Maintaining the Identity of Multiple Vehicles as They Travel Through a Video Network

Gregory T. Kogut, *Member, IEEE*, and Mohan M. Trivedi, *Member, IEEE*

Abstract-- **Large vision infrastructures have recently been installed, or are under construction in many major metropolitan areas[1]. These networks of cameras offer enormous volumes of data from large areas. However, most work in ITS computer vision is constricted to the use of either single cameras, or small numbers of cameras[2,3]. There is an enormous amount of information available from the fusion of data from multiple cameras that is not available from any single sensor. This paper presents an algorithm that provides an example of such data fusion: maintaining the identity of a vehicle, or groups of vehicles as they pass through multiple cameras sites. The algorithm uses a combination of color features and the spatial organization of vehicles within platoons to minimize false positives. The algorithm could be used in an application to track individual cars as they travel long distances through a traffic system, allowing the measurement of point-to-point traffic parameters, such as travel time.**

Index terms-- vehicle, matching, video, network, identity

I. Introduction

Large-scale computer vision infrastructures are becoming increasingly common with the increasing availability of cheap digital cameras and ubiquitous networking. There are large-scale networks of cameras currently installed on major freeway systems. There are three types of coverage possible:

1. Ubiquitous, complete coverage (Type I): all, or nearly all the environment is within the field of view of a stationary camera. This type of coverage is rare, due to expense and lack of need to cover an entire environment at once.
2. Qualified, complete coverage (Type II): Only a fraction of an environment is covered at any one time, but the sensors may be reconfigured on demand to cover any given area.
3. Incomplete coverage (Type III): only strategic, predetermined positions within the environment are covered. This is common in ITS vision systems, where only a small area of major freeway systems is covered. The locations must be strategically chosen to satisfy the demands of a given application.

So far, the most ITS vision infrastructures only offer viewing of raw data from individual sensors, usually to visually check the flow of traffic along a given route[1]. Bandwidth limitations of the Internet usually restrict these applications to low quality, low frame-rate video. Some systems offer tracking and traffic density, but these are generally not currently available to the public, and are usually limited to either single, or a few, cameras[2]. There is very little work on systems that fuse visual data from multiple cameras to provide information not available from single cameras. Once example, however, is Simone Santini's "Very Low Rate Video Processing" work that provides estimate of both local and global traffic density from video data provided by Web traffic cameras in the Seattle area[4]. There are significant qualitative differences among the three types of sensor coverage. While single sensor views are useful, and the most common current use of large networks of cameras, the restriction to a single view severely limits the quantity and quality of data available from the viewable environment. Current research in sensor networks is concentrating on data fusion techniques that allow data from multiple cameras to be combined. There are many advantages to data fusion, as outlined by UW's ITS Research Program[5]:

- Increased confidence: more than one sensor can confirm the same target

---

Gregory T. Kogut and Mohan M. Trivedi are with the Electrical and Computer Engineering Department, University of California-San Diego, 92093-0407, USA.

[1] http://www.wsdot.wa.gov/PugetSoundTraffic/cameras/.

[2] D. Koller, K. Daniildis, and H. Nagel, "Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes," *Int. Jour. Of Computer Vision*, vol. 3, 257-281, 1993.

[3] D. Beymer, P. McLauchlan, B. Coifman, J. Malik, "A Real-time Computer Vision System for Measuring Traffic Parameters," *Proc. Computer Vision and Pattern Recognition*, 1997, pages 495-501.

[4] S. Santini, "Very Low Rate Video Processing," *Proceedings of SPIE Vol. 4311, Internet Imaging II,* San Jose, Jan. 2001.

[5] D. Dailey, P. Harn, and P. Lin, "ITS Data Fusion," Final Research Report, Research Project T9903, ITS Research Program.

- Reduced ambiguity: joint information from multiple sensors reduces the set of hypotheses about the target
- Improved Detection: integration of multiple measurements of the same target improves signal-to-noise ratio, which increases the assurance of detection
- Increased robustness: one sensor can contribute information where others are unavailable, inoperative, or ineffective
- Enhanced spatial and temporal coverage: one sensor can work then or where another sensor cannot
- Decreased costs: a suite of "average" sensors can achieve the same level of performance as a single, highly-reliable sensor, and at a significantly lower cost

The vehicle-matching problem is the task maintaining the identity of a vehicle that travels through multiple, non-contiguous, sensor sites. This problem has been studied extensively using inductive-loop sensors that are widely deployed in many road systems[6],[7]. Most of these systems are based on matching calculated vehicle signatures.

While most research has been done using loop-based sensors, the use of vision sensors has also been explored. Most vision-based vehicle matching system use individual car features, such as color and shape, as matching criteria[8,9]. These methods have proved to be somewhat limited in the vehicle-matching problem, with the highest reported true positive rate of 16.42% by the AutoColor Matching System of Zeng and Crisman for "real" environments with wide-angle views of the roadway, and changing lighting conditions. Some of the problems with using solely color for vehicle matching include such things as the low of variability in car colors (i.e., the large number of white cars), and non-uniform variability in lighting conditions between camera sites by such things as specular reflection and widely varying visible surface areas of cars. Another video-based vehicle signature analysis system achieves impressive results in vehicle identification, however the system uses highly specialized hardware, and top-down camera views which preclude the use of the sensors for other tasks[10]. The system proposed in this paper uses camera hardware and views typical of general-purpose traffic cameras.

Some research involves using more than just individual vehicle signatures[11]. These systems exploit a characteristic of much vehicle traffic: vehicles tend to form platoons. There is information contained in the vehicle ordering within a platoon as long as this ordering is maintained reasonably well between sensor sites. This information can be employed to greatly reduce the ambiguity in matching cars that travel between two sensor sites that are relatively close. This paper uses platoons similarly, only relaxing the need for strict ordering of cars, and taking into account side-to-side positioning as well as front-to-back ordering.

This paper proposes a hybrid method, using color and spatial information to construct labeled graphs of small groups, or platoons, of cars, as they travel through a camera site. Then, the task of matching the cars at following cameras sites can be cast as a problem of probabilistic graph matching. In this implementation, a probabilistic hill-climbing algorithm is used to perform the graph matching. Once matching has been preformed, vehicles and platoons of vehicles may be matched between camera sites.

II. Algorithm Description

The vehicle identification system employs a sequential, modular architecture typical of many computer vision applications. The algorithm consists of two primary modules, each containing several sub-modules. The first module performs low-level computer vision consisting of video capture, segmentation, feature calculation and tracking, and platoon detection. The second, higher order, module consists of a graph creation algorithm, and the matching algorithm.

Figure 4, at the end of paper, shows the data flow and overall processing infrastructure.

Segmentation. Moving vehicles are segmented from half resolution (320x240) full-rate (30fps) RGB video input streams. Two segmentation algorithms are performed independently, and the results are then fused together to obtain a more accurate segmentation.

An adaptive region-based segmentation algorithm, capable of detecting shadow, is used [12]. This allows for accurate segmentation under a wide range of lighting conditions, and uses the less error-sensitive region information.

[6] Kühne, R. Immes, S., Freeway Control Systems for Using Section-Related Traffic Variable Detection, *Pacific Rim TransTech Conference Proc.*, Vol 1 ASCE 1993, pp 56-62.

[7] Levine, S., McCasland W., Monitoring Freeway Traffic Conditions with Automatic Vehicle Identification Systems, *ITE Journal*, Vol.64, No 3, pp 23-28.

[8] Zeng, N.; Crisman, J.D., "Vehicle matching using color," Proc. ITSC '97, p. 206-11.

[9] Rojas, J.C.; Crisman, J.D., "Vehicle detection in color images," ITSC '97 Proceedings, p. 403-9.

[10] MacCarley, C.A., Videobased Vehicle Signature Analysis and Tracking Phase 1: Verification of Concept and Preliminary Testing. UCB-ITS-PWP-98-10, PATH, University of California, Berkeley, CA.

[11] Coifman, B., Cassidy, M., Vehicle Reidentification and Travel Time Measurement on Congested Freeways, Transportation Research-B (submitted)

[12] I. Mikic, P. Cosman, G. Kogut, M. Trivedi, "Moving shadow and object detection in traffic scenes," *International Conference on Pattern Recognition*, Barcelona, Spain, September 2000.

An edge-based segmentation algorithm is used to help resolve errors in the region-based segmentation. For example, sometimes region-based segmentation will merge falsely merge two vehicles into one "blob." An edge segment can be used to re-split the segment into two individual vehicles.

Feature Calculation. 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 AutoColor Matching System. The AutoColor Matching System compensates for differences between illumination at cameras sites and between cameras. The system uses mean and variance values for the R, G, and B channels of a vehicle's color as the feature model. The details of the AutoColor Matching System are provided in [8]. The matching and scoring modules of the AutoColor system are not used in this algorithm.

Tracking. A simple vehicle-tracking scheme identifies identical vehicles in successive video frames from the same camera site. The tracking algorithm uses the color models described above and the blob centroids from the segmentation module, to help solve the data association problem. Full rate video, and predictable target motion in road scenes makes the tracking module very accurate, however, problems such as vehicle occlusion introduce some error within this module. A more robust tracking algorithm should be used in future implementations, such as that shown in [3]. Vehicles are assigned vehicle IDs (valid only within one camera site, at this stage), and their locations and velocities are recorded. A simple camera calibration provides a world-coordinate system, and a framework reducing the perspective effect in calculating vehicle size[13].

Platoon Detection. Platoons of vehicles are detected. A platoon is a vehicle, or group of vehicles, traveling in close proximity. While a platoon in most of the ITS literature often refers to groups of cars traveling in the same lane, while in this system the vehicles may be traveling in any lane, and need maintain strict ordering or relative positioning between camera sites.

The platoon detection algorithm recognizes groups of cars that are entirely within a pre-defined region of the road scene. The pre-defined area serves to avoid including vehicles that are near the horizon, or that are only partially within the image plane. This avoids including vehicles that tend to have less reliable feature data, such as erroneous size estimates, or missing occluded pixels. New platoons are created only when one or more vehicles either leave or enter the defined region. This avoids the creation of redundant platoons. However, a vehicle may be assigned to multiple different platoons during its trip through the defined region, and every vehicle is assigned to at least one platoon.

[13] Bas, E.K.; Crisman, J.D., "An easy to install camera calibration for traffic monitoring," *ITSC '97,* p.362-6

The high-level vision module consists of the algorithms that construct graphs based on the input from the low-level vision module, and the algorithm which matches graph between the two cameras sites.

Graph Creation. The graph creation module models groups, or platoons, of cars in a road scene as a labeled, undirected graph. Labeled graphs are graphs in which each node and edge has a label. In this paper, nodes vehicle models, and edges encode information about the spatial arrangement of vehicles in the platoon. In the current implementation, the nodes contain the color information calculated in the feature calculation model, and the edges are the Euclidean distance between vehicles.

The module constructs a connected graph from the platoon and spatial information, so that edges exist between a vehicle, and all other vehicles in the platoon. While this increases the computational load on the matching module, described, below, the total complexity is relatively low if platoon sizes are restricted to 1-6 vehicles.

Platoon Matching. This module matches graphs between the two camera sites. A close match between two graphs suggests a strong likelihood that the graphs correspond to the same, or roughly the same, platoon of cars. A probability threshold is used to determine the matches.

The first step in the matching algorithm is to create sets of data graphs from Site 2 for each pattern graph from Site 1. Both the distance between the sites, and the platoon velocities are known, so a predicted travel time may be calculated for platoons traveling from Site 1 to Site 2. All platoons at Site 2 that fall within a reasonable time of their predicted arrival time are used as data graphs to match with the pattern graph.

Once a set of data graphs has been assembled, they are matched individually with the pattern graph from Site 1. Because of the noisy nature of computer vision data, and the unpredictable nature of freeway traffic an approximate graph matching algorithm is used, which allows for disparities between the graphs, such as missing or extra nodes. These might result from such common vehicle tracking problems such as occlusion.

The graph-matching module is based on the approximate graph matching algorithms presented in [14]. The algorithm constructs a mapping between the pattern graph and each data graph by finding a sequence of edit operations that would convert the data graph to the pattern graph. The edit operations used are: *node relabeling, node deletion, node insertion, edge relabeling, edge deletion, and edge insertio*n. A cost function, λ, is associated with each edit operation. The cost functions used in this implementation

[14] Wang, J.T.L.; Kaihong, Z; Gung-Wei, Ch, "Approximate graph matching using probabilistic hill climbing algorithms," *Proc. Sixth International Conference on Tools with Artificial Intelligenc*e, 1994. p.390-6.

are shown below. a and b are either edges or nodes in the following notation.

- Node relabeling: $\lambda(a \to b) = (x_a - x_b)' C^{-1} (x_a - x_b)$, the Mahalanobis distance between the color features of the two nodes.
- Node deletion: $\lambda(a \to \Lambda) = |x_a|$, the magnitude of the vector of the node being deleted.

- Node insertion is the same as the above, substituting the inserted node for the deleted node.

Edge cost functions are analogous to node cost functions, with the arithmetic differences and magnitudes substituted for the vector functions.

The total cost function for a graph match is there the sum of all cost functions for a mapping between the data and pattern graphs. It can be shown that the minimum-cost mapping corresponds to the least cost sequence of edit operations[14]. Calculating the minimum-cost mapping can be cast as a state-space search problem, the details of which are shown in [14].
Minimum-cost mappings are calculated for all graph-matching combinations. The double threshold evaluation method of was used to determine true matches. In this method, the two least-cost scores are used[8]. The best score must pass an absolute cost threshold, and also the difference between the two scores must pass another threshold. This method serves to reduce the number of false positive matches.

If the mapping with the lowest cost passes a pre-defined matching threshold value, the data graph associated with the pattern graph is assumed to be a true match to the pattern graph. However, the graphs are not necessarily identical, due to the node and edge operations allowed in the matching algorithm.

Once the best-fitting graph for the pattern graph has been calculated, and has passed the threshold, then the site-to-site tracking data can be calculated for both the platoon, and for individual vehicles within a platoon. Such data include point-to-point travel times, changes in velocity, or lane-changing behaviors for the region between the two camera sites.

III. Test Bed Infrastructure

Vehicle matching algorithms depend highly on the quality of calculated vehicle features, such as color, shape, and location. The quality of calculated features, in turn, depends highly on the quality of the raw video stream used as an input. Most publicly available live traffic video lacks the dynamic range, resolution, and frame rate necessary for performing vehicle matching using current algorithms. The CVRR Lab at UCSD has constructed its own on-campus ITS test-bed, with the goal of providing high quality real-time video to the CVRR lab, as well as to the Internet[15,16]. This data has proved instrumental in providing the large quantities of traffic data between from two camera sites necessary in the development of the algorithm described by this paper.

This test-bed is currently operational, and consists of two pan-tilt-zoom cameras, and one ODVS (Omni Directional Vision Sensor) overlooking a traffic scene. The pan-tilt zoom cameras are Pelco Spectra II surveillance cameras. The test bed provides Type I coverage, as defined above, with the entire scene being entirely covered at all times by a combination of the three sensors. 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. However, the quality of the Internet data is subject to degradation due to the nature of Internet traffic.

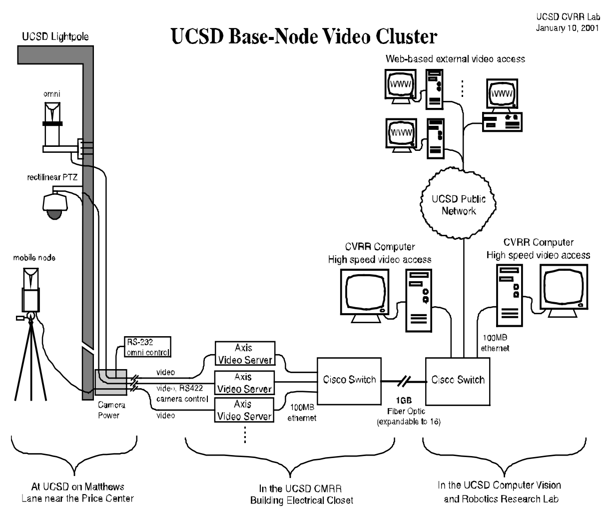

Figure 1: ATON ITS Video Network

IV. Data Collection And Experiment Design

The 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 ITS video network test bed, described above. This data consists of images from two cameras with similar, overlapping views of a two-lane road. The data from this site consisted of two to three vehicle platoons moving slowly. The second data set consists of samples from a 20-minute segment of video

---

15 M. M. Trivedi, I. Mikic, and G. Kogut, "Distributed video networks for incident detection and
management", *Proceedings of the IEEE Conference on Intelligent Transportation Systems,* Dearborn,
Michigan, October 2000.

16 http://cvrr-axis[1-4].ucsd.edu (only 1,2, and 4 active as of 2/15/2001)

taken with two freeway overpasses, located approximately 150m apart with non-overlapping views. The views from the two cameras also have significantly different viewing angles: one is viewing oncoming traffic, and the other departing traffic. The data was collected with handheld Hi-8 cameras, and then digitized on a standard PC using a Matrox Meteor II frame capture card. Test data sets are stored as AVI movie files.

Data from both sources was used in the development and testing of the above algorithm. The test bed data provides “easy” scenarios in a highly controlled one-lane environment, avoiding or minimizing many common problems in vehicle tracking, such as vehicles changing lanes, types of vehicle occlusion, and high-speed vehicles passing one another. The test bed data consists mostly of 2-3 vehicle platoons, with a high number of the vehicles being campus shuttles that are large and easy to track. The high perspective view of the traffic also minimizes problems with occlusion. One challenge, however, is the large amount of pedestrian traffic typical in the scene. Live data from the test bed site is available at [16].

The freeway data is, on the other hand, extremely challenging. The freeway traffic exhibits high speed, traffic density, and an off-ramp immediately after the second overpass. The off-ramp tends to destabilize platoon behavior, as individual vehicles maneuver to position themselves in the right lane to take the off-ramp.
There is also a great incidence of error due to occlusion and shadow in test data in the earlier stages of vision processing, such as the segmentation and tracking stages. Also, the perspectives at the two camera sites are significantly different, compared to the test-bed data. As well as viewing vehicles from very different angles, a wider perspective is used, resulting in less resolution. However, this data is more typical of data acquired from ITS vision sensor systems. A typical image from the freeway data may contain up to 5-10 vehicles traveling at high speed and maneuvering among lanes.

Samples from these two data sets, covering a variety of traffic conditions, were extracted from recorded video. The samples were selected with the goal of representing a wide variety of the matching scenarios present at both traffic scenes. . Each sample consists of 10-15 seconds of video during which a platoon passes through the cameras site, as well as an equal “buffer” of video before and after the sample to prevent making the identification trivial. A ground truth was acquired by manually identifying matching platoons in the two cameras views in both data sets. This ground truth was used to calculate accuracy statistics. Accuracy is reported in the number of true positive identifications.

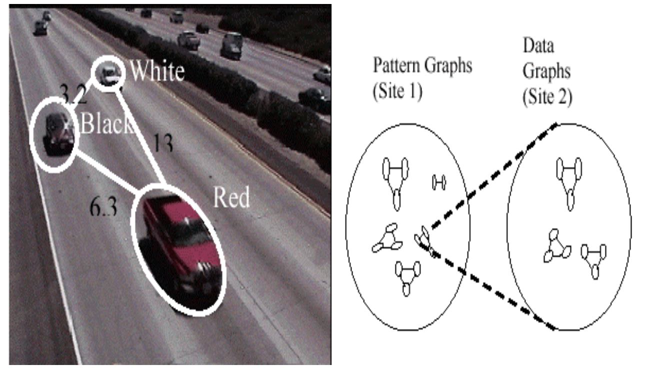

Figure 2 A simplified graph structure and outline of graph matching process

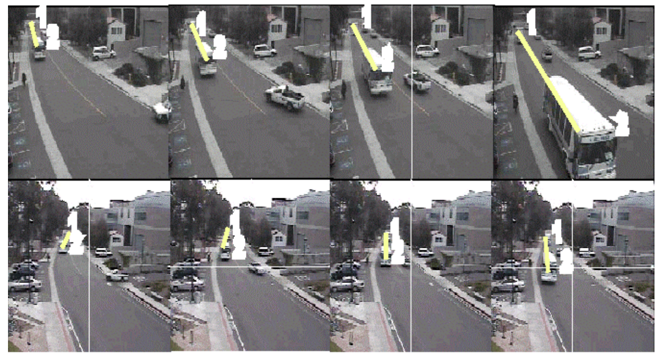

Figure 3: A successful matching sequence from the test bed video network. This sequence shows a simple 2-node platoon being tracked through two camera views.

V. RESULTS

Data Set	*# Samples*	*Mean Platoon Size*	*# True Positive Matches*	*Match %*
Test Bed	31	2.2	27	87%
I-5	22	3.5	10	45%
Totals	53	2.6	37	65%

Errors were caused by both errors propagated from lower-level steps in the vision processing system, and from the inability of the algorithm to deal with some vehicle and platoon behavior. Examples of error in low-level processing include errors in segmentation due to shadow and occlusion. Error in the identification module was usually caused by where the arrangement of vehicles in the platoon underwent great change between the two sites. Predictably, these occurred more frequently in the high-speed, high-volume freeway data than in the test-bed data. For example, the “fast” left-hand lane contained many cars traveling much faster than the general traffic. Vehicles from this lane tended to outrun their platoon from
Site 1, and entered a previously detected platoon, reducing the quality of the match for graphs created from both platoons. The aforementioned difference in perspective

between the two camera views in the freeway data also reduced the overall quality of the resulting data.
Another cause of error was an off-ramp just after Site 2. Vehicles in the center lane often moved suddenly toward the outer lanes in order to take this off-ramp, resulting in different platoon configurations between Sites 1 and 2.

VI. Discussion

This paper has considered the use of platoons of vehicles, and the spatial structure of such platoons, in the problem of maintaining the identity of vehicles as they pass through multiple camera sites. This approach differs from previous work that uses only features of individual vehicles. The use of information about the relative position of cars helps reduce the ambiguity in matches that results from the relatively homogenous colors of vehicles, such as the large number of white vehicles. The initial results show that the system may be a useful implementation of data fusion in ITS infrastructures with large numbers of visual sensors. However, the results also show that sensor placement and traffic behavior in the sensor environment will be very important factors in implementations of algorithms of this nature. Accuracy improves greatly with close sensor proximity and predictable vehicle behavior. The sensitivity of the algorithm to the quality of data from low-level vision processing also indicates the need for the system to be based on a robust vision architecture.

VII. Future Work

The results presented were based on only a small number of samples from two traffic scenes, and without the unexpected difficulties introduced by real-time traffic sensing. The performance of the presented system should further be tested with much larger volumes of data from a variety of scenes and traffic environments. Also, other methods of modeling the spatial relationships among vehicles in a platoon should be explored, as Euclidean distance contains only limited information. Vehicle features other than color should be explored as well. Potentially useful features include texture features such as wavelets.
Future work also includes incorporating the identification system into a complete, real-time ITS vision system, rather than working offline on recorded samples.

VIII. Acknowledgements

Our research is supported in part by the California Digital Media Innovation Program in partnership with The California Department of Transportation (Caltrans). We wish to thank our colleagues from the Computer Vision and Robotics Research Laboratory who are also involved in related research activities.

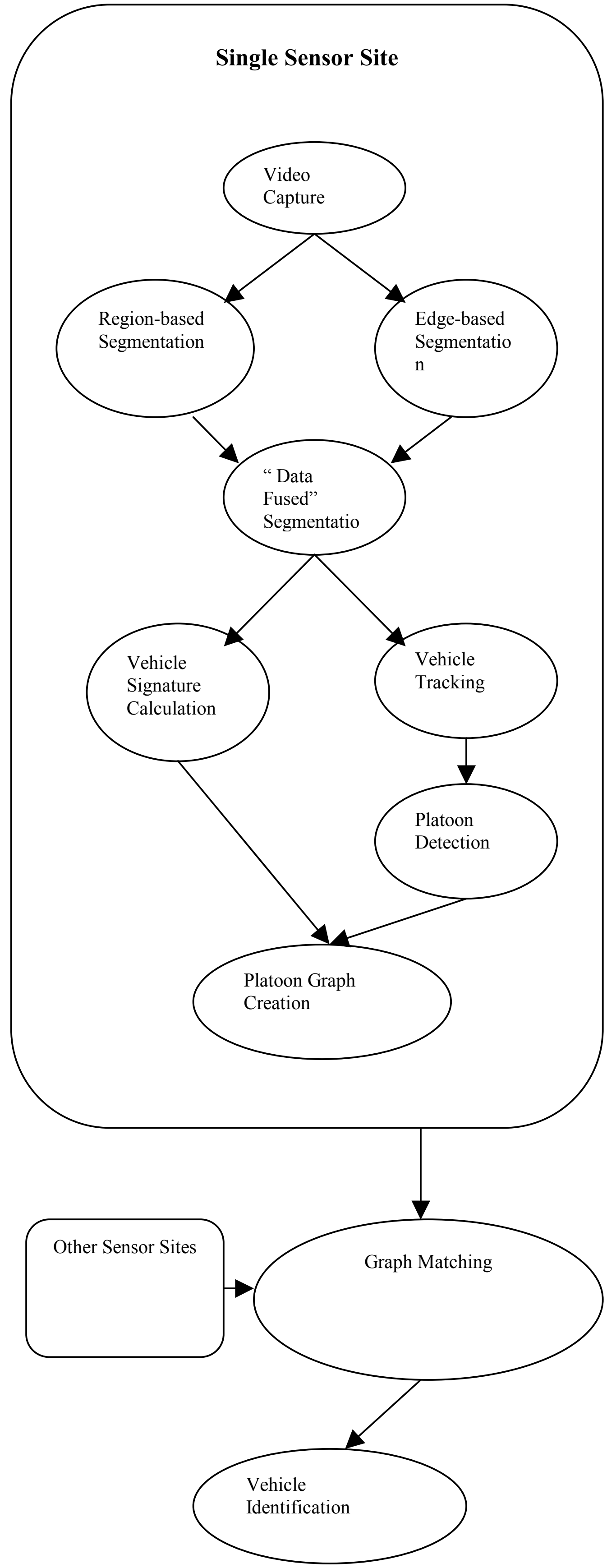

Figure 5: Data Flow Diagram. Two arrows entering the same node indicates data fusion