Segmentation and Matching of Vehicles in Road Images

MARIE-PIERRE DUBUISSON, ANIL K. JAIN, AND WILLIAM C. TAYLOR

The use of image processing for the segmentation and matching of vehicles in road images is described. Two cameras sense images of the road at different sites in order to estimate the travel time required for a vehicle to travel between two points in a road network. The color images are digitized and analyzed so that moving vehicles can be extracted and segmented into identifiable parts such as the roof, hood, trunk, sides, and wheels. These parts and their attributes such as shape and color are then used to match the vehicles observed from the two different sites. A good match implies that the same vehicle has been sensed by both cameras.

Many research projects are being conducted under the general title of intelligent vehicle-highway systems (IVHS). One of the two primary objectives of IVHS is to reduce travel time by helping the traveler avoid congestion and find the minimum travel time path through a street network. The ability to measure changes in travel time is central to the evaluation of all advanced traveler information systems (ATIS) and advanced traffic management systems (ATMS).

Current techniques for estimating travel time through a network include the floating car and the average speed procedures. These techniques are both labor-intensive and data poor. A vehicle and driver can collect only one observation of the travel time from Point A to Point B in a network on each run. If the network is large or the travel speed low, it can result in only one observation per day in the period of interest (rush hour). Thus, a statistical analysis of changes in the travel time due to a change in traffic signal timing may not be possible because of a limited data set.

License plate matching techniques provide a more robust data base, but they too are labor-intensive. This technique requires observers stationed at each point of interest and coding of the data before a matching algorithm can be exercised. The intent of this research is to develop a procedure that uses remote data collection techniques (using video cameras) and an algorithm that requires no intermediate data coding.

Another potential application of the video image processing technique is the estimation of a real-time origin-destination (O-D) matrix. Several recent studies have looked at algorithms to estimate an O-D matrix using link traffic counts (*1,2*). However, no solution has been found that is accurate but not data-intensive. These algorithms require prior knowledge (or simultaneous data collection) on link volumes and intersection turning volumes. Video image processing has the potential of providing data on the flow from each entry point to all exit points in the network simultaneously—thus an O-D matrix. Although it would require multiple cameras, it would not be labor-intensive for either data collections or data reduction.

This study addresses the data collection problems inherent in current techniques for evaluating ATMS or ATIS. The objective is to explore visual image processing to determine automatically the average travel time from any entry point to any exit point using individual vehicles as probes. The information can then be used in before and after studies to evaluate traffic signal system algorithms and vehicle routing algorithms and to serve as input to driver information systems. To solve this problem, it is necessary to be able to recognize individual vehicles from these images. A block diagram of the system is shown in Figure 1.

Vehicle recognition is a well-studied problem in dynamic scene analysis in the computer vision literature. A segmentation algorithm is applied to the sensed image to extract some features or attributes from the detected vehicle, such as its shape, wheel location, and color. At exit points, images are scanned to search for the same attributes and match them with entering vehicles; a good match means that the same vehicle was seen twice, and the travel time can be estimated by computing the difference in time of the two images.

It is not necessary to match every vehicle, since a large data set can be assembled from the traffic stream, and the present interest is only average travel time data. In this study, the images were obtained where the volume was sufficiently low to have only one vehicle per image. However, the approach will be generalized to the case of multilane highways and higher-volume traffic as the algorithms are refined. The image processing algorithms used here are briefly depicted in Figure 2 and detailed in the rest of the paper.

DETECTION OF MOVING OBJECTS

The simplest method for detecting a moving object in a fixed background is by image subtraction. In a method proposed by Jain et al., the detection of a moving object requires the object to have moved by an amount at least equal to its length in the image (*3*). The boundary of the object can then be extracted perfectly. This assumption was considered too restrictive, since it requires a large field of view that could contain two vehicles. To extract the details of individual vehicles, it would be desirable to have the vehicle occupy as much of the image as possible. In other words, a compromise between the perfect contour of the moving object and a small field of view is required. The method developed in this study does not produce a perfect contour of the moving object, but

Department of Civil and Environmental Engineering, Michigan State University, East Lansing, Mich. 48824.

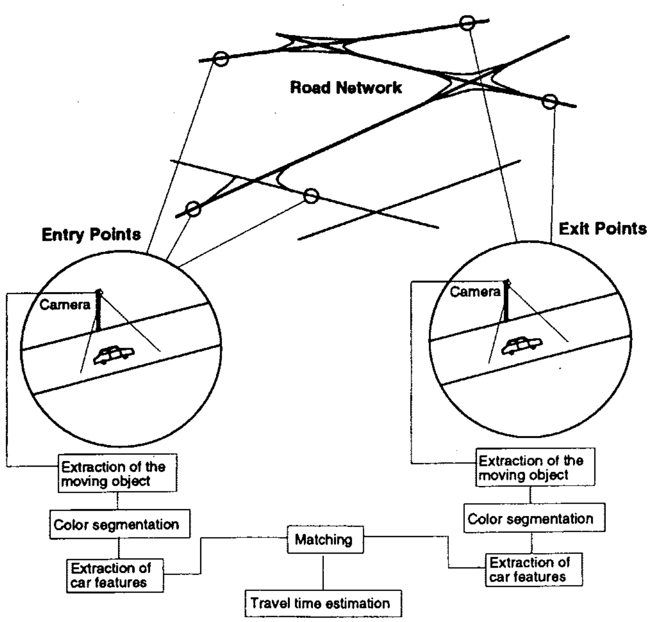

FIGURE 1 Road network and associated sensors and processing modules.

it has the advantage of using a smaller field of view, resulting in a larger area for the vehicle in the image frame.

Two color images are input to the algorithm, the second image being a few frames later than the first image in the sequence (interframe time is 1/30 sec). The images are 512 × 512, and each pixel is represented by a vector of intensity values in the three-dimensional color space (red, green, and blue). Each color component is quantized into 256 levels. Let

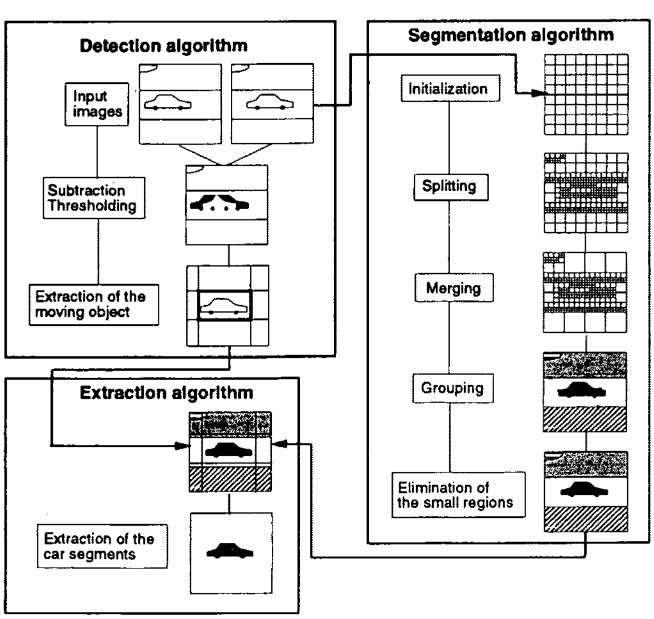

FIGURE 2 Image processing algorithms.

$\mathbf{f}^1$ and $\mathbf{f}^2$ represent the two images captured at times t_1 and t_2:

$$\mathbf{f}^1 = [\mathbf{f}^1(i,j)] = [f_R^1(i,j), f_G^1(i,j), f_B^1(i,j)] \tag{1}$$

$$\mathbf{f}^2 = [\mathbf{f}^2(i,j)] = [f_R^2(i,j), f_G^2(i,j), f_B^2(i,j)] \tag{2}$$

where $\mathbf{f}^1(i,j)$ is the intensity vector at pixel (i,j). The two images are subtracted at each pixel and thresholded to produce a binary image I in the following manner.

If

$$|f_R^1(i,j) - f_R^2(i,j)| \geq t \vee |f_G^1(i,j) - f_G^2(i,j)| \geq t \vee |f_B^1(i,j) - f_B^2(i,j)| \geq t \tag{3}$$

then

$$I(i,j) = 1, \qquad \text{else } I(i,j) = 0 \tag{4}$$

where $\vee$ refers to the logical OR operator. The parameter t is called the threshold; this case, t has been fixed at 30.

Thresholding the difference image produces a binary image in which most of the "on" pixels (pixels with the value 1) belong to the moving vehicle. Some other pixels take the value 1 because of the noise or the movement in the background (for example, wind in the tree foliage or the grass), but these pixels either are too isolated or belong to small groups that cannot be part of a vehicle. A rectangular area containing the moving vehicle is extracted by summing over the rows and columns of the binary image $I(i,j)$ and finding the rows and columns where these sums are significant. The row sums must be greater than 1/10 of the image length, and the column sums must be greater than 1/20 of the image height. These numbers would be modified if a larger field of view is used where the vehicle sizes appear to be smaller in the image.

Figure 3 summarizes the process of extracting moving objects. Figures 3*a* and 3*b* show the red components of frame i and frame $i + 3$ in the input image sequence. Figure 3*c* shows the binary image obtained after subtraction and thresholding and Figure 3*d* shows the rectangular area surrounding the moving vehicle in frame $i + 3$.

The initial vehicle image data base used in this study contains 22 road images. Each color image contains one vehicle. These images were captured by recording a road scene outside the engineering building on the Michigan State University campus. A videocassette recorder was used in combination with an image digitizer on a workstation to acquire the still images. For each of the 22 images, a bounding rectangle for the moving object was correctly identified. Thus, this method can be used to accurately determine which part of the image contains the moving vehicle.

COLOR SEGMENTATION

Color segmentation is the process that divides the image into homogeneous regions (called segments) using the color information at each pixel. Three methods are commonly used to perform this task.

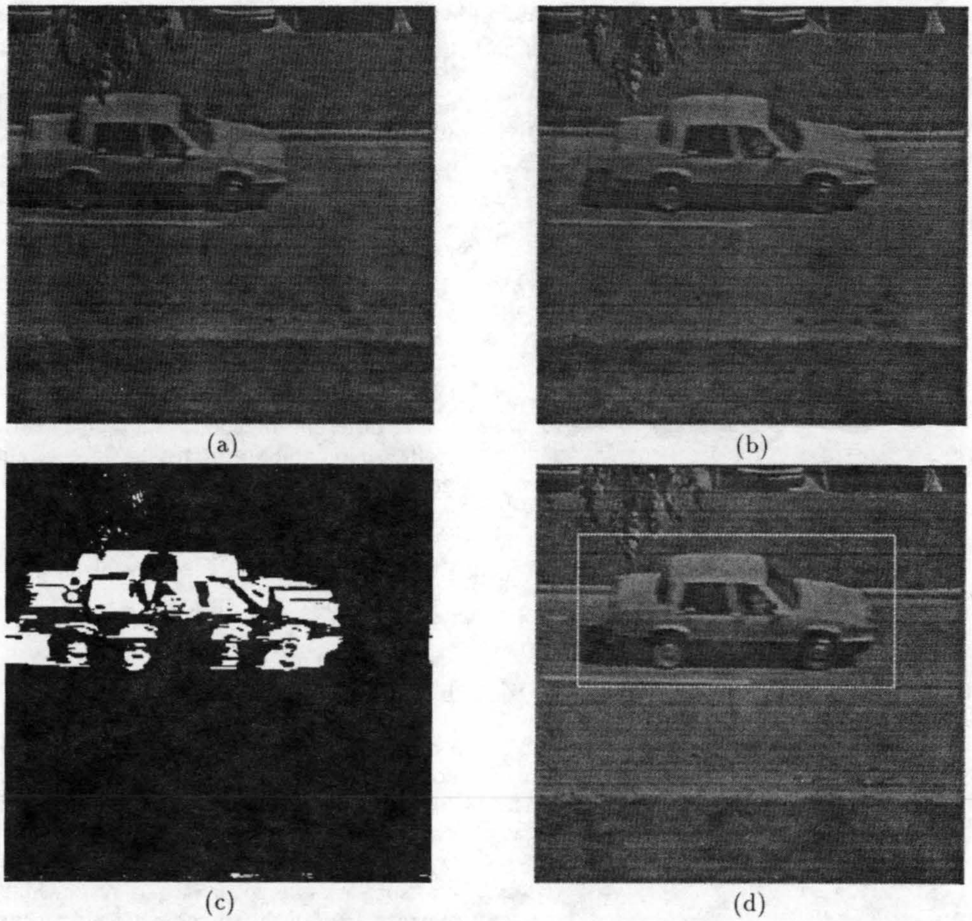

FIGURE 3 Detection of moving vehicle: *a*, Frame *i*; *b*, Frame $i+3$; *c*, output of subtraction and thresholding; and *d*, rectangular area outlining moving object.

Clustering

Clustering consists of finding clusters of points in the three-dimensional color space and assigning each cluster to a different segment (*4*). The Euclidean distance metric is commonly used to find points that are sufficiently similar to each other to define a cluster. The main disadvantage of this method is that the number of clusters (or segments) needs to be specified in advance.

Recursive Region Splitting

Recursive region splitting takes a region of the image and determines a threshold in one feature by finding the best peak in the set of feature histograms (initially the entire image is considered) (*5*). The subregions defined by this threshold are then further segmented if necessary. Ohlander used the following features: R, G, B (original tristimuli red, green, and blue), I, H, S (intensity, hue, and saturation), and the linear combination Y, I, Q (color system for TV signal). See the work by Ballard and Brown for more details on these color components (*6*). This algorithm can be very unstable because of the nonlinearity of the functions used to compute the hue and saturation components. Ohta et al. showed, using a Karhunen-Loeve transformation, that the transformed features

$$I1 = (R + B + G)/3$$
$$I2 = (R - B)/2$$
$$I3 = (2G - R - B)/4 \qquad (5)$$

produced the best segmentation results (*7*).

Split-and-Merge Algorithm

The main advantages of the split-and-merge algorithm (*8*) are that the number of segments need not be known in advance, allowing the background to be arbitrarily complicated, and the detected homogeneous regions are connected—the spatial location of a pixel plays a role in its assignment to a region. In this project, a modification of this algorithm was used as described in the following. The color segmentation algorithm is divided into five steps.

Initialization

The $N \times N$ image (where $N = 2^k$) is divided into a number of square subimages. For example, a 512×512 image is divided into 64 squares that are 8 × 8. In each square subimage, the mean and variance of the intensity values for each of the three color components are computed as follows:

$$m_R = \frac{1}{z^2} \sum_{i=x}^{x+z} \sum_{j=y}^{y+z} R(i,j)$$

$$m_G = \frac{1}{z^2} \sum_{i=x}^{x+z} \sum_{j=y}^{y+z} G(i,j)$$

$$m_B = \frac{1}{z^2} \sum_{i=x}^{x+z} \sum_{j=y}^{y+z} B(i,j) \qquad (6)$$

and

$$v_R = \frac{1}{z^2} \sum_{i=x}^{x+z} \sum_{j=y}^{y+z} [R(i,j) - m_R]^2$$

$$v_G = \frac{1}{z^2} \sum_{i=x}^{x+z} \sum_{j=y}^{y+z} [G(i,j) - m_G]^2$$

$$v_B = \frac{1}{z^2} \sum_{i=x}^{x+z} \sum_{j=y}^{y+z} [B(i,j) - m_B]^2 \qquad (7)$$

where (x,y) are the coordinates of the upper left corner of the square and z is the length of the subimage (initially $z = 8$).

Splitting

Subimages that are not homogeneous are further split into four squares of length $z/2$ and new means and variances are computed for each new square. This is repeated until the individual subimages have reached an area of 4 pixels. A subimage is not homogeneous if

$$(\sigma_R^2 + \sigma_G^2 + \sigma_B^2) > t_\sigma \qquad (8)$$

Merging

If four adjacent subimages have not been split in the second step and satisfy the homogeneity condition

$$\{|m_{R_i} - m_{R_j}| + |m_{G_i} - m_{G_j}| + |m_{B_i} - m_{B_j}|\} < t_m \quad \forall i,j \qquad (9)$$

then they are merged to form a larger subimage of length $2z$.

Grouping Subimages To Form Regions

Grouping subimages eliminates the arbitrary region boundaries imposed by the initialization process. Square subimages are merged into regions if they satisfy the criterion

$$\{|m_R - M_R| + |m_G - M_G| + |m_B - M_B|\} < t_g \qquad (10)$$

where the M's are the means for the region currently being formed. A region grows by the addition of neighboring subimages when they satisfy this criterion. The region means are updated every time a new subimage is added to the region. When no more subimages can be added, a new region is created.

Deleting Small Regions

During this segmentation process, some very small regions are created, which are then removed by merging them with neighboring larger regions. This is done on the basis of a minimum size threshold, t_s.

The different threshold values given earlier are parameters to the segmentation program. Since no universal segmentation algorithm has been found yet, these parameters must be adjusted to properly segment the specific images being analyzed. In this study, the parameters were fixed at $t_m = 50$, $t_v = 50$, $t_g = 40$, and $t_s = 250$ for all the images used in the experiments.

Figure 4 shows the segmentation results for the vehicle image shown in Figure 3*b*. A total of 38 segments have been identified in this image. One segment corresponds to the road and two others correspond to the grass. The curb and marking on the road can also be extracted. Different segments appear for the leaves in the tree and the vehicles parked in the parking lot in the background. Meaningful segments are also extracted from the vehicle. The roof, hood, trunk, two windows, and wheels can be distinguished. The use of color information in the split-and-merge process is responsible for this excellent performance of the segmentation algorithm.

EXTRACTION OF VEHICLE

Once the image has been segmented and the moving object has been detected, those segments that are part of the moving object can be extracted from the segmented image to obtain a segmented image of the vehicle alone. This is done by considering all the segments and discarding those that are outside the bounds of the rectangular area. The segments for which

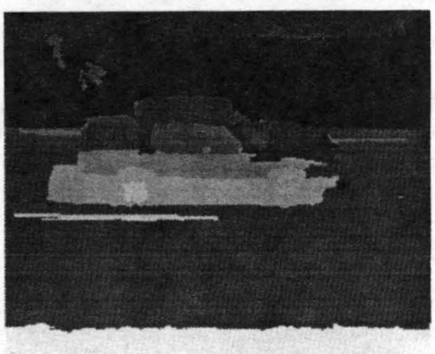

FIGURE 4 Color segmentation results.

the intersection of their area with the bounding rectangular area is less than 80 percent are also eliminated. Also, elongated regions that belong to either the curb or the markings on the road can be suppressed by looking at each column of the image and setting to the background color the regions (in the background) of height fewer than 15 pixels. The result of this process is shown in Figure 5. Note that the vehicle in Figure 5 was extracted by taking the intersection of the bounding rectangle in Figure 3*d* and each of the 38 segments in Figure 4. It can now be seen that only the segments belonging to the vehicle are retained.

This algorithm was tested on 22 car road images, each containing one vehicle. The vehicle segments have been extracted successfully in all the images on which this algorithm has been tried. Results for two other vehicles are shown in Figures 6 and 7. Again, only regions belonging to the vehicle are kept, and these regions represent meaningful parts of the vehicle.

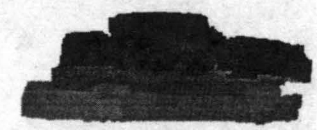

FIGURE 5 Extraction of vehicle from background.

INITIAL MATCHING RESULTS

For each vehicle, a low-level description now exists in terms of a number of segments. These segments can be used by a matching algorithm to determine if a given vehicle has been sensed at an earlier time. The matching algorithm used is based on a graph matching approach (*6*). A graph is built to obtain a high-level description of each vehicle. A node in this graph corresponds to a segment in the segmented image. The node attributes are the number of pixels in the segment, the average color of the segment (three components: red, green, and blue), and the perimeter of the segment. Two nodes are connected by an edge in the graph if the corresponding two segments are neighbors in the segmented image. The arcs in the graph also have attributes, which are the distance between the centroids of the two corresponding segments and the orientation of the line linking those two centroids. An example of this relational graph is shown in Figure 8.

Matching two vehicles is equivalent to matching the corresponding two relational graphs. A similarity measure can be computed by comparing the node attributes and the arc attributes in the two graphs. This is done as follows: the best matching two nodes are found by comparing their attributes, and the next best matching two nodes are found by comparing their attributes and their relationship (link attributes) with the other nodes already matched. This process stops when all the regions in at least one of the vehicles have been matched.

Among the 22 road images that were available, 3 showed the same vehicle at different times (Vehicles 5, 15, and 21). For experimentation purposes, a known vehicle was driven three times around the building. A similarity measure was obtained for each possible pair of vehicles. Table 1 presents the similarity measures for some of those pairs of segmented images. The similarity measures are 0.95 between car5 and car15, 0.95 between car5 and car21, and 0.94 between car15 and car21. The high similarity measures indicate that the vehicles must be the same, and indeed they are, as explained earlier. The other similarity measures, however, are quite low, indicating that the two vehicles are not the same.

One of the remaining questions is the definition of the level of the similarity measure required to conclude that a match has been made. In Table 1, car8 has similarity measures of 0.90 with car5 and car21 and 0.86 with car15, yet these are not matches. Using a threshold of 0.91 would eliminate these false matches, but if the threshold were set at 0.95, the true match between car15 and car21 would have been missed.

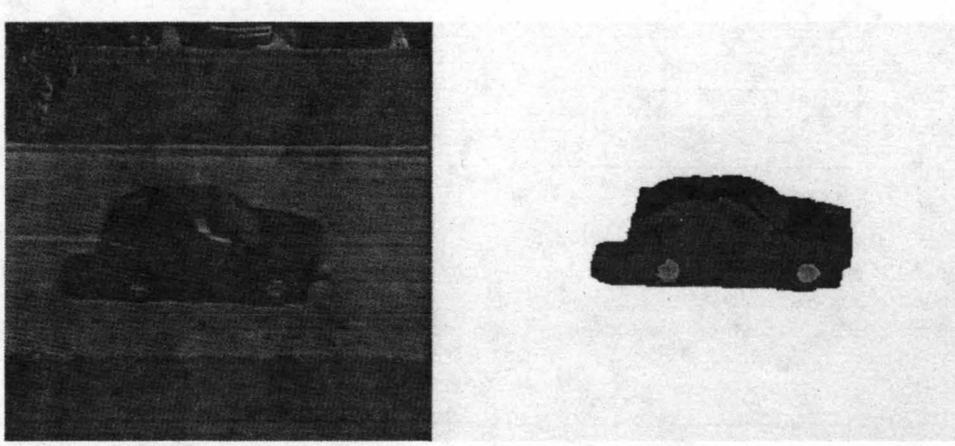

FIGURE 6 Results of region segmentation, Vehicle 1: *left*, original image; *right*, region segmentation of vehicle only.

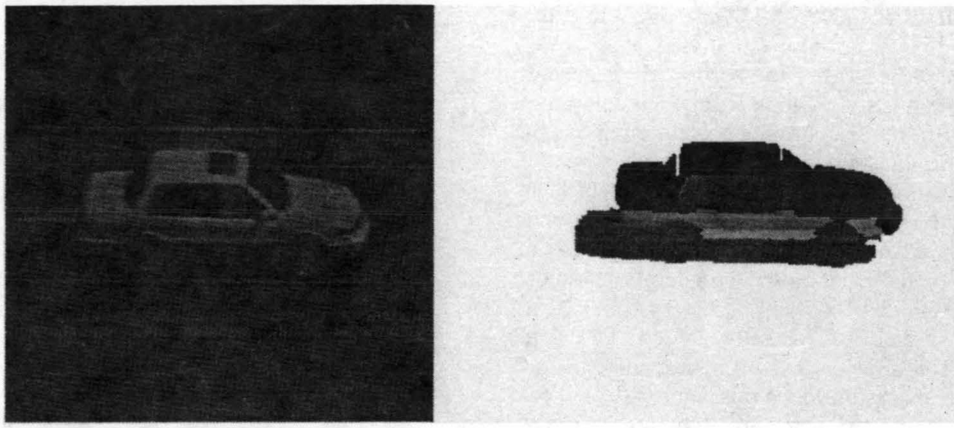

FIGURE 7 Results of region segmentation, Vehicle 2: ***left*****, original image,** ***right*****, region segmentation of vehicle only.**

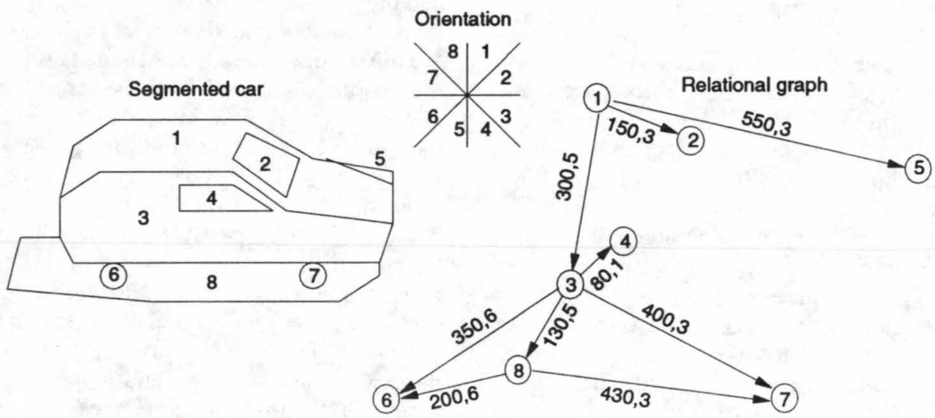

FIGURE 8 Relational graph.

TABLE 1 Initial Matching Results

	car4	car5	car6	car7	car8	car13	car15	car17	car18	car21
car4	1.00	0.08	0.34	0.42	0.09	0.39	0.07	0.12	0.07	0.05
car5		1.00	0.04	0.36	0.90	0.06	**0.95**	0.58	0.34	**0.95**
car6			1.00	0.06	0.31	0.12	0.05	0.32	0.17	0.05
car7				1.00	0.09	0.27	0.07	0.17	0.48	0.08
car8					1.00	0.54	0.86	0.58	0.05	0.90
car13						1.00	0.05	0.10	0.08	0.26
car15							1.00	0.56	0.31	**0.94**
car17								1.00	0.42	0.56
car18									1.00	0.05
car21										1.00

These results show that the segmented image of a vehicle can be used for matching. Once pairs of vehicles have been matched, the travel time for a given vehicle can be computed if the time it was sensed by the first camera and the time it was sensed by the second camera are known.

CONCLUSIONS

It is possible to extract the moving vehicle from the background and segment it into meaningful regions such as the roof, hood, trunk, and wheels. The quality of the segmented images is very good, because color has been used during the segmentation process. These segmented regions—along with their shape, color, and the relationships between them—describe the structure of the vehicle fairly well. A relational graph is then constructed to summarize this information. A graph matching algorithm is used to determine whether a vehicle observed by the second camera at an exit point has been seen by the first camera at an entry point of the road network.

A number of matching algorithms have been developed and presented in the literature (9–11). Future work will involve an investigation of a better matching algorithm for this problem and extending the process to handle images of multilane highways. The problem of handling multilane highways can be solved either by ignoring the vehicles that do not appear in the front lane (they would still need to be separated from the vehicle in the first lane in the image, which is a difficult image segmentation problem) or by monitoring each lane separately, which would require many more cameras. It is almost certain now that the first solution will be chosen.

ACKNOWLEDGMENTS

This research was supported by a grant from the U.S. Department of Transportation through the Great Lakes Center for Truck Transportation Research.

REFERENCES

1. Moshe, B.-A. Methods To Combine Different Data Sources and Estimate Origin-Destination Matrices. In *Transportation and Traffic Theory* (N.H. Gartner and N. H. M. Wilson, eds.), 1987, pp. 459–481.
2. Cremer, M., and H. Keller. A New Class of Dynamic Methods for the Identification of Origin-Destination Flows. *Transportation Research*, Vol. 21B, No. 2, 1987, pp. 117–132.
3. Jain, R., and H.-H. Nagel. On the Analysis of Accumulative Difference Pictures from Image Sequences of Real World Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 1, No. 2, April 1979, pp. 206–214.
4. Jain, A. K., and R. C. Dubes. *Algorithms for Clustering Data.* Prentice-Hall, Englewood Cliffs, N.J., 1988.
5. Ohlander, R., K. Price, and R. Reddy. Picture Segmentation Using a Recursive Region Splitting Method. *Computer Graphics and Image Processing*, Vol. 8, 1978, pp. 313–333.
6. Ballard, D. H., and C. M. Brown. *Computer Vision.* Prentice-Hall, Englewood Cliffs, N.J., 1982.
7. Ohta, Y.-I, T. Kanade, and T. Sakai. Color Information for Region Segmentation. *Computer Graphics and Image Processing*, Vol. 13, 1980, pp. 222–241.
8. Horowitz, S., and T. Pavlidis. Picture Segmentation by a Tree Traversal Algorithm. *Journal of the Association for Computing Machinery*, Vol. 23, No. 2, April 1976, pp. 368–388.
9. Price, K., and R. Reddy. Matching Segments of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 1, No. 1, Jan. 1979, pp. 110–116.
10. Davis, L. Shape Matching Using Relaxation Techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 1, No. 1, Jan. 1979, pp. 60–72.
11. Cheng, J. K. *Algorithms for Matching Relational Structures and Their Applications to Image Processing. Ph.D. thesis. Purdue University, West Lafayette, Ind., Dec. 1980.*

Publication of this paper sponsored by Committee on Transportation Data and Information Systems.