TRACKING AND ACTIVITY ANALYSIS IN WIDE AREA AERIAL SURVEILLANCE VIDEO

A Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By

Ryan K. Feather, B.S.

Graduate Program in Computer Science & Engineering

The Ohio State University

2011

Master’s Examination Committee:

Prof. James W. Davis, Adviser

Prof. Eric Fosler-Lussier
ABSTRACT

In this work, we provide tracking and activity analysis of an aerial video data set. We propose algorithms that are both scalable and able to handle the additional challenges presented by this form of data. Specifically, we present a method that consists of two main parts: track extraction and traffic activity analysis. After a preprocessing stabilization step, we use a constrained interest point matching algorithm to generate tracking data of vehicles in the scene. Finally, we present algorithms that use this data to recognize traffic activity patterns such as traffic direction, bidirectional roads, bidirectional stops, and accelerations/decelerations via analysis of average speed patterns. We provide a thorough analysis of our results, including quantitative analysis of our activity analysis algorithms.
For my parents
ACKNOWLEDGMENTS

During the course of my Master’s, I have had the pleasure of working under Professor James W. Davis. His guidance in the process of problem solving, insight into the nature of the problems at hand, and constant encouragement to expand my knowledge base have been indispensable and will surely be a positive influence in all of my future endeavors.

I would like to thank my peers in the computer vision lab. They’ve always been willing to answer questions and dispense insight when needed. I also would like to thank Matt, Michael, Karthik, Kevin, and Zhili for the ever entertaining conversations that have made this lab more than just a place to work.

I would like to thank Meredith. She has provided love and companionship for the last two years and been there to both boost and check my ego as required.

Finally, and surely most deservedly, I thank my parents Dennis and Betty Jo. They taught me the value of hard work and have unfailingly supported me in the pursuit of my interests. I know I owe my past and future successes in a large part to their involved parenting.
VITA

August 2, 1986 .......................... Born in Elkins, WV, USA

May, 2008 ............................. B.S., Computer Science
                  West Virginia University Institute of Technology, Montgomery, WV, USA

FIELDS OF STUDY

Major Field: Computer Science and Engineering

      Studies in Computer Vision: Prof. James W. Davis
TABLE OF CONTENTS

Abstract . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ii
Dedication . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . iii
Acknowledgments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . iv
Vita . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . v
List of Tables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . viii
List of Figures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ix

Chapters:

1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
   1.1 Approach . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
   1.2 Contribution and Significance . . . . . . . . . . . . . . . . . . . . . . 7
   1.3 Outline . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

2. Related Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
   2.1 Tracking Methods in Wide Area Aerial Surveillance . . . . . . . . . . . 9
   2.2 Scene Activity Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . 11
   2.3 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

3. Tracking in Wide Area Aerial Surveillance Video . . . . . . . . . . . . . . . . 14
   3.1 Stabilization . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
   3.2 Tracking . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18
      3.2.1 Motion Segmentation . . . . . . . . . . . . . . . . . . . . . . . . 18
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Results for bidirectional stop localization experiments across several track numbers</td>
<td>52</td>
</tr>
<tr>
<td>4.2</td>
<td>Results for bidirectional stop localization experiments across several track breakage probabilities</td>
<td>53</td>
</tr>
<tr>
<td>4.3</td>
<td>Results for bidirectional stop localization experiments across several observation noise levels</td>
<td>53</td>
</tr>
<tr>
<td>4.4</td>
<td>Results for bidirectional stop localization experiments with varying levels of vehicles that pass through without stopping</td>
<td>54</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>A full frame of the Greene 07 dataset. The original dimensions are about 13300x10500 pixels.</td>
</tr>
<tr>
<td>1.2</td>
<td>This cropped 500x500 patch demonstrates the small pixel area of vehicles in the Greene 07 dataset. It is shown in contrast to the portion of the data we analyze in this work.</td>
</tr>
<tr>
<td>1.3</td>
<td>System of overlapping grid cells used to allow parallelization of our tracking approach.</td>
</tr>
<tr>
<td>3.1</td>
<td>Overview of tracking pipeline from pre to post-processing.</td>
</tr>
<tr>
<td>3.2</td>
<td>The region of interest we analyze. This region of interest has a ground plane area of $3.7km^2$ and has an image size of 22 megapixels.</td>
</tr>
<tr>
<td>3.3</td>
<td>Challenges created by stitching multiple cameras. (a) Differences in intensity. (b) Misalignments.</td>
</tr>
<tr>
<td>3.4</td>
<td>Images created by plotting all track segments after length thresholds. Different shades represent individual grid cells.</td>
</tr>
<tr>
<td>3.5</td>
<td>The output of our tracking algorithm viewed as one data set.</td>
</tr>
<tr>
<td>3.6</td>
<td>Map of track density per pixel created by exponentially spatially weighting the count of objects near each pixel. This map represents roughly the number of observations near any given location.</td>
</tr>
<tr>
<td>3.7</td>
<td>Map of track speed per pixel created by exponentially spatially weighting the speed of objects near each pixel. This map represents the average speed of objects traveling near any given location.</td>
</tr>
</tbody>
</table>
3.8 Effects of grid system on tracking observations when combined with large occlusions or small cell size. (a) The combination of an overpass breaking tracks and short path length through the grid cells leads to lower track density. (b) This poorer tracking yields a decreased average speed. 28

4.1 Activity analysis pipeline from tracks and metadata to activity. 32

4.2 Road mask used to aid in activity analysis. This was manually created in image form. Indexes are assigned in the RGB channels of each pixel corresponding to the roads present at that location. 33

4.3 Direction detection. (a) Bidirectional road with two directions apparent. (b) A two bin histogram is used to measure location counts corresponding to traffic direction relative to the center line. 35

4.4 Segmentation process for a unidirectional road. 40

4.5 Segmentation process for one lane of a bidirectional road. 42

4.6 Two bidirectional stops characterized by checkerboard-like patterns. 43

4.7 Bidirectional stops give evidence of lower level acceleration zones. 44

4.8 All acceleration zones. Light boxes indicate a net increase in speed and dark a decrease. 47

4.9 Acceleration zones comprising bidirectional stops. Different shaded boxes represent different stops. 48

4.10 Results of selected ground experiments. (a) The ideal zones that our detector should find. (b) Zones outlines found with 1000 tracks in the track number experiment. White zones are accelerations and black are decelerations. (c) Results from a break probability of 30%. (d) Results from a location noise $\sigma$ of 2 pixels. (e) Results of letting 95% of traffic pass through unstopped. 57

4.11 In this diagram, white roads were the subject of analysis, black roads were not analyzed and are only provided for reference, circles indicate bidirectional stop detections, darker arrows are decelerations, and lighter arrows are accelerations. 58
4.12 Straightened road representations. Isolated black bands are due to occlusions or low track density. (a) I-675 S. (b) I-675 N. (c) I-675 S. Exit (d) I-675 N. Exit (e) Indian Ripple Road and Dorothy Ln. (f) Indian Ripple E. To I-675 S. (g) Indian Ripple W. To I-675 N. (h) Indian Ripple E. To I-675 N. (i) Indian Ripple W. To I-675 S. (j) County Line and Stroop Roads.
CHAPTER 1: INTRODUCTION

Persistent wide area aerial video is a relatively new form of surveillance. In this work we examine the Greene 07 dataset [15]. This dataset consists of frames stitched together from six sensors which capture overlapping grayscale images at a frame-rate of approximately 1.2 Hz. The resulting frames have a resolution that can exceed 10,000x10,000 pixels.
pixels. However, as is typical of high altitude aerial video, this data has a low spatial resolution. Vehicles can range in area from a few dozen to a couple of hundred pixels (see Fig. 1.2). The data is gathered from a fixed area over a timespan of about eight and a half minutes from an airplane circling an area of interest.

Analysis of this persistent aerial video lends itself to assessing activity in large areas simultaneously and automatically. Such analysis could find application in domains such as force protection, traffic management, and urban planning. Using machines to perform this task is an important goal as the number of objects to monitor and size of the area monitored prohibits manual analysis.

Figure 1.2: This cropped 500x500 patch demonstrates the small pixel area of vehicles in the Greene 07 dataset. It is show in contrast to the portion of the data we analyze in this work.
Tracking of objects and activity recognition are both heavily studied and important tasks in computer vision. Tracking of objects is the process of associating points on a specific object temporally. Activity recognition is the task of understanding the behavior of objects. While both of these activities are heavily and actively studied, the successful application of tracking techniques in persistent wide area aerial surveillance video has been limited, and, as far as we are aware, the application of activity recognition techniques in such data has thus far not been addressed in the literature. This thesis introduces a novel scalable approach to handling the challenges inherent to tracking objects in this data source and introduces novel techniques for detecting locations where specific types of traffic activity are likely to occur.

Tracking of objects in wide area aerial video presents numerous challenges in addition to those presented by standard surveillance video. Common challenges include unstable video, the size of the imagery, stitching misalignments, numerous occlusions, low spatial resolution, and intensity differences between the images composing a frame. Another phenomenon present in this video is parallax motion (the apparent motion of an object caused by viewing it from multiple lines of site). This occurs in aerial data when, due to the circular path of the airplane mounted sensor around the region of interest, the tops of relatively tall structures appear to move in a circular fashion relative to the ground plane. Additionally, a low frame-rate means that vehicles may have large displacements between frames.

Existing approaches to handle these challenges such as [13, 17] rely on graph based cost minimization schemes where blobs are matched using numerous constraints. However, accurate blob detection requires methods such as median background modeling [13], which can be computationally expensive. These methods also require some method of assignment cost minimization such as the $O(n^3)$ Hungarian algorithm. Additionally, both [13] and [17]
make use of contextual information such as derived road orientation or geographic information system (GIS) data. While this information may be useful for disambiguation of track assignments in crowded scenes, it limits the applicability of the tracker to environments where traffic flows on well-defined paths.

Our goal is to efficiently handle these challenges in a way that produces accurate tracking results suitable for behavior analysis and that makes few assumptions about the nature of the region of interest’s environment. The goal of making as few assumptions as possible is due to the potential application of wide area aerial surveillance video analysis in diverse environments ranging from remote or rural areas where GIS information such as road maps may be in inaccurate or unobtainable to suburban parking lots which are not typically detailed in maps and can contain traffic whose flow is not characterizable enough to yield helpful contextual information.

We present a tracking algorithm that handles these challenges robustly enough to provide data that activity analysis can be performed on. It is based on appearance descriptor matching augmented by motion masking and constraints based on the expected motion of real world objects. Our algorithm provides “weaker” tracks. There are often multiple tracks per object that may individually break but still collectively provide information about the full motion of the object.

As the goal of this work is to provide activity analysis in wide area activity analysis, we measure the success of our tracking approach in terms of whether it can efficiently provide suitable data for activity analysis. To this end we both assess the qualitative aspects of our collective tracking results and further validate this approach through successful application of activity analysis algorithms.
We attempt to handle the challenges of dealing with activity that occurs over large spatial areas and that must be understood through tracking data that may be sparse or contain noise. Additionally, the timespan covered is too short to attempt any long term behavior trend analysis. Given few data, we provide a set of algorithms which recognize activity patterns visible in relatively short video segments which can be both useful on their own and serve as a starting point for further analysis. To that end we aim to detect direction of traffic flow, whether roads consist of unidirectional or bidirectional traffic flow, zones where vehicles tend to accelerate or decelerate, and areas where bidirectional traffic comes to a stop. We imagine one possible application of such detection as being anomaly detection.

In this thesis, we present an approach to traffic activity pattern analysis that incorporates metadata and is based on distilling average speed and trajectory information for a given location from the tracking observations present near that location. This averaged data is then processed in a pipelined manner to obtain semantically meaningful information about roads and the activity that characterizes regions on them.

Success of our activity analysis algorithms is measured by their ability to uncover their respective intended activities from tracking observations. To measure this we discuss the qualitative results of our algorithms on the Greene 07 dataset as well as quantitative results on synthetic test data.

1.1 Approach

Our work can been divided into two main parts. The first is the extraction of tracking data from wide area aerial surveillance data. The second is the processing of this data to extract semantic information.
In the tracking stage, we first stabilize our imagery using an interest point matching algorithm. A motion segmentation mask is then generated based on normal flow. An interest point matching algorithm then uses motion based constraints to narrow and disambiguate possible matches and only attempts to detect interest points in areas of motion as indicated by the motion mask. A final cleanup step is applied to filter out tracks that are likely noisy observations. These steps are dealt with in a system of overlapping grid cells as in [13] (see Fig. 1.3). This system allows us to parallelize the algorithm and helps to isolate the negative effects of image stitching discontinuities (see Fig. 3.3). While we do not attempt to track between grid cells, we use an overlap to help avoid any “dead” spots that may occur due to missing observations.

Figure 1.3: System of overlapping grid cells used to allow parallelization of our tracking approach.
The ultimate goal of our tracking algorithm is to provide high quality information to activity analysis algorithms. We show that we meet this goal by examination of the quality of data collected.

Activity analysis is based on analyzing the average velocity information at a given location on a road. In order to facilitate this, tracking observations are first registered to the road they were detected on via an external road map. Velocity information from the registered tracking data is then mapped and averaged for each location in the region of interest. Central lines are generated based on the supplied road map and these are then used to direct sampling of road velocity data to detect lanes and traffic direction as well as create a straightened version of roads on an average speed map. These straightened roads are then converted to a set of segments based on splitting where maximums and minimums occur in a smoothed velocity profile. Velocity based criteria are used to rate whether these segments correspond to zones of acceleration/deceleration and/or bidirectional stops. Effectiveness of our algorithm is evaluated based on the known layout of the road system.

1.2 Contribution and Significance

This thesis introduces multiple novel approaches in the processing of persistent wide area aerial surveillance that allow for effective, efficient vehicle tracking and analysis of traffic patterns. Specifically the novel contributions of this work are:

• An interest point matching based tracker that operates in a parallel manner, utilizes normal flow motion segmentation, and enforces matching constraints for track disambiguation.

• An approach to detect traffic flow direction from tracking data.
• An approach to distinguish uni from bidirectional roads.

• An approach to detecting zones of acceleration/deceleration that uses an absolute velocity difference criteria to label road segments.

• An approach to identifying bidirectional stops based on averaged speed data.

These items are of particular interest to the surveillance community. Tracking algorithms for this data type are few and the behavior detection algorithms represent what is some of the first work on behavior analysis in this data source. This work is, therefore, an important step towards utilizing a potentially powerful data source.

1.3 Outline

We will cover each concept discussed above in subsequent chapters. We begin with a brief overview of related work in Chapter 2. Our methodology is then divided into one chapter for tracking, Chapter 3, and one for activity analysis, Chapter 4. In the tracking chapter, we discuss the specific details of our algorithm and the reasoning behind each step. We go on to demonstrate the success of this approach via qualitative analysis of the tracking data gathered. When discussing activity analysis, we will overview the pipeline of steps that we perform from track registration and traffic direction detection through straight road representation generation. We end with a discussion of the methodology used to detect bidirectional stops and zones where changes in speed occur followed by the results thereof. We conclude with Chapter 5 wherein we discuss the progress made in this work. We go on to suggest directions for future research.
CHAPTER 2: RELATED WORK

In this chapter we will briefly cover related work. We will first discuss work on tracking in general followed by work specific to low frame-rate wide area aerial surveillance. We then discuss activity analysis approaches specific to road traffic. Finally, we discuss more general, unsupervised approaches to scene understanding.

2.1 Tracking Methods in Wide Area Aerial Surveillance

Tracking is a highly studied and important problem in the field of computer vision. A few of the many approaches to the problem available are found in [5, 7, 9, 12, 13, 17]. We begin by discussing why several general tracking approaches may not be well suited for application to wide area aerial surveillance data and what ideas we can still use from these approaches.

One popular approach to tracking is the Kanade-Lucas-Tomasi(KLT) tracker [9]. This approach applies a gradient descent approach to tracking where optical flow is used to guide the search process. Calculation of such flow relies on close pixel proximity of an object’s location between frames. When such proximity is not available, a pyramid based approach is used where image pixel scale is reduced to effectively bring an object’s frame to frame locations closer together. As objects in wide area aerial surveillance video may often move many (over 60) pixels between frames this pyramiding would prove necessary. However, when applying such pyramiding, one must be careful the type of smoothing and subsampling used will not remove too much information about an object’s appearance. Due
to low spatial resolution, the types of objects we attempt to track are often only a few dozen pixels in area and are therefore effectively absorbed by surrounding data after only a few pyramid levels.

The authors of [12] use an efficiently calculable appearance descriptor based on covariance of numerous features along with a model update mechanism to track objects. This work requires a way to obtain an initial model. Given the wide variety of vehicles that exist, we would likely need to build many such models. Additionally, tracking on appearance description alone assumes that a sufficient amount of visual information exists, which may not be the case in the low spatial resolution data we work with.

In [5], a multi object tracking method is proposed that first detects objects and then attempts to connect detections temporally by probabilistically determining the most likely matches. This approach combines size, position, and appearance information in order to generate hypotheses about object matches and will therefore rely on these features providing distinctive information for each object. While we also would like to connect multiple object detections across time, our work focuses on data where the size and, again, appearance of many objects will likely be very similar. We therefore must include other means to distinguish between possible matches.

The KLT [9] algorithm is augmented in [14] to include programmed constraints based on expected object behavior to determine when to terminate tracks. These constraints are based on the assumed typical motion of objects of interest. We adapt this constraint based approach to augment visual information, but for reasons previously discussed, we must replace the KLT portion of this tracker with some other means to connect points on objects temporally.
The authors of [7] use interest point detection to locate objects and matches feature descriptors across frames to produce tracks. This approach is similar to the one we adapt, but the small pixel area of our objects again does not yield enough visual information to track by appearance alone.

Current approaches to track in low frame-rate, wide area data are typically focused on building tracks using motion detection schemes and cost minimization to match blobs [13, 17]. [13] performs median background modeling to allow detection of motion. An optimal frame-to-frame assignment of blobs is found by using the Hungarian algorithm and contextual information based on nearby object trajectories is employed in order to avoid tracks jumping between objects. In [17] blob detection is performed via three-frame subtraction and GIS information is used to provided contextual information to the optimal assignment scheme.

While these approaches are appropriate for gathering tracks in wide area aerial surveillance data, they assume that the consumer of tracking output will require near perfect tracking results. Our approaches to activity analysis do not require such strong tracking data which allows us to opt for more efficient tracking means. Additionally, contextual information about road structure and surrounding traffic flow plays an important role in the success of these algorithms. Our goal is to show the ability of a weaker tracking approach to provide useful information even without this added context.

2.2 Scene Activity Analysis

Many approaches aimed at gaining an understanding of the behavior (i.e. activity) of objects in scenes. An overview of approaches can be found in [11]. As our work mainly
deals with traffic flow, we will begin by discussing work that specifically deals with vehicular activity.

Gaining an understanding of road structure and general traffic behavior through activity analysis is an actively studied problem. The authors of [10] model vehicular tracks as polynomials and uses K-means to cluster tracks in polynomial coefficient space for lane detection. The work then goes on to categorize lanes as entries or exits and as belonging to primary or secondary roads. While this approach could prove useful for discovering lane directionality, it assumes a greater quality of tracking data than we believe can be readily extracted from the Greene 07 data source. Specifically, image stitching errors could cause vehicles to unexpectedly shift as much as a lane width.

The approach in [6] proposes an occlusion reasoning algorithm and learns vehicle count speed from tracking data. The occlusions that this work deals with specifically are those where one vehicle occludes another. This type of occlusion is largely not present in our data and therefore of less interest to us. Additionally, we believe it would be difficult to obtain an accurate vehicle count or per vehicle speed in wide area aerial surveillance data due to issues such as occlusion by clouds and the difficulty of obtaining accurate, strong tracks.

A popular approach to activity analysis is to attempt to learn “kinds” of activities in an unsupervised way via clustering techniques. In [8], the authors compute similarity between tracks based on a directional histogram for input to a graph based clustering algorithm. The output clusters are then split into finer level clusters using a similarity measure based on a track feature calculated by finely resampling an input track. For the resulting clusters to give an accurate representation of the underlying behavior trends, both of the features of this algorithm require that tracks accurately follow objects for many frames. We believe
that this quality of input is difficult to achieve in many surveillance scenarios, including the one dealt with in this work.

Another clustering based approach is presented in [16]. This work introduces a spatial similarity score that allows frequently broken tracks to be found similar despite existing for varying spatial distances. This work also employs a confidence measure to facilitate the comparison of varying sized objects across a scene affected by projective distortion. The effective use of such spatial information would likely require spatially dense tracking data or an increased quantity of less dense tracking data. However, our work deals with a short timespan of data with a low frame-rate, which makes the extraction of such data difficult.

Our goal of recognizing activity from weak tracking data is justified by the success of such attempts in prior work. In [14] the author’s propose a method that uses weak tracks as input to a method that clusters a state space using information about track origin and destinations as well as temporal similarity. This approach is employed on more typical surveillance videos where a higher tracking data density is available than we can expect.

2.3 Conclusion

This review of related work shows that the application of tracking algorithms to the kind of video dealt with here is still in its infancy, and, while activity analysis from surveillance video is a popular research subject, activity analysis on this data is still largely unexplored. Due to growing interest in working with this variety of data, it is the intent of this thesis to begin to address this gap. Our work will do so by introducing a tracking algorithm that produces weak tracking data that is more readily gathered than traditional, stronger tracking output. This weaker tracking input will be used as input to algorithms that are able to effectively detect activities in limited wide area aerial surveillance data.
CHAPTER 3: TRACKING IN WIDE AREA AERIAL SURVEILLANCE VIDEO

In this chapter we introduce our algorithm for tracking in wide area aerial surveillance data. This algorithm is able to produce enough accurate tracking data to perform activity analysis. An overview of the pipeline can be found in Fig. 3.1.

We start by discussing the preprocessing stabilization steps necessary for the specific dataset we work with. We then discuss our interest point matching algorithm and how it overcomes the challenges present in this dataset. We briefly discuss how we filter the resulting tracking data before activity analysis. We end with a discussion of the results of applying our algorithm to the Greene 07 dataset.

In tracking and throughout the rest of this thesis we will be working with a subset of the Greene 07 dataset extracted using the GDAL tool set. This region of interest is approximately 4700x4700 pixels and covers an area of approximately $3.7km^2$ (see Fig. 3.2).

3.1 Stabilization

To produce accurate tracking results, we first must stabilize the frames of the Greene 07 dataset. In order to find suitable point-wise relationships to generate a transformation from, we proceed by matching SURF [2] interest point descriptions between frames. The SURF algorithm is a popular interest point detection and description approach. It first finds the locations of blob-like structures in an image via use of a Hessian based point locator. A
Figure 3.1: Overview of tracking pipeline from pre to post-processing.

Pixel size based scale is also assigned to these structures. A description vector is then built from a region around the located points with the area of the region based on the assigned scale. This vector contains summations of gradient approximations calculated for multiple regions near the interest point. Description vectors can be compared by Euclidean distance to assign matches to points.

We observed that there is a possibility of SURF interest points clustering about certain highly structured areas and not providing even coverage across the entire scene. In order to
counter this, a grid system is used where only one point is randomly selected from each of evenly spaced grid cells on a particular section.

Given interest points, RANSAC [4] is used to robustly fit a transformation model for a patch between frames. The RANSAC algorithm takes as input data points, a type of model to fit, and a few other tuning parameters. In our case, the data points consist of
matched pairs of interest point locations. After experimentation, the best model for the stabilization needs in this data was found to be one that only includes rotation, translation, and scaling parameters. These were the sorts of transformations observed and any more complicated model was observed to produce unsatisfactory results, possibly due to the effects of imperfectly stitched input imagery (see Fig. 3.3(b)).

The RANSAC algorithm attempts to discriminate inlier data points from outliers. In our case inliers would correspond to point matches on stationary or ground plane objects. Outliers are all others. The algorithm iteratively selects a relatively small subset of test data points and fits a model to these points. An error measure is then calculated for each data point based on the fit model. Those points with a low enough associated error are chosen as inlier points. If there are enough inlier points and the mean associated error is the lowest so far, a new model is fit to the inlier data points. The output of the algorithm is the best model fit after some number of trials.
After a transformation is found for a frame, that image is transformed to match the original frame in the patch. This stabilization is performed in a gridded fashion as previously mentioned. It is worth noting that larger grid overlaps are used in stabilization than in tracking in order to assure that there is no missing data between patches.

It is important to remember that, in the general case, stabilization is an optional step. Future data sources may well be expected to come in a pre-stabilized form in which case this step could be easily omitted from the pipeline.

3.2 Tracking

After stabilized imagery is obtained, the following tracking algorithm is applied. It involves use of a motion segmentation mask followed by use of SURF interest point detection and description to provide appearance models that can be matched between frames in a constrained manner. The stabilized image patches are once again processed independently.

3.2.1 Motion Segmentation

Motion segmentation is a common technique to help distinguish moving foreground objects from the background in video where the background is static. In our tracking algorithm, it is used to help reduce the number of possibly confusing point detections by limiting the locations that are valid for interest points to be tracked from. Areas of possible vehicular motion are those such that

\[
\left| \frac{I_t}{\sqrt{I_x^2 + I_y^2}} \right| > m
\]

(3.1)

where \(I_t\) is the temporal derivative at an image location, \(I_x\) and \(I_y\) are the spatial derivatives at that location, and \(m\) is the minimum amount of absolute normal flow considered to indicate valid motion. This absolute normal flow criteria is used to filter out false detections.
caused by buildings, which could potentially generate small amounts of motion due to parallax. This motion typically generates minute normal flow magnitudes which are easier to suppress than the potentially large pixel intensity differences which may be generated by the same behavior. We found this normal flow based technique to produce better results in our data than regular image differencing, though we recognize that the buildings we deal with are only a few stories high and taller structures may create larger flow magnitudes which cannot be easily distinguished from the sort generated by vehicular traffic.

3.2.2 Interest Point Matching and Track Behavior Constraints

Given a motion segmentation mask, we perform interest point matching based tracking. In our interest point matching based tracking, SURF’s Fast Hessian interest points are detected in both frame $F_i$ and frame $F_{i+1}$, and feature vectors are calculated for points in both $F_i$ and $F_{i+1}$. In this step we could use the interest points detected for stabilization instead of detecting points again. We instead choose to describe this step as if we are detecting again to enforce the removability of the stabilization step from the rest of the pipeline given adequate data. After detection, distance is calculated between the feature vectors of each point in $F_i$ and each point in $F_{i+1}$ in order to find the points which match best and, as a result, find the motion of objects between $F_i$ and $F_{i+1}$. In this framework, there is no requirement that every point from either $F_i$ or $F_{i+1}$ has a match. It should be noted that, since we will ultimately use tracking output in a frame-to-frame manner, our goal is not to produce long unbroken tracks. We therefore do not attempt to match interest points between patches or provide any location based “track stitching” after track collection.

One weakness of interest point based tracking is that many points in a given frame may have similar appearance feature descriptions. While we recognize the validity of using
contextual information, such as GIS, to help resolve this problem and believe it could serve as useful prior information, we would like to demonstrate the ability of a weak tracking approach to provide useful input to an activity analysis algorithm without this additional information. Instead of such detailed context, we rely on several heuristics. These include limiting valid points to those within motion regions, limiting matches by encoding domain knowledge about the objects being tracked into the matching algorithm, and disambiguating cases where multiple $F_i$ points match to a single $F_{i+1}$ point using assumptions about track behavior. Each of these will be described in turn.

In the first step of the interest point matching framework, the motion segmentation mask is used to limit the locations in which interest points can be detected in each frame. The motion segmentation mask for a given frame must necessarily contain information about movement from the previous frame and to the next frame when available. For example, interest points in $F_1$ can be detected anywhere in the motion segmentation mask between $F_1$ and $F_2$, which we will denote as $M_{1\to2}$. For any given $F_i, i > 1$ the area where input points can be detected, denoted as $A_i$, is described by

$$A_i = M_{i-1\to i} \cup M_{i\to i+1}. \quad (3.2)$$

This process serves to potentially speed up the detection stage, limit the number of descriptions needed, and narrow down the search space for matching.

There are some assumptions about object motion that have been encoded into the matching the step. Along with using the Laplacian to limit matches as proposed in [2], all points are subject to a maximum matching distance. If a potential match point is greater than this Euclidean spatial distance away, it is not even considered for matching. The distance chosen is dependent on whether the track has a motion history, i.e. there is a velocity vector calculated from the last two observations. If such a history is not available, a default
of 100 pixels is used. This value was chosen as it corresponds to approximately 110mph, which is a reasonable upper bound for the expected speed of interstate traffic. If a track is long enough, the object’s previous velocity plus an acceleration factor is used to determine the maximum matching distance. This strategy has been observed to greatly increase the ratio of correct matches and is necessary in dense scenes. After it has been decided that two points are within a close enough distance to match, their descriptor distance is calculated. The point with the lowest distance is chosen as the best match. The ratio between the top two matches is used at this point to determine if a point has a valid match, again as proposed in [2]. It is possible that multiple input points may match to one point in the next frame. In this case we need to choose a strategy to decide which point is the correct match. To this end, if a set of points $P$ match a point $r$ from the previous frame and every track $T_j$ such that $P_j \in T_j$ has length greater than 1, we choose the correct match, $P_c$, such that

$$c = \underset{j}{\text{argmin}} \{\|r - (P_j + v_j)\|\}$$

(3.3)

where $v_j$ is the velocity calculated from the last two points in $T_j$. If such previous velocities are not available due to track lengths, the point whose SURF descriptor matches best is chosen.

After the best match for all points has been decided, a step is taken where the behavior of the track is used to determine if it is a valid track and terminate it if it is not. However, due to the nature of interest point tracking, it is possible for a track to end on what is actually a valid point to start another track. Therefore, I shall describe the process for both breaking and terminating tracks, indicating what reasons are valid for each.

The reason to terminate a track is often that the frame to frame displacement is too small or not. This is generally the case with background points or vehicles that have come to a stop. In contrast, it was observed that a track switching from one object to another often
creates a sharp change in velocity in one or both of the speed and direction components. Therefore, a maximum angle and a maximum deceleration criterion are employed. It often desirable to end the previous track and start a new one in the case of switching. Therefore, a track is “broken”. This entails terminating the current tracking and creating a new track with the first point being the last point of the track that was just terminated.

3.2.3 Post Tracking Track Removal

While our activity analysis algorithms will ultimately use only the frame-to-frame properties of tracks, multi-frame tracking data is still preferable to frame-to-frame optical flow. When attempting to analyze wide area aerial surveillance data, it is possible evidence of object motion may be erroneously observed due to various aspects of the data which include track stitching errors, sensor noise, moving clouds, large parallax motions, and georegistration errors. The tracking step of our algorithm is designed to stop tracking a point if it exhibits behavior indicative of this sort of error, so these forms of error are typically temporally short lived - existing for only a few frames. It is therefore reasonable to set a minimum frame-length threshold to remove many unwanted observations. We perform this step after all tracks are collected.

3.3 Results

We tested our tracking algorithm on the Greene 07 dataset. Given the goal of producing results that are, as a whole, suitable for further analysis, we believe it is valid to compare averages of the collected data to the expected results. To that end, and to facilitate further analysis, we have created several visualizations of the data. These include average speed, observation count, and velocity component maps. Each of these was created on a half
scale resolution for efficiency. The averaging maps were created by taking a spatially exponentially weighted average of the observed data in the vicinity of each pixel location.

We shall start with two images of raw tracking output. These images show track coverage of the entire region of interest.

Figure 3.4: Images created by plotting all track segments after length thresholds. Different shades represent individual grid cells.
3.3.1 Track Coverage

In Fig. 3.4 and Fig. 3.5 we see the output of our tracking algorithm. These results demonstrate how the gridded system provides full coverage of the area of interest when occlusions are not a limiting factor. These maps also begin to show how our tracking output can indicate the presence of potentially interesting behavior in a region. For example, it is possible to make out paths generated by vehicles traversing the unmapped parking lot in the upper center of Fig. 3.5.

3.3.2 Mapped Average Results

We look to average speed and track densities as one measure of the success of our tracking algorithm for data gathering. The track density map was created by adding weight for each tracking observation \( o \) to an array of locations \( D \) such that

\[
D_{i,j} = D_{i,j} + e^{-\frac{\sqrt{(i-o.y)^2+(j-o.x)^2}}{2*\sigma^2}}
\]  

(3.4)

where \( o.x \) and \( o.y \) indicate respective coordinates of the observation.

The average speed map is constructed similarly with an observation’s speed at each location weighted against other observations.

These maps give some confirmation that the tracking algorithm is gathering data as expected. The density map (see Fig. 3.6) confirms that only small amounts of noise data exist outside of the locations where we would expect activity to occur (e.g. parking lots, roads). Additionally, it is apparent that the more heavily traveled roads, such as the interstate, and Indian Ripple Road (the long horizontal road) produce more data.

From the average speed map (see Fig. 3.7), we see an expected distribution of speed values. Traffic on the interstate averages in the 60-70mph zone whereas side roads will
be below 45 or 25mph. It is apparent that some noisy observations still do exist, but that they are in the minority. These observations could often be removed with the aid of GIS information (e.g. removal of observations outside of road systems), but this approach is outside the scope of our work.
3.3.3 Limitations

- It can be observed that some noisy observations exist outside of the road system. Often these are a result of undesirable tracking due to image stitching errors which are difficult to handle through normal flow image masking or the tracking of cloud movement.
Figure 3.7: Map of track speed per pixel created by exponentially spatially weighting the speed of objects near each pixel. This map represents the average speed of objects traveling near any given location.

- The algorithm is vulnerable to environmental effects such as partial occlusions and lighting changes. This is largely a limitation on the ability of an interest point descriptor to capture robust visual information at such a low spatial resolution.
Figure 3.8: Effects of grid system on tracking observations when combined with large occlusions or small cell size. (a) The combination of an overpass breaking tracks and short path length through the grid cells leads to lower track density. (b) This poorer tracking yields a decreased average speed.

- Some vehicles at low contrast compared to the road proved difficult to detect via interest point detection. This problem is due to the reliance of the interest point detector we selected on comparison of relative intensities across pixels.

- One potential pitfall of not including a cross-grid cell stitching component to our algorithm exists. In Fig. 3.8(a) it seems that an occlusion, our discarding of short
tracks, and no long path through any overlapping grid cell combine to create a situation where it is difficult to collect valid tracking data from fast moving vehicles, thus leading to a lower track density on the upper side of the right interstate lane. This difficulty is reflected by a lower average track speed in this area (see Fig. 3.8(b)). Therefore, we believe it would be reasonable to add a stitching component in future implementations.

- One limitation of our constrained interest point matching tracking algorithm is its reliance on several tuning parameters such as an initial maximum displacement. These parameters are dependent on the nature of the data in terms of both spatial resolution and the types of activity present. For example, while our experiments were performed on a dataset containing traffic traveling in the range of speeds typical of suburban and interstate traffic, we would expect tracking to fail on vehicles traveling at high speeds on raceways or on airplanes at takeoff speed. On the other end, a much tighter set of parameters could yield potentially better results for vehicles crawling slowly over rough terrain.

### 3.3.4 General Applicability

We believe that our algorithm could be applied to other aerial surveillance datasets to achieve a similar quality of results. In particular, our tracking algorithm is especially suited to monitoring suburban and interstate traffic from a high altitude overhead view. In this case, the limiting factor was observed to be primarily the size and number of occlusions present. In other aerial surveillance situations, various parameters of our constrained interest point algorithm may need to adjusted to deal with differing spatial resolutions or different expected vehicle activities.
As a general weak tracking algorithm, our algorithm could be applied to a wider array of surveillance situations where the location of the sensor is static. However, the ability of many of our constraint parameters to effectively control tracking quality could be affected by factors such as perspective.
CHAPTER 4: ACTIVITY ANALYSIS

When attempting to understand activity patterns in low frame-rate, wide area aerial surveillance video, the quality and quantity of middle level inputs, such as tracking data, that can be extracted is an important consideration. We are constrained to using only around eight and a half minutes of data. With this data we would like to build a model of behavior that is both useful on its own and a potential stepping stone to future analysis. In order to make this model useful for applications such as anomaly detection, it should not require data from dense traffic over a long duration (i.e. hours) to learn. Additionally, we must make the assumption that this tracking data will come from an imperfect tracker which may give multiple observations per target and may produce frequently broken tracks.

Given these assumptions, we will present our attempt to detect information about the number of lanes, traffic direction, slow downs, speed ups, and bidirectional stop locations. We will discuss and justify our use of road map like metadata, including how we extract central lines for each road from this metadata. We then go on to discuss how our tracking output is used in conjunction with the central lines to discover traffic flow direction and whether a road consists of one lane or two. Next, we will discuss how, by using lane and direction information, we create straightened road speed map representations to facilitate further analysis. Then, we discuss how these straightened representations are processed to first detect zones of acceleration and deceleration and then bidirectional stops. We conclude with a discussion of the results of these efforts.

For an overview of the process, see Fig. 4.1.
Figure 4.1: Activity analysis pipeline from tracks and metadata to activity.
Our traffic activity analysis approach utilizes a road map-like mask where individual roads are given unique indexes (see Fig. 4.2). While our tracking approach was designed to be more general purpose and demonstrate the strengths of weaker tracking for activity analysis input, our activity analysis algorithms focus on traffic activity patterns occurring...
in the context of an established road system which makes the use of contextual information appropriate for this portion of our work. We do not believe that the availability of such a mask is an unreasonable assumption as GIS information may be available, and in lieu of such, many approaches exist to extract road structure information from aerial imagery (e.g. [3, 1]). Given this information and tracking output from our tracking approach, we register each tracking observation to the road it occurred on based on pixel location.

4.2 Center Lines

When we attempt to find the orientation of a road and create straightened road projections, our algorithms will need to an effective way to sample data for a given road. To do this, we need a path upon which to base a traversal. We use morphological operations to obtain such a path from the supplied road mask.

This is done by first performing a skeletonization operation. This reduces the road to a representation consisting of pixel thick connected strokes. The skeletonization algorithm may leave multiple branches at each actual end of the road as well as some smaller branches at points between. As our algorithms will require a single start and end, we iteratively use a spur removal routine to remove the branch end points until we only have two end points left. The resulting set of eight connected pixels is the center line for a given section of road. Note that a road may have multiple such lines in the event that it is split for some reason such (e.g., an overpass).

When we traverse these lines, it is often desirable to measure the local direction of line as well as a normal vector to that angle. We do not use the difference of the current point, \( p_i \), and its following point \( p_{i+1} \) when calculating the current direction. This is due to that fact that only multiples of 45°are possible when comparing eight-connected neighbors.
Instead, we use the vector

\[ W = p_{i+k} - p_{i-k} \]  \hspace{1cm} (4.1)

where \( k \) is some positive integer. In addition to allowing a greater number of possible directions, this also provides some smoothing effect. When calculating the normal \( N \) of \( W \), we always take the normal vector that would be the result of rotating \( W \) by 90° clockwise - in other words the right normal vector.

### 4.3 Road Orientations

Knowing information about the lane structure and traffic flow direction on a road is both necessary for further analysis and useful information in of itself. We now describe our method for detecting such. For any given road, it is reasonable to assume that as we

![Figure 4.3: Direction detection. (a) Bidirectional road with two directions apparent. (b) A two bin histogram is used to measure location counts corresponding to traffic direction relative to the center line.](image)

Figure 4.3: Direction detection. (a) Bidirectional road with two directions apparent. (b) A two bin histogram is used to measure location counts corresponding to traffic direction relative to the center line.
travel along it, traffic will either be flowing towards us or away from us (see Fig. 4.3(a)).

Using this assumption, we can characterize traffic flow information with two bin angular
histograms (see Fig. 4.3(b)).

These histograms are created by traversing the center line of the road using the con-
tentions described above. At each sample, a cross section of the road normal to the center
line is taken from a mapping of average direction vectors created similarly to the average
speed maps discussed previously. The angle represented at each index in this cross section
is calculated and the angle of the current center line direction is subtracted. Negative angles
that may result are re-mapped as
\[ \theta = 2\pi + \theta. \]

An angle, \( \theta \), such that \( \frac{\pi}{2} < \theta \leq \frac{3\pi}{2} \) represents traffic flow away from the current
location in the direction of the orientation line and is mapped to bin one of the histogram.
All other angles represent traffic flow towards the current location and are mapped to bin
two. We create three such histograms, \( H_l, H_r, \) and \( H_t \) where \( H_l \) represents traffic flow to
the left of the center line, \( H_r \) represents traffic flow to the right, and

\[ H_t = H_l \cup H_r. \]  \hspace{1cm} (4.2)

If

\[ \frac{\min(H_{t1}, H_{t2})}{\max(H_{t1}, H_{t2})} > s, \]  \hspace{1cm} (4.3)

where \( s \) is the minimum ratio between the bin counts and is currently set to 0.75, then the
road is considered bidirectional. In the case of bidirectional roads, the maximum value in
\( H_l \) determines whether a traveler moving in the left lane in the direction indicated by the
center line would be traveling with or against traffic. \( H_t \) is used similarly for unidirectional
roads.
4.4 Straight Road Projections

Using road centers as a guide, we would like to create a straightened road projection of roads on our average speed map to facilitate further analysis. While some distortion is unavoidable when turning a curved object into a straight one, some approximation is acceptable in this projection. It is reasonable to use such a projection as roads are essentially linear structures.

It is also desirable for these projections to have a standard orientation. As such, we have chosen to standardize such that, traffic flow moves from left to right on the projection. This is the flow on the right side for bidirectional roads. Also, for a traveler moving in the direction of traffic flow, the top of the projection would be to the left and the bottom to the right. To accomplish left to right flow, we use the previously derived road orientations to reverse any road center paths where forward traversal would result in moving into oncoming traffic.

Once we standardize the direction of road traversal, we use Algorithm 1 to sample and normalize.

In this, we again use the method of direction calculation discussed in 4.2. To provide a more accurate representation, we do not simply traverse each point in the path. Instead we use the direction vectors derived from the path to guide a unit step. Cross sections of the road normal to the direction vector are sampled from top to bottom and placed as column vectors in an array. The lengths of each sample in this array are then normalized.

Due to reasons previously discussed, we may encounter areas of weak tracking data and these areas are often accompanied by a lower track density. To provide the best observations for our analysis, we do not include in our sampling any pixel within a road where the observed track density is within the bottom 5% for that road.
Algorithm 1 Road Straightening: In this above algorithm, $M$ is the road id mask, $D$ is the region’s speed map, $C$ is the road’s center line, $id$ is the road’s identification number, $k$ is the same width discussed in 4.2, and $s$ is a sample half-width chosen to be wide enough to adequately sample our widest roads.

1: procedure STRAIGHTENROAD($M, D, C, id, k, s$)
2:    Let $SP$ be an array of height $2 \times s + 1$
3:    $D_{M \neq id} \leftarrow \text{NaN}$
4:    $i \leftarrow 1$
5:    $j \leftarrow 1$
6:    $\text{curPoint} \leftarrow C_1$
7:    while $i < \text{len}(C)$ do
8:        $d \leftarrow \text{Direction}(C, i, k)$
9:        $n \leftarrow \text{RightNormal}(d)$
10:       $pTop \leftarrow \text{curPoint} - s \times n$
11:       $pBottom \leftarrow \text{curPoint} + s \times n$
12:       Set column $j$ of $SP$ to be an interpolated, pixel wide sample of length $2 \times s + 1$
13:       taken from $D$ between $pTop$ and $pBottom$
14:       $\text{curPoint} \leftarrow \text{curPoint} + d$
15:       $h \leftarrow i$
16:       $\text{totalChecked} \leftarrow 0$
17:       while $h \leq \text{len}(C)$ & $\text{totalChecked} < k$ do
18:           if $\|\text{curPoint} - C_i\| > \|\text{curPoint} - C_h\|$ then
19:               $i \leftarrow h$
20:       end if
21:       $\text{totalChecked} \leftarrow \text{totalChecked} + 1$
22:    end while
23:    $j \leftarrow j + 1$
24:    $\text{for } j \leftarrow 1 \text{ to } \text{width}(P) \text{ do}$
25:        $\text{let } \text{curCol} \text{ be the column } j \text{ of } SP$
26:        $\text{set } m \text{ to be the index of the first value } \neq \text{NaN} \text{ in } \text{curCol}$
27:        $\text{set } n \text{ to be the index of the last value } \neq \text{NaN} \text{ in } \text{curCol}$
28:        $\text{let } h\text{NormCol} \text{ be } \text{curCol}_{m:n} \text{ interpolated to height 101}$
29:        $\text{let column } j \text{ of } P \text{ be } h\text{NormCol}$
30:    $\text{end for}$
31:    $\text{return } P$
32: $\text{end procedure}$
After sampling, the bounds of the road are found in each column and the sample is resized to a standard width. These straightened road projections are used to perform additional activity analysis.

4.5 Road Segmentation

Our bidirectional stop and acceleration/deceleration zone detection algorithms are based on the labeling of segments of lanes. One way to define such segments is by splitting the lane where minimums and maximums in average speed occur. To accomplish this, we first create a speed-profile of the lane. This is performed by taking a Gaussian smoothed average at each location in the straightened road projection. The Gaussian kernel used for this average has

$$\sigma_y = \frac{\text{roadwidth}}{6}$$

and

$$\sigma_x = 25.$$  \hspace{1cm} (4.4)

The acceleration at each point in the output vector $V$ is calculated as

$$A_i = V_{i+1} - Vi : 1 \leq i < n$$

where $n$ is the number of samples in a road projection. Zero crossings are found in $A$ where $A_i$ and $A_{i+1}$ do not share the same sign. These sign changes find many of the desired segment endpoints, however there may be situations where no zero crossings occur when an average velocity profile reaches a minimum or maximum and velocity then stays respectively slightly negative or positive, likely due to a combination of steady observation velocities and small numerical errors. To handle this case we augment our definition of zero
crossing to include $i$ such that $|A_i| > \epsilon$ and $|A_{i+1}| \leq \epsilon$ or $|A_i| \leq \epsilon$ and $|A_{i+1}| > \epsilon$ where

$$\epsilon = .01 \ast \max(A).$$

(4.7)

Figure 4.4: Segmentation process for a unidirectional road.
The values 1 and \( n \) are added to this set to include end segments. Given total set of segment endpoints, \( Z \), for a lane, the set of segments is defined as

\[
S_i = V_{j\leftarrow k} : j = Z_i, k = Z_{i+1}.
\]  
(4.8)

At each segment, we calculate the difference in speed. If the lane this segment corresponds to has a forward traffic flow, this difference is calculated as

\[
\Delta V_i = S_{im} - S_i
\]  
(4.9)

where \( m \) is the length of the segment. In the case that traffic flow is oncoming, we reverse the subtraction -

\[
\Delta V(S_i) = S_i - S_{im}.
\]  
(4.10)

This convention ensures that apparent increases in speed are treated as positive and decreases as negative. Additionally, segments are ordered depending on the traffic flow direction of the lane. Fig. 4.4 and 4.5 give an overview of the process.

4.6 Acceleration/Deceleration Detection

Given a set of segments, \( S \), we would like to detect in which segments vehicular motion can be characterized by an increase or decrease in speed. We believe that the most valid way to detect this behavior is by checking if \( |\Delta V(S_i)| > w \) where \( w \) is set to 10mph. We view this simple constraint as being an appropriate way to determine this as we can not expect any typical change pattern to characterize velocity over a segment. This difficulty in characterization is due to noisy observations and the wide variety of possible traffic patterns that could create such a change.
When the above criteria is met, the sign of $\Delta V(S_i)$ determines the difference in an acceleration and deceleration segment. A positive sign indicates acceleration, and a negative one deceleration.
4.7 Bidirectional Stop Detection

Bidirectional stops, as caused by traffic lights at crossroads, tend to give a distinctive “checkerboard” like pattern in traffic flow (see Fig. 4.6). This pattern yields itself to a detection scheme based on the identification of a particular segment pattern. In particular, our criteria are:

- A bidirectional stop consists of four total segments from two lanes on a road \((A_i, A_{i+1}, B_j, B_{j+1})\).
  Here \(A\) denotes the oncoming lane and \(B\) the other.

- The speed changes associated with the segments are such that \(\Delta V(A_i) < 0, \Delta V(A_{i+1}) > 0, \Delta V(B_j) < 0, \Delta V(B_{j+1}) > 0\).

- \(A_i \cap B_{j+1} \neq \emptyset\) and \(A_{i+1} \cap B_j \neq \emptyset\).

- \(\text{median}(|\Delta V(A_i)|, |\Delta V(A_{i+1})|, |\Delta V(B_j)|, |\Delta V(B_{j+1})|) > u\) where \(u\) is a threshold set to 10mph.
Figure 4.7: Bidirectional stops give evidence of lower level acceleration zones.

If any set of segments meet all of these criteria, the set is considered to be a unique bidirectional stop detection.

We can also use this higher level phenomenon as strong evidence of underlying, weak magnitude activities. Each section detected as part of a bidirectional stop is treated as an acceleration or deceleration detection as well (see Fig. 4.7).

4.8 Results

Our activity analysis algorithms were tested on the output of our tracking algorithm. In particular, we chose ten of the most highly traveled roads in our area of interest for analysis. We shall present an overview of the correctness of our orientation detection algorithm, which worked correctly on every instance in our test set. As the bidirectional stop and acceleration/deceleration zone detections are the primary goal of all algorithms, we shall focus on these results they provide as they demonstrate the effectiveness of the previous
algorithms such as straightening and segmentation. We shall show the success of these algorithms qualitatively by comparison to what behavior we have observed in the data and quantitatively through experiments with synthetic data. We will also discuss the modes of failure for each.

4.8.1 Orientation Detection

We measure the success of our orientation detection through two measures. One is the ratio of correct decisions on whether traffic is uni or bidirectional. The other is the ratio of correct decisions on whether traffic tends to flow towards or away from an observer traveling along our center lines. Out of the ten sample roads, both bidirectional roads were detected as such with all others detected as unidirectional for a success rate of 100%. The case of flow direction detection also yielded a 100% success rate with flow being correctly assigned as towards or away as per the discussion in 4.3.

4.8.2 Acceleration/Deceleration Detection

Quantitative measurement of the correctness of acceleration and deceleration detection is a difficult task. We define such a zone as a section of a lane where some significant portion of the vehicles that pass through that section experience a significant change in velocity. As defining a significant portion or a significant change relies on choosing some threshold, I leave this to future consideration. Instead, I will discuss the qualitative aspects of our results. Please refer to Fig. 4.8 and Fig. 4.11. Outside of bidirectional stop zones, it is clear that our algorithm is able to detect areas where the majority of vehicles will change speed significantly such as onramps and exits. In fact, there was at least one deceleration zone found on both exits (see Fig. 4.8(c,d)) as well as an acceleration on each onramp (see Fig. 4.8(f-i)). Additionally, an area of acceleration was found on the “J” shaped road where
it intersects another. Analysis of the data confirmed that there was a significant amount of traffic entering the road and accelerating at that point.

Another noteworthy aspect of the acceleration detection algorithm is the ability of our simple criterion to avoid detection of non-intended areas. In fact, the two areas with false detection, an area on the interstate (Fig. 4.8(b)) and an area at bottom end of the “J” shaped road (the right end of Fig. 4.8(j)) both correspond to areas where poor tracking data was received for reasons covered in Sec. 3.3.3.

4.8.3 Bidirectional Stop Detection

One way we can measure bidirectional stop detection success by comparing detections to the number of known stops in our data. On the roads that we analyze, there are eight bidirectional stops caused by traffic lights with one intersection being common to two roads. From Fig. 4.9 we see a total of eight with one missed set of segments and one false detection. These results correspond to correctly identifying all but one bidirectional stop present and one false detection. The missed stop comes from a section where two such stops are close enough together that track speed observations blended together in a way that masked a set of acceleration and deceleration zones. The false detection is in the bottom left portion of the map (see Fig. 4.11). We believe this corresponds to the presence of an intersection combined with poor tracking in this area due to occlusions.

An overview of our acceleration/deceleration and bidirectional stop results from the Greene 07 dataset can be seen in Fig. 4.11. We believe that detection of the sorts of activities mapped here is a good first step in understanding scene activity present in wide area aerial surveillance video.
Figure 4.8: All acceleration zones. Light boxes indicate a net increase in speed and dark a decrease.
Figure 4.9: Acceleration zones comprising bidirectional stops. Different shaded boxes represent different stops.
4.8.4 Ground Truth Experiments

While we have provided some qualitative analysis of our bidirectional stop detection algorithm, we would also like to also provide quantitative analysis. We believe this is an important step to help measure the accuracy of our activity analysis, and one that is often overlooked in the field of scene activity analysis. Particularly, we would like to measure the ability of our bidirectional stop algorithm to find the correct locations for each of the four zones involved when subjected to various possibly limiting factors. However, in order to provide such results, we need some notion the ground truth. As estimating locations where vehicles tend to start accelerating or decelerating from real data could prove difficult and possibly prone to arbitrary judgements, we instead take the approach of creating synthetic data upon which to test our algorithms. This approach also offers the advantage of allowing us to test the robustness of our algorithm by adjusting the qualities of the synthetic data.

Our synthetic data is aimed to model a road with a single bidirectional stop. Specifically, we model a road that has dimensions of 1000x70 pixels. The bidirectional stop in question has speed up and slow down zones of 200 pixel length and is designed to mimic the pattern that would be caused by intersection with road of width 100 centered at pixel 500 (see Fig. 4.10(a)).

We model the ideal paths of objects across our road as two lines equal in distance from the widthwise center of our road on opposite sides. Each path has a starting point and the starting points for these two paths are on opposite ends of the road. Each index of our model paths is assigned an average speed that an ideal object would travel at at that point. Traffic is modeled by generating tracks that move along these linear paths from start point to end point at the indicated rates. Each modeled track starts in a random location chosen uniformly randomly from the first \( n \) path indexes where \( n \) is the indicated speed at the start
of that path. In addition to a randomized start location, each tracking observation may also have its two dimensional location perturbed by noise generated from a normal distribution whose variance, $\sigma$, we may choose. Additionally, to simulate weaker tracking data, each track may also be broken randomly based upon a frequency parameter, $b$.

Once tracking data is generated, it is run through the mapping, width normalization, segmenting, and bidirectional stop detection stages of our algorithm. It should be noted that we omit road directionality testing from this setup as any failure of this step would only serve as a complicating factor that distracts from the goal of testing localization ability of our bidirectional stop detection algorithm. All of the other steps mentioned can be viewed as required preprocessing that directly affects the strength of our bidirectional stop detection method.

We run several experiments with this setup. In these, we vary the number of tracks, observation noise, and track breakage parameters. We also run experiments with track sets that contain both tracks that stop and those that do not. In each experiment except for the ones where we vary the number of tracks, we simulate a total of 1000 objects traveling in each direction of traffic flow. To measure the localization performance of our algorithm under varying conditions, we calculate precision, recall, and the F1 score. Due to the random nature of our input, each experiment was run 100 times. We report the average results calculated across all runs.

Each of our localization performance measurements is calculated in terms of the number of lengthwise road lane indexes correctly assigned to an appropriate zone. As previously discussed, there should be four total zones detected for each bidirectional stop: two acceleration zones and two deceleration zones. Collectively, precision ($p$), recall ($r$), and F1 score measurements require true positive ($tp$), false positive ($fp$), and false negative
\((fn)\) counts for calculation. The true positive count for our test is the number of indexes correctly assigned to their respective zones. The false positive is the total number of indexes incorrectly assigned to either the wrong zone or an area where there should be no zone. False negative indexes are those to which a particular zone should be assigned but is not. These could be indexes where either no zone was found, or where the incorrect zone was assigned. The formulas for these quantities are

\[
p = \frac{tp}{tp + fp} \tag{4.11}
\]
\[
r = \frac{tp}{tp + fn} \tag{4.12}
\]

and

\[
F1 = 2 \times \frac{p \times r}{p + r} \tag{4.13}
\]

We first calculate the accuracy of our detection algorithms given a varying number of objects moving from each direction. In this test, we hold the amount of track location noise constant at a \(\sigma\) of 2 pixels and the chance of a track breaking constant at 10\%. This test is meant to determine the minimum number of tracks needed to reach a localization accuracy for detection of bidirectional stop behavior. These tests should not be interpreted as being aimed at determining how many observations are needed to detect a trend in behavior as the notion of “trend” is largely dependent on the nature of a given scene.

The results of varying track number (see Tab. 4.1) show that, even with a random track start location and location observation noise, our algorithm’s accuracy (F1 Score) is largely independent of the number of tracking observations. While there is some improvement over orders of magnitude, the F1 score from 1000 objects only increases a few percent over the results from 5 objects. For this test and all following, our lowered precision scores
<table>
<thead>
<tr>
<th>Track Count</th>
<th>F1 Score</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.759</td>
<td>0.624</td>
<td>0.971</td>
</tr>
<tr>
<td>10</td>
<td>0.752</td>
<td>0.615</td>
<td>0.972</td>
</tr>
<tr>
<td>25</td>
<td>0.763</td>
<td>0.630</td>
<td>0.971</td>
</tr>
<tr>
<td>50</td>
<td>0.769</td>
<td>0.649</td>
<td>0.971</td>
</tr>
<tr>
<td>100</td>
<td>0.778</td>
<td>0.655</td>
<td>0.971</td>
</tr>
<tr>
<td>250</td>
<td>0.782</td>
<td>0.655</td>
<td>0.971</td>
</tr>
<tr>
<td>500</td>
<td>0.783</td>
<td>0.657</td>
<td>0.971</td>
</tr>
<tr>
<td>1000</td>
<td>0.783</td>
<td>0.657</td>
<td>0.970</td>
</tr>
</tbody>
</table>

Table 4.1: Results for bidirectional stop localization experiments across several track numbers.

can generally be attributed to an extension of deceleration/acceleration segments into areas where consistent speeds occur due to the smoothing involved in both mapping and calculation of speed profiles. The few percent recall error will typically be cause by ambiguity about the where a deceleration zone ends and an acceleration zone begins. In these areas, track speeds are effectively zero for a few pixels width which makes choosing the correct end/start point difficult (see Fig. 4.10(b)).

A related test is the ability of our algorithm to handle frequently broken tracks. This test was performed by removing track observations with a varying probability while holding track number and location noise steady \( (n = 1000, \sigma = 2) \). To remove points, we choose a random number between 0 and 1 when we generate each observation. If that number is less than our break probability \( b \), the next observation will be ignored. We break the current track, and start a new track when we eventually choose a number greater than \( b \). If this process results in any isolated points (i.e. single frame tracks), these points will automatically be ignored by the rest of the algorithm.
<table>
<thead>
<tr>
<th>Break Chance</th>
<th>F1 Score</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>0.784</td>
<td>0.657</td>
<td>0.971</td>
</tr>
<tr>
<td>10%</td>
<td>0.784</td>
<td>0.657</td>
<td>0.971</td>
</tr>
<tr>
<td>20%</td>
<td>0.784</td>
<td>0.657</td>
<td>0.971</td>
</tr>
<tr>
<td>30%</td>
<td>0.784</td>
<td>0.657</td>
<td>0.971</td>
</tr>
<tr>
<td>40%</td>
<td>0.783</td>
<td>0.657</td>
<td>0.971</td>
</tr>
<tr>
<td>50%</td>
<td>0.783</td>
<td>0.656</td>
<td>0.971</td>
</tr>
<tr>
<td>60%</td>
<td>0.782</td>
<td>0.654</td>
<td>0.971</td>
</tr>
<tr>
<td>70%</td>
<td>0.779</td>
<td>0.650</td>
<td>0.971</td>
</tr>
</tbody>
</table>

Table 4.2: Results for bidirectional stop localization experiments across several track breakage probabilities.

<table>
<thead>
<tr>
<th>σ</th>
<th>F1 Score</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.775</td>
<td>0.645</td>
<td>0.970</td>
</tr>
<tr>
<td>1</td>
<td>0.783</td>
<td>0.657</td>
<td>0.970</td>
</tr>
<tr>
<td>2</td>
<td>0.783</td>
<td>0.657</td>
<td>0.970</td>
</tr>
<tr>
<td>4</td>
<td>0.781</td>
<td>0.654</td>
<td>0.971</td>
</tr>
<tr>
<td>8</td>
<td>0.760</td>
<td>0.623</td>
<td>0.972</td>
</tr>
<tr>
<td>16</td>
<td>0.736</td>
<td>0.595</td>
<td>0.970</td>
</tr>
<tr>
<td>32</td>
<td>0.655</td>
<td>0.512</td>
<td>0.917</td>
</tr>
</tbody>
</table>

Table 4.3: Results for bidirectional stop localization experiments across several observation noise levels.

The results of this set of tests (see Tab. 4.2) show that our algorithm seems to not be greatly affected by even greater than a 50% probability of track breakage. This is expected due to our sole reliance on frame to frame information from tracking observations. Removal of tracking observations only amounts to a decreased amount of tracking data, which we have already shown to have little effect on the results of our algorithm.
<table>
<thead>
<tr>
<th>Pass Through Ratio</th>
<th>F1 Score</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>0.777</td>
<td>0.646</td>
<td>0.975</td>
</tr>
<tr>
<td>5%</td>
<td>0.779</td>
<td>0.649</td>
<td>0.973</td>
</tr>
<tr>
<td>10%</td>
<td>0.779</td>
<td>0.650</td>
<td>0.975</td>
</tr>
<tr>
<td>15%</td>
<td>0.781</td>
<td>0.651</td>
<td>0.978</td>
</tr>
<tr>
<td>25%</td>
<td>0.786</td>
<td>0.658</td>
<td>0.975</td>
</tr>
<tr>
<td>50%</td>
<td>0.793</td>
<td>0.672</td>
<td>0.978</td>
</tr>
<tr>
<td>75%</td>
<td>0.810</td>
<td>0.689</td>
<td>0.980</td>
</tr>
<tr>
<td>95%</td>
<td>0.839</td>
<td>0.729</td>
<td>0.988</td>
</tr>
</tbody>
</table>

Table 4.4: Results for bidirectional stop localization experiments with varying levels of vehicles that pass through without stopping.

We next calculate the accuracy of our detection algorithms given a varying level of location observation noise (see Fig. 4.10(c)). Track number and break chance are held constant ($n = 1000, b = 0.1$). The results of this test show that our localization performance is dependent on the location noise present in tracking observations (see Tab. 4.3). The precision score drops as $\sigma$ becomes large. This can be interpreted as a result of decreased segmentation quality. However, in this case, the largest $\sigma$ is equivalent to over half of the actual track speed, which would likely would be the case only in extremely noisy input data.

Our final test involves simulation of two modes of traffic: one that stops and one that passes through at an even speed. Our test involves varying the ratio of the two kinds of traffic while holding all other factors constant ($n = 1000, b = 0.1, \sigma = 2$). The results of this test (see Tab. 4.4), may seem surprising. Even up to 95% traffic pass through, we still consistently find a bidirectional stop zone, and that zone is actually better localized at higher levels (see Fig. 4.10(e)). However, these results can easily be explained.

Regarding the ability to detect zones with so many objects passing through, we must remember that we ultimately analyze with our tracking results in terms of the average value.
around a location. Slower tracks will generate more observations in an area, thus lowering the average value more than tracks from an object moving through that location at a higher rate will raise it.

The tracks that move through without stopping are not without effect, however. These tracks do serve to raise the average around the borders of our segments enough to compensate for some of the previously described loose segmentation, thus raising precision.

### 4.8.5 Limitations

- Close proximity of bidirectional stops can cause poor segmentation. This poor segmentation in turn yields missed detections of accelerations, decelerations, and bidirectional stops.

- The current acceleration/deceleration segment detector is not suited to finding zones where a small percentage of vehicles undergo a major speed change while most simply pass through.

- The acceleration/deceleration detection algorithm currently makes no distinction between a drop in speed caused by an actual deceleration and the start of a zone where oncoming traffic enters at a lower rate of speed and accelerations. This could potentially lead to false detections. The same is true of the distinction between an increase caused by actual acceleration and an apparent increase caused by the end of a zone where traffic decelerates to turn off to another road.
4.8.6 General Applicability

Our activity analysis algorithms are designed for analysis of traffic activity patterns from an overhead aerial view. As such, we believe that our algorithms will be most successfully applied on this constrained domain. Within this domain, we believe our algorithms could be successfully applied to a variety of other scenarios. In particular, once information is known about road structures, our road directionality, acceleration/deceleration detection, and bidirectional stop algorithms should all be applicable. Both the acceleration/deceleration and bidirectional stop detection algorithms were defined based on very general specifications of the types of behavior to be observed and should therefore generalize well within this domain. For example, bidirectional stop detection does not distinguish between the pattern induced by vehicles stopping at a stoplight and other stopping patterns such as that which may be induced by vehicles slowing due to a railroad crossing or an accident.
Figure 4.10: Results of selected ground experiments. (a) The ideal zones that our detector should find. (b) Zones outlines found with 1000 tracks in the track number experiment. White zones are accelerations and black are decelerations. (c) Results from a break probability of 30%. (d) Results from a location noise $\sigma$ of 2 pixels. (e) Results of letting 95% of traffic pass through unstopped.
Figure 4.11: In this diagram, white roads were the subject of analysis, black roads were not analyzed and are only provided for reference, circles indicate bidirectional stop detections, darker arrows are decelerations, and lighter arrows are accelerations.
Figure 4.12: Straightened road representations. Isolated black bands are due to occlusions or low track density. (a) I-675 S. (b) I-675 N. (c) I-675 S. Exit (d) I-675 N. Exit (e) Indian Ripple Road and Dorothy Ln. (f) Indian Ripple E. To I-675 S. (g) Indian Ripple W. To I-675 N. (h) Indian Ripple E. To I-675 N. (i) Indian Ripple W. To I-675 S. (j) County Line and Stroop Roads.
CHAPTER 5: CONCLUSION AND FUTURE WORK

In this work, we have presented novel techniques for tracking vehicles and analyzing activity in persistent wide area aerial surveillance video. We began with an introduction to the problem in Chapter 1. We briefly discussed related work in Chapter 2. Next, we described our approach to the problem of tracking in this type of data and the results thereof in Chapter 3. In Chapter 4, we presented our approach to processing and analyzing this data. We discussed approaches to traffic flow direction detection, uni versus bidirectional road discrimination, bidirectional stop detection, and acceleration/deceleration zone detection as well as the results of these approaches.

We have shown that weak tracking data can be successfully collected from low frame-rate wide area aerial surveillance data in a way that is robust to the sorts of challenges posed by such data while being highly amenable to parallel computation. We believe this is an efficient and powerful approach to gathering data for activity analysis, especially in datasets that pose a high degree of tracking challenge. We went on to show that useful activity can be learned from accumulations of weak tracking data through the results of our road orientation, acceleration/deceleration detection, and bidirectional stop detection algorithms. Overall, our work shows that it is possible to both extract useful tracking data in this new dataset and process this data to automatically detect activities that occur in a scene. We view this work as a first stepping stone to successful analysis of wide area aerial surveillance data.

We will reiterate the contributions of this work and then discuss potential future work.
5.1 Contributions

We have presented several techniques to improve tracking in persistent wide area aerial surveillance data as well as several techniques to gather information about the nature of activity occurring on roads in this data. Our contributions can be summarized as follows.

- We presented a novel approach to tracking in a highly challenging variety of video. Our approach handled low frame-rate and multiple objects by using an interest point matching approach. Our algorithm utilized a normal flow based motion segmentation to narrow the possible interest points while minimizing the amount of corruption caused by parallax motion. A system of behaviorally driven constraints were used to further narrow possible matches and disambiguate vehicle matches in crowded environments.

- We introduced a novel approach to traffic flow detection based on analysis of radial traffic direction histograms that worked perfectly on our dataset.

- We introduced a novel approach to determining whether a road is uni or bidirectional based on analysis of radial traffic directions that was able to distinguish roads with opposing lanes of traffic from one way roads.

- We introduced a novel approach to detection of bidirectional stops based on labeling of road segments that was able to detect many of the four way traffic lighted intersections in our data.

- We presented a novel approach to labeling road segments as areas of acceleration or deceleration based on a simple absolute velocity change criterion. This approach
proved especially effective at detecting the sort of activity present on onramps and exits.

5.2 Future Work

Future work in this area would likely revolve around three areas - tracking improvements, activity analysis improvements, and applications of activity analysis.

5.2.1 Tracking Improvements

While our tracker was able to provide valuable data, there are still some weaknesses that could be addressed.

- **Occlusions** The most visible cause of tracking errors seems to be caused by momentary occlusions. One possible solution would be a multi-frame matching algorithm where the tracker attempts to find the best visual match and lowest prediction error among points for up to \( n \) frames. Additionally, road structure and traffic flow history learned from some initial section of data could be used as prior knowledge to attack this problem in a probabilistic manner.

- **Speed** While we believe that our tracker could operate in a real time manner given enough CPU’s to distribute its load across, it is always desirable to be able to process more quickly on less hardware. To that end we propose a method based on the observation that areas with more traffic tend to have a larger area included in the motion segmentation region and more objects to track which yields a significantly slower frame-rate. To that end, one possibility would be a dynamic partitioning of the scene, where the shape and size of individual patches in our gridded system are
modified to decrease the area of patches containing heavy flow. This sort of system could be thought of as a “focus of attention” model.

5.2.2 Additional Activity Analysis

One weakness of the activity detection methods presented in this thesis is the inability to distinguish between two or more spatially overlapping modes of activity (e.g., start, stop, and pass through at an intersection). One way to overcome this weakness would be to performing activity recognition through a clustering approach that allows multiple types of activity to overlap spatially. However, it is our belief that any accurate clustering approach would require more data than was available to us in this work.

5.2.3 Activity Analysis Applications

We view our work as containing good building blocks for further analysis. Here we list two of the many potential research directions that could expand on our work.

• **Anomaly Detection** One direct application of our acceleration and deceleration zone detection algorithm is anomaly detection. This application would involve first building a map of these zones over a substantial period of normal traffic flow. A rolling detection window could then be used where zones are detected over shorter time periods’ worth of data. Any deviation in zone location or number could indicate anomalous traffic behavior.

• **Road Exit Discovery** Our work currently only analyzes established road networks. However, it may be that common paths exist for traffic outside of established roads. Examples include parking lot traffic and off-road activities. Often these sorts of paths will have a connection to an established road. Through analysis of differing
tracking densities and/or presence of acceleration/deceleration zones, it may possible to pinpoint where these sorts of activities occur.
BIBLIOGRAPHY


