Interactive off-line segmentation of moving objects in real traffic conditions

Vladimír Jeleň
Dept. of Cybernetics and Artificial Intelligence, FEI TU of Košice, Slovak Republic
vladimir.jelen@tuke.sk

Abstract – Exploitation of image processing methods in traffic control is becoming more popular in recent years. This article deals with segmentation in real traffic conditions. We used modern "graph-cut" segmentation method which is able to solve more complicated situations like e.g. occluding objects.

Keywords – motion tracking, segmentation, graph-cuts, threshold, energy minimalization

I. INTRODUCTION

The number of people using personal motor vehicle to move from point A to point B is increasing and the traffic situation in big cities is near the collapse. As a result, more and more car accidents occur. One of possible solutions is to automate traffic control systems. Semi-automatic segmentation with motion prediction can help to analyze critical situations off-line.

In complicated situations like car accidents, objects or their parts are occluded. It is very difficult to segment occluded objects using simple methods, like thresholding which assumes that brightness values of pixels belonging to objects is significantly different than pixels belonging to the background. Therefore, we have to use method which can incorporate apriori information about regions, borders, occluding and shape. One of such modern methods is Graph-cuts which is able to distinguish occluded objects with the same brightness levels as the background.

II. MATERIALS AND METHODS

A. Acquisition

We used computer with 2GHz Dual Core CPU, 3GB of RAM and 64MB VGA for our segmentation method. It is a common configuration. Several types of optical cameras and webcams for image acquisition was tested in experiment. (Vivotek IP camera, Logitech Web camera…) A common industrial camera placed above a crossroad was used too.

The situation on the crossroad was recorded and saved to AVI format for next processing and it was transferred into the computer, where it was subsequently analyzed by our software.

B. Software

Our application is a plug-in module for image processing software Ellipse and it was created in Microsoft Visual Studio 2008 using C++ compiler. This plug-in module combines segmentation with low level object tracking. Classes supporting Graph-cut calculations are available on web site of Vladimir Kolmogorov [8].

C. Labeling

The application of Min-cut/Max-flow algorithm in image processing first time was described in [1]. This algorithm is very effective for combinatory optimization and for energy minimalization of many types of energy functions in computer vision.

In the next part basic information will be described about graphs and flows in the context of energy minimalization.

An oriented weighted graph \( G = (\mathcal{V}, \mathcal{E}) \) consists of a set of nodes \( \mathcal{V} \) and a set of oriented edges \( \mathcal{E} \) that represents connections between the nodes. The nodes usually represent pixels or voxels.

A graph contains several additional special nodes called terminals. These nodes are usually marked \( s \) (source) where \( s \in \mathcal{V} \) and \( t \) (sink) where \( t \in \mathcal{V} \).

Terminal nodes, in computer vision, correspond with separated sets of labeled pixels, which can be divided into the \( s-t \) categories. You can see an example of a traditional \( s-t \) graph in the figure 1. Values of edges are represented by width of line.

Usually two types of edges occur in a graph: \( n \)-links and \( t \)-links.

Fig. 1. Visualization of \( s-t \) graph with source and sink terminals. Green line represent cut on position with minimal energy.
The first one connects pairs of neighboring pixels or voxels. It is a representation of neighboring system in the picture. Weights of n-links are responsible for penalization of difference in brightness values of neighboring pixels [2] [3].

The t-links connect pixels and terminals and their values represent the classification of pixels to the nodes s and t.

D. Segmentation

If G = (V, E) is undirected bipartite graph defined as set of nodes or vertices and edges ergo connections of neighboring pixels. Then it is possible to describe each pair of connected nodes in this graph by edge e = [p,q] ∈ E.

The nodes in our case represent pixels and we have two terminals in the graph. Source terminal tagged as S is the representation of all pixels corresponding with an object in the picture and Sink terminal tagged as T which is the representation of the background. Each pixel is connected to both S and T terminals.

Non-negative weights w_e are assigned to each of the edges e ∈ E which contain n-links and t-links.

The s-t cut C ∈ E is a subset of edges in a graph. The cut divides nodes between terminals of a graph G(C) = (V, E\C). This divides the picture to object and background. The energy cost is given as a sum of weights placed on the cut.

\[ |c| = \sum_{e \in E} w_e \]

In our case n-links are placed on the boundary of the object to be segmented therefore their costs represent the cost of that boundary. On the other hand the separated t-links represent regional properties of the segment. Ergo cut with minimal cost is balance between border and region properties.

E. Interactive segmentation based on Graph-Cuts

The user marks regions in the picture that belongs to the object and regions that definitely are parts of the background [5] [4]. Number of marked regions depends on the user and the type of the task. Segmentation consists of three parts:

1. For each pixel inside the object is given a value that represents similarity of its intensity with the model. Low values represent better results.
2. For each pixel inside the background is given the value that represent similarity of its intensity with the model. Low values represent better results.
3. For each pair of neighboring pixels, where one is inside the object and the second is outside, is given a value that represents the similarity of intensities of both pixels. Low values represent their similarity.

Let it p be a pixel from the set of pixels P and A_p=0 or 1 indicating that p is part of the background or the object.

Let it \( R_{p}(A_p) \) be a function of similarity of pixels p (with values 0 and 1).

Let \( B_{p,q} \) is the variance of the intensity changes of pixels p, q.

Then segmentation value is given by:

\[ E = \mu \sum_{p \in P} R_{p}(A_p) + \sum_{(p,q) \in N: A_p \neq A_q} B_{p,q} \]

where N is the set of neighboring pixels. The first part of the equation represents the regional property and the second part represents the smoothness of the border and its continuity.

III. IMAGE ANALYSIS AND MOTION TRACKING

A. Image analysis strategy

The first step is conversion of video signal into the stack of images. Each video frame is converted into 8-bit grayscale picture, because information about color is not important and this operation increases rapidly the efficiency of image processing in next step. We proceed with segmentation after this necessary preprocessing.

The user sets a seed line in the first two images from stack. That line must belong to objects. These two regions are used by the algorithm to automatically predict the motion and to calculate regions around the line that will be used for segmentation.

![Fig. 2:](image)

This setting defines three regions that represent:

- Part of the object (it is the expansion of the seed line and it is shown on Fig.2, it is the region labeled by “1”)
- Part of the background (it is the frame of the scanning region shown on Fig.2, it is frame labeled by “2”)
- Unallocated region (it is shown in Fig.2 and it is labeled by “0”)

The algorithm finds the best border between regions representing the object and background in unallocated region labeled by zeros.

B. Motion tracking strategy

Prediction is the main feature typical for motion tracking. Extrapolation is the simplest form of trajectory prediction where the position of the object in the next frame is given by the position on previous frame shifted by vector (dx, dy) representing the differences of positions on last two frames. There are two important parameters in the model.

\[ \ddot{v} = (x, y) \]

where x, y defines the position of point B.
The difference vector
\[ \vec{d} = (dx, dy) \]
this represents differences of individual parameters in last two frames. The predicted values determining the model position, rotation, shape for the present and for the next frame are given
\[ \vec{v}_{t+1} = \vec{v}_t + \vec{d} \]

Interactive control of the program is a clear request of people who analyze images [6], [7] of traffic. Program should work as autonomously as possible but under the control of human operator. In critical situations user should be able to correct the algorithm by entering a new corrected seed region.

IV. EXPERIMENTAL RESULTS

We found that the graph-cuts based method is Nx slower than the method based on thresholding, but it is more precise. Their comparisons are shown in fig.4 (graph-cuts) and fig.3 (thresholding). The latter method is not useful to analyze of situation on crossroads.

V. CONCLUSION

Segmentation of real traffic conditions is a very complex problem. It is necessary to consider many factors like natural constrains, quality of recording, objects size or their properties and many similar factors. In this contribution we developed a simple method which is able to trace moving objects off line. It can be useful e.g. for analysis of car accidents where the contour of studied object is detected. First pair of series requires manual labeling of seed region, then algorithm finds the contours on the subsequent frames automatically. Graph-Cuts method in the context of computer vision is robust to situations like occluding parts of objects where simple segmentation methods based on thresholding fail.

We plan to increase the efficiency of the algorithm and fully automate the process of segmentation in the future.

ACKNOWLEDGMENT

This work was supported by Research and Development Support Agency under project APVV-0682-07.

REFERENCES