

Distributed Interactive Video Arrays for Event Based Analysis of Incidents

Mohan M. Trivedi, Andrea Prati, and Greg Kogut

Computer Vision and Robotics Research Laboratory
University of California at San Diego
La Jolla, CA 92093-0434

The Distributed Interactive Video Array (DIVA) system is developed to provide a large-scale, redundant cluster of video streams to observe a remote scene and to supply automatic *focus-of-attention* with *event-driven servoing* to capture desired events at appropriate resolutions and perspectives. Installing multiple sensors introduces several new research issues related to the system design, including handoff schemes for passing tracked objects between sensors and clusters, methods for determining the "best view" given the context of the traffic scene, and sensor fusion algorithms to best employ the strengths of a given sensor or sensor modality. This paper describes our research focused on the development of DIVA system for traffic and incident monitoring. The paper describes the overall architecture of the DIVA system. Algorithms for vehicle and platoon tracking using multiple cameras, and experimental results using novel distributed video networks deployed on the campus and the interstate I-5.

I. INTRODUCTION

There has been limited research in using multiple sensors and sensor modalities in traffic scenes to provide information not available from a single camera. One example, however, is the work presented in [1] that estimates both local and global traffic density from video data provided by Web traffic cameras in the Seattle area. Basically, most systems use single rectilinear CCD cameras, and use simple linear transforms to translate from image to world coordinates. While single sensor views are useful, dependence on a single view severely limits the quantity and quality of data available from the viewable environment, as already stated in Section I. Also, cars are tracked from a single, fixed perspective, while the best perspective with which to view the scene may change with time of day or traffic density. Current systems also use single, dedicated processors to analyze and record data, and do not provide the ability to distribute processing, select from an array of available sensors, or access real-time or archived data at multiple remote [2]. Past works in cross-camera correspondence can be divided into two categories: *geometry-based* and *recognition-based* [4]. In the first case, geometric features are transformed into the same spatial reference in order

to allow uniform matching. In this case, explicit camera calibration is required [5][6].

II. DIVA SYSTEM CAPABILITIES

The distributed interactive video array supports the following capabilities:

- a) *Distributed video networks*: to allow complete coverage the sensors must be placed in a wide area. The system has *televieing* capability, i.e. all the sources of information are available through a TCP/IP connection to the distributed computer(s).
- b) *Active camera systems*: exploitation of redundant sensing is mandatory. For this reason, this framework must have one, or more, central "monitors" able to select the camera with the best view of a given area in response to an event. Focus-of-attention in multiple camera systems is a relevant, and relatively new, research area.
- c) *Multiple object tracking and handoff*: to create a model of the environment and interact with it, the objects in the scene must be detected, segmented and tracked not only in each view but also among different views. This problem is usually referenced as the "camera handoff" problem or the "re-identification" problem.
- d) *3-D localization*: once that the object has been detected, tracked in different views and re-identified, the system should be able to assert *where it is* in the 3-D world coordinates. 3-D camera coordination in a multicamera system in an effective way is still a challenging research topic.
- e) *Multisensor integration*: how to exploit information from rectilinear CCD cameras, omnidirectional cameras and infrared cameras in an integrated and effective way is one of the key objectives of the system.

An example is shown in Figure. 1. Figure. 1(a) shows a possible setup. The omnidirectional camera is placed on the median, whereas the four rectilinear cameras are at the sides of the road. Let us assume that an incident occurs in the zone indicated as (1) in Figure. 1(a): while rectilinear cameras do not cover that area, the omnidirectional does, even if with a low-resolution image. Once the incident has been

detected, the omnidirectional camera commands the rectilinear camera to move towards the incident area and, perhaps, to zoom on it (Figure. 1(b)). The OD camera is the *primary view*, while the PTZ cameras are the *secondary views*.

In Figure 1(c) another example of multisensor coordination is reported. Referring to Figure. 1(a), an incident occurs in the area indicated with (2).

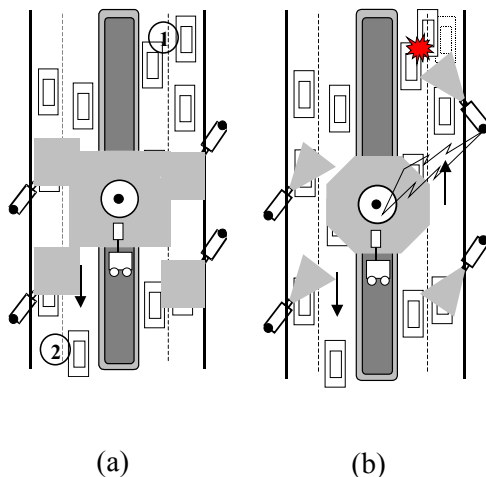


Figure 1. Example of multisensor coordination (a) reports a possible setup on a freeway. The omnidirectional camera is placed on the median, whereas the four rectilinear cameras are at the sides of the road. A robot equipped with an omnidirectional camera is stored in a box in the median. Let us assume that an incident occurs in the zone indicated as (1) in fig. (a): while rectilinear cameras do not cover that area, the omnidirectional does, even if with a low-resolution image. Once the incident has been detected, the omnidirectional camera commands the rectilinear camera to move towards the incident area and, perhaps, to zoom on it (b). The same situation arises in the area (2).

III. DIVA ARCHITECTURE

A. System overview

We envision a system that covers the highways and intersections with many sensor clusters that communicate with each other. Each cluster would include microphones, rectilinear and omni-view CCD cameras, infrared cameras and real-time range sensing cameras. As discussed in the previous sections, fusion of information from the sensors within each cluster and between different clusters would allow for monitoring of the traffic, recognition of individual behaviors and group behaviors, incident detection and intervention management. In addition to triggering appropriate responses, results from such analysis would be stored in a database. This would allow statistical analysis of past events and addition of standing queries for behaviors that were not defined at the time the system was designed.

Figure 2 (attached) shows the block diagram that illustrates the design of the system. It has five major

parts: *the DIVA sensor system, processing layers, system states, database and the interface.*

System states contain data produced by processing layers, such as segments, tracks, video frames, etc. Each layer takes input from sensors or from the system states (outputs from other layers). Each layer produces results, which are included in the system states.

The core of this architecture is the DIVA sensor system. This architecture is very convenient for dealing with multiple sensors. Some layers would operate on results that come from only one sensor (segmentation for example), while others would be responsible for integrating information from multiple sensors (3D tracking).

The primary-secondary (or “master-slave”) paradigm of DIVA system has been described above. Figure 2 reports a possible configuration in which all the omnidirectional cameras and one rectilinear PTZ camera are assumed primary views (note the letters on the lower right corner of the boxes). The data provided by these cameras is processed by the *event detection* module of the Central Monitor (CM). Figure 3 (attached) shows the details of the CM. The event detected is used as index to access to the *Event-Action Database (EAD)*. Three examples of event based servoing using the UCSD testbed are presented in Figure 4.

The system uses two camera clusters for these experiments. In part (a), the primary camera has detected a “stalled vehicle” event on the road. The secondary camera provides a close-up view of the passenger and the vehicle. In part (b) the primary camera has detected a vehicle in the emergency lane. The secondary camera provides the close-up of the vehicle license plate. Finally, in part (c), the primary camera has detected a stalled vehicle and the secondary camera provides a close-up of the traveler in need.







	Primary view	Secondary View
(a)		
	Event: Incident detected	Action: zoom to the area of the incident
(b)		
	Event: Car stopped in a reserved area	Action: zoom to the license plate number
(c)		
	Event: Flat tire detected	Action: zoom to the tire.

Figure 4: Three examples of Event-based servoing: a) stalled vehicle; b) view from emergency response; c) traveler in-need

Both individual and group behaviors lend themselves to statistical analysis and classification techniques, due to the inherent unpredictability of driver behavior and the error associated with vision data. Techniques such as HMM-based classification, Bayesian inference, and statistical clustering have worked well with other computer vision applications and will likely make good tools with which to analyze traffic scenes. Statistical classifiers such as HMMs or Bayes nets can be trained to recognize specific behaviors, such as lane changing, or deviations from a one of several "standard" behaviors. Clustering techniques could be used to analyze large amounts of data to determine use patterns of a section of freeway, and to recognize possible inefficiencies in traffic flow. This database associates the corresponding action to the event and sent it to the *action decision maker*, which has the function to interpret the action and to redirect it either to the *focus-of-attention* module or to the *driving directions* module. The former commands to the

secondary PTZ cameras to act in reaction of the event, the last sends via wireless network to the robots the information necessary to drive to the location computed by the action decision maker. The interface allows the users to add, modify and remove tuple in the

IV. EXPERIMENTAL TEST BED AND RESULTS

The CVRR (Computer Vision and Robotics Research) Lab at UCSD has constructed its own test beds on campus as well as on Interstate 5, with the goal of providing high quality real-time video to the CVRR lab, as well as to the Internet (Figure 5, attached). This data has proved instrumental in providing the large quantities of traffic data from the camera sites necessary in the development and test of the algorithm described in this paper. This test bed is currently operational, and consists of four PTZ cameras, one static ODVS, one infrared camera and one mobile ODVS camera. These sensors are hooked up to a dedicated gigabit Ethernet network, which provides up to 16 full-rate, full-resolution video streams to the CVRR lab. This dedicated network is also connected to the Internet, allowing for public use of the traffic data and possibility of use the PTZ commands of the cameras.

The modular design of this architecture allows for different algorithms for the same task to be tested without any difficulty in a plug-and-play manner. To systematically evaluate the goodness of our distributed architecture, we compare different methods of shadow detection and of multiple camera tracking.

The detailed comparison and evaluation of moving shadow detection algorithms has been reported in [7].

For multiple camera tracking, we implement two novel approaches. The first approach is reported in 0 and is based on graph matching. A model of the color of each detected vehicle is calculated. The system employs a color matching system that is a partial implementation of the Auto Color Matching System [3], in which the differences between illumination at cameras sites and between cameras are compensated. The mean and variance values of the R, G and B channels are used as feature model. This is used as signature to identify the object. A simple vehicle-tracking scheme identifies identical vehicles from the same camera site (single camera tracking) by using this color model and the blob centroids from the segmentation module, to help solve the data association problem. Then, platoons of vehicles are detected. A platoon is a vehicle, or group of vehicles, traveling in close proximity 0. Vehicles that are entirely within a pre-defined region of the road scene are detected as platoon. Matching identical vehicles

in different camera sites can be a challenging problem, since visual information can drastically change between two views. In particular, in a freeway environment the difference between the aspect of the objects in the upstream view and in the downstream view is relevant, both in shape and in color. For this reason, this method uses a symbolic representation of the information. Taking the perspective distortion into account, the composition and relative distances inside a platoon is the same on the two views. Indeed, a labeled, undirected graph is created from this data.

This matching system was tested with samples from data taken from two sites. The first data set, offering “easy” data, is from images taken from the UCSD test bed described above where platoons move slowly. The second data set consists of samples from a 20-minute segment of video taken with two freeway overpasses, located approximately 150m apart with non-overlapping views.

The test bed data provides “easy” scenario in a highly controlled one-lane environment, avoiding or minimizing many common problems in vehicle tracking, such as vehicle changing lanes, vehicle occlusions (minimized by the high perspective view of the traffic), and high-speed vehicles passing one another. The freeway data is, on the other hand, extremely challenging. The freeway traffic exhibits high speed, traffic density and, in our case, an off-ramp immediately after the second overpass. This tends to destabilize platoon behavior, as individual vehicles maneuver to position themselves in the right lane to take the off-ramp. Also, the perspectives at the two camera sites are significantly different, compared to the test-bed data.

A ground truth was acquired was acquired by manually identifying matching platoons in the two camera views in both data sets. This ground truth was used to calculate the *matching accuracy* as the percentage of true positive matches on the total of samples. Results for the two data sets are reported in Table I (attached)

Unlike the first method, the second multiple camera tracking method explored assumes uncalibrated, overlapped cameras. This is more properly a camera handoff method. The system requires the manual (or semi-automatic) drawing of the field of view (FOV) overlap between the two (or more) cameras. The algorithm performs the following steps:

Step 1: Find moving objects using background subtraction with the segmentation process above described

Step 2: Correlate objects with previous frames’ objects using Fieguth color calculation 0 and proximity to previous position.

Step 3: Check if any objects exist in the FOV area for first camera. If they do, look for matching objects in the FOV area for second camera. Matching is based on Fieguth color calculation and relative area proximity.

Step 4: If matching objects are found mark both with the same ID number, choosing the ID number of the object that has existed for a longer duration. This should assign the ID associated with the object in the originating camera to the object that has just appeared in the other camera.

Step 5: If a match is found in the area of overlap, mark it as such so that further attempts at matching this object will not be made.

Step 6: If an object leaves the area of overlap and has been matched, reset the matched flag so it can be matched again if it re-enters the area of overlap.

Step 7: Perform a background image update and repeat the process.

Even though the test bed data set is easier than the freeway environment, results (reported in Table II) are promising. The low performance of data set 2 and 3 are due to the white large shuttle buses in the scene: the auto iris of the cameras adjusts to a smaller aperture, making the rest of the image appear darker. Since the segmentation is based on background subtraction, this sudden variation causes many problems at the segmentation level.

V. CONCLUDING REMARKS

The main goal of the overall research is the realization of a powerful and integrated traffic-incident detection, monitoring and recovery system based on distributed active multicamera video-based architecture. Installing multiple sensors introduces several new issues into the system design, including handoff schemes for passing tracked objects between sensors and clusters, methods for determining the “best view” given the context of the traffic scene, and sensor fusion algorithms to best employ the strengths of a given sensor or sensor modality. The limitation of the field of view of a single camera system or of a non-active multicamera system is overcome with an active system with event-driven servoing based on an event-action paradigm. The flexibility is assured by the event-action database (EAD) and its interface that allows for dynamic modification of the event-action tuples.

VI. ACKNOWLEDGEMENTS

Our research is supported in part by the California Digital Media Innovation Program (DiMi) in partnership with the California Department of Transportation (Caltrans). We also wish to thank our colleagues from the lab who are also involved in related research activities.

VII. BIBLIOGRAPHY

- [1] S. Santini, "Very low rate video processing", in *Proceedings of SPIE*, Vol. 4311, *Internet Imaging II*, Jan. 2001.
- [2] S. Bhonsle, M. Trivedi, A. Gupta, "Database-Centered Architecture for Traffic Incident Detection, Management, and Analysis," *IEEE Conference on Intelligent Transportation Systems*, Dearborn, Michigan, October 2000.
- [3] N. Zeng, and J.D. Crisman, "Vehicle matching using color", in *Proceedings of IEEE Intl Conference on Intelligent Transportation Systems (ITSC)*, 1997, pp. 206-211
- [4] T-H. Chang, S. Gong and E-J. Ong., "Tracking multiple people under occlusion using multiple cameras", in *Proceedings of British Machine Vision Conference (BMVC)*, vol. 2, pp. 566-575, 2000.
- [5] Q. Cai and J.K. Aggarwal, "Tracking Human Motion in Structured Environments Using a Distributed-Camera System", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 12, Nov. 1999, pp. 1241-1247
- [6] K. Sato, T. Maeda, H. Kato and S. Inokuchi, "CAD-based object tracking with Distributed Monocular Camera for Security Monitoring", in *Proceedings of 2nd CAD-based Vision Workshop*, pp. 291-297, 1994
- [7] A. Prati, I. Mikic, R. Cucchiara, and M. M. Trivedi, "Analysis and Detection of Shadows in Video Streams: A Comparative Evaluation," *IEEE CVPR Workshop on Empirical Evaluation Methods in Computer Vision*, Kauai, December 2001.
- [8] <http://cvrr.ucsd.edu:88/aton/testbed/>
- [9] M.M. Trivedi, I. Mikic and G. Kogut, "Distributed video networks for incident detection and management", in *Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC)*, 2000, pp. 155-160
- [10] G. Kogut and M.M. Trivedi, "Maintaining the identity of multiple vehicles as they travel through a video network", *Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC)*, 2001.
- [11] P. Fieguth, and D. Terzopoulos, "Color-based tracking of heads and other mobile objects at video frame rates", in *Proceedings of the IEEE Intl Conference on Computer Vision and Pattern Recognition (CVPR)*, 1997, pp. 21-27

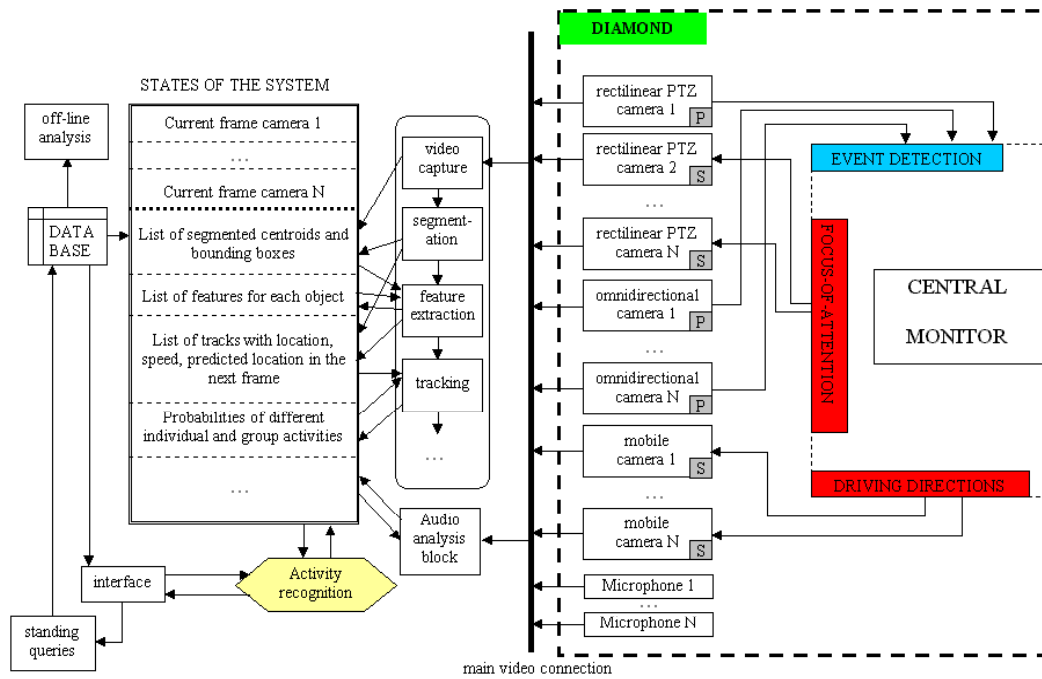


Figure 2. Sensory cluster block diagram. The five components included in the ATON architecture are: the DIVA sensor system, processing layers, system states, database and the interface.

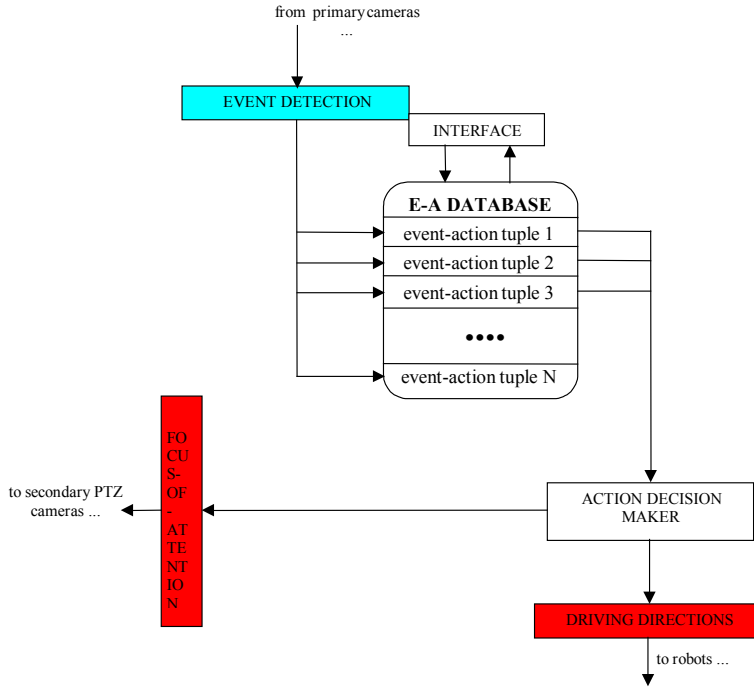


Figure 3. The Central Monitor of the DIVA system. The data provided by the primary cameras is processed by the *event detection* module. The event detected is used as index to access to the *Event-Action Database (EAD)*. This database associates the corresponding action to the event and sent it to the *action decision maker*, which has the function to interpret the action and to redirect it either to the *focus-of-attention* module or to the *driving directions* module. The former commands to the secondary PTZ cameras to act in reaction of the event, the last sends via wireless network to the robots the information necessary to drive to the location computed by the action decision maker.

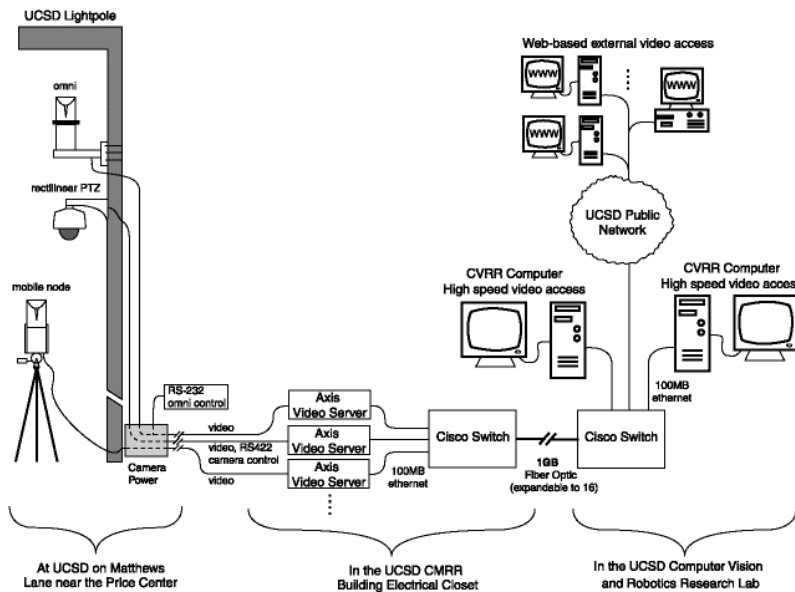


Figure 5. The architecture of the test bed constructed in the UCSD campus.

TABLE I. TRACKING ACCURACY FOR THE PLATOON MULTIPLE CAMERA TRACKING ALGORITHM.

Data Set	Nr. samples	Mean platoon size	# true positive matches	Match Accuracy %
Test bed	31	2.2	27	87%
I-5	22	3.5	10	45%
Totals	53	2.6	37	65%

TABLE II. TRACKING ACCURACY FOR THE CAMERA HANDOFF ALGORITHM.

Data Set	Nr. Samples	# true positive matches	Match Accuracy %
Test bed 1	4	3	75%
Test bed 2	18	10	56%
Test bed 3	9	4	44%
Test bed 4	12	10	83%
Test bed 5	32	25	78%
Totals	75	52	69%