INTRODUCTION

Although mobile 3D displays are ready for market, there is not enough 3D content for these displays. Therefore, conversion of 2D content to 3D is crucial in the intermediate period. For conversion of mono images to stereo, an interactive segmentation based method is proposed in [1]. The results of that method show that with accurate image segmentation, it is possible to generate 3D content with acceptable quality. However, fully automatic and accurate image segmentation is still a difficult problem, despite the advancements during recent years. When compared to fully-automatic methods, semi-automatic interactive segmentation algorithms have improved performance due to the user involvement. Recent interactive segmentation algorithms were shown to be successful in various applications [2, 3] but these methods are not very suitable for mobile touch-screen applications. Firstly, correction stage commonly used in these algorithms is not desired due to the tedious zooming operations. Moreover, the complexity of some of the algorithms make them not very suitable for mobile use-cases due to limited processing power of mobile devices. After extensive investigation of existing literature on interactive segmentation, it is argued that there is no candidate which satisfy all of these requirements. Therefore, an interactive segmentation algorithm based on dynamic and iterative graph-cut specifically tailored for mobile touch screen applications is proposed in [4] and utilized in our paper for our 2D-3D conversion.

After the image segmentation stage, depth image based rendering (DIBR) techniques are used in literature to obtain stereoscopic data from a single image and corresponding depth map [5]. Several algorithms, including using highly sophisticated image inpainting ones [6] are proposed for visually plausible hole filling. In this paper, we propose two novel hole handling techniques, namely foreground enlargement, covering holes by scaling the foreground to a slightly bigger scale, and background blurring, hiding the artifacts resulting from hole filling by imitating the blurry background effect in photographs having short depth of field.

This paper is organized as follows. In Section 2, we provide the details of our proposed algorithm. Firstly the details of the interactive segmentation algorithm is proposed, followed by the proposed hole filling techniques. In Section 3, we describe the conducted subjective experimental results of our algorithm. Section 4 concludes the paper.

2. PROPOSED METHOD

2.1. Interactive Segmentation

In order to decrease the computation time, input image is initially oversegmented via simple linear iterative clustering (SLIC) algorithm [7] in order to decrease the number of nodes in the graph. Then, grayscale version of the image is shown to user and dynamic-interactive image segmentation process takes place. User starts to colorize the object of interest by finger strokes on the screen. With each input stroke, a dynamic and iterative algorithm is executed, and the intermediate results of the algorithm is updated on the display in real-time. User continues to colorize the object of interest until he/she is satisfied with the result.

Color model used in the algorithm is mixture of Gaussians in RGB color space [3]. With each stroke of the user, foreground color model is learned from interacted super-pixels,
while interaction is background free. Background is assumed to be the non-interacted super-pixels and background model is learned from these regions. As a result, there exist actual foreground regions in the background model and this makes the algorithm more conservative against expanding the foreground. Energy minimization problem is defined by using Gibbs energy and this energy is minimized by using min-cut/max-flow algorithm [8]. In order to apply min-cut/max-flow efficiently, the algorithm runs on the sub-image around the interaction instead of the whole image [4]. Size of this sub-image is determined using the method proposed in [4]. Some interactions and their corresponding results are shown in Figure 1.

![Interaction](image1.png) ![Segmentation Result](image2.png)

**Fig. 1:** User interactions and corresponding segmentation results

In order to conceal user interaction errors, a novel method is also proposed in [4]. While coloring the object of interest, user generally goes beyond the borders of the interested object. Moreover, most of the times, user returns back. In order to solve these type of interaction errors, a cost function based on color similarities is defined and minimized by using A* algorithm.

![Path correction algorithm for 3 main scenarios](image3.png)

**Fig. 2:** Path correction algorithm for 3 main scenarios

In order to conceal user interaction errors, a novel method is also proposed in [4]. While coloring the object of interest, user generally goes beyond the borders of the interested object. Moreover, most of the times, user returns back. In order to solve these type of interaction errors, a cost function based on color similarities is defined and minimized by using A* algorithm.

![Path correction algorithm for 3 main scenarios](image4.png)

**Fig. 2:** Path correction algorithm for 3 main scenarios

In order to conceal user interaction errors, a novel method is also proposed in [4]. While coloring the object of interest, user generally goes beyond the borders of the interested object. Moreover, most of the times, user returns back. In order to solve these type of interaction errors, a cost function based on color similarities is defined and minimized by using A* algorithm.

![Path correction algorithm for 3 main scenarios](image5.png)

**Fig. 2:** Path correction algorithm for 3 main scenarios

In order to conceal user interaction errors, a novel method is also proposed in [4]. While coloring the object of interest, user generally goes beyond the borders of the interested object. Moreover, most of the times, user returns back. In order to solve these type of interaction errors, a cost function based on color similarities is defined and minimized by using A* algorithm.

![Path correction algorithm for 3 main scenarios](image6.png)

**Fig. 2:** Path correction algorithm for 3 main scenarios

In order to conceal user interaction errors, a novel method is also proposed in [4]. While coloring the object of interest, user generally goes beyond the borders of the interested object. Moreover, most of the times, user returns back. In order to solve these type of interaction errors, a cost function based on color similarities is defined and minimized by using A* algorithm.

![Path correction algorithm for 3 main scenarios](image7.png)

**Fig. 2:** Path correction algorithm for 3 main scenarios

In order to conceal user interaction errors, a novel method is also proposed in [4]. While coloring the object of interest, user generally goes beyond the borders of the interested object. Moreover, most of the times, user returns back. In order to solve these type of interaction errors, a cost function based on color similarities is defined and minimized by using A* algorithm.
3. SUBJECTIVE TESTS

3.1. Interactive Segmentation

We have proposed and practiced a subjective evaluation for interactive segmentation algorithms. We have compared 3 different interaction methods, namely intelligent scissors [2], grab-cut[3] and the proposed method [4]. Grab-cut uses only a rectangle around the object of interest as an input, whereas intelligent scissors requires the approximate boundary of the object. Our method requires coloring of the object using the approach detailed in Section 2.1.

3.1.1. Subjective Test Methodology

In order to evaluate the algorithms, four different evaluation metric is used: performance, entertainment, easiness and overall. A demo of three interaction methods is shown to the subject. After the demo, the subject is asked to segment four random images with all three algorithms in a random order. When finished, the subject is asked to rate each algorithm by the four aforementioned metrics. Rating is done by grading each test from 1 to 5. Grade 1 corresponds to the worst score within the metric, whereas grade 5 corresponds to the highest score. The tests were conducted in a regular office environment with capacitive touch screen tablet PC. 15 subjects, composed of undergraduate engineering students, participated in the tests.

3.1.2. Test Results

Performance: The proposed method has the best performance result, while intelligent scissors is the second best. This result is expected due to the clear performance difference between algorithms. Extensive analysis on the performance of each algorithm is presented in [4].

Easiness: Grab-cut only requires a rectangle around the object. As a result, it scored to be the easiest one in the subjective tests. Our method becomes second since coloring is considered to be much easier than choosing a boundary.

Entertaining: Our method has the best entertainment result subjectively. This is possible due to the coloring gesture. Intelligent scissors was ranked third since selection of landmarks and moving around the boundary is a quite hard task.

Overall: All the subjects have selected the proposed algorithm as the one having the best overall experience.

Median ratings for each algorithm as well as inter-quartile ranges (IQR) and standard deviations (STD) are summarized below. Dependent ANOVA test is applied and resulting p-values are the same for each metric and equal to 0.0005.

Table 1: Interactive Segmentation Test Results (Median:IQR:STD)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Perf.</th>
<th>Easiness</th>
<th>Entertain</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed Method</td>
<td>5:1:.45</td>
<td>4:0:.86</td>
<td>5:1:.74</td>
<td>4:1:.45</td>
</tr>
<tr>
<td>Int. Scissors[2]</td>
<td>3:1:.51</td>
<td>2:1:.74</td>
<td>3:2:.89</td>
<td>2:1:.76</td>
</tr>
</tbody>
</table>

3.2. 3D Rendering

In 3D Rendering tests, user preferences for foreground enlargement and background blurring methods were evaluated.

3.2.1. Subjective Test Methodology

The double-stimulus continuous quality-scale (DSCQS) method [11] was adopted. 3D images from 12 different test images were created by the experimenter beforehand using the system presented. For each of the tests, two stereoscopic images are shown to the subject side by side and the subject is asked to choose the better one. One of the stereoscopic images is plain and the other one is generated using one or two of the algorithms presented in Section 2.2. The subjects are not told which one of the images is generated by
the proposed methods, also the placement of the images are random.

The tests were conducted in a regular office environment. Each subject was shown 30 different image pairs. All pairs were shown twice at random times for a consistency check. 36 subjects, composed of undergraduate engineering students, participated in the tests. Images were shown on a 22” 3D computer screen with shutter glasses.

### 3.2.2. Results

Some of the images scored particularly well when foreground was enlarged, while others performed worse. It is observed that if the foreground object is actually closer to the camera in real 3D space (i.e. when the synthesized depth map is more realistic), foreground enlargement is preferred by subjects. On the other hand, if there exist some objects actually closer to the camera than the selected foreground object, foreground enlargement is not preferred due to complications in 3D perception. Moreover, if a visual inconsistency due to replacement of the foreground object occurs, the technique is unfavored. Several images belonging to these two subsets can be seen in Figure 5. Overall test results for these two subsets can be seen in Table 2.

![Fig. 5: Images which scored high (top) and low (bottom) in foreground enlargement tests](image)

### Table 2: Foreground Enlargement Test Results

<table>
<thead>
<tr>
<th>Test Image Subset</th>
<th>Positive</th>
<th>Negative</th>
<th>Neutral</th>
</tr>
</thead>
<tbody>
<tr>
<td>FG Enlargement Compatible</td>
<td>%60</td>
<td>%35</td>
<td>%5</td>
</tr>
<tr>
<td>Not-FG Enl. Compatible</td>
<td>%41</td>
<td>%47</td>
<td>%12</td>
</tr>
</tbody>
</table>

### Table 3: Background Blurring Test Results

<table>
<thead>
<tr>
<th>Used Algorithm</th>
<th>Positive</th>
<th>Negative</th>
<th>Neutral</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background Blurring</td>
<td>%19</td>
<td>%74</td>
<td>%7</td>
</tr>
<tr>
<td>BG Blur &amp; FG Enlargement</td>
<td>%22</td>
<td>%73</td>
<td>%5</td>
</tr>
</tbody>
</table>

Background blurring scored quite low in the tests. It can be argued that, the apparent degradation of image quality disturbs subjects. This also affected user preferences when foreground enlargement and background blurring were used together. Nevertheless, including an option to blur the background should be preferable in such an 2D-3D conversion system. Results are presented in Table 3.

### 4. CONCLUSION

This paper presents a novel 2D-3D image conversion system, optimized for mobile devices. The proposed system involves a novel interactive image segmentation algorithm and a 3D rendering stage. Proposed interactive segmentation algorithm is found out to be the most preferable through subjective tests. Among proposed rendering techniques, foreground enlargement appears to be preferable for some specific data, whereas background blurring should remain as a user preference.

### 5. REFERENCES


