

Project Report

Team: Sentinels

“GrabCut” — Interactive Foreground Extraction using Iterated Graph Cuts

Proposed Timeline

Week 1

1. Iterative Segmentation UI (HowTO and API)
2. Understanding the paper and project specifics.

Week 2

1. Image segmentation by Graph Cut
 - Report creation.

Week 3

1. The GrabCut segmentation algorithm
 - Report updation

Week 4

1. Benchmarking and testing.
2. Report and Project finalization.
3. Border Matting (and Transparency) if time permits.

Work Done

Understood Segmentation Algorithm

The segmentation is done using the graph mincut algorithm. The pixels of the image are represented as the nodes of the graph where each pixel is connected to its 4 (or less for bordering ones) neighboring nodes. The weights of each edge is defined by how much the two pixels match in terms of their colors (RGB value). Thus similar colored pixels have a high weighted edge between them. Two extra nodes, Background Terminal and Foreground Terminal, are added in the graph and connected to every pixel each, where the weights of

the edges are determined by how much likely is one pixel to be in background and foreground respectively (this is determined with the help of the GMMs and the User interactions). Now this graph is ready to be used in mincut algorithm where we divide the graph into two parts such that Background Terminal and Foreground Terminal are in two parts and each other node is in exactly one partition. The cut is made such that the sum of the weights of the cut edges is minimum. The pixels in foreground partition are marked as foreground and pixels in background partition are marked as background.

User Interface

We have completed work for a Window Manager which allows the user to give necessary hints to the GrabCut algorithm regarding segmentation. The features that we allow currently are:

1. The user can draw an initial box which should contain the foreground objects.
2. Allowing the user to view original image and the proposed segmentation side by side.
3. The user can use a brush like feature to give 4 kind of hints to the algorithm that are:
 - i. Sure background.
 - ii. Sure foreground.
 - iii. Probably background.
 - iv. Probably foreground.
4. User can save the current output image.

This was tested completely using the inbuilt openCV grabCut function.

Gaussian Mixture Model

Gaussian Mixture Model were used for having <i>soft segmentation</i>
--

1. `GMM` class defined in `gmm.py` (`./gmm.py`) contains the functionalities required for *Grabcut segmentation* algorithm:
 - Initialization of the parameters are done using the *K-means clustering algorithm* to get better results.
 - Functionalities include:
 - `GMM.score` : Predict probabilities of samples belong to a particular component in the *mixture model*.
 - `GMM.prob` : Predict probability (weighted score) of samples belong to the GMM.
 - `GMM.component` : Predicts which GMM component the samples belong to in the *Gaussian mixture model*.
 - `GMM.fit` : Computes the mean and co-variance.