – Proceedings of ACM Conference on Human Factors in Computing Systems, Vancouver, Canada, April 1996, pp. 205–212.

[27] B. Suh, B. B. Bederson, Semi-automatic photo annotation strategies using event based clustering and clothing based person recognition, Interacting with Computers 19 (4) (2007), 524–544.

[28] M. Kendall, A new measure of rank correlation, Biometrika 30 (1-2), (1938) 81–93.

About the Author — Kevin McGuinness received his B.Sc. degree in Computer Applications and Software Engineering from Dublin City University in 2005. He subsequently joined the Centre for Digital Video Processing and is now pursuing a Ph.D with the School of Electronic Engineering. His research interests include image and video segmentation, segmentation evaluation, content based multimedia information retrieval and software development.

About the Author — Noel E. O'Connor graduated from Dublin City University with a B.Eng. in Electronic Engineering in 1992 and a PhD in 1998. He is now a Senior Lecturer in the School of Electronic Engineering and a Principal Investigator in the Centre for Digital Video Processing, which he co-founded. Since 1999, he has published over 130 peer-reviewed publications. His current research projects include scene-level classification, multi-spectral video analysis, smart AV sensored environments, and 2D/3D visual capture.