SEMANTIC LABELING OF TRACK EVENTS USING TIME SERIES SEGMENTATION AND SHAPE ANALYSIS

Josh Harguess and J. K. Aggarwal

Computer & Vision Research Center / Department of ECE
The University of Texas at Austin
{harguess, aggarwaljk}@mail.utexas.edu

ABSTRACT

This paper presents a novel framework for applying semantic labels to events within a track. A track is a two-dimensional (2D) or a three-dimensional (3D) signal in time where each point of the signal is the x and y (and z) centroid spatial coordinate of an object at a specific frame of the video. The track may be generated by the movement of a vehicle, person, or object. In the 2D case, the signal is decomposed into x and y time series for use in one-dimensional time series segmentations. Then the results of the two segmentations are combined to produce a 2D signal segmentation of the track which results in unique events to be labeled. The Procrustes measure, from shape analysis, is employed along with template matching to find the most likely trajectory of each individual event. Once each event is labeled with a semantic description from the template, we enhance the label using other basic measurements based on the track. The application of our framework on 4 vehicle tracks from original videos is shown to display the efficacy of our method.

Index Terms— time series, signal resolution, image shape analysis, shape measurement

1. INTRODUCTION

Much attention has been paid to methods for identifying atomic actions of a vehicle (or other object) in a video scene, such as if a right or left turn has been made. Previous methods of event labeling have used statistical shape theory and Autoregressive and Moving Average (ARMA) for activity recognition [1]. However, little attention has been paid to trying to semantically describe a more complex vehicle track or path. Unmanned aerial vehicles (UAV), global positioning satellite (GPS) tracking devices, and other sensors have the ability to track vehicles over a long period of time recording many complex activities. Currently, human operators are responsible for describing an object's actions in a complex video track. We propose a framework for automatically segmenting an object's track into meaningful events and applying semantic labels to those events within a given track.

In this paper, we define a track of an object to be a two- or three-dimensional (2D or 3D) signal in time where each discrete value of the signal is the x and y (and z) spatial coordinate of the centroid of the object at a specific frame in a video sequence. Therefore, the framework is not limited to tracks from a video sequence and can be used on other datasets where the spatial and temporal coordinates of an object are known, such as GPS coordinates.

The objective of the proposed framework is to segment a track into multiple, meaningful, 'unique' events. The Procrustes distance is used to find the best template match for each segment. A semantic label is then applied to each event, such as 'turned right', 'straight', etc. Additionally, since the spatial and temporal coordinates of the track are assumed to be known, we can perform other measurements to enhance the semantic description of each event, such as the distance traveled and the change in the object's orientation.

2. TRACK SEGMENTATION

Given a set of spatial and temporal coordinates of an object, the first step in applying semantic labels to events is to segment the track into unique events that occur within the track. In general, one wishes to identify a complete (including every coordinate of the track) and unique (non-overlapping) segmentation of the track. This segmentation will include spatial segments, such as maneuvers made by the object, and temporal segments, such as when the object has stopped moving.

We first identify temporal segments, such as an object that has stopped movement, by searching for frames where the x and y positions do not change (or the total change is less than a predefined ϵ) from one frame to the next. Once these frames are identified, the x and y points corresponding to these frames are labeled as 'stopped' and the remaining points are grouped sequentially into one or more tracks for segmentation and labeling.

Next, track segmentation on the remaining spatial coor-

This work was funded in part by the Air Force Office of Scientific Research under contract FA9550-07-C-0021, as a subcontract under 21st Century Technologies, Inc. Its release has been approved by the AFOSR: contact the corresponding author for details.

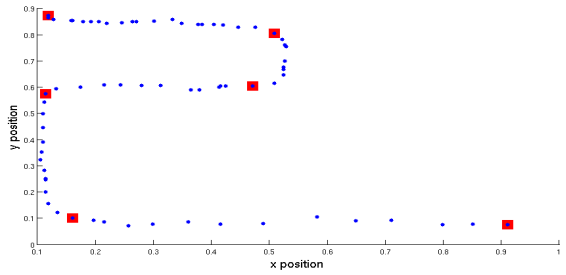

Fig. 1. 2D track segmentation of simulated data.

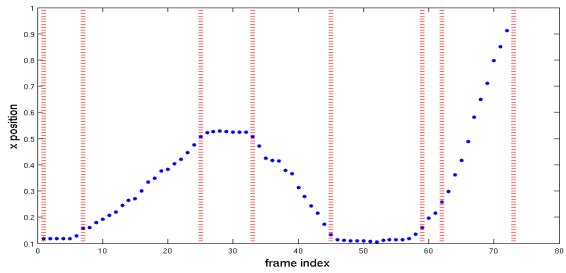

Fig. 2. X time series segmentation.

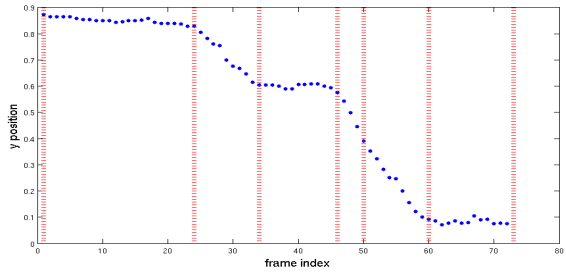

Fig. 3. Y time series segmentation.

dinates is performed. For simplification, let us assume that the coordinates are 2D, such as in the example track of an object shown in Figure 1. The proposed framework may be extended to segment and label tracks of 3 spatial dimensions. Our method is based on decomposing the 2D track into two one-dimensional (1D) x, and y time signals. The resulting 1D signals are segmented using a time series segmentation algorithm. Galati and Simaan [2] implemented an algorithm known as ALESDA (Automatic Least Squares Error Decomposition Algorithm) which performs an automatic decomposition of time series into step, ramp, and impulse primitives. Motivated by this work and others, Lemire [3] created an optimal dynamic programming solution to the problem of piecewise linear time series segmentation. The author proposes an adaptive time series model that allows the polynomial degree of each interval to vary. This leads to an algorithm that uses two parameters to segment the time series optimally. The first parameter is the maximum degree of the polynomial used in each interval, which is used in the algorithm as a penalty so that a lower degree of polynomial is chosen when possible. The second parameter is the maximum number of segments which helps to determine the scale of the segmentation. In our implementation of Lemire's algorithm, only constant and linear functions (maximum order of 1) are used. Figures 2 and 3 display the segmented x and y 1D signals obtained from the example 2D track in Figure 1.

Once each 1D signal has been segmented, we combine the results to produce a segmentation for the original 2D track. Our approach is to use a sliding temporal window of size w (determined by test data) and average the results (starting and stopping times of each segment) found in both the x and y segmentations that fall within the window. The final 2D segmentation may be thought of as a smoothed combination of the x and y time series segmentations and is representative of the natural segments as seen by human vision.

The result of the 2D track segmentation of the example track can be seen in Figure 1, where the segmentation of the track is identified by red squares. As seen from Figure 1, the 2D track has been appropriately segmented into natural segments.

3. EVENT LABELING

We now have a 2D track which has been segmented both temporally (stopped events) and spatially. These segments include the x and y coordinates and the start and stop frames of each segment.

A template matching scheme is used for labeling the segments, since we would like to observe and label only a finite set of atomic events within each track. The atomic events (and thus labels) include *travel straight, turn right, turn left, u-turn left, u-turn right, change lanes right, change lanes left,* and *stopped*. Template matching is a flexible method for classifying events, so more maneuvers may be added to this set to obtain a finer description of the track behavior. All of the above atomic events can be described in a purely spatial context (given the correct scaling and time information) except for 'stopped', which has been labeled in the segmentation portion of our framework. Figure 4 displays the templates that were generated for the experiments in this paper (assuming 2D data where the first frame index is at the origin).

Before performing template matching, we interpolate each 2D maneuver with a cubic spline. Both the input event from our segmentation and the template maneuvers are interpolated so that one can compare the input maneuver with the templates using the same number of points.

3.1. Procrustes Distance

Several researchers [1, 4] have used Hidden Markov Models (HMM), ARMA, and/or shape analysis to characterize and

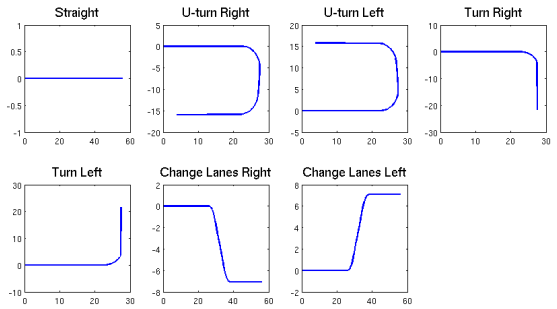

Fig. 4. Example template maneuvers.

recognize trajectories of objects. The motivation for using the Procrustes distance measure in this paper comes from [1], where the authors applied shape analysis to recognizing group activity. Other authors have used shape analysis and the Procrustes distance for various recognition activities, such as human gait [5].

In [6] shape is referred to as "the geometrical information that remains when location, scale and rotational effects are filtered out from an object." In our case, we are representing shape as a collection of points, or landmarks. Each finite number of ordered points (such as an input sequence from a single event) constitutes a shape.

Let $\boldsymbol{U}$ and $\boldsymbol{V}$ be two configuration matrices of size $N \times 2$, where N is the number of points, $\boldsymbol{U}$ is the collection of coordinates from the input sequence, and $\boldsymbol{V}$ is the stored template. The method begins by translating both configuration matrices so that their centroids are located at the origin. Then each point is expressed as a complex number with the x coordinate as the real component and the y coordinate as the imaginary component. (i.e. our configuration matrix $\boldsymbol{U}$ of size $N \times 2$ becomes vector $\boldsymbol{u}$ of size $N \times 1$ complex numbers). We will now perform 2D Procrustes Analysis, or planar Procrustes Analysis on the resulting vectors. The full Procrustes fit of $\boldsymbol{u}$ onto $\boldsymbol{v}$ is

$$\boldsymbol{u}^{Proc.} = (a + ib)\mathbf{1}_N + \beta e^{i\theta}\boldsymbol{u}, \tag{1}$$

where $\mathbf{1}_N$ is a vector of ones of length N and the parameters a, b, β and θ are chosen to minimize

$$\|\boldsymbol{v} - \beta e^{i\theta}\boldsymbol{u} - (a + ib)\mathbf{1}_N\|^2. \tag{2}$$

The resulting parameters from this minimization are

$$a + ib = 0, \tag{3}$$

$$\theta = arg(\boldsymbol{u} * \boldsymbol{v}), \tag{4}$$

$$\beta = \frac{(\boldsymbol{u} * \boldsymbol{v}\boldsymbol{v} * \boldsymbol{u})^{1/2}}{\boldsymbol{u} * \boldsymbol{u}}. \tag{5}$$

For more information on the derivation and an example, refer to [6].

This full Procrustes fit allows us to use a metric known as full Procrustes distance, which calculates the distance between the configurations in shape space. The full Procrustes distance between two complex configurations $\boldsymbol{u}$ and $\boldsymbol{v}$ is given by

$$\begin{aligned} d_F(\boldsymbol{u}, \boldsymbol{v}) &= \inf_{\beta,\theta,a,b} \left\| \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|} - \frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}\beta e^{i\theta} - a - ib \right\| \\ &= \left(1 - \frac{\boldsymbol{v} * \boldsymbol{u}\boldsymbol{u} * \boldsymbol{v}}{\boldsymbol{u} * \boldsymbol{u}\boldsymbol{v} * \boldsymbol{v}}\right)^{1/2}. \end{aligned} \tag{6}$$

This full Procrustes distance fulfills our comparison metric requirements for template matching since the measure prescales the vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ to unit size based on scale, rotation and translation [6].

4. ADDITIONAL MEASUREMENTS

Given the spatial coordinates of the track as well as the start and stop frame indices of each event, several additional measurements can be easily obtained to enhance the label of each segment of an object's track. In our experiments, orientation change (the difference between the starting orientation angle and ending orientation angle of the event), the distance traveled and the elapsed time of the segment were measured and reported. Many more measurements, such as velocity and acceleration, can be easily added to the results of our framework.

The orientation change calculation may be useful because many times the segmentation and labeling will produce several consecutive straight paths that, when combined together, do not describe the path as a whole. Therefore, the orientation change can identify a turn of the vehicle that was not found during segmentation and labeling. For instance, if an object travels straight and then quickly turns left, the result could be an event labeled 'straight', an orientation change of roughly 90 degrees to the left, and then another event labeled 'straight'.

5. EXPERIMENTS

Our framework is used to segment and semantically label several video sequences of varying lengths taken from an overhead view of a parking lot. The centroid of the vehicle was labeled in each frame manually, since we assume that the object has been successfully tracked. The results of our algorithm applied to four of the tracks are shown in Figures 5 - 8. As one can see from the plots, the algorithm segmented each track and applied an appropriate semantic label to each of the segments. For instance, in Figure 8, our method identified 3 different vehicle turns, 2 different straight segments, and 1 stopped segment. The only misclassification is the 'change lanes right', which should be labeled as 'straight'. Therefore,

Track #	1	2	3	4
# of Segments	5	8	5	7
# Correct	4	7	4	6

Table 1. Accuracy of labeled segments in each of the 4 tracks.

we accurately classified 6 of the 7 segments for track 4. The recognition results on all tracks are displayed in Table 5. The overall accuracy was found to be 84%. However, even in instances where the framework did not get the semantic label correct, there is enough information present to correctly identify the track of the vehicle. This is apparent from Figure 8 where the straight segments have orientation changes that correspond to the correct track of the object.

6. CONCLUSION

We have developed a framework for applying semantic labels to track events. The proposed solution divides this problem into two main subproblems; 2D track segmentation and single event labeling. It is shown that our framework provides successful results on tracking sequences from four original videos. The current 2D track segmentation method relies on two parameters which are chosen by hand. Future work on this problem should include a method for the automatic generation of these segmentation parameters. We would like to integrate 3D tracks into our framework for completeness. The above framework may be integrated with vehicle tracking to build a complete system for video applications.

7. REFERENCES

[1] Y. Wang, K. Huang, and T. Tan, "Group Activity Recognition Based on ARMA Shape Sequence Modeling", International Conference on Image Processing (ICIP), 2007, pp. 209-212.

[2] D. G. Galati and M. A. Simaan, "Automatic decomposition of time series into step, ramp, and impulse primitives", Pattern Recognition, Volume 39, Issue 11, November 2006, pp. 2166-2174.

[3] D. Lemire, "A Better Alternative to Piecewise Linear Time Series Segmentation", SIAM Data Mining, 2007.

[4] N. Vaswani, A.R. Chowdhury, and R. Chellappa, "Statistical shape theory for activity modeling", IEEE International Conference on Multimedia and Expo (ICME), 2003, pp. 181-184.

[5] L. Wang, H.Z. Ning, W.M. Hu, and T.N. Tan, "Gait recognition based on procrustes shape analysis", International Conference on Image Processing (ICIP), 2002, pp. 433-436.

[6] I. L. Dryden and K. V. Mardia, "Statistical Shape Analysis", John Wiley & Sons, 1998.

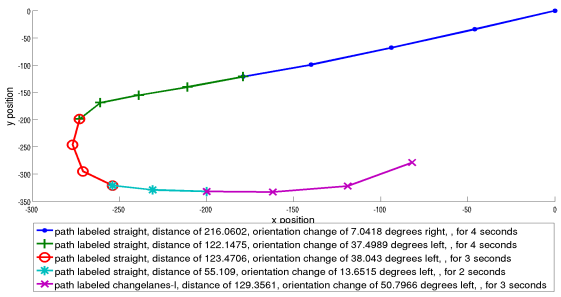

Fig. 5. Algorithm applied to track 1.

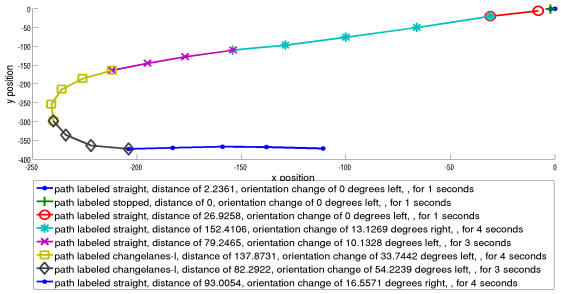

Fig. 6. Algorithm applied to track 2.

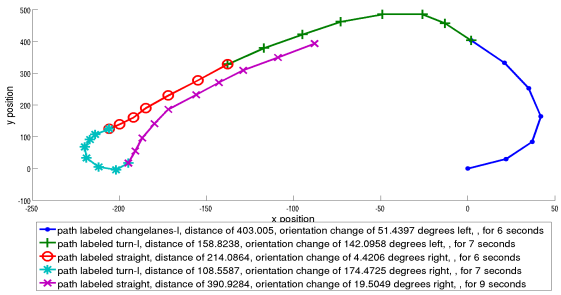

Fig. 7. Algorithm applied to track 3.

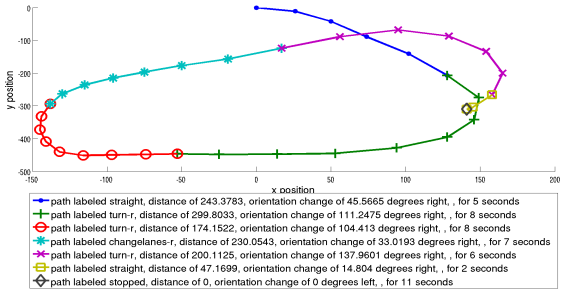

Fig. 8. Algorithm applied to track 4.