SEMI-AUTOMATIC SEMANTIC OBJECT EXTRACTION FOR VIDEO CODING

Zhitao Lu and W. A. Pearlman

Electrical Computer and System Engineering Department
Rensselare Polytechnic Institute Troy NY 12180
luz2, pearlw@rpi.edu

ABSTRACT

A semi-automatic algorithm to extract the semantic video object in image sequences is proposed. Different schemes are used to get the initial video object in the first frame and other frames of a sequence. In the first frame, two polygons are input by the user to specify the area in which the object boundary is located. Then the video object is extracted automatically based on only the first frame. In the following frames, the image frame is segmented into intensity homogeneous regions. The moving regions are detected by a morphological filter, non-moving regions are selected by the object model from the previous frame. These regions form the initial video object. In each frame, after the initial object is available, the edges which belong to the video object of interest are selected by a local object contour model. Finally, an active contour model (snake) is applied to extract the final object contour.

1. INTRODUCTION

In the new video coding standard MPEG-4 and the future MPEG-7, besides coding efficiency, new functionalities, such as manipulating, searching and interacting with meaningful video objects, are required. Since most digital images and video signals are in pixel format without semantic information, how to get the semantic video object in each frame becomes a very important issue.

In this work, we present a semi-automatic semantic video object extraction technique. The user inputs the approximate initial shape and location of the object. The accurate closed object contour is extracted automatically. This object contour is also used as the object model in the following frames for automatic object extraction. The advantages of this scheme are:

- No complex definition of "semantic" is needed, since it is easy for humans to identify the scope of the video object of interest.
- A fairly accurate boundary of the object can be extracted in the first frame.

2. OVERVIEW OF THE PROPOSED ALGORITHM

First, the algorithm finds the initial video object. Different schemes are used for the first frame and other frames. In the first frame, the initial object is input by the user. Two polygons, outer and inner polygon, are input to specify the shape and location of the object boundary. Then the video object is extracted automatically based on only the first frame. In other frames, each frame is segmented into intensity homogeneous regions by a region growing technique. The moving regions are detected by a morphological motion filter; and non-moving regions are selected by the object model from the previous frame. These regions form the initial video object.

After the initial object is available, edges in each frame are detected by the Canny edge detector [2]. An edge selection module is used to select edges which belong to the object of interest. The snake technique is used to find the accurate location of the boundary points of the object. Finally, a linking process forms the closed object contour by using an edge based distance transform. The flow chart of the algorithm is shown in Fig. 2. Each component in the diagram is explained in detail in the following sections.

3. FINDING THE INITIAL VIDEO OBJECT

3.1. Initial Video Object in the First Frame

In any semi-automatic technique, initial information needed from the user should be easy to input. Here two polygons are required to specify the shape and location of the object of interest. The outer (inner) polygon must be outside (inside) the video object. Another important requirement is that each line of the outer polygon approximates the local direction of the object boundary. A better result is expected if the user inputs a better initial outer polygon. We have found that 10-30 vertices are needed for all sequences used in our experiments.

3.2. Initial Video Object in Other Frames

3.2.1. Region Growing Technique

Each frame is first segmented into intensity homogeneous regions. The pixel (x, y) belongs to the homogeneous region R when it satisfies the following equation.

$$(x, y) \in R \quad \text{if } |I(x, y) - \mu_R| \leq \sigma$$

where $I(x, y)$ is the intensity of the pixel at (x, y), μ_R is the average intensity value of the region, R is the homogeneous region, and σ is the threshold. We start from one pixel p, and check all its neighborhood points. The point that satisfies the above criterion is inserted into the region. The region will expand to the whole homogeneous area until no more neighborhood points can be added. New regions are created by starting from the remaining points. This process stops when all the pixels in the frame are assigned to a certain homogeneous region.

3.2.2. Morphological Motion Filtering

The moving regions are the regions whose motion is different from the global motion. The dense optical flow field is estimated by the Horn-Schunck algorithm [4]. The global motion is modeled by a six-parameter affine model as below:

$$\hat{u}(x, y) = a_0 + a_1 * x + a_2 * y$$
$$\hat{v}(x, y) = a_3 + a_4 * x + a_5 * y$$

where $\hat{u}(x, y)$ and $\hat{v}(x, y)$ represent the motion vector at (x, y) and a_0 to a_5 are the affine parameters. The parameters a_0 to a_5 are calculated by the least median square method [7]. A morphological motion filter that removes components which do not follow the dominant global motion has been proposed in [8]. It merges the so-called flat zones in the image according to a specified criterion. Here we use the difference between the local motion vector,$u(x, y)$ and $v(x, y)$, and global motion vector as the criterion [6].

$$H(x, y) = (\hat{u}(x, y) - u(x, y))^2 + (\hat{v}(x, y) - v(x, y))^2$$

We apply this morphological motion filter to the regions obtained from the region growing process.

For non-moving regions, we match them backward to the object model in the previous frame. The regions which belong to the object in the last frame are also parts of the object in the current frame. These moving and non-moving regions form the initial video object in the current frame. Its contour is used as the initial object model in the current frame.

4. SELECTION OF EDGES BELONGING TO OBJECT

Edges in each frame are detected by the Canny edge detector. The main problem is to determine which edges belong to the video object of interest. First, we eliminate the edges which are not in the object boundary area, because we know that they are definitely not part of the object boundary. We assume that the direction between object edges and background edges is different . A local contour model, which is a line segment with length d and direction θ, is used to check if the edge has similar direction with the model or not. The process is as below:

1. For each edge point between the object boundary area, find the nearest model point.
2. Generate the local contour model according to the direction θ at this object model point.
3. Turn the local contour model by angle $n * \delta\theta$, where $-N < n < N$ and $\delta\theta$ is the least angle to turn. Check if the current edge segment matches the direction of the local contour model or not.
4. If not, go back to last step. If yes, denote the current edge point as the object edge point.
5. If the current edge segment does not match the local contour model at any angle within $\pm N * \delta\theta$, denote it as a background edge point and remove it.

5. ACTIVE CONTOUR MODELING

After the edge information has been selected, an active contour model (snake) is used to find the accurate location of the object contour.

The Snake technique was originally proposed in [5]. A snake is a set of ordered points, called control points. By moving these control points, the snake can approach any shape at any location. The behavior of the snake is determined by an energy function defined as:

$$E_{snake} = \alpha * E_{int} + \beta * E_{ext}.$$

The internal energy, E_{int}, is usually defined to determine the shape of the snake, and the external energy,E_{ext}, is used to determine the location of the snake. α and β are the weight coefficients to balance these two terms. Appropriate energy terms and the searching algorithm to find the final location of control points are two key components in the snake technique.

Two techniques are applied here to make the searching algorithm easier. First, an object edge selection module presented in the last section is used. Most unwanted edges are removed. Second, we define the external energy as the distance to the nearest edge. The distance from the edge is calculate by the distance transform [1]. With this definition for external energy, only the neighbors of the control points need to be checked, and the control points move to the neighbor with lowest value, which means nearest to the

object edge. This reduces the search area and makes the searching algorithm very simple and fast compared to the traditional external energy definition.

6. CREATE CLOSED CONTOUR OF OBJECT

After we get the final snake, the closed contour of the video object is formed by linking the discrete snake control points. During the linking process, object edge points are first chosen, otherwise the points with least Euclidean distance are chosen. We accomplish this by an edge based distance transform. The difference between it and traditional distance transform in [1] is that we assign a small value d_{edge} to the distance between two edge points, and a relative large value $d_{non-edge}$ to the distance between a non-edge point pair and an edge point to a non-edge point pair. As shown in Fig. 1. The linking process is as follows: suppose we link two control points p1 to p2.

- Set p1 as zero distance.
- Get the distance of every pixel to p1 by the edge-based distance transform.
- From p2, choose the neighbor point that has least distance to p1 as a point of the closed contour.
- Repeat last step until p1 is reached.

7. EXPERIMENT RESULT

We have used several test sequences to test the performance of the proposed algorithm. The results of sequence Akiyo and Claire are presented here. Figure 3 and Fig. 4 show the initial polygons we input for both sequences. The number of vertices of the outer and inner polygon for each sequence are shown in table 1. Figure 5 and Fig. 6 show the object contour in the first frame of sequence Akiyo and Claire. It shows that the algorithm can get accurate object contour based on only the first frame in the sequence. Figure 7 and Fig. 9 show the results for frame 10 and 30 of the sequence Akiyo. Figure 8 and Fig. 10 show the results for frame 4 and 9 of the sequence Claire.

8. CONCLUSION

A new algorithm to extract the semantic video object is proposed in this work. It generates accurate an video object model for automatic video object extraction. The local contour model is used to select the edges which belong to the interested video object. The accurate closed object contour is extracted by snake model. The use of local contour model and newly defined energy function reduce the complexity of the snake optimization. The performance of proposed algorithm is demonstrated by the experiments on several widely used test sequences.

Table 1: Number of vertices of the user input object model

Sequence	outer	inner
Claire	16	11
Akiyo	12	10

9. REFERENCES

[1] G. Borgefors,"Distance Transformations in digital images" , Computer Vision, Graphics and Image Processing, vol. 34, pp. 344-371 , 1986

[2] J. Canny, " A computational Approach to Edge Detection", IEEE Trans. PAMI, pp. 679-698, Nov. 1986

[3] C. Gu and M. C. Lee, "Semi-automatic Segmentation and Tracking of Semantic Objects", IEEE Trans. on Circuits and System for Video Technology, vol. 8, No. 5 pp. 572-584, sep. 1998

[4] B. Horn and B. Schunck, "Determining Optical Flow", Artificial Intelligence 17, pp. 185-203, 1981

[5] M. Kass, A. Witkin, D. Terzopoulos, "Snakes:Active Contour Models", International Journal of Computer Vision, pp. 321-331, 1988

[6] T. Meier, "Segmentation for Video Object Plane Extraction and Reduction of Coding Artifacts" Ph.D Thesis, Dept. of Electrical and Electronic Engineering, Univ. of Western Australia, 1988

[7] P. Rousseeuw and A. Leroy, Robust Regression and Outlier Detection, John Wiley&Sons, NewYork, NY, 1987

[8] P. Salembier, A. Oliveras, L. Garrido, "Anti-extensive Connected Operators for Image and Sequence Processing", IEEE Trans. Image Processing, vol 7, No. 4 pp. 555-570, Apr. 1998

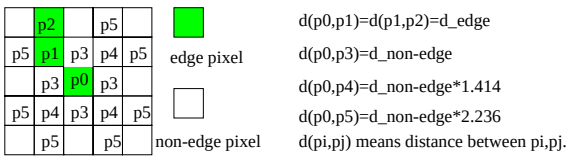

Figure 1: The Unit Distance of The Edge Based Distance Transform

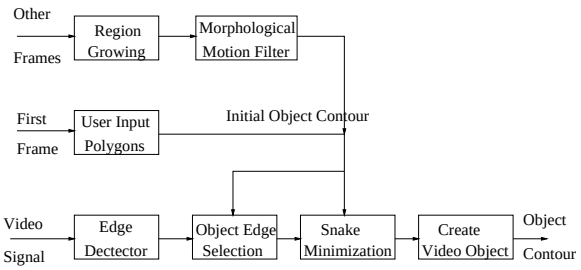

Figure 2: Diagram of The Video Object Extraction Algorithm

Figure 3: User Input Initial Polygons for The Akiyo Sequence

Figure 4: User Input Initial Polygons for The Claire Sequence

Figure 5: Object in The First Frame of The Akiyo Sequence

Figure 6: Object in The First Frame of The Claire Sequence

Figure 7: Object in Frame 10 of The Akiyo Sequence

Figure 8: Object in Frame 4 of The Claire Sequence

Figure 9: Object in Frame 30 of The Akiyo Sequence

Figure 10: Object in Frame 9 of The Claire Sequence