Image Background Segmentation

March 8, 2023

1 Our approach

1.1 Image conversion to grayscale

Let I(x,y) be the original color image, where (x,y) denotes the pixel coordinates. We can convert this image to grayscale using the following formula [3]:

$$I\_gray(x,y) = 0.299 \times I\_red(x,y) + 0.587 \times I\_green(x,y) + 0.114 \times I\_blue(x,y) \tag{1}$$

where

$$I\_red(x,y), I\_green(x,y) \& I\_blue(x,y) \tag{2}$$

are the intensities of the red, green, and blue color channels at pixel (x,y), respectively

1.2 Application of the Canny Edge Detector [5], [2]

The grayscaled image I_gray(x,y) is convolved with a Gaussian filter G(x,y) to reduce noise and smooth the image.

$$I\_smooth(x,y) = I\_gray(x,y) * G(x,y) \tag{3}$$

The gradient magnitude and orientation of the smoothed image I_smooth(x,y) are computed using the Sobel operator. This involves convolving the image with two separate filters, G_x and G_y, in the horizontal and vertical directions, respectively:

$$Ix(x,y) = I\_smooth(x,y) * Gx$$
$$Iy(x,y) = I\_smooth(x,y) * Gy$$

where Gx and Gy are the 3x3 Sobel kernels:
Gx = [-1 0 1; -2 0 2; -1 0 1], Gy = [-1 -2 -1; 0 0 0; 1 2 1]

The edges detected are thinned out by suppressing non-maximum pixels along the direction of the gradient. The resulting edge map is denoted as E(x,y), which is defined as:

$$E(x,y) = 0, \tag{4}$$

if I_gray(x,y) is not a local maximum along the direction of the gradient

$$E(x,y) = I\_gray(x,y), otherwise. \tag{5}$$

The edge image E(x,y) is thresholded using two thresholds, T_low and T_high, to separate strong edges from weak edges.

$$B(x,y) = 1, if E(x,y) \geq T\_high$$
$$B(x,y) = 0, if E(x,y) < T\_low$$
$$B(x,y) = 2, if T\_low \leq E(x,y) < T\_high$$

Finally, weak edge pixels that are connected to strong edge pixels are also classified as edge pixels.

1.3 Edged Image Dilation

Image dilation [1] is a morphological operation that expands the boundaries of objects in an image by adding pixels to their edges. Let E(x,y) be the input image and B be the structuring element, then the dilation of I(x,y) by B is given by:

$$(E \oplus B)(x,y) = max[E(x-i, y-j) + B(i,j)] \tag{6}$$

where $\oplus$ denotes the dilation operation, and the maximum is taken over all pixel coordinates (i,j) within the support of the structuring element B. This operation thickens the edges and creates a more complete boundary of the object in the image.

1.4 Contours and mask creation

We use a contour detection algorithm from the OpenCV [4] libary to find the contours of the object, and select the largest contour to obtain the outer boundary of the object. Finally, we create a binary mask of the object by drawing the largest contour onto a new image, which can be used to isolate the object from the original image or to perform further analysis on the object.

1.5 Mask Erosion and Application to the original image

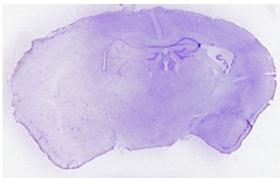
Image erosion [1] is a morphological operation that shrinks the boundaries of objects in an image by removing pixels from their edges. Let M(x,y) be the input image and B be the structuring element, then the erosion of M(x,y) by B is given by:

$$(M \ominus B)(x,y) = min[M(x+i, y+j) - B(i,j)] \tag{7}$$

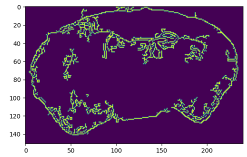
where *ominus* denotes the erosion operation, and the minimum is taken over all pixel coordinates (i,j) within the support of the structuring element B. This operation can be used to remove small or thin features from an image or to separate objects that are close together. Finally, the mask is applied to the original image.

2 Results

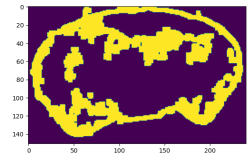
In the figure below, we can see the different stages of the algoritm's application.



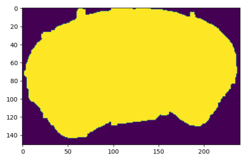
(a) Original image



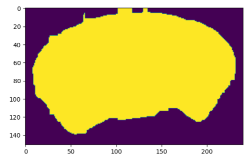
(b) Edged Image



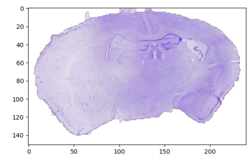
(c) Dilated edged image



(d) Binary Mask



(e) Eroded binary mask



(f) Segmented image

Figure 1: The intermediate stages of the algorithm described above

References

[1] Diya Chudasama, Tanvi Patel, Shubham Joshi, and Ghanshyam Prajapati. Image segmentation using morphological operations. *International Journal of Computer Applications*, 117:16–19, 05 2015.

[2] Lijun Ding and Ardeshir Goshtasby. On the canny edge detector. *Pattern Recognition*, 34(3):721–725, 2001.

[3] Shaymaa Hantoosh and Ahmed Alridha. Dynamic weights equations for converting grayscale image to rgb image. *Journal of University of Babylon for Pure and Applied Sciences*, 26, 10 2018.

[4] Naveenkumar Mahamkali and Vadivel Ayyasamy. Opencv for computer vision applications. 03 2015.

[5] Wen-kui Zheng and Kun Liu. Research on edge detection algorithm in digital image processing. 01 2017.