Assignment 1 — Foreground Segmentation and Poisson Blending

1. **Introduction:**

On this assignment we implemented foreground segmentation using GrabCut algorithm and used Poisson Blending technique to blend in cut objects with given images. In most cases, our results were good with high precision on the GrabCut algorithm, and the main factor for misfunction was similar color pallets of background and foreground. when using the Poisson Blending algorithm, we found that \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*.

1. **GrabCut Algorithm:**Our implementation for the GrabCut algorithm \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*. We achieved an average time of \*\*\* minutes until convergence for the images given.
   1. **Algorithm results:**

|  |  |
| --- | --- |
| Image name | Result |
| banana1.jpg |  |
| banana2.jpg |  |
| book.jpg |  |
| bush.jpg |  |
| cross.jpg |  |
| flower.jpg |  |
| fullmoon.jpg |  |
| grave.jpg |  |
| llama.jpg |  |
| memorial.jpg |  |
| sheep.jpg |  |
| stone2.jpg |  |
| teddy.jpg |  |

* 1. **Failure cases:**The GrabCut algorithm tends to produce unsatisfactory results when the subject in the given image is too close in color to the background as can be seen in the example image 'banana1.jpg'.  
     Also, when the initial bounding box around the subject includes a large portion of the background, the time until convergence increases. This issue is especially critical when the object's shape is not well contained within a small rectangle.
  2. **Effects of Blur:**we compared the effects of different blur intensities on the GrabCut algorithm. In our test we used two types of blur kernels: and of 1's. We also used gaussian blur filter which produced similar results to the bigger mask, so it is not displayed here. We found that \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*. This is compatible with logic since \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*. Actual findings are presented in *Table 1.*

Table 2 - Effects of blur

|  |  |  |  |
| --- | --- | --- | --- |
| Image name | Blur | Accuracy | Iteration until converge |
| Image1 | No blur |  |  |
| Low blur |  |  |
| High blur |  |  |
| Image 2 | No blur |  |  |
| Low blur |  |  |
| High blur |  |  |
| Image 3 | No blur |  |  |
| Low blur |  |  |
| High blur |  |  |

* 1. **Effects of GMM components count:**we compared the effects of GMM components count on the GrabCut algorithm. We found that \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*. Actual findings are presented in *Table 2.*

Table 3 - Effects of GMM components count

|  |  |  |  |
| --- | --- | --- | --- |
| Image name | Components | Accuracy | Iteration until converge |
| Image1 | 2 |  |  |
| 5 |  |  |
| 10 |  |  |
| Image 2 | 2 |  |  |
| 5 |  |  |
| 10 |  |  |
| Image 3 | 2 |  |  |
| 5 |  |  |
| 10 |  |  |

* 1. **Effects of Bounding box initialization:**we compared the effects of initial bounding box on the GrabCut algorithm. We found that a tighter box around the object produce better results and improves the time until convergence. Also, the accuracy increases. Actual findings are presented in *Table 3.*

Table 4 - Effects of box initialization

|  |  |  |  |
| --- | --- | --- | --- |
| Image name | Box | Accuracy | Iteration until converge |
| Image1 | Tight box |  |  |
| Loose box |  |  |
| Image 2 | Tight box |  |  |
| Loose box |  |  |
| Image 3 | Tight box |  |  |
| Loose box |  |  |

As can be seen in the table, most images had good results with both Accuracy and Jaccard value above 97%.

1. **Poisson blending:**

Our implementation for the Poisson Blending algorithm \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*.

* 1. **Blending results:**

|  |  |  |  |
| --- | --- | --- | --- |
| Target image | Source image | Accuracy | Iteration until converge |
| Grass mountains |  |  |  |
|  |  |  |
|  |  |  |
| table |  |  |  |
|  |  |  |
|  |  |  |
| wall |  |  |  |
|  |  |  |
|  |  |  |

1. **Appendices:**