Contour Extraction of Moving Objects

Lili Qiu, Li Li
{lqiu, lili}@cs.cornell.edu
Department of Computer Science
Cornell University
Ithaca, NY 14850

Abstract

This paper proposes a novel approach to the contour extraction of moving objects in an image sequence. Our approach is to fuse information from color segmentation, motion segmentation, and active contour (snake) to achieve accurate extraction of the boundary of moving objects. It works well not only for a single moving object, but also for images having mutiple moving objects. Several experiments have been conducted to show the promise of our algorithm.

Keywords: *color segmentation, reflectance ratio, sequential labeling, motion segmentation, census transform, morphological filter, snake algorithm.*

1. Introduction

Extracting the contour of moving objects is of great interest in computer vision and many practical applications. The contour of an object contains information about the shape of the object and has been widely used in object recognition. The extraction of motion boundary is very useful in video surveillance to detect suspicious objects. Medical imaging also heavily uses it to track the motion of organs inside human body.

However to accurately extract the boundary of moving objects is a very challenging problem. Many researches have been conducted to tackle the problem. Lots of previous approaches use either motion field or edge information to solve the problem. Recently it has been realized that fusion of these information together can produce better results. Yet it is not clear what kind of fusion is effective in general. In this paper, we propose a fusion approach which is based on information from color segmentation, motion segmentation and active contour (snake) to achieve accurate extraction of the boundary of moving objects.

Below are detailed descriptions of our approach. Section 2 discusses color segmentation based on reflectance. Section 3 presents motion segmentation using census transform, boundary information, and morphological filter. Section 4 describes the fusion of color and motion segmentation. Section 5 talks about final contour extraction. Section 6 analyzes our experimental results. Finally section 7 presents our conclusion.

2. Color Segmentation

Color segmentation is the process that divides an image into homogeneous regions using the color information at each pixel. We employ the sequential labeling algorithm based on reflectance proposed by Nayar and Bolle [5] in this step. The main idea here is to use the reflectance ratio as a measure of similarity of two neighboring pixels, and partition the whole image into portions where each portion has similar reflectance coefficients. As shown in [5], the ratio of the reflectance coefficients of the two points 1 and 2 can be obtained by $p(1,2) = \frac{I_1}{I_2} = \frac{\rho_1}{\rho_2}$, where $p(1,2)$ is the reflectance ratio of the two points 1 and 2, I_1 and I_2 are their intensities, ρ_1 and ρ_2 are their reflectance coefficients. To prevent the denominator from being 0, we use the following revised definition:

$$p(1,2) = \frac{\rho_1 - \rho_2}{\rho_1 + \rho_2 + \triangle} = \frac{I_1 - I_2}{I_1 + I_2 + \triangle'}$$

where both $\triangle$ and $\triangle'$ are relatively small. We choose $\triangle' = 1$ in our experiments, where $I \in [0, 255]$. For a color image, the reflectance ratio can be represented as a vector of three components. For example, for an image in RGB color space, the reflectance ratio is $p = [p^r, p^g, p^b]$ [5].

The computed reflectance ratio is then used as a measure of connectivity between two neighboring pixels. Small reflectance ratio suggests these two pixels belong to the same region. Therefore we define the following definition of connectivity of the two neighboring pixels 1 and 2:

$$c(1,2) = \begin{cases} 1 & \text{if } \|p(1,2)\|_2 < T \\ 0 & \text{otherwise} \end{cases}$$

where $\|p(1,2)\|_2$ is the 2-norm of $p(1,2)$ defined above, T is a threshold and in our experiments it is chosen to range from 0.03 to 0.1. This threshold can be picked automatically in such a way that an image is partitioned into reasonable number of segmented regions. According to the definition of connectivity, sequential labeling algorithm proceeds to label all the connected pixels with the equivalent labels. Finally in order to eliminate small, invalid regions appearing on and around the two regions' boundary, another threshold is used, assuming all the regions smaller than the threshold are invalid. The threshold we choose in our experiments is 30 pixels. After the elimination, the entire image is partitioned into regions of similar reflectance. This information will later be integrated with motion segmentation results.

3. Motion Segmentation

Motion segmentation is a well studied problem in computer vision. The most popular methods for extracting moving objects are based on image subtraction and optical flow.

Image subtraction approach uses a reference frame which is obtained by averaging a large number of frames. This approach suffers from the difficulties to obtain an accurate reference image. It is also very sensitive to the change of illumination conditions.

The approach based on optical flow uses the segmented velocity field to identify different moving objects in the scene. There are two different ways to compute the optical flow. One is based on energy minimization, and the other is based on correlation. The problem with the energy minimization approach is that the physical constraints are not always satisfied, and may often fail in area of great interest. Furthermore it is also very sensitive to noise such as image gain and bias. The correlation approach is done by either matching features extracted from adjacent frames, or finding dense disparity map. Since the number of image features are relatively small, and the result from any match can be wrong, a dense disparity map is necessary for extracting a relatively accurate contour of moving objects. Therefore we choose to use the non-parametric approach proposed by Zabih and Woodfill [6] to compute a dense disparity map because of its robustness and efficiency.

3.1. Census Transform Correlation with Color Images

The non-parametric approach to disparity map computation is based on the local ordering of intensities. It can be shown that ordering information is robust to outliers and invariant to monotonic intensity distortion such as image gain and bias [6].

We use census transform proposed in [6]. Its main idea is to first apply an ordering-based local transform to the image, and then to use correlation. The way we extend the census transform for color images is to associate three census bit strings S's with each pixel, namely R, G, B census bit strings S_R, S_G, and S_B. The best matched window will be the one whose R, G, B census strings are closest to the current window's, that is i is the best match if

$$H(S_{R_i}, S_{R_{target}}) + H(S_{B_i}, S_{B_{target}}) + H(S_{G_i}, S_{G_{target}})$$

is minimum in the search window, where H denotes hamming distance. This compution can be done very efficiently.

3.2. Post-Process the result of Census Transform

Any kind of correlation algorithm, regardless of whether it is based on the intensity or rank, does not work well in very smooth area, because searching a match in a homogeneous area is not meaningful. Hence the disparities computed at smooth regions are not reliable. Post process of the results from the Census Transform is thus necessary.

The basic idea of our post-process is to use the disparity of the boundary pixels as the disparity of the entire region. From the color segmentation, we obtain a list of regions with very similar reflectance. For each of these regions, we find the most popular disparity of the pixels on and around its boundary, and assign all the pixels inside the region with that disparity. The reason we use the most popular disparity of boundary pixels rather than the average is that the former is more noise tolerant. By using this process, the reliability of the optical flow result is enhanced.

3.3. Mask Generation of Moving Objects

3.3.1. From disparity image to mask image of all objects

The result from the post-processed disparity image is still quite noisy as shown in Figure 3. The noisy pixels are generally due to lighting changes, illumination changes and false matches in the computation of disparity images. We need to obtain homogenous regions as masks of moving objects. Morphological filter is applied to the initial noisy mask image.

The basic morphological operations are dilation and erosion. Dilation is the process of enlarging the original image. Denote A as the original image, B as the structuring element. The dilation of image A by structuring element B denoted by $A\oplus$B, is defined as

$$A \oplus B = \{x | (\hat{B}_x \cap A) \neq \emptyset\}$$

where $\hat{B}_x$ is the reflection of B shifted by x.

Erosion is the process of shrinking the original image. It is similarly defined as

$$A \ominus B = \{x|(B)_x \subseteq A\}$$

Mophological filter is defined as

$$(A \circ B) \bullet B$$

where $A \circ B$ is defined as $(A \ominus B) \oplus B$, and $A \bullet B$ is defined as $(A \oplus B) \ominus B$. Morphological filter eliminates background noise and fills small gaps inside an object. This property makes it well suited to our objective since we are interested in generating masks which preserve the object boundary. As shown in Figure 4, through morphological filter, small gaps inside an object boundary are filled in, and background noise is removed.

3.3.2. Extraction of masks for each moving objects

The extraction of mask for each moving object is done by the following morphological operation:

$$X_k = (X_{k-1} \oplus B) \cap A \quad k = 1, 2, 3 \cdots$$

X_0 contains a point of the connected component representing a given object. The iterative process finds all pixels belonging to that component. As each mask is extracted, it also gets labeled so that the fusion process can take it as initial input.

3.3.3. Dealing with objects in the same connected component

There is a special case to be considered where different moving objects are close or even connected in the image. If the moving objects are just close and not physically connected, opening operation of the morphological operation will usually disconnect the moving objects. But if the objects are connected, we need to treat it as a special case. Our approach is to compute the histogram of the disparity image, and find the threshold(s) to partition one mask into multiple masks, each with similar disparity.

4. Fusion of Motion Mask and Color Segmentation Regions

Our goal now is to fuse a list of regions obtained from color segmentation with the mask(s) obtained from motion segmentation to produce a new mask (masks) which contain(s) only the regions that belong to the moving object(s). Dubuisson and Jain proposed a way of fusion motion mask with color segmentaion [2]. Based on their idea, we propose a fusion scheme extended to be used with multiple masks as well as a single mask. For each region R_i from color segmentation, we use the following criterion to decide the relationship between a region and mask:

1. If over 90% pixels of the region lie inside a mask, then the entire region belongs to the mask;
2. If less than 10% of its pixels are inside one mask, then the entire region does not belong to the mask;
3. If neither of the above, the decision will be made by using snake algorithm.

Snake algorithm is proposed by Kass $et\ al$ [4]. It is an energy minimizing spline guided by external constraints. These constraining forces along with internal forces are combined into an expression of the total energy which needs to be minimized. The total energy of an active contour in a parametric representation $v(s) = (x(s), y(s))$ is defined as:

$$E_{total} = \int (E_{int}(v(s)) + E_{image}(v(s)))ds$$

The internal energy we use is

$$E_{int}(v(s)) = \frac{\alpha|v_s(s)|^2 + \beta|v_{ss}(s)|^2}{2}$$

where $v_s(s)$ and $v_{ss}(s)$ are first and second order partial derivitives of $v(s)$, and α and β are the parameters to control the sensitivity to the first and second derivative respectively. The image force we use is

$$E_{image} = w_{line} * E_{line} + w_{edge} * E_{edge}$$

where $E_{line} = I(x, y)$, $E_{edge} = \|\nabla I(x, y)\|^2$, w_{line} and w_{edge} are the parameters to control the weight of E_{line} and E_{edge} respectively. This allows snake to track lines and edges. The snake is initialized with the boundary of a motion mask from the motion segmentation, and its final contour will be used to guide our decision for the regions falling into case (3). In the case of multiple motion masks, multiple snakes will be used, with each attracting to the boundary of one moving object.

We have experimented with several rules to determine the relationship between regions and motion masks in case (3), and found the following one leads to best results in our test.

Suppose color segmentation divides the image into regions R_i's, motion segmentation produces the masks M_j's, and for each mask M_j we run the snake algorithm by intializing the snake with the boundary of M_j to obtain a new mask S_j whose boundary is the final snake contour. Then a region R_i of case (3), which has larger than 10% and less than 90% of its pixels inside a mask M_j, belongs to the mask M_j **if** $20\% * area(R_i) \leq \|R_i \cap M_j\|$ **and** $\|R_i \cap M_j\| \leq \|R_i \cap S_j\|$. If there are multiple masks M_j's satisfy the condition, then pick the mask in which $\|R_i \cap S_j\|$ is the largest. If there is no mask M_j satisfying the condition, then R_i belongs to the background. The intuition behind this is if the region has reasonable amount of intersection with the original mask, and running snake algorithm increases the intersection even further, then it is more likely the region belongs to the mask.

5. Final Contour Extraction

After getting the final mask(s), we then extract the boundary of the mask(s). We first find the center of the mask, and then following the line radiating from the center we look for the point lying at the last transition from the label of the current mask to a different label. We do this for every 2 degrees, until we cover 360 degrees. These points are then connected using lines and will be the final contour of the moving objects.

6. Experimental Results

We test our algorithm on several image sequences. The results included in the appendix show two image sequences in our test.

The first image is a girl walking inside the campus store. As shown in Figure 2, the sequential labeling has partitioned the image into uniform regions with similar reflectance. The shape of the girl's body stands out from the background. However there are far more regions than the number of the objects in the images. (Altogether there are 5958 regions.) Clearly this result needs further processing.

Figure 3 shows the disparity images in x and y directions. The left image in Figure 4 is the initial mask of the walking girl, which has noise present both in the background and inside the object. Applying morphological filter to this image, we can get a uniform mask for the moving object as shown in the right of Figure 4. The boundary of this mask is still not very accurate due to the error introduced in the computation of the disparity image and morphological distortion. Nevertheless it is good enough to be the snake initial contour.

Snake can attract to intensity edges. But the final snake contour is still not close enough to the actual boundary of the moving person, because snake balances between internal and external energy, and does not lock onto the exact object boundary as shown in the left of Figure 5. Thus further processing is necessary to improve accuracy. Fusing information obtained from snake and color segmentation technique fulfills our goal. It gives much more accurate motion boundary as shown in the middle and right of Figure 5.

The other image sequence is two men walking towards each other. Multiple moving objects in the scene makes the extraction more difficult. However, using census transform along with post processing, the moving object mask are approximated fairly well, especially it detects there are two moving objects. Using this information along with the color segmentation result, the moving people's contour are extracted as shown in the right of Figure 6.

7. Conclusion

Our contour extraction of moving objects is based on information from color segmentation, motion segmentation, and active contour. Reflectance ratio, which is photometric invariant, is first used to partition the image into regions having similar reflectance. Then invalid regions are eliminated. The valid regions are later fused with the motion mask(s) generated from the a series steps of motion segmentation by using active contour. Experiments show promise of our approach.

The key idea here is the fusion of different segmentation techniques. Good heuristics to fuse different techniques allow us to explore more information in the scene and thus achieve more accurate results.

8. Acknowledgement

This paper was done as a course project of computer vision course at Cornell. We greatly appreciate our instructor, Dr. Ramin Zabih, who has taught us fundamental concepts, and given us interesting ideas to stimulate our work. His suggestion of using cencus transform is particularly useful.

References

[1] "Using Dynamic Programming for Solving Variational Problems in Vision", Amir A. Amini, Terry E. Weymouth, and Ramesh C. Jain. IEEE transactions on Pattern Analysis and Machine Intelligence, pp. 855 - 867, 1990.

[2] "Contour Extraction of Moving Objects in Complex Outdoor Scenes", Marie-Pierre Dubuisson and Anil K. Jain, International Journal of Computer Vision, pp. 83 - 104, 1995.

[3] "Digital Image Processing", Rafael C. Gonzalez and Richard E. Woods, Addison-Wesley, 1992.

[4] "Snakes: Active Contour Models", Michael Kass, Andrew Witkin, and Demetri Terzopoulos, IEEE, pp. 259 -268, 1987.

[5] "Reflectance based object recognition", Shree K. Nayar and Ruud M. Bolle. International Journal of Computer Vision, 1996.

[6] "A Non-Parametric Approach to Visual Correspondence", Ramin Zabih and John Woodfill, IEEE PAMI.

A. Appendix

Figure 1. Original Image Sequence: left is 1st image, and right is 2nd one.

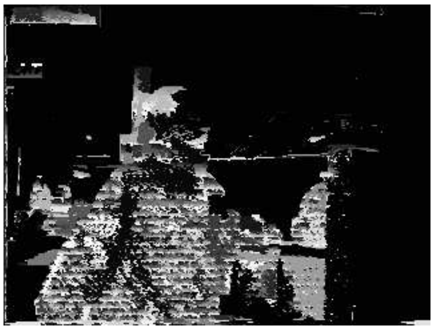

Figure 2. Sequential labeling result.

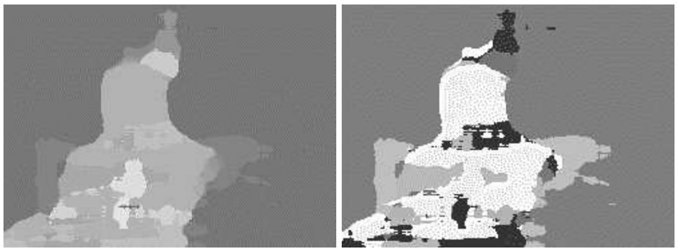

Figure 3. Left: Optical flow in x direction, scaled from -12 to 12; Right: Optical flow in y direction, scaled from -2 to 2.

Figure 4. Left: Initial Motion Mask obtained from disparity image; Right: Motion mask after morphological filter operation.

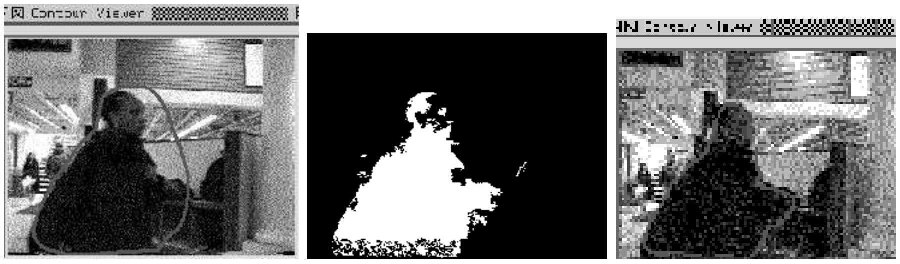

Figure 5. Left: Final position of snake contour obtained from initializing the snake with the boundary of motion mask; Middle: Result from fusion of motion mask and sequential labeling; Right: final extraction result.

Figure 6. Another Image sequence: Left and middle: the 1st and 2nd images in the original sequence; Right: final extraction result.