2003-34

Final Report

Sensor-Based Ramp Monitoring

Minnesota Department of Transportation

Research

Technical Report Documentation Page

1. Report No.	2.	3. Recipients Accession No.
MN/RC 2003-34		

4. Title and Subtitle	5. Report Date
SENSOR-BASED RAMP MONITORING	May 2003
	6.
7. Author(s) Nikos Papanikolopoulos, Osama Masoud, Eric Wahlstrom	8. Performing Organization Report No.
9. Performing Organization Name and Address University of Minnesota Department of Computer Science and Engineering Artificial Intelligence, Robotics and Vision Lab 5-216 Electrical Engineering and Computer Science 200 Union Street SE Minneapolis, MN 55455	10. Project/Task/Work Unit No. 11. Contract (C) or Grant (G) No. (c) 81655 (wo) 7
12. Sponsoring Organization Name and Address Minnesota Department of Transportation 395 John Ireland Boulevard Mail Stop 330 St. Paul, Minnesota 55155	13. Type of Report and Period Covered Final Report 14. Sponsoring Agency Code

15. Supplementary Notes
http://www.lrrb.gen.mn.us/PDF/200334.pdf

16. Abstract (Limit: 200 words)

This report covers the creation of a system for monitoring vehicles in highway on-ramp queues. The initial phase of the project attempted to use a blob tracking algorithm to perform the ramp monitoring. The current system uses optical flow information to create virtual features based on trends in the optical flow. These features are clustered to form vehicle objects. These objects update themselves based on their statistics and those of other features in the image. The system has difficulties tracking vehicles when they stop at ramp queues and when they significantly occlude each other. However, the system succeeds by detecting vehicles entering and exiting ramps and can record their motion statistics as they do so. Several experimental results from ramps in the Twin Cities are presented.

17. Document Analysis/Descriptors		18. Availability Statement
Ramp queues Blob tracking Vehicle tracking	Ramp monitoring Occlusion	No restrictions. Document available from: National Technical Information Services, Springfield, Virginia 22161

19. Security Class (this report)	20. Security Class (this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	23	

Sensor-Based Ramp Monitoring

Final Report

Eric Wahlstrom
Osama Masoud
Nikos Papanikolopoulos
Artificial Intelligence, Robotics, and Vision Laboratory
Department of Computer Science and Engineering
University of Minnesota

May 2003

Published by:
Minnesota Department of Transportation
Research Services Section
MS 330
395 John Ireland Boulevard
St. Paul, MN 55155

Executive Summary

This report covers the creation of a system for monitoring vehicles in highway on ramp queues. The initial phase of the project attempted to use a blob tracking algorithm to perform the ramp monitoring. The current system uses optical flow information to create virtual features based on trends in the optical flow. These features are clustered to form vehicle objects. These objects update themselves based on their statistics and those of other features in the image. The system has difficulties tracking vehicles when they stop at ramp queues and when they significantly occlude each other. However, the system succeeds by detecting vehicles entering and leaving ramps and can record their motion statistics as they do so. Several experimental results from ramps in the Twin Cities are presented.

Table of Contents

List of Figures

CHAPTER 1

INTRODUCTION

This report summarizes the work done so far to collect and process image sequences taken from freeway on-ramps. The ability to monitor the congestion and throughput of an on-ramp is an important component of traffic monitoring. Past research in the area of traffic flow modeling and control of freeway ramps has led to higher total freeway traffic volumes being served. Simulations and real traffic flow data are often used to select suitable on-ramp traffic metering strategies. We have used vision sensors for gathering data as a first step. This type of information can be distributed in real-time to customers via the internet or via variable message signs. Eventually, this sensory information can be integrated into appropriate on-ramp traffic metering strategies. Possible implementation of this system is expected to positively impact the traffic average speed and freeway traffic volume.

We have divided the task into two areas: optical flow grouping and vehicle tracking. Optical flow grouping contains the previously completed phases of this project including optical flow computation and virtual feature creation. The vehicle tracking phase contains efforts to group virtual features into recognizable vehicles using clustering techniques. This approach differs from the previous attempts in that there is no image segmentation phase to separate foreground from background. Instead, this technique relies on the perceived motion of the pixels in the image to provide candidates for vehicles.

CHAPTER 2

OPTICAL FLOW GROUPING

Cluttered environments (having many objects) make it difficult to isolate and track objects from visual data. Techniques that rely on solving the problem of separating foreground objects from the background find themselves facing a more difficult problem: separating foreground objects from one another. This is because in a cluttered environment, foreground objects are often occluded (partially or fully) by other foreground objects or by static background objects. In a dynamic environment, however, foreground objects rarely exhibit identical motion patterns. This is particularly true in a cluttered environment. For example, vehicles in congested traffic have more unique motion patterns than in free-flowing traffic. Motion cues are therefore a valuable resource when considering the problem of objects tracking.

Optical flow is one of the most popular representations of motion in an image sequence. Methods that compute optical flow aim to find the projection of the velocities of 3-D surface points on the image plane. Because of certain limitations, however, only the apparent motion in the form of velocities at pixel locations can be estimated. There has been a very large body of work in the literature concerning optical flow starting with Horn and Schunk [1]. The original hypothesis is that image intensity of the moving region remains more or less constant for at least a short period of time. This is known as the brightness consistency constraint and leads to the famous optical flow constraint equation:

$$\nabla I \cdot \mathbf{v} + I_t = 0,$$

where $\nabla I = (I_x, I_y)$ is the image gradient, I_t is the image temporal derivative, and $\mathbf{v}$ is the image velocity for some pixel (the optical flow). This equation alone is not sufficient to find both components of $\mathbf{v}$ (one equation and two variables) and additional constraints have to be used. The most widely used constraint is the local smoothness assumption [2]. This causes the computed optical flow to be erroneous around object boundaries and motion discontinuities in general.

To handle the issue of motion discontinuities, many researchers reformulated the problem as a layer model [3,4,5,6]. In this model, the image is assumed to be composed of patches whose motion can be parameterized. The problem is usually formulated as an optimization and results in a segmentation of motion between two frames. The main application here is motion-compensated image encoding.

Regardless of which method is used, when the temporal separation between the two frames is small, the measured flow will be noisy (see Figure 1). This is due to pixel discretization, noise in the image and camera jitter. When the temporal separation is large, the brightness consistency constraint is less likely to hold. In tracking, the goal is to compute object trajectories over a long period of time. This means one has to look beyond the solution to the two-frame problem of optical flow.

Figure 1. Optical flow computed from two frames in a traffic scene

We propose a method that will look beyond two frames. By somehow aggregating optical flow measurements over a long duration, one can look at trajectories and furthermore, associate a measure of reliability to them. An optical flow resulting from noise will not remain consistent through time and can be detected as such. Our basic idea is to create a virtual feature at every pixel having a high optical flow confidence. Then we can use a Kalman filter with a constant-velocity model where measurements are provided by the optical flow. This allows us to deal with noisy data as well as to associate an error covariance to the estimation. Finally, we group these features based on their motion and spatial characteristics taking into consideration their error covariances. We expect these groups will form a coherent moving object. Figure 2 shows some preliminary results depicting the trajectories of the virtual features.

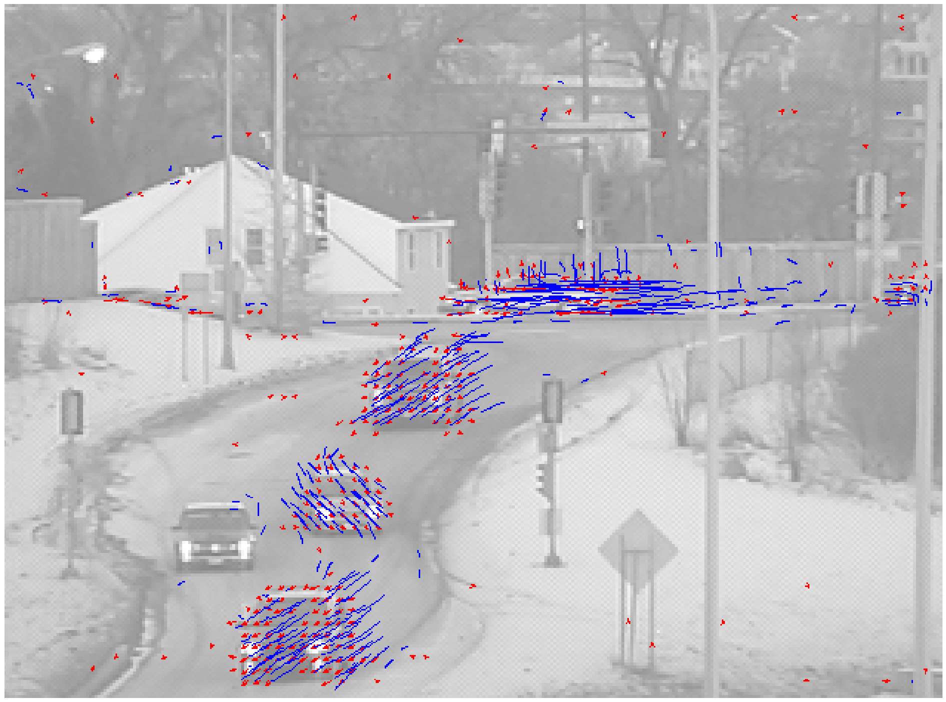

Figure 2. Virtual feature trajectories

CHAPTER 3

VEHICLE TRACKING

After features have been created and tracked, the system groups them into vehicle objects. This is done in three phases: clustering, vehicle creation, and vehicle splitting. The desired result is that each vehicle object represents a single unique vehicle in the image.

VEHICLE CREATION

First, the system tries to cluster the features with similar nearby features. The clustering algorithm used here matches a given feature against other features based on the magnitude and angle of the optical flow of the feature. Each feature is examined to determine if it has properties similar to its neighbors. If many features can be clustered together, a vehicle object is created to hold these features. The clustering process depends on four parameters. These are three tolerance settings for how different features may be in terms of speed, direction, and position and a numerical threshold for how many features must be clustered in order to create a vehicle object. The vehicle object itself is responsible for keeping a list of the features it holds and reporting statistics such as average speed in the vertical and horizontal directions and its centroid.

VEHICLE MAINTENANCE

Next, any features that exist in the image but are not members of a vehicle are compared to the current vehicles to determine if they belong. This allows vehicles that have already been created to continue to collect features, but will not allow vehicle objects to merge together. This is important because as the vehicle moves through the image, new features may be found that hold significant information about the vehicle. At the same time, features that already belong to another vehicle are excluded to allow for tracking when vehicle begin to occlude each other.

VEHICLE REPORTING

After a vehicle has been created, its statistics are reported to a file. The statistics tracked are current speed and position. In addition, the system logs entry and exit events to another file. These events correspond to vehicle creation and loss.

CHAPTER 4

RESULTS

The ramp monitoring software was tested using four separate video sequences, which contained footage of two ramps. The videos incorporated between 4 and 104 vehicles on the ramp. In one video, there is a queue in existence at the beginning that encompasses about one-third of the ramp.

In the first video, which was also used for development of the software, four vehicles enter the ramp. The software successfully detects each vehicle and tracks it. Figure 4.1 shows the software detecting the first vehicle. Tracking is lost on each vehicle before it exits the ramp. In two cases, this is because the optical flow features disappear as the vehicle slows for the queue. Figure 4.2 shows the software after it has lost tracking on the first vehicle. In the remaining two cases, the vehicle is lost due to occlusions by other vehicles in the cue.

In the second video, six vehicles enter the ramp. Four of the vehicles are tracked successfully until they reach the semaphore at the end of the ramp. Figure 4.3 shows the software detecting two vehicles. Due to its small size, one car is not detected as it travels the ramp. This is caused by the necessity for a minimum number of features detected to find a vehicle. The remaining vehicle is tracked until shortly before the semaphore when it stops behind another vehicle.

In the third video, two vehicles are in the ramp at the beginning and two vehicles enter later. The first two vehicles are not detected because they have already slowed for the semaphore and no significant features are detected on them. Figure 4.4 shows the starting positions of these missed vehicles in the ramp. The remaining two vehicles are detected and tracked successfully through the ramp. Figure 4.5 shows these vehicles being tracked.

On the final video, 104 vehicles enter the ramp. Of these vehicles, 20 enter the ramp already occluded by another vehicle to some degree. The software counted a total of 99 vehicles entering the ramp. Of these 99, four were caused by traffic flow on the adjoining highway. In three instances, large trucks were counted twice as they progressed through the ramp. This indicates that 92 of the 104 vehicles were correctly counted as they entered the ramp. This shows an 88% success rate for counting vehicles. The performance for tracking the vehicles as they moved through the ramp showed similar performance to that of the previous videos.

The results of these tests show that that ramp monitoring software can complete several of its objectives. The software finds and tracks vehicles as they enter the ramp. In doing this, it also finds the speed of each vehicle as it enters the ramp. The software reports the statistics of each vehicle that is present in each frame. It also logs events that occur, such as vehicles entering or leaving the ramp. With this data, the system can record usage statistics for ramps as well as record statistics of driving behavior.

Figure 4.1 Video 1: Finding a vehicle

Figure 3.2 Video 1: Lost first vehicle due to occlusion

Figure 4.3 Video 2: Finding vehicles

Figure 4.4 Video 3: Missing vehicles

Figure 4.5 Video 3: Finding vehicles

CHAPTER 5

CONCLUSION

The ramp monitoring application has been designed to process video in a unique environment. The traffic pattern that occurs in a ramp queue requires a system that can deal with many occlusions and stop-and-go traffic conditions. The initial phase of this project attempted to use a blob tracking algorithm to track vehicles in highway on ramps. While it met with some successes, it had difficulty because of the nature of traffic in ramps. That phase hypothesized that a model for processing video using optical flow characteristics could solve the problems encountered.

This phase has encompassed the creation of an algorithm for creating features based on the optical flow and the tracking of those features into vehicle objects. Several key successes were made. First, the model for tracking based on optical flow presents a useful source of information for image processing algorithms. Second, the system can track vehicles entering and leaving highway on ramps. However, due to the nature of the solution, the application does not track the vehicle when it stops in a ramp queue. The feature tracking algorithm requires high certainty in the optical flow in order to track features. This certainty appears to fail in situations where vehicles slow dramatically and occlude each other. Also, occlusions between vehicles cause the virtual features to "detach" themselves from the correct vehicle and follow some other vehicle. This occurrence then causes the vehicle statistics to be skewed and will ultimately result in the loss of tracking.

Given the issues that influenced this project, an implementation that incorporates multiple algorithms could be useful. The feature tracking algorithm alone does not seem to offer a complete solution to ramp queue monitoring. A combination of a motion tracking algorithm with an algorithm for remembering motionless features could offer hope for queue monitoring.

References

1. B.K.P. Horn and B.G. Schunk, “Determining optical flow,” Artificial Intelligence, 17:185-204, 1981.

2. B.D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” *in Proc. the 7th International Joint Conference on Artificial Intelligence*, pp. 674-679, Vancouver, 1981.

3. Y. Weiss, “Smoothness in layers: Motion segmentation using nonparametric mixture estimation,” *in Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 520-527, 1997.

4. J.Y.A. Wang and E.H. Adelson, “Representing moving images with layers,” *IEEE Trans. Image Processing.*, 3(5):625-638, 1994.

5. S. Hsu, P. P Anandan, and S. Peleg, “Accurate computation of optical flow by using layered motion representations,” *in Proc. ICPR*, pp. 743-746, Jerusalem, Israel, October 1994.

6. S.X. Ju, M.J. Black, and Y. Yacoob, “Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency,” *In Proc. IEEE Conf. CVPR*, pp. 307-314, San Francisco, CA, June 1996.