Proceedings of the 2007 IEEE
Intelligent Transportation Systems Conference
Seattle, WA, USA, Sept. 30 - Oct. 3, 2007

Multi-camera Based Traffic Flow Characterization & Classification

Ramsin Khoshabeh, Tarak Gandhi, Mohan M. Trivedi
University of California, San Diego
La Jolla, California, 92093-0434
{ramsin,tgandhi,mtrivedi}@ucsd.edu

Abstract— **We describe a system that employs the use of an omnidirectional camera in tandem with a pan-tilt-zoom (PTZ) camera in order to characterize traffic flows, analyze vehicles, and detect and capture anomalous behaviors. The framework is such that we can generate long-term statistics of traffic patterns while still monitoring unusual activity, even apart from the traffic zone. We use the PTZ in conjunction with the omni camera in order to perform classification analysis at coarse and fine levels. The omni performs the coarse classification and using information from that camera, the PTZ is able to perform refined classifications while actively moving throughout the scene.**

I. INTRODUCTION

Improving efficiency and safety of the road network is one of the main goals of intelligent transportation systems. Analysis of traffic parameters such as flow, speed, and density are important for transportation planning in order to enhance the efficiency of the roadways. On the other hand, analysis of behavior of individual vehicles has potential to enhance the safety by detecting accidents and unsafe situations in advance. Vehicle detection, tracking, and classification form the basis of such system for traffic analysis. In recent times, video-based analysis has gained popularity for these tasks. Considerable research has been performed and many good systems have been designed [1][2][3]. However, most current systems are based on a single camera which results in a tradeoff between the field-of-view and resolution. Pan-Tilt-Zoom (PTZ) cameras, which have been common in surveillance applications, can obtain high resolution videos of user-specified locations. However, they cannot obtain the full picture of the scene at the same time. Hence, a system combining a wide field-of-view (FOV) camera for complete scene coverage and a PTZ camera for selectively zooming into interesting parts of the scene forms an ideal combination.

In this paper, we propose a traffic flow analysis and classification system which keeps track of vehicles in the scene using a wide FOV camera. Using the PTZ camera, high resolution snapshots are captured at multiple locations on the vehicle track by repeatedly controlling the camera. Capturing images at multiple viewpoints has the potential to give information which can be useful for detailed classification, identification, and higher level analysis. In particular we use tracking information acquired from the omnidirectional camera in order to detect vehicles in the PTZ images. We then perform analysis of that detection region in order to improve the vehicle classification. Vehicle tracking in the wide FOV camera is also used to generate traffic statistics over a long period of time. Additionally, parking lot activity is analyzed with the same camera because of its large field of view. Section 2 gives the overview of the system and its major components. In section 3, each of the components is discussed in detail. Experimental results are in Section 4.

II. SYSTEM OVERVIEW

This paper describes a fully functional traffic monitoring system. Similar uni-camera systems have been discussed previously, such as in [4]. Yet our system is unique in the fact that we have two very different cameras working in unison to, not only continuously analyze traffic patterns and statistics, but to also actively detect events and perform refined classification.

Fig. 1. The system deployed. Top-left quadrant: A context-aware map actively showing the PTZ camera direction. Top-right quadrant: The PTZ image with buttons for camera control. Bottom half: Omni traffic analysis. Statistics are in the top-left corner and stored for further analysis. Detected objects are marked with green boxes.

III. SYSTEM ARCHITECTURE

A. Equipment

Our monitoring system uses a high-resolution (1600×1200) omnidirectional video sensor that has the capability of seeing objects in a 360 degree field-of-view. Because this omni camera is capable of capturing images over a very large field-of-view, we use it to monitor traffic statistics over a stretch of road that is greater than one-hundred meters in length. Furthermore, the camera has an excellent view of a large parking lot adjacent to the road, as can be seen in Fig. 1. In addition to analyzing road traffic, we are able to concurrently analyze activity in the parking lot. While a conventional rectilinear camera might be able to perform one of these tasks alone, it is nearly impossible to do both with a single camera.

The second camera is a PTZ camera mounted nearby the omni. It has high zoom capabilities that prove to be desirable for performing detailed analysis of objects captured with the omnidirectional camera.

Calibration: To calibrate the cameras, we use the fact that they are positioned in spatial proximity and relatively far from the scene. We find corresponding points for a number of PTZ positions in the omni image as shown in Fig. 2. Using these points we are able to compute a first-order functional relationship that approximately maps coordinates in the omnidirectional image to pan and tilt values in the PTZ camera without using camera geometry. For a fixed zoom value, the pan and tilt values (θ and ϕ) for the PTZ are given in terms of omni coordinates (x,y) as follows:

$$Pan: \theta = \alpha - \arctan\left(\frac{x - c_x}{y - c_y}\right) \quad (1)$$

$$Tilt: \phi = \beta - \gamma\sqrt{(x - c_x)^2 + (y - c_y)^2} \quad (2)$$

where c_x and c_y are the coordinates of the center of the omni image and α, β, and γ are experimentally evaluated.

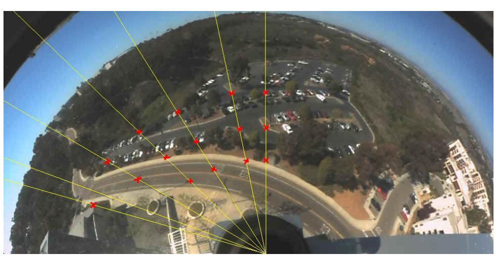

Fig. 2. Calibration correspondence points. Yellow rays represent changes in pan position and red points represent tilt positions for that pan value.

B. Video-based Processing

1) Segmentation & Tracking: Tracking is performed using established image segmentation techniques with dependable results. We first segment the image retrieved from the omni-camera by generating a mixture of Gaussians background model as discussed in [5], [6], and [7]. Once the image is segmented, blobs identified as foreground are tracked by applying a Kalman filter at each frame as described in [4].

2) Classification: Objects in the omni images are classified into one of five predefined groups 'person,' 'crowd,' 'car,' 'bus,' or 'no_label.' Because the relationship between object position and size is not uniform in the omni image, we plot track sizes as a function of image position (x-coordinates) and observe that the behavior is a nearly linear function. We perform a linear regression on several tracks to compute their functional estimate. There is a clear margin of separation between smaller objects (cars, vans, trucks, SUV's) and larger objects (buses and semis). We take the mean of all of the functional estimates of these tracks to be the class-separating line. Since pedestrians are much smaller than any vehicle, we compute the estimated boundary between cars and pedestrians as a line with a slope significantly smaller than the smallest car. Fig. 3 shows the plot of the various vehicle track sizes as a function of position. The class boundaries are shown in green. The overall classification is done with the following heuristic:

- Obtain the approximate area of the object, A, as the area of the bounding box of the tracked detection.
- If track length is less than N, then the class is 'no_label.'
- Else, for $T_1 < T_2$, classify the track as 'pedestrian' if A$< T_1$, 'car' if $T_1 \leq A < T_2$, and 'bus' if A$\geq T_2$, where T_1 and T_2 are boundary lines obtained in linear regression.
- Obtain the direction of travel of the track based upon the direction it has travelled in the furthest.
- If the label was 'car' or 'bus' and its direction is perpendicular to the road, then relabel it as 'crowd.'

Thus, we are able to find class separations between pedestrians, smaller objects (cars, vans, trucks, SUV's), larger objects (buses and semis), and crowds.

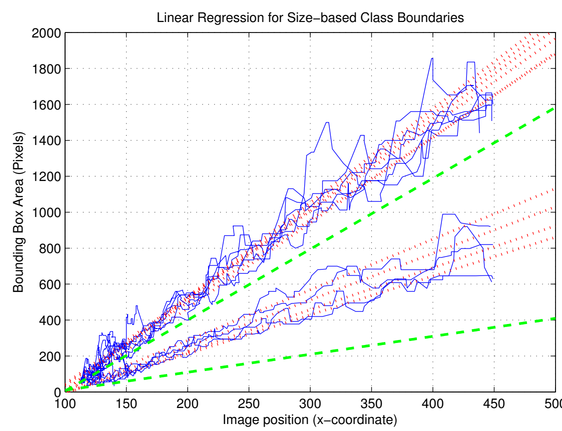

Fig. 3. Class boundaries for video based classification. The blue lines are the areas of the tracks as a function of position. Red lines represent their linear estimate. Green lines are the final class boundaries, T_1 and T_2.

3) Statistics Generation: Based on the tracking in the omnidirectional video, we collect information about individual detections as well as general traffic statistics. Due to the large field-of-view of the omni camera, vehicles can be tracked for a long stretch of road to acquire more accurate vehicle statistics. In each direction of travel, we compute vehicle counts and velocity estimates. We store this information along with track histories and sizes for further analysis.

4) Event Detection: The system has the capability of detecting many types of events. Possible events of interest can be to capture vehicles speeding, making U-turns, or other illegal maneuvers. For demonstration of system capabilities, we detect two kinds of events (in addition to the PTZ events we describe in the next section). The first event is to monitor a user-specified "virtual fence" region. Whenever an object enters this region an alarm is triggered to report the breach to whoever might be monitoring it. The second event detection is when a vehicle stops anywhere on the road for more than a few seconds. We consider this event a stalled car or possibly an accident and so again want to trigger an alarm.

5) Parking Lot Activity Analysis: To demonstrate the advantages of using an omni camera, we also analyze the parking lot area. We collect tracking data and use it to determine the lanes that are used by the vehicles. We also examine behaviors in the lot. Furthermore, we potentially could perform parking occupancy measures throughout the day to learn usage patterns. This could be useful in order to possibly alleviate parking congestion during peak times or for planning of future parking lots.

C. Event-based Active PTZ Control

Using the relative calibration of the omni and PTZ cameras, we are able to actively servo the PTZ camera to locations of interest as detected with the omni tracking. One interesting usage of event-based servo-ing is to capture the events we previously labelled as stalled cars or accidents with a finer degree of resolution. Our system does so by informing the PTZ of the occurrence of the critical event so we can capture and store the activity.

Additionally we define an event whenever a new westward-bound vehicle is detected and a second corresponding event when the vehicle is going out of the range of the omni camera. When either of these events occurs, the PTZ automatically servos and captures higher resolution images to refine the classification of the vehicle.

D. Camera Based Analysis

Due to the fact that the PTZ is capable of moving to a location and displaying scenes at a finer granularity than the omni, we use to it provide classification refinement on top of what the omni can provide.

Vehicles are captured at a lower resolution in the omni image. Therefore, it is difficult to identify features that distinguish vehicles. However, in the PTZ images, they are much more clearly evident. The following subsections describe the algorithms we use to classify vehicles into three categories: small (sedans), medium (trucks, vans, SUVs), and large (buses, semis). Further class separation is certainly possible if one identifies features that distinguish between, for example, trucks and vans.

1) Detection: To detect vehicles, we initially capture snapshots of the road when no cars are on it (this is identified from the tracking done in the omni). We use the median of these images as a model for our background scene. This is subtracted from the image being processed and pixel values greater than a threshold, T, are deemed foreground objects. Alternatively, we could take multiple snapshots over a window of time when capturing a vehicle and use those additional images as the background. We next perform morphology to smooth away noise and close blobs. Afterwards we select the largest blob as being the vehicle detection according to background subtraction.

However, segmentation alone does not rely on the additional knowledge the omni can provide, and is therefore more prone to error. To add semantic knowledge to the detection, we first make the assumption that, for any given perspective, the region in the omni camera is approximately planar. Since the PTZ is generally observing a small area and the cameras are high above the scene, it is a reasonable assumption. We therefore use the four-point algorithm as defined in [8] to compute a homography between the bounding box in the omni image and the one in the PTZ.

Homography without segmentation would not generate bounding boxes accurate enough to use on their own because the PTZ camera mechanics introduce a nonuniform time delay between the time the omni image is captured and when the PTZ actually moves and captures its image. However, by using homography with segmentation, we should get a final, reliable region of interest.

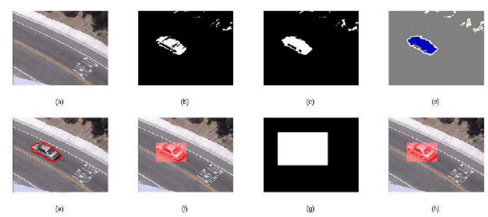

Fig. 4. PTZ Object Detection (a) Background (b) Foreground (c) Morphology (d) Blob (e) Blob on Original (f) Segmentation Box (g) Homography Box (h) Final Bounding Box

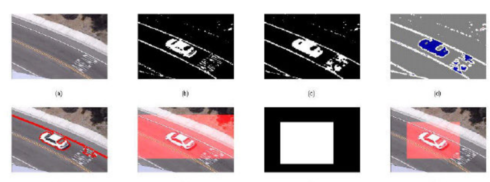

Fig. 5. Jitter provides poor results. Homography improves the detection.

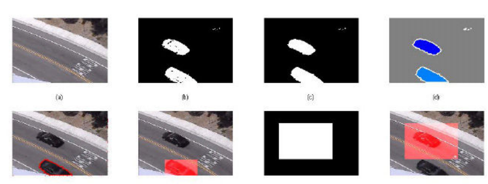

Fig. 6. Segmentation falsely identifies the vehicle closer to the camera as the detection. However, homography rectifies the false labelling again.

Fig. 4 shows the procedure graphically. In this case, the segmentation was sufficient for identifying the bounding box. Fig. 5 and Fig. 6 show two potential problems of using just segmentation. In Fig. 5, slight camera jitter introduces much more foreground than is expected. Therefore, the bounding box is inaccurately labelled. However, images (g) and (h) show the improved, albeit larger, bounding box by using the omni information. More interestingly, Fig. 6 depicts a scenario with two vehicles in the scene. With segmentation alone, we choose the incorrect vehicle as the vehicle being tracked. Yet the homography bounding box corrects the mistake and isolates the correct vehicle.

2) Feature Selection: Once we obtain the cropped image of the vehicle, we compute a feature vector in order to classify the image using Support Vector Machines (SVMs). Texture descriptors such as the Gradient Localization-Orientation Histogram (GLOH) (also known as Histogram of Oriented Gradient (HOG)) have been proposed in [9] to classify objects. This approach divides the image into rectangular cells and computes the histogram of the gradient orientations in each cell. These histograms are used as feature vectors for the SVM to distinguish between objects. The approach has also been extended to finding other objects. Koch and Malone [10] use the GLOH at multiple scales to distinguish between vehicles and other objects, such as animals and people, in thermal infrared images. The results from individual frames are fused over the entire vehicle track using the sequential probability ratio test.

The procedure for computing a GLOH is as follows:

- Compute the gradients in x and y dimensions.
- Subdivide the image into M×N discrete blocks.
- For each block element, quantize the gradient orientations into K bins. For each bin, increment the corresponding histogram bin.
- Collect the histograms into an M×N×K array and smooth spatial and orientation directions to avoid aliasing.
- Normalize the histogram array to unit vectors.
- Clip all values greater than c=0.2 to reduce the effect of large gradients due to spurious illumination and other changes. Renormalize.
- Stack the resulting array into a B=M×N×K dimensional feature vector.

3) Classification: After a feature vector is generated by GLOH for each image, it is passed on to the SVM algorithm for classification. SVM projects this data into a higher (potentially infinite with kernel functions) dimensional space where there exists a separating hyperplane between classes. Then the algorithm attempts to maximize the margin between the two classes by constructing two parallel hyperplanes on either side of the separating hyperplane. The hyperplane that has the largest margin is deemed the maximum-margin hyperplane and is a function of only those data points that lie on the margin (i.e., the support vectors). A guide to SVMs is included in the library package available from [11].

IV. EXPERIMENTAL RESULTS

A. Experimental Setup

To examine the functionality of the system, we ran a three-hour experiment from about 2:00 pm to 5:00 pm to gather data for the statistics experiments.

For SVM classification, we used 629 images of small, medium, and large vehicles as the training set. We then tested the classification using 164 separate test images.

Furthermore, we tested each of the functionalities we have thus far described.

B. Statistics

For traffic flow statistics, we generated vehicle counts of the eastbound and westbound lanes. Every minute the number of vehicles travelling in each direction was stored in a local database. Fig. 7 displays the flow patterns on the eastbound lane. We see that the traffic density remained fairly regular except when more vehicles were detected in the second half of the first hour. This experiment was performed on a Sunday, and so we see that weekday rush hour behavior is not observed in the evening as would be expected. Instead, there is actually more activity earlier in the afternoon.

In Fig. 8 we see that while traffic had a fairly regular flow pattern, more vehicles drove by in the first hour than in the next two hours. Again, the explanation could be the same as that of the eastbound statistics.

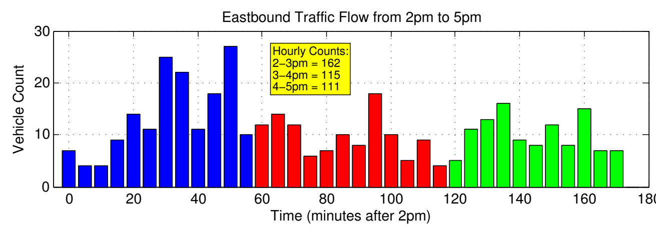

Fig. 7. Vehicle counts on the eastbound lane in 5-min intervals. Blue is the 1st hour, red the 2nd, and green the 3rd.

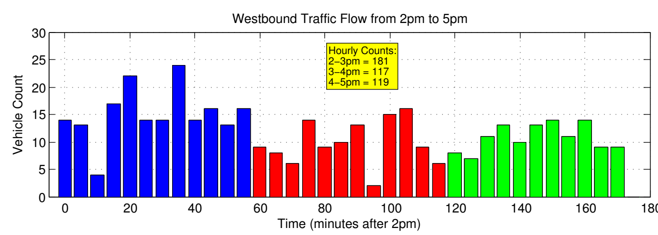

Fig. 8. Vehicle counts on the westbound lane in 5-min intervals.

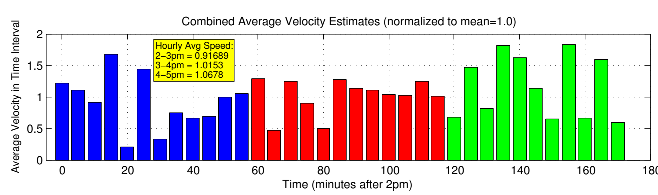

Fig. 9. Velocity statistics on both lanes in 5-min intervals. Velocities are normalized to the average velocity over the 3-hours.

We also maintain statistics on vehicle velocities. Fig. 9 shows the plots for average velocities over five-minute intervals. Interestingly, when we had the most vehicles, the average velocity was the lowest.

C. Parking Lot Activity

Fig. 10 illustrates the paths vehicles followed over a three-hour period in the parking lot adjacent to the road. From this we clearly see the lanes. We also notice the discontinuities where there are occlusions by the trees. These path statistics could be used to perform occlusion reasoning for tracking vehicles in the lot.

Also, in Fig. 11, we show an image of parking lot activity analysis for one of the three hours. At each node, we display the number of vehicles that were going further into the parking lot through that node as values in blue and vehicles exiting as values in red. We note that few vehicles drove all the way into the back of the lot. This is probably because this portion of the lot was already full at this time. The majority of the vehicles entered into the middle area and drove out (or possibly circled around) from the section closest to the camera.

Fig. 10. Vehicle paths over a 3-hour time period overplotted onto an image of the scene

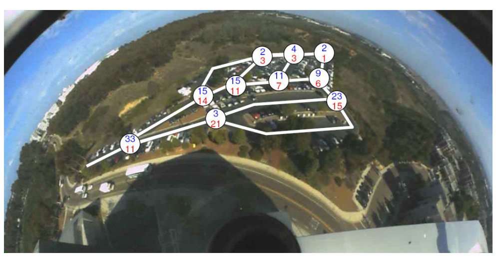

Fig. 11. Estimates of parking lot activity from 10-11am. Blue numbers indicate tracks that were going further into the parking lot through the node, while red numbers indicate tracks that were exiting through the node.

D. Event Detections

1) Stopped/Stalled Car: Fig. 12 depicts some thumbnails where a vehicle has been captured as being considered either "stopped" or "stalled." There are two snapshots for each track, one when it first enters the scene and one when it is leaving it. Thus, we show that we can in fact correctly locate and identify a vehicle in distress and respond accordingly.

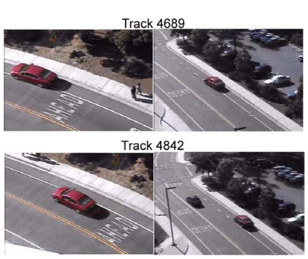

Fig. 12. Vehicle captured as stopped/stalled on the roadside

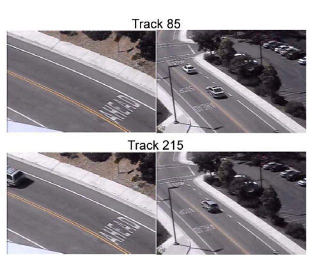

Fig. 13. Scenes where vehicle capture was missed

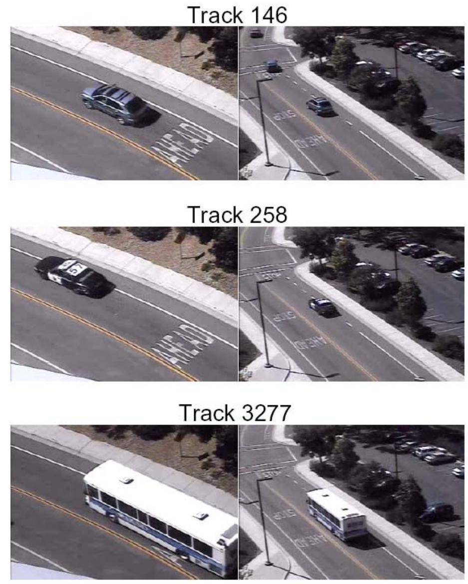

Fig. 14. Correctly captured vehicles

2) Active PTZ: In Fig. 14 we show a sequence of typical cars captured correctly using the system. In Fig. 13 we see that at times it will miss a car in one of its snapshot sequences. A possible explanation for this behavior is that if the camera was busy capturing the end of a nearby track, it will not have sufficient amount of time to move back in time to capture the current track. This can be seen in the first example (track 85) where we notice that in the snapshot on the right that there was a car directly in front of the currently tracked vehicle.

3) Size-Based Classification: Table I shows the results for the size-based classification with the omni camera. The "other" category refers to vehicles that were labelled as 'pedestrian', 'crowd,' or not given a label. Since this test set only contained vehicles, all three categories were combined to signify a mislabelled vehicle. With these three categories, the omni achieved a total classification rate of 80.49%.

TABLE I
CONFUSION MATRIX FOR HEURISTIC-BASED OMNI CLASSIFICATION

Classification Results						
		Predicted				
		Sm/Med	Large	Other	Totals	**Accuracy**
Actual	Sm/Med	124	0	31	155	80.00%
	Large	0	8	1	9	88.89%
	Other	0	0	0	0	—
	Totals	124	8	32	**164**	**80.49%**

E. PTZ Analysis

1) Classification Results: In table II the results for classification at one of the predefined snapshot locations are shown. The results are shown for bounding boxes derived from background subtraction and from the combined homography and segmentation approach. The test set contained 164 vehicles (97 small ones, 58 medium ones, and 9 large ones).

We note that even with the additional categorical breakdown between small and medium-sized vehicles, the PTZ performs better than the omni does when we use well-defined bounding boxes.

Since the comparison metric is unclear when we are comparing between two different types of classification, we repeated the classification method we used with the PTZ images on the omni images. Table III shows the confusion matrix for that method. In this test set, the PTZ classification performed 8.53% better than the omni. While this number is significant, we note that the results on the omni are somewhat skewed since this set contained more small vehicles than anything else. Clearly the omni has difficulty distinguishing trucks, vans, etc. from smaller vehicles like sedans. This is quantized in the second row of the confusion matrix where we see the poor results of classifying medium-sized vehicles. Most of the vehicles are classified as small vehicles as the totals show. So even though the test data favored the omni's poor classification, the PTZ still performed better. Additionally, were our cameras fully calibrated, the PTZ classification would perform even better with more accurate bounding boxes.

TABLE II

CONFUSION MATRICES FOR SVM-BASED CLASSIFICATION FROM PTZ

Using Segmentation						
		Predicted				
		Small	Medium	Large	Totals	**Accuracy**
Actual	Small	75	17	5	97	77.32%
	Medium	9	44	5	58	75.86%
	Large	0	0	9	9	100.00%
	Totals	84	61	19	**164**	**78.05%**
Segmentation with Homography						
		Predicted				
		Small	Medium	Large	Totals	**Accuracy**
Actual	Small	83	12	2	97	85.57%
	Medium	5	52	1	58	89.66%
	Large	0	0	9	9	100.00%
	Totals	88	64	12	**164**	**87.80%**

TABLE III

CONFUSION MATRIX FOR SVM-BASED CLASSIFICATION FROM OMNI

Classification Results						
		Predicted				
		Small	Medium	Large	Totals	**Accuracy**
Actual	Small	93	4	0	97	95.88%
	Medium	30	28	0	58	48.28%
	Large	0	0	9	9	100.00%
	Totals	123	32	9	**164**	**79.27%**

V. CONCLUDING REMARKS

We have shown that this system can reliably monitor traffic flows and respond to various event triggers. In addition we have demonstrated the strengths of an omni camera and the even greater synergistic strength of combining both omni and PTZ cameras. By taking advantage of the large viewing area of the omnidirectional camera we showed that traffic patterns could continuously be monitored while additional event detections took place. Similarly, by a simple calibration, the PTZ camera's higher resolution showed that we could capture a scene in greater detail for analysis refinements.

In the future, we plan on performing more detailed classification analysis on the PTZ images. For this we will require highly accurate camera calibration which will have to take into account the individual camera geometries and separations. We also hope to monitor parking lot activity much more closely to potentially learn patterns of behavior in that area. We will also gather statistics over much longer periods of time to test the system performance over those time periods. Additionally, rather than predefining set detection events, we would like to learn anomalous events based upon "normal" behavior learned from long-term track patterns.

VI. ACKNOWLEDGMENTS

We are grateful for the support of TSWG and the NSF-RESCUE Project for making this research possible. We are highly appreciative of the assistance received from all colleagues at the CVRR laboratory, particularly Mr. Erik Murphy-Chutorian for his assistance in the work of classification. The authors gratefully acknowledge the reviewers' comments.

REFERENCES

[1] Z. Zhu, G. Xu, B. Yang, D. Shi, and X. Lin, "VISATRAM: a real-time vision system for automatic traffic monitoring", *Image and Vision Computing*, 2000, pp. 781-794.

[2] S. Gupte, O. Masoud, R. Martin, and N. Papanikolopoulos, "Detection and Classification of Vehicles", *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, Mar 2002, pp. 37-47.

[3] P. Kumar, S. Ranganath, H. Weimin, and K. Sengupta, "Framework for Real-Time Behavior Interpretation From Traffic Video", *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 1, Mar 2005, pp. 43-53.

[4] B. Morris and M. Trivedi, "Robust Classification and Tracking of Vehicles in Traffic Video Streams", *in 9th International IEEE Conference on Intelligent Transportation Systems*, Sept 2006.

[5] C. Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking", *1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, vol. 2, pp. 2246.

[6] O. Javed, K. Shafique, and M. Shah, "A hierarchical approach to robust background subtraction using color and gradient information", *in Workshop on Motion and Video Computing*, pp. 22-27, Dec 2002.

[7] O. Achler and M.M. Trivedi, "Real-Time Traffic Flow Analysis using Omnidirectional Video Network and Flatplane Transformation", *in 5th International IEEE Conference on Intelligent Transportation Systems*, Sept 2002.

[8] M. Yi, S. Soatto, J. Kosecka, and S. Sastry, An Invitation to 3-D Vision. New York: Springer-Verlag, 2004, 139.

[9] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, p. 886-893, June 2005.

[10] M. Koch and K. Malone, "A Sequential Vehicle Classifier for Infrared Video using Multinomial Pattern Matching", *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop*, June 2006.

[11] C. Chang and C. Lin, "LIBSVM: a library for support vector machines", 2001, Software available at:
`http://www.csie.ntu.edu.tw/~cjlin/libsvm`