ECE 6310 Introduction to Computer Vision
Semester Project - Food segmentation

Chunheng Zhao

1 Introduction

This report considers the problem of food segmentation. In this report, an active contour algorithm is implemented to allow a user to semi-automatically segment food items in a photograph of multiple foods on a dinner plate. An active contour, also known as a "snake", is a semi-automated method for segmentation. It takes an initial contour and iteratively moves it based upon a set of criteria. The goal is to start with an initial contour close to a desired location, and then use the active contour algorithm to drive the contour to the final desired location.

The active contour algorithm works as follows. The contour v is modeled with a discrete set of points. The contour is moved by searching in the neighborhood of each pixel on the contour for a better location.

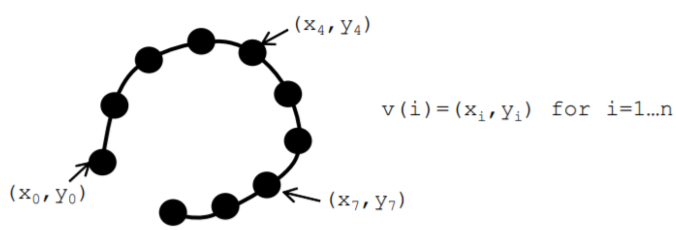


Figure 1: Active Contour Algorithm

There are a number of interesting frameworks depending on the constraints of the problem. The rubber band model operates like a rubber band that is first opened wide and then allowed to contract. Balloon model starts inside an object and makes the contour expand, which mimics a balloon. Ringed model starts 2 initial contours, one outside the object o interest, and one inside. The external contour uses a contracting internal energy, while the interior contour uses an expanding internal energy. Template model formulates the internal energy in order to maintain the known shape of the object. Also, active contours can be helpful for tracking a shape through a video sequence. The template model and contour tracking can be combined. They have been applied extensively to problems involving the tracking of humans, including face tracking, lip tracking, body motion tracking, and hand tracking.

In this project, five images are provided: bacon-eggs-toast.pnm (3 items), eggs-pancakes-milk.pnm (3 items), hushpuppies-biscuits.pnm (2 items), fish-lemon-rice-greens.pnm (4 items) and macaroni-kale.pnm (2 items). The image names indicate the number of food items in each image, which is also the number of segmented regions expected to be produced. This report also discussed the influence of different parameters and the way to improve performance.

2 Methods

In the program, it will first load a color PNM image and convert it into the greyscale version using the average of the three color bands. Due to the large size of the original image, resizing is also implemented here to automatically GUI (Graphic User Interface) is also created and the user has the following options: left-click to draw around a food item that subsequently automatically shrinks to wrap to the food boundary; right-click a point within a food item that subsequently automatically grows a contour to its outer boundary; shift-click (either button) to manually drag a contour point to a new location.

2.1 Rubber band model

Specifically, for the left click, GUI will run the following steps. Left click and hold the mouse button to draw a contour anywhere in the image. The contour will store every pixel location that the user moves the mouse through while the button is held down. When the button is released, the contour should be downsampled to every fifth point. It is assumed that the contour encloses an area, so that the final point connects to the first point for subsequent active contour processing.

After a left-clicked contour is initialized, an active contour will be iterated implementing the rubber band model. An internal energy term will cause the contour to try to shrink. We are searching the new location for pixels in a 19x19 window. For every location in the window, we calculate internal and external energy terms. The contour point is then moved to the location that has the smallest total energy. Internal energy terms are defined by constraints upon the shape of the contour. They are not dependent on the image data, they only depend upon the locations of the other contour points. External energy terms depend only upon the image data, and are not affected by the shape of the contour.

I used 2 internal energy terms and 3 external energy terms. The internal energy terms are the square of the distances between points, and the square of the deviation from the average distance between points. The latter term can be found by first calculating the average distance between all contour points, and then taking the square of the difference between that average and the distance between the current contour point and the next contour point. It is assumed that the contour encloses an area, so that the last contour point can be connected to the first contour point to calculate internal energy terms. The first internal energy term is calculated based on:

$$E_{internal1} = (x_i - x_{i+1})^2 + (y_i - y_{i+1})^2 \tag{1}$$

The second internal energy term is calculated based on:

$$E_{internal2} = (\sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} - average)^2 \tag{2}$$

The first external energy term is the square of the image gradient magnitude, and is calculated using convolution with a Sobel template. The negative is taken so that minimizing it moves it towards the brightest location. The Sobel's operator is defined as:

$$f_1 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad f_2 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{3}$$

wheref_1 and f_2 compute the horizontal and vertical gradients. Then the first external energy term is calculated as:

$$E_{external1} = |gradient|^2 \tag{4}$$

The second external energy term is the square of the deviation from the average gradient. It will minimize the deviation in edge strength and reward points that have a similar gradient. It makes the contour to move towards the same edge because in this project plate is really close to foods and sometimes the edge of plate will be regarded as edge of foods. To some degrees, rewarding points that move towards similar edges would make it better. The second external energy term is calculated as:

$$E_{external2} = (|gradient| - average)^2 \tag{5}$$

The third external energy term is the square of the variance of intensity of points. The idea here is to reward the points that have the same color. It makes sense because the food used here has the consecutive same color on its surface and it has different color from the plate. It is calculated as:

$$E_{external3} = (color - average)^2 \tag{6}$$

The total energy at each pixel is the sum of the internal and external energies. Each energy term is normalized by rescaling from min-max value to 0-1. This process gets repeated for every points in the contour. An iteration of the active contour completes once every point has moved. Typically, the active contour algorithm runs for 10-100 iterations. It can also be stopped once the total or average distance moved by all the contour points falls below a threshold. For the rubber band model, the iteration size is 50. Because the initial contour is drawn by user roughly which would not be close to the actual contour, internal energy should be weighted most to make contour shrink and then weight gradient terms a little more to make the contour stop shrinking at the edge. The total energy is calculated as:

$$E = 2 * E_{internal1} + 4 * E_{internal2} - E_{external1} + 1.2 * E_{external2} + E_{external3} \tag{7}$$

2.2 Balloon model

For the right click, GUI will run the following steps. Right click (detected after release) any point in the image. The right click will create a small circular contour centered at the clicked point with radius 10 pixels. The contour should be downsampled to every third point on the circle. After a right-clicked contour is initialized, an active contour will be iterated implementing the balloon model. An internal energy term should cause the contour to try to expand.

We are searching the new location for pixels in a 23x23 window. 2 internal energy terms and 3 external energy terms are also used here. However, the internal energy term which are the square of the distances between points should be negative in order to make the contour to try to expand. All the other terms keep the same. The total energy at each pixel is still the sum of the internal and external energies and each energy term is also normalized by rescaling from min-max value to 0-1. For the balloon model, the iteration size is 80. Because the initial contour is a circle with radius 10 pixels which is usually far from the actual contour, the internal energy still needs to be weighted most in order to make contour expand. Then weight gradient a little more to make the contour stop expanding at the edge. For the first 40 iterations, the weight for gradient keeps the same as other external energy terms which is 1 in order to help contour expand. For the next 40 - 80 iterations, the weight for gradient is increased to 1.3 to help contour stop expanding. The total energy is calculated as:

$$E = -2 * E_{internal1} + 1.5 * E_{internal2} - (1.1 \ or \ 1.3) * E_{external1} + 1.2 * E_{external2} + E_{external3} \quad (8)$$

2.3 Neutral state model

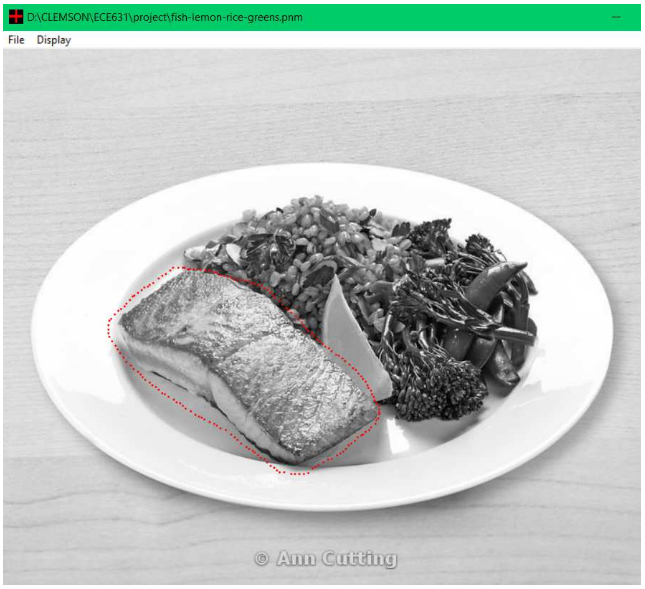
Upon completion of the active contour algorithm (Rubber band or balloon), GUI allows the user to grab a point and manually move it to a new location. Manually moving a point, should initialize another active contour. This third active contour has neither the rubber band (shrink) nor balloon (grow) energy term. It will be neutral with respect to size. It fixes the manually moved point to the given location, not allowing it to move. All the other points are allowed to move normally.

There are two situations for the shift-click, one is grabbing a point in rubber band model and the other is grabbing a point in balloon model. For both situations, the window sizes keep the same as before which are 19x19 and 23x23, but the energy terms change. Only two energy terms are used in this neutral state model. The internal energy term is the square of deviation from the average distance between points and the external energy term is the square of the image gradient magnitude. The total energy at each pixel is still the sum of the internal and external energies and each energy term is also normalized by rescaling from min-max value to 0-1. The idea here is to relocate the contour points evenly, so the internal energy will be weighted more heavily. This works fine with grabbing a point towards an actual contour of foods. However, when grabbing out a point which lies on the actual contour, some other points will follow that point to be out of actual contour after several iterations. In order to make all the points to try to stay on the right path, the weight of internal energy will be set to zero after 10 iterations and let external energy to pull points back. The total energy is calculated as:

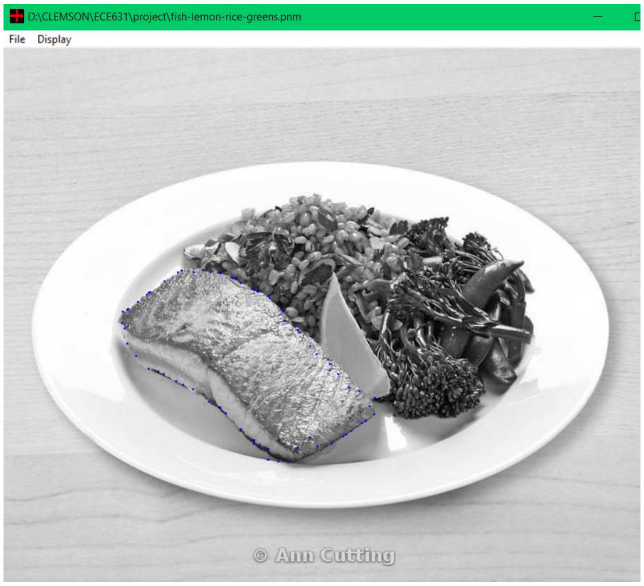
$$E = (2 \ or \ 0) * E_{internal2} - E_{external1} \quad (9)$$

3 Results

After implementing the above energy terms in C and building the GUI in MS Visual Studio, the results of food segmentation are shown here. Figure 2 (a) shows an initial contour for fish and figure 2 (b) shows the final contour after running the rubber band algorithm. For convenience, the rest results of rubber band model only shows the final contours. As we could see in the figures, the red contour is the initial contour and all the blue contours are the final contours. The rubber band algorithm in this project is able to segment foods and show their actual contour. Also, as shown in figure 3 (b) and (d), multiple pieces of the same food could be segmented into one region.



(a) Initial contour for fish



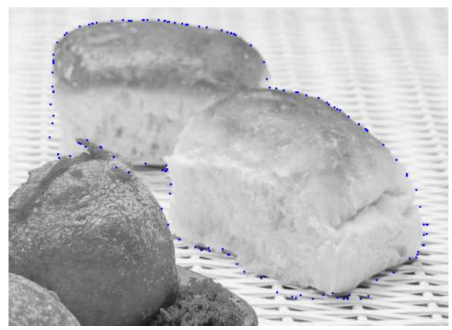
(b) Final contour for fish

Figure 2: Results for rubber band model

(a) milk

(b) hush

(c) greens



(d) biscuits

(e) kale

(f) lemon

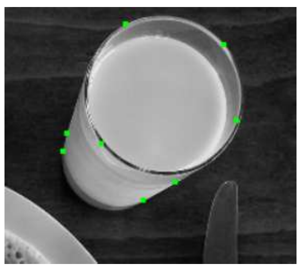
Figure 3: Results for rubber band model

(a) Initial contour for kale

(b) Final contour for kale

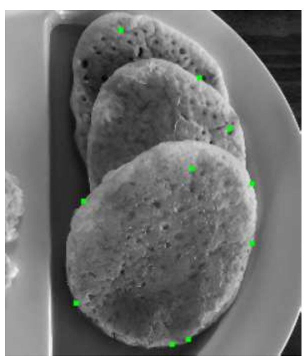
Figure 4: Results for balloon model

Figure 4 (a) shows an initial contour for kale and figure 4 (b) shows the final contour after running the balloon algorithm. For convenience, the rest results of balloon model only shows the final contours. As we could see in the figures, the red contour is the initial contour and all the green contours are the final contours. The balloon algorithm in this project is able to segment foods and show their actual contour. Also, as shown in figure 5 (b) and (d), multiple pieces of the same food could be segmented into one region.

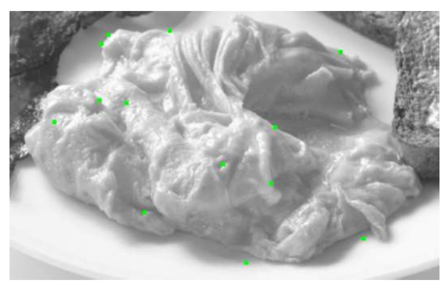


(a) milk

(b) hush



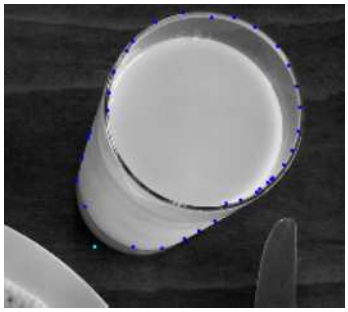
(c) greens



(d) biscuits

Figure 5: Results for balloon model

Figure 6 (a) (c) shows an initial contour for milk and kale after grabbing a point from rubber band model and figure 6 (b) (d) shows the final contour after running the neutral state model. Figure 6 (e) shows an initial contour for milk after grabbing a point from balloon model and figure 6 (f) shows the final contour after running the neutral state model. The blue or green contour points are the normal points after running rubber band or balloon algorithm and the light brilliant blue point is the manually moved point. As we could see in the figures, the neutral state model is able to fix the manually move point to the given position, not allowing it to move, and relocate the other contour points evenly.

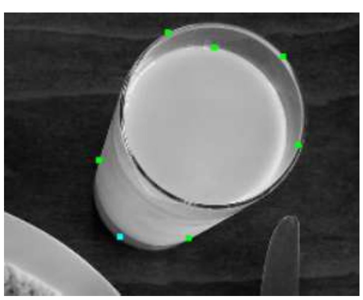


(a) milk (before)

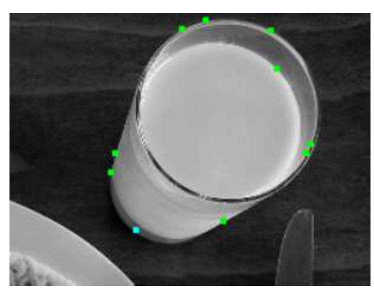
(b) milk (after)

(c) kale (before)

(d) kale (after)



(e) milk (before)



(f) milk (after)

Figure 6: Results for neutral state model

4 Conclusion

In this project, we defined the energy terms so that the contour acted like either a rubber band stretching or balloon. The internal energy was defined to pull all the contour points towards each other or expand respectively. The external energy was defined to pull the contour towards edge pixels or same color pixels.

We could conclude that an active contour can also have multiple internal and external energy terms, with each pulling the contour in a different manner. And energy terms are usually squared, to provide a steeper gradient towards minima. Specifically, in this food segmentation project, the texture is another good way to be set as an energy term and it rewards the points that have similar texture, which would also work better because the simple food has similar texture on the surface. The external energy of variance of color used in this project is a simple implementation of this idea.

The window size must be large enough so that the desired image content can have an effect through external energy terms, but not so large that it allows the contour to get pulled to an alternate local minimum. Usually the window size is balanced against the number of iterations. All the energy terms are usually normalized within the window, so that they carry equal distributions. This can be done simply by calculating the min and max values for an energy term within the window and normalizing across that range. In addition, the energy terms may be weighted differently, depending on the desired behavior of the active contour and the actual type of food. For example, if the food is milk, then the color terms could be weighted more heavily. If the food is bread in the white dinner plate, then the color terms wouldn't work well and the gradient terms should be weighted more heavily.