Masking Objects Using Segmentation

Mrinal K. Rawool

Topics in Image Processing CS6420

Professor Sanjiv Bhatia

Friday, May 17, 2019

Masking Objects Using Segmentation

Abstract

Image segmentation is one of the important areas of research in the field of computer vision. In the case of real world digital images, challenges stem from the fact that the object of interest may be surrounded by a lot of noise thereby making it difficult to semantically separate the object from the background. Another challenge encountered is due to the nature of the physical devices capturing the image. As the intensity changes are gradual, the object boundaries are not sharp. This causes difficulty in determining where the boundary pixels belong. In other words, where does the object end and the background start. Lastly, the sharper, or detail oriented image you have, the more processing power it takes to process and segment the image. In this project, I have implemented image segmentation using two algorithms. SLIC super-pixels and seeded region growing algorithm.

1. Introduction

The main objective of the project is to successfully segment all the objects of interest from a given image with limited interaction from an external user. The interaction will be in the form of a scribble mask that will indicate the position(s) of the object(s) of interest. The scribble mask is a grayscale image with same dimensions as the original image. Furthermore, for an image with multiple objects of interest, each object is annotated by a different intensity level. This project uses images and scribbles provided by the CV-DAVIS challenge (https://davischallenge.org/challenge2018/interactive.html). A sample image and the corresponding scribble mask is provided below.

FIG 1: BLACK SWAN

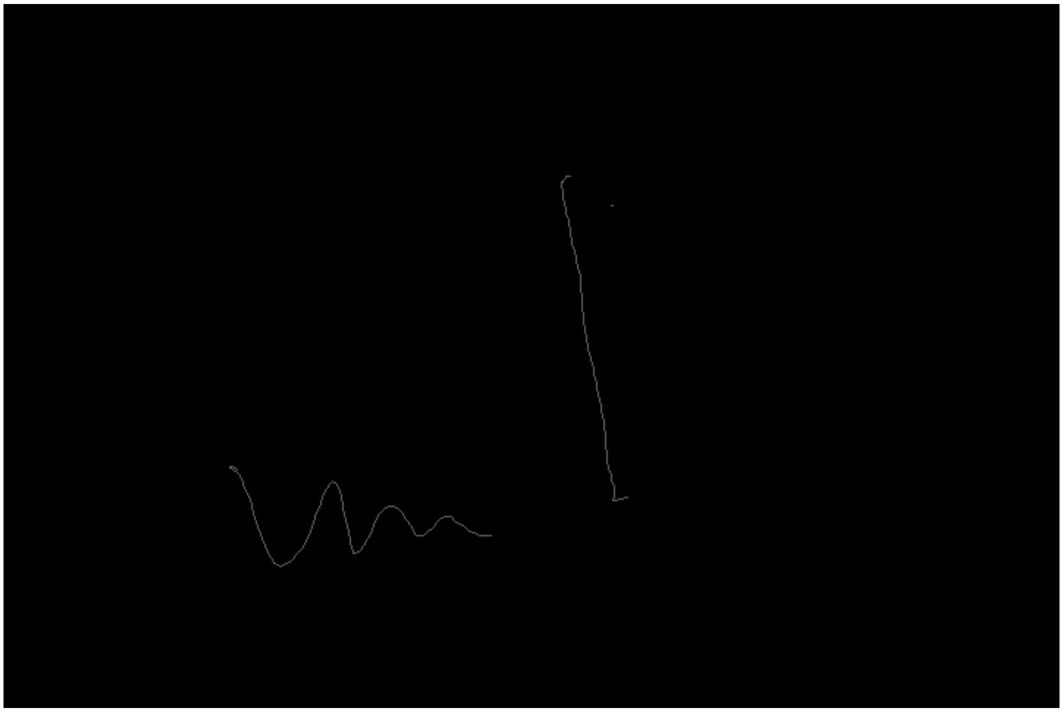

FIG 2: ANNOTATED SCRIBBLE FOR BLACK SWAN

The overall approach is to tackle the task in two steps. The first step is to reduce the level of detail in the image. This helps in speeding up the process due to fewer number of pixels to process. Moreover, it also helps in enhancing the image attributes like object borders that would eventually help in segmentation. I have used the OpenCV implementation of SLIC super pixels to achieve this objective.

The second step is to use the super pixels and the scribble masks to create a mask for each object of interest. To achieve this, I have used a seeded region growing algorithm along with the annotated scribble masks. I have used scribble masks to generate statistical information about the object of interest. This information will help us determine the nature of the foreground (object) from the background (environment).

2. Creating Super Pixels

The main objective behind creating super pixels was to reduce the amount of processing data. I have implemented the SLIC super-pixel algorithm provided by OpenCV for this project. The algorithm is gradient based and it uses both distance in the CIELAB colorspace and spatial proximity to decide on how pixels are to be grouped.

The foundation of the SLIC super-pixel algorithm is the calculation of distance between two pixels. This distance is calculated in a 5 dimensional space to take into account both the physical proximity of the pixels as well as the color distance between them. CIELAB colorspace ensures that difference is pixel values correspond to similar amount of perceived visual change in color. Prior to calculating the distance to determine color similarity, the SLIC algorithm takes an input of the number of super-pixels desired. Let K be the number of desired super-pixels and N be the total number of pixels in the image. Therefore, the approximate size of each super-pixel is N/K. Furthermore, assuming all the super-pixels are roughly of the same size, the grid interval S between two super-pixels would be $\sqrt[2]{N/K}$.

The 5D color similarity distance D_s of a pixel P_i from a super-pixel center P_k is given by the equation

$$D_s = d_{lab} + (m/S)d_{xy}$$

Where d_{lab} is the color distance in CIELAB given by the equation

$$d_{lab} = \sqrt{((l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2)}$$

, and d_{xy} is the euclidean distance given by

$$d_{xy} = \sqrt{((x_k - x_i)^2 + (y_k - y_i)^2)}$$

D_s takes into account both CIELAB distance and spatial distance because Euclidean distances in CIELAB color space are perceptually meaningful for small distances. If spatial pixel distances exceed this perceptual color distance limit, then they begin to outweigh pixel color similarities. The variable m determines the compactness of the super-pixel which in turn is determined by the amount of emphasis on spatial proximity relative to CIELAB color proximity.

The algorithm used for creating super pixels is called simple linear iterative clustering algorithm. This algorithm is a variant of the K-means clustering algorithm. At the start of the algorithm, K cluster centers are space equally throughout the image. Then, they are moved to their seed position that has the lowest gradient in a 3 X 3 neighborhood. This is done to avoid using a noisy pixel. Image gradients are calculated using the equation

$$G(x, y) = ||I(x+1, y) - I(x-1, y)||^2 + ||I(x, y+1) - I(x, y-1)||^2$$

where I(x, y) is the LAB vector corresponding to the pixel at position (x, y), and ||.|| is the L2 norm. This takes into account both color and intensity information. Each pixel in the image is

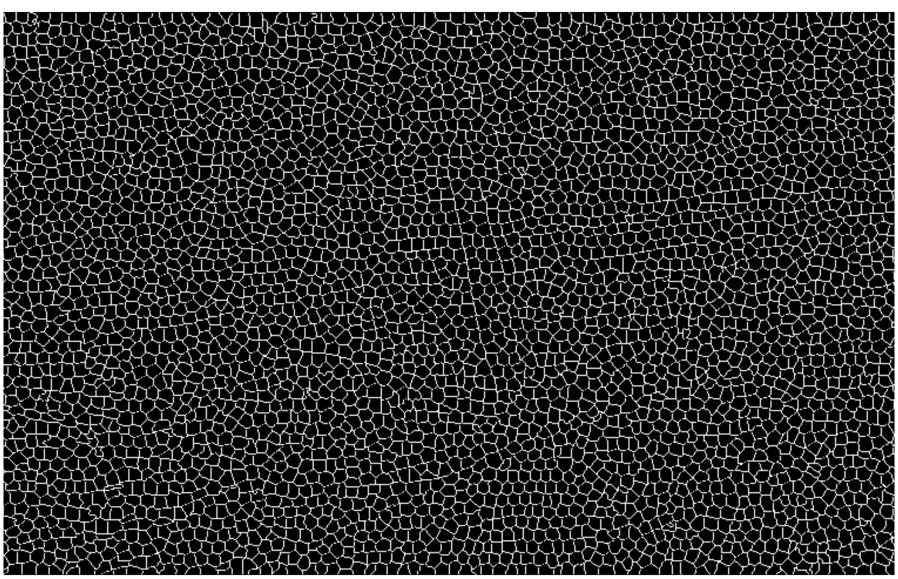

FIG 3: LABEL MASK FOR BLACK SWAN

associated with the nearest cluster. Once all the assignments are done, a new cluster center is calculated as the average $labxy$ vector of all the pixels belonging to a cluster. This process is repeated until convergence.

The OpenCV implementation of SLIC super pixels yields an array of labels with the same dimensions as that of the original image. However, the value at any given pixel P(x,y) is the super-pixel label that P(x,y) belongs to. Visually, the super pixel boundaries look as shown in Fig 3 (shown above)

I have used these labeled pixels to create a new image where every pixel value is replaced with the average pixel value of the super pixel that the pixel under consideration belongs to. This average is calculated using RGB values. The following figure shows the lower resolution image created using super pixels.

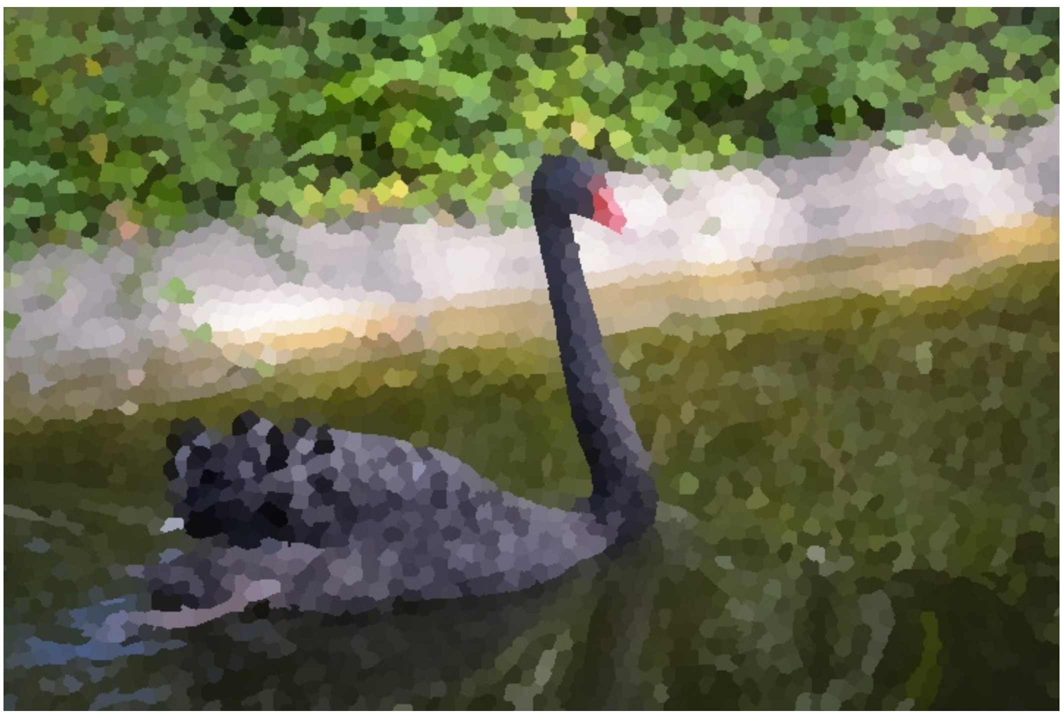

FIG 4: BLACK SWAN IMAGE WITH SUPER-PIXELS

3. **Creating Masks**

We start with the segmented image obtained from the previous section to create mask for the desired object. The algorithm used for this task is the seeded region growing algorithm. This algorithm starts grouping pixels based on some statistical properties of the image of the object of interest. For my implementation, I have used the mean and the standard deviation of the object to determine if a pixel or in this case, a super pixel belongs to the object or not. The foundational equations used for creating masks are

Mean of all points P_i belonging to scribbles is $\mu = (1/n) \sum_{i=1}^{n} I(P_i)$

Standard Deviation of a neighboring point $P_k$is $\sigma^2 = (1/n) \sum_{k=1}^{n} (I(P_k) - \mu)^2$

A brief overview of the algorithm is as follows. The algorithm starts with a seed as the starting point. A stack called *frontier* keeps a list of all the neighboring pixels to be considered. Each pixel from the *frontier* is examined for similarity with the seed. If found similar, it is added to the list of pixels that are considered as the object. The algorithm stops when all of the pixels in the *frontier* are examined and the *frontier* is empty.

In my implementation, I have used super-pixels instead of pixels to reduce the number of computations. I start with the pixels provided by scribble masks to get an idea about the mean and standard deviation of the object of interest. The *frontier* is also initialized with these pixels. The frontier in my case is implemented as a vector of points instead of a stack. The algorithm starts by examining the neighbors of every point P(x,y) in the frontier. If the neighbors are found similar to the pixel under consideration, they are added to another vector that keeps track of the super pixels that belong to the object. Additionally, these neighbors are also added to the frontier so that their neighbors can be examined for similarity. The processing completes when all the points $P(x, y)$ belonging to the *frontier*. It is to be noted that CIELAB space is used for checking

FIG 5: BLACK SWAN MASK

the similarity of the super pixels. Fig 5 (above) shows the mask generated for the "Black Swan" image using seed growing algorithm.

4. Multiple Objects

The masking algorithm has been extended for multiple objects by simply iterating the process for the number of objects to be segmented. Since each object is annotated by a different pixel value in the scribble mask, this knowledge can be used to generate separate masks for each object. The following images demonstrate the transition from the original image to segmented image for multiple object.

FIG 6: DOG JUMP IMAGE

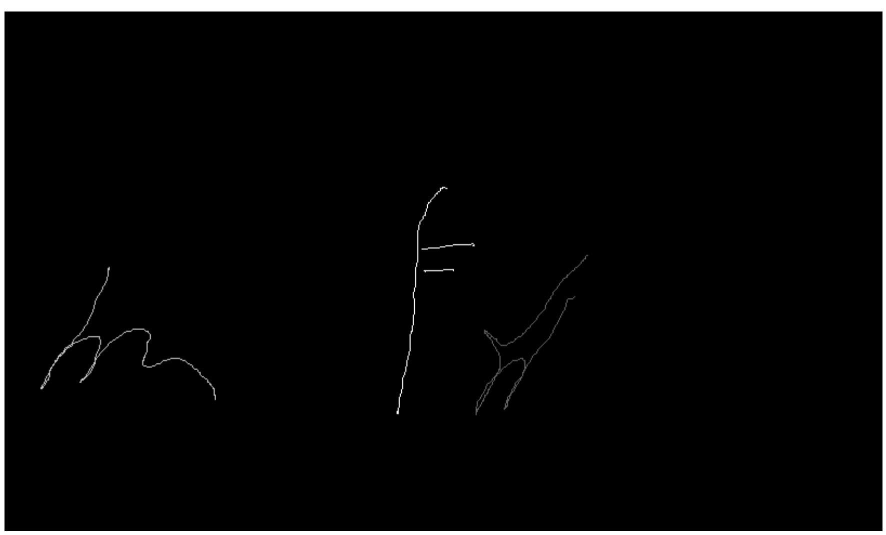

FIG 7: SCRIBBLE MASK FOR DOG JUMP IMAGE

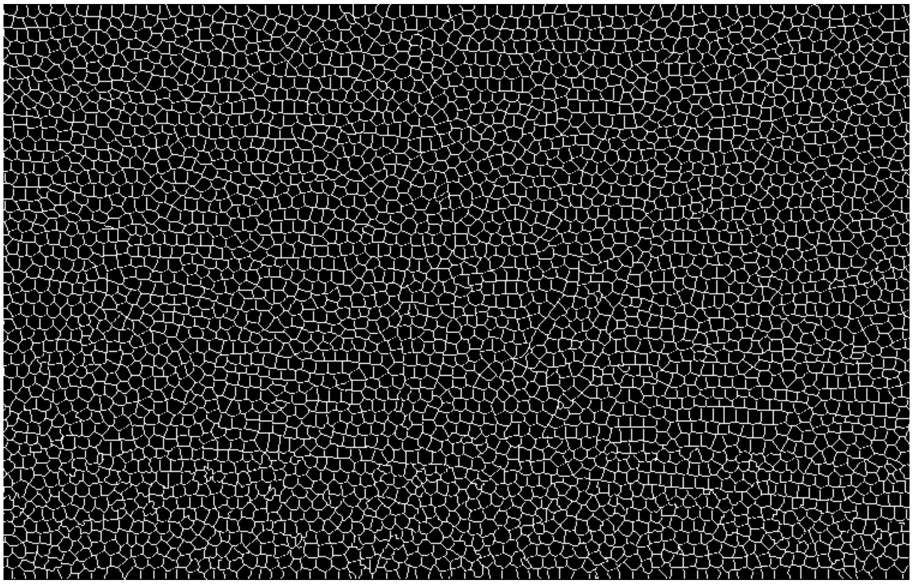

FIG 8: LABEL MASK FOR DOG JUMP

FIG 9: SEGMENTED IMAGE FOR DOG JUMP

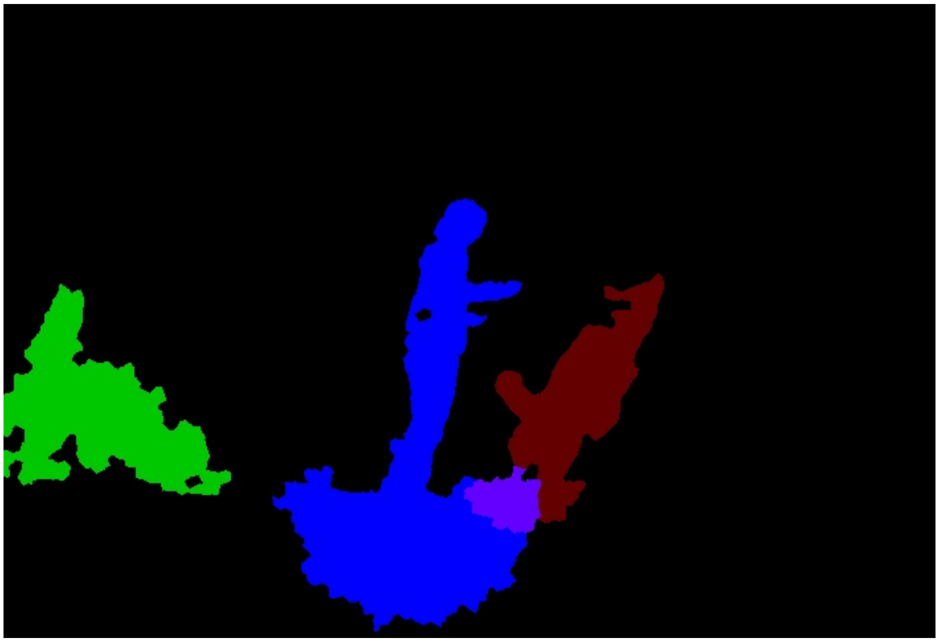

FIG 10: MASKS FOR ALL THREE OBJECTS FROM DOG JUMP IMAGE

5. Conclusion

This project aimed at implementing SLIC super pixels and seeded region growing algorithms for creating object masks. Further research can be directed towards making the objects masks more accurate at the pixel level. The seeded region growing algorithm implementation can be enhanced to achieve a better output for object masks. In conclusion, the project gives a good starting point for developing more accurate object segmentation masks.

References

1. Birchfield, S. (2018). *Image processing and analysis*. Boston, MA: Cengage Learning.
2. R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels. Technical report, 2010.