Extracting Roadway Background Image: a Mode-Based Approach

Jianyang Zheng (Corresponding Author)
Research Assistant
Box 352700
Department of Civil and Environmental Engineering
University of Washington
Seattle, WA 98195-2700
Tel: (206) 616-6056
jianyang@u.washington.edu

Yinhai Wang, Ph.D.
Assistant Professor
Box 352700
Department of Civil and Environmental Engineering
University of Washington
Seattle, WA 98195-2700
Tel: (206) 616-2696
Fax: (206) 543-5965
yinhai@u.washington.edu

Nancy L. Nihan, Ph.D.
Professor and Director
Transportation Northwest (TransNow)
Box 352700
Department of Civil and Environmental Engineering
University of Washington
Seattle, WA 98195-2700
Tel: (206) 543-8255
nihan@u.washington.edu

Mark E. Hallenbeck
Director
Washington State Transportation Center (TRAC)
University of Washington, Box 354802
University District Building
1107 NE 45th Street, Suite 535
Seattle, WA98105-4631
Tel: (206) 543-6261
Fax: (206) 543-5965
tracmark@u.washington.edu

Submission date: August 1, 2005, re-submission data: November 15, 2005
Word count: 3646 + 6 * 250 = 5146

ABSTRACT:
Traffic monitoring cameras are widely installed on streets and freeways in U.S. metropolitan areas. Video images captured from these video cameras can be used to extract many valuable traffic parameters through video image processing. A popular way to capture these traffic data is to compare the current traffic images with the background image, which contains no vehicles or other moving objects, just background such as pavement. Once the moving vehicle images are separated from the background image, measurements of their number, speed, etc. can be obtained.

Background images are typically extracted from a video stream through image processing because, for normal traffic streams on urban streets, it may be hard to find a frame without any vehicles. This paper introduces a new method that can quickly extract the background image from traffic video streams for both freeways and intersections in a variety of prevailing traffic conditions. This method has been tested with field data and the results are promising.

1 INTRODUCTION

Traffic data, such as vehicle volumes, are critical for traffic management and other transportation applications. Of the prevailing traffic sensors for collecting such data, video cameras are among the most popular because they have the ability to capture not only traffic volumes, but also speeds, vehicle classifications, queue lengths, control delays, and other traffic parameters. These parameters can be obtained through detecting and tracking vehicles based on their features *(1)* or profiles *(2)*. Another popular approach to obtain these parameters is called "image subtraction." It subtracts background image from current image to detect objects moving across a constant scene *(3)*. The constant scene is referred to as the background, which contains merely static objects (e.g. road pavement, roadside buildings, etc.) and clear from moving vehicles or pedestrians. Some of the methods detect vehicles by directly comparing the three color values (measures of red, green, and blue). To eliminate the effects from the changes of illumination, Rojas and Crisman proposed to compare color in the chromaticity plane, where the measured colors of the same object under different illuminations will concentrate to one value. The color values in the chromaticity plane can be easily transformed from the RGB color space *(4)*. The background image can also be updated with a Kalman filter-based adaptive model to accommodate the change of lighting conditions *(5)*. Therefore, the background image must be known to run the image subtraction algorithm efficiently. To get the background image of a scene, we need to use video image processing to extract the background information from a series of traffic images, because it is nearly impossible to find a frame in a traffic video stream without a single moving vehicle or pedestrian on a busy urban road.

There are several ways to extract background images from traffic video streams. Avery et al *(6)* introduced a background image extraction algorithm based on the changes of pixel colors from frame to frame. Each pixel in the previous frame was compared with the same pixel in several consecutive frames. If their color values stayed the same, it was assumed that this pixel was not occupied by moving vehicles and the color values were assigned to the corresponding pixel on the background image. This process was iterated until color values for all pixels in the background image were determined. This algorithm proved to work well on freeways under non-congested conditions. Gupte et al *(7)* used a similar method to extract background images. They first calculated a binary motion mask, which is the subtraction of two successive frames. Any pixel with different color values between these two frames was assumed to be part of a moving object. A motion mask was used as a gating function to extract the background image from traffic images by filtering out moving objects. After a sequence of frames is processed, the entire background image could be extracted. Elgammal et al. *(8)* studied each pixel's value in the three color channels (red, green, and blue) in an image sequence, and tried to find the distribution of these values. They assumed that the values formed a Gaussian distribution, where the probability of a pixel being a background pixel could be estimated. Then the decision on whether it is a background or a foreground pixel is made by comparing the estimated probability with a given threshold. This method can be used for both freeways and intersections. Cucchiara et al *(9)* also looked at pixel values in an image series, but they suggested using the medians of color values in the series as the background values. This method can also be applied to both freeways and intersections. Zheng et al. *(10)* used median background extraction in a video image application for detecting signal-cycle failures at urban intersections. The background extraction method requires sorting the image series to obtain the median values and hence demands more computing time.

In this paper, we introduce a new method that can quickly extract background images by finding the mode of the pixel value series. The details of this method are presented in the following section. Then field tests on this method are described and the test results are discussed. Conclusions about the utility of this method are drawn at the end of this paper.

2 METHODOLOGY

In a series of traffic image frames, the color values of a pixel at location (x, y) at time t are I(x, y, t). In most cases the pixel's color values are in the Red, Green, and Blue (RGB) color space, and I(x, y, t) is a vector with three elements: $I^R(x, y, t)$, $I^G(x, y, t)$, and $I^B(x, y, t)$. For a given pixel (x, y), its color values from time t_i to time t_{i-n} are represented by matrix M(x, y) as follows:

$$M(x, y) = \{I(x, y, t_i), I(x, y, t_{i-1}), I(x, y, t_{t-2}), ..., I(x, y, t_{i-n})\} \tag{1}$$

This given pixel has only two possible states in the traffic image: obstructed by a moving vehicle or clear from obstruction. If it is obstructed by a moving vehicle (common referred as foreground), the observed color values are actually from the obstructing vehicle. The pixel's color values in the periods of being obstructed can be expressed as:

$$M_F(x, y) = \{I(x, y, t_{F1}), I(x, y, t_{F2}), I(x, y, t_{F3}), ..., I(x, y, t_{F.j})\} \tag{2}$$

where t_{F1}, t_{F2}, t_{F3}, …, and $t_{F.j}$ are the times when the pixel is occupied by the foreground.

If this pixel is not obstructed by a moving object, its observed color values should be those of the background. The pixel's color values in this case are in the following set:

$$M_B(x, y) = \{I(x, y, t_{B1}), I(x, y, t_{B2}), I(x, y, t_{B3}), ..., I(x, y, t_{B.k})\} \tag{3}$$

where t_{B1}, t_{B2}, t_{B3}, …, and $t_{B.k}$ are the times when the pixel is not occupied by the foreground.

$M_F(x, y)$ and $M_B(x, Y)$ have the following relationships:

$$M(x, y) = M_F(x, y) \cup M_B(x, y) \text{ and } \phi = M_F(x, y) \cap M_B(x, y) \tag{4}$$

where ϕ represents an empty set.

Since the background is motionless, the color values of this pixel would approximately be the same during the entire analysis time (here we assume the analysis time is short enough to ignore the luminance change of the scene):

$$I(x, y, t_{B1}) = I(x, y, t_{B2}) = I(x, y, t_{B3}) = ... = I(x, y, t_{B.k}) \tag{5}$$

For the foreground, however, since the moving vehicles occupying the pixel may be in different colors and shapes at different times, we cannot expect that:

$$I(x, y, t_{F1}) = I(x, y, t_{F2}) = I(x, y, t_{F3}) = ... = I(x, y, t_{F.j}) \tag{6}$$

Therefore, we assume that it should be the background-pixel color vector that occurs the most frequently in M(x, y) and this concludes that:

$$I(x, y, t_{Bi}) = \text{Mode}\ (M(x, y)) \tag{7}$$

where t_{Bi} is the time when the pixel is not occupied by the foreground and Mode(M(x, y)) represents the mode of color values in M(x, y).

By calculating the mode of the M(x, y), the background color values at location (x, y) can be determined. After applying the same process to each pixel in the image, a background image will be extracted.

The assumption may be violated when the analysis time period, $t_A = (t_i – t_{i-n})$, is not long enough and a stopped vehicle occupies the pixel during most of the period t_A. The possibility to violate the assumption can be lowered by increasing t_A to a reasonable level. For some applications (e.g. image extraction at a congested intersection) that require a longer t_A, we can use a lower sampling rate to balance the computing work load. With a lower sampling rate, the system can process fewer amount of frames over even a longer period t_A. In the field test we used a sampling rate of 1 fps (frame per second).

Although the background is motionless, the background color values are not always the same because of disturbances in lighting, atmosphere, cameras, etc. Consequently, Equation (5) is not usually 100% true. This is illustrated in Figure 1, which shows the histograms of a given pixel's background color values over a two minute interval. To accommodate the small variations of background pixel values due to these disturbances and make the proposed algorithm more robust, we introduce a function to aggregate several color value in a small range into one bin. The function in our algorithm is:

$$h_i = \sum_{j=i\times s}^{(i+1)\times s-1} frequency_j \text{ , } i = 0, 1, 2, \ldots, (256/s-1) \quad (8)$$

Where frequency_j is the frequency of color value j in the series M(x, y), s is the size of the range, and h_i is the aggregated bin value. With the function (8), 256 color levels can be regrouped into 256/s bins. Using the frequencies of these regrouped bins, the impacts of disturbances are reduced. This robustness is not at the cost of reduced color resolution. If bin i is identified to be the mode, then we do a secondary mode operation to find the best color value of the pixel. That is we calculate the mode again for the s color values: $i \times s$, $i \times s + 1$, $i \times s + 2$, … , $(i + 1) \times s - 1$. This secondary mode is considered the best background color value.

In this study, we used $s = 4$ according to the quality of our images. This corresponds to a total bin number of 64.

3 TEST AND DISCUSSION

All the field test data were obtained from traffic monitoring cameras that were not specifically installed and calibrated for video image processing. Although such cameras may not generate the best video data for traffic analysis purposes, they are already widely deployed across the country for day-to-day traffic management operations, and are, therefore, a potentially robust source of traffic video data. Image processing algorithms that can be proven to be sufficiently accurate in extracting traffic data from such cameras will be more robust and have a much larger application domain than those that require data from specially placed and specially calibrated cameras. Consequently, the field test data were all taken from cameras that were already in the field for traffic detection or other traffic management purposes. The Smart Transportation Applications and Research Laboratory (STAR Lab) at the University of Washington has a fiber optic line connected to the Traffic System Management Center (TSMC) at the Washington State Department of Transportation (WSDOT). This fiber optic line brings full motion video to the STAR Lab from nearly 300 surveillance video cameras deployed in the Greater Seattle area. Our freeway test video images were captured at the STAR Lab from these surveillance video cameras.

The first field test was conducted on a freeway with free-flow traffic. Figure 2 shows the extracted backgrounds for this test. Figure 2 (a) is a typical snapshot of the traffic stream at this test site. It shows that there was no traffic congestion during the test period. Figure 2 (b) ~ (e) shows the extracted backgrounds for the same video stream using different analysis time periods, t_A. Figure 2 (b) shows that when t_A = 10 seconds, the extracted background contained wrong pixel values for a significant portion of the background image. When t_A was increased to 30 seconds, most background pixels received the correct color values except for some locations at the far side of the image, identified by the author-added arrow in Figure 2 (c). When t_A was further increased to 60 seconds, the extracted background image (Figure 2 (d)) was almost perfect. As shown in Figure 2 (e), when t_A was three minutes, the results were nearly the same as when t_A was one minute. The test shows that, for a t_A longer than one minute, the algorithm will work well for freeway applications in free flow conditions.

The second field test was performed on a section of congested freeway. The traffic condition for this test site is shown in figure 3 (a). Obviously, traffic density is higher at this site than at the fist test site. Therefore, a longer t_A is desired. Figure 3 (b) ~ (e) shows the background images extracted when t_A = 2 minutes, 5 minutes, 10 minutes, and 20 minutes, respectively. The figures indicate that, when traffic is congested, the extracted background will not be as good as that for free flow condition, especially on the far side of the image. Figure 3 (e) and (f) show that, when t_A was greater than 10 minutes, the quality of the extracted background did not increase. The fact that the background quality does not improve and may even become worse as t_A grows to too long may be a result of changing factors such as the position of the sun that can, over time, affect pixel colors significantly. The quality of the background image appeared to be best when t_A was six minutes. Though the colors for the far side background pixels were not correct, they may not hurt follow-up image processing steps because traffic detections are most likely done at the near side where image resolution is better. Therefore, this method can also extract useful background image under congested conditions.

The location of the last field test was at a signalized intersection. At a signalized intersection, a vehicle may stay still for minutes even though there is no congestion. Figure 4 shows the results for this final test with different values of t_A. Figure 4 (a) shows a snapshot of the traffic when the signal was red. Figure 4 (b) shows the extracted background image when about half of the analysis period t_A had a red signal and the other half had green or yellow signals. Figure 4 (c) ~ (e) shows the extracted background when t_A was two minutes, five minutes, and 10 minutes, respectively. These images provide most of the background with good quality, especially as t_A gets longer. However, these are not perfect background images, and flawed pixels are located near the stop line, where most of the image processing is performed. Conversely, Figure 4 (f) shows a perfect background image extracted from the same traffic video stream with t_A = 1 minute. The only difference is that this image was extracted when the traffic signals were green for most of the time period t_A. Therefore, if the user can select an analysis period in the video stream that has a sufficient amount of green time, the algorithm can quickly extract a perfect background image.

The field test data can also demonstrate the effectiveness of the proposed background extraction method. A pixel in the congested freeway images was studied for this purpose. Position of the pixel was identified by the author-added square and arrow in Figure 3 (d). The pixel's histograms of three color channels in one minute were shown in Figure 5. Traffic video showed that in the test period 23 vehicles passed over the pixel, with eight vehicles in black, six in gray, five in red, and four in white. All of these vehicles were passenger cars or pickups. The frame rate was 4 frame/second in the test, and the manual counting showed that of the 240 frames in the test minute, the pixel was not obstructed by any vehicles in 116 frames, covered by black vehicles in 37 frames, by gray vehicles in 31 frames, by red vehicles in 29 frames, and by white vehicles in 27 frames. Because of these vehicles, the pixel's color histograms extended to a wider range. The distribution was close to multiple Gaussians, as identified by Stauffer and Grimson *(11)*. Based on the scenario described, the highest columns showed the correct background color values, and the lower curves showed the colors of vehicles. As shown in Figure 5, the modes of each color channel were the colors that correspond to the highest columns.

The median color values were interfered by vehicle colors and did not sit in the right ranges of the true background colors. The mean color values were the least accurate because they were interfered the most by the vehicle colors. This example indicated that the proposed background extraction algorithm using color modes was more robust than the background extraction algorithm using color medians. However, the comparison is only based on the data we collected, and we cannot conclude that the proposed method performs better in all conditions.

The proposed algorithm is not only more robust, but also faster than the median background extraction algorithm. To find the median value of a data set, the data need to be sorted first. There are several sorting algorithms, such as bubble, insertion, selection, shell, heap, merge, and quick sorts (please refer to *(12)* for details of these sorting algorithms). For bubble, insertion, selection, and shell sorts, the algorithmic complexity, or relative efficiency, is O (n^2), where O (Big O notation) is a mathematical notation used in computer science to measure the complexity of an algorithm and n represents the size of the data set. If an algorithm has O (n^2) on average, then its execution time increases quadratically with n. These sorting algorithms are the least efficient, especially when the size of the data set is large. For heap, merge, and quick sorts, the algorithmic complexity is O ($n \times \log (n)$), which is relatively faster than other sorting algorithms *(12)*. The algorithmic complexity of the mode algorithm, however, is only O (n) because it takes just one scan to find the mode of a data set. Therefore, the mode background extraction algorithm will be faster than the median algorithm, no matter which sorting algorithm is used to find the median values. This performance difference between the median algorithm and the mode algorithm proposed in this study has been demonstrated with the field data. Table 1 summarizes the computing times consumed by the two algorithms to extract background from the same video data sets. The experiments were conducted on a DELL personal computer (Pentium 4 at 3.0 GHz, 1.00GB of RAM, Windows XP). Three test video data sets, each recorded at a different location, were used for this test. The median background extraction algorithm was implemented based on the quick sort algorithm. As can be seen in Table 1, the mode algorithm performed approximately the same when the data set is small. However, when the number of processed frames gets bigger, the saving of computing time with the mode algorithm becomes more significant.

4 CONCLUSIONS

In this paper, a new algorithm to extract background images from traffic video streams is introduced. This algorithm analyzes each pixel’s color values in a series of frames captured during a particular time period, and then uses the mode of the series as the correct color value for the background image. To handle small disturbances of background color over a short time period, a function is used to aggregate neighbor color values into one bin for mode calculation. With this function, the algorithm is more robust and lowers the impact of the disturbances resulted from environmental factors. This algorithm does not require sorting and hence is easy to implement and fast to extract background images.

Field tests were performed with video data from traffic monitoring cameras that were not specifically configured for video image processing. The test results showed that the algorithm could extract nearly perfect backgrounds on freeways when traffic was in free flow condition. For congested freeways, the algorithm could still generate background images for image processing using the near side of the image. Test results also demonstrated that the algorithm is effective to generate clear and nearly perfect backgrounds at urban signalized intersections.

ACKNOWLEDGEMENTS

This research was funded by the Washington State Department of Transportation (WSDOT), Community Transit (CT), and Transportation Northwest (TransNow), the US Department of Transportation University Transportation Center for Federal Region 10, at the University of Washington. The authors also appreciate helps from Larry Senn from the WSDOT and Zohred Zandi from CT.

REFERENCES:

1. Beymer, D., P. Mclauchlan, B. Coifman, and J. Malik. A Real Time Computer Vision System for Measuring Traffic Parameters. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 495-501*, 1997.
2. Kim, Z. and J. Malik. Fast Vehicle Detection with Probabilistic Feature Grouping and its Application to Vehicle Tracking. *Proceedings of IEEE International Conference on Vision, vol. 1, pp 524-531*, 2003.
3. Shapiro, L.G. and G.C. Stockman. *Computer Vision*. Prentice Hall, New Jersey, 2001.
4. Rojas, Juan Carlos and Jill D. Crisman. Vehicle Detection in Color Images. *Proceedings of the IEEE Intelligent Transportation System Conference*, 1997.
5. Malik, Jitendra and Stuart Russell. Traffic Surveillance and Detection Technology Development: New Sensor Technology Final report. *Research Report UCB-ITS-PRR-97-6, California PATH Program*, 1997.
6. Avery, R. P., Y. Wang, and S. G. Rutherford. Length-Based Vehicle Classification Using Images from Uncalibrated Video Cameras. *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems*, October 2004, pp. 737-742.
7. Gupte, S., O. Masoud, R. F. K. Martin, and N. P. Papanikolopoulos. Detection and Classification of Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 3, No. 1, March 2002, pp. 37-47.
8. Elgammal, A., D. Harwood, and L. Davis. Non-parametric Model for Background Subtraction. *Proceedings of IEEE International Conference, Computer Vision '99 FRAME-RATE Workshop*. 1999.
9. Cucchiara, R., C. Grana, and M. Piccardi, and A. Prati. Detecting Moving Objects, Ghosts and Shadows in Video Streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 10, 2003, pp. 1337-1342.
10. Zheng, J., Y. Wang, N. L. Nihan, and M. Hallenbeck. Detecting Cycle Failures at Signalized Intersections Using Video Image Processing. *Preprint CD-ROM, the 84th Annual Meeting of Transportation Research Board, Paper 05-2002*. January 2005.
11. Stauffer, C. and W. E. L. Grimson. Adaptive Background Mixture Models fro Real Time Tracking. *Proceedings of IEEE international conference, Computer Vision and Pattern Recognition*, 1999, pp. 246-252.
12. Carrano, Frank M., Paul Helman and Robert Veroff. Data Abstraction and Problem Solving with C++. Addison Wesley Longman, 2nd edition, 1998.

LIST OF FIGURES

LIST OF TABLES

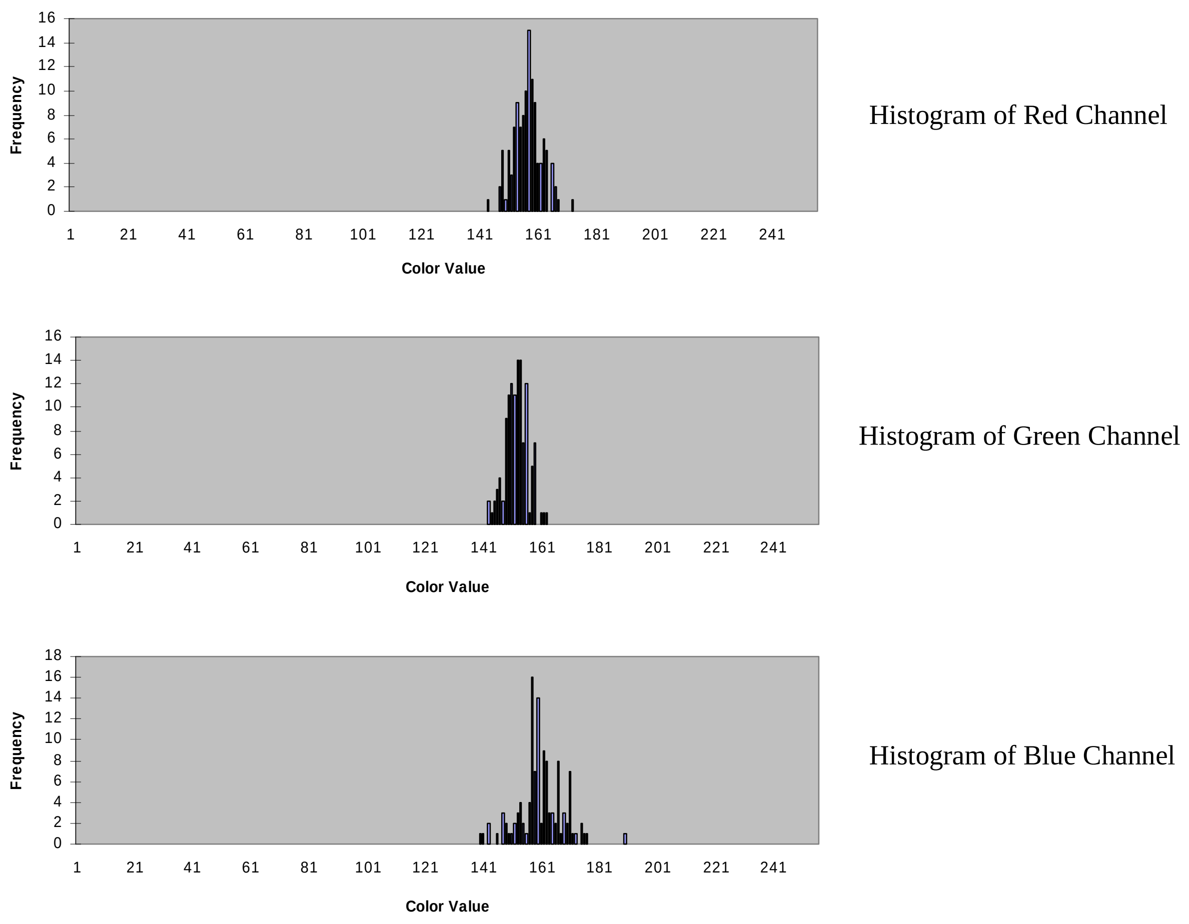
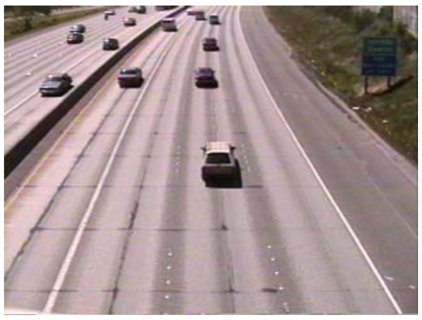
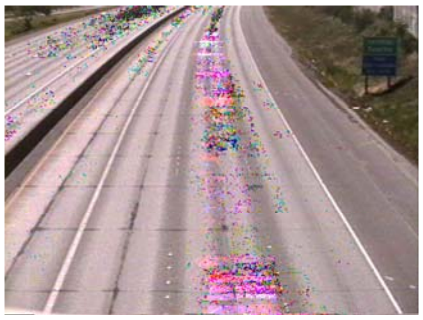


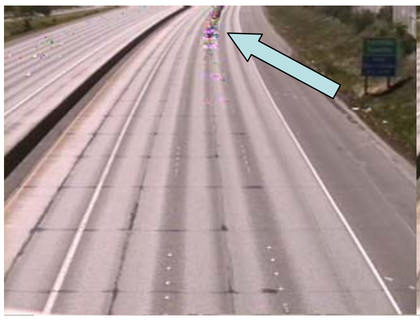
FIGURE 1   Background variance in two minutes.



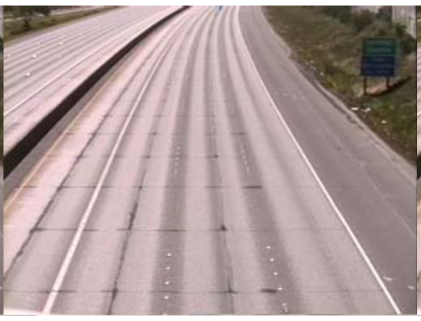
(a) Traffic Image



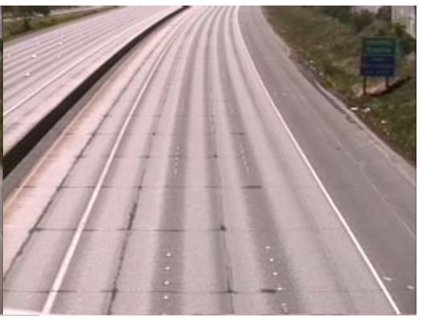
(b) Background extracted in 10 seconds



(c) Background extracted in 30 seconds

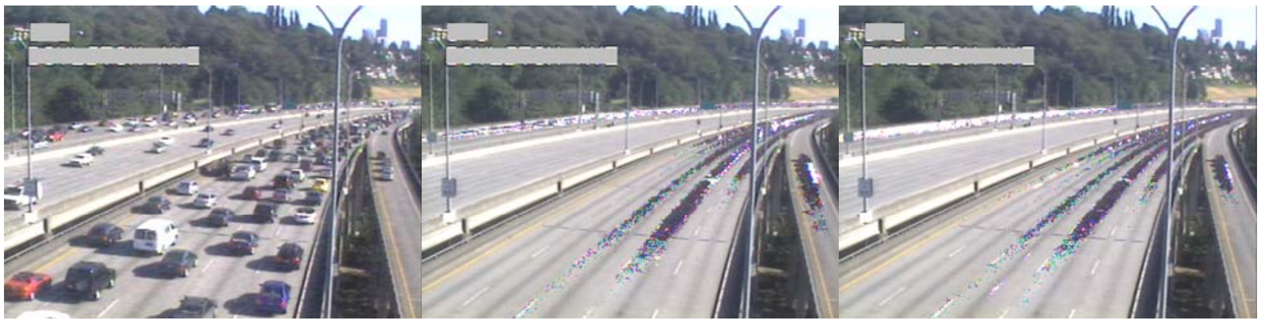


(d) Background extracted in one minute



(e) Background extracted in three minutes

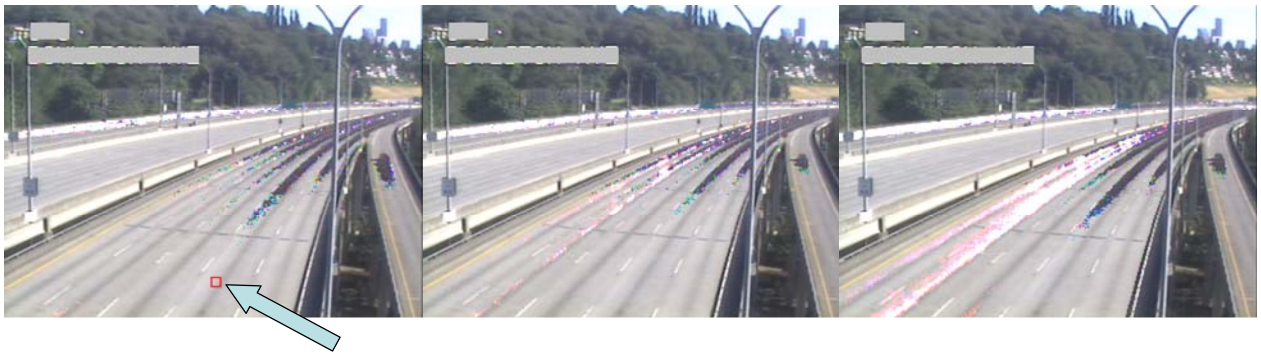
FIGURE 2 Background extraction on freeway in free flow.



(a) Traffic Image

(b) Background extracted in two minutes

(c) Background extracted in four minutes

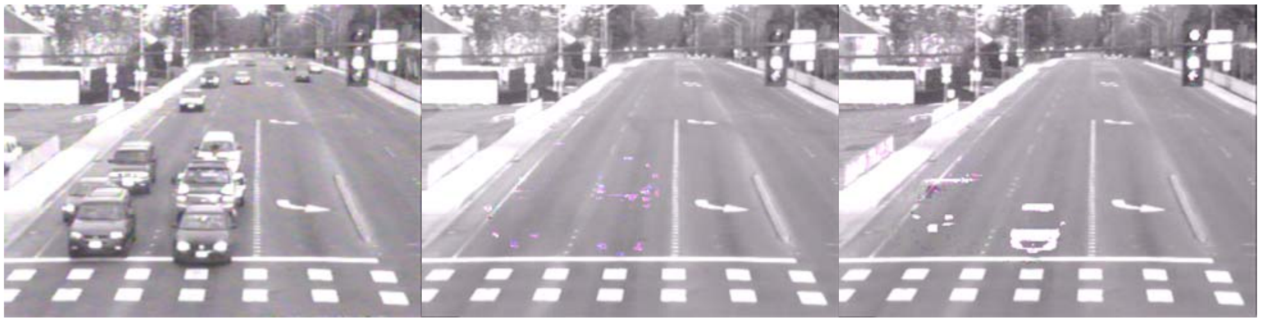


(d) Background extracted in six minutes

(e) Background extracted in 10 minutes

(f) Background extracted in 20 minutes

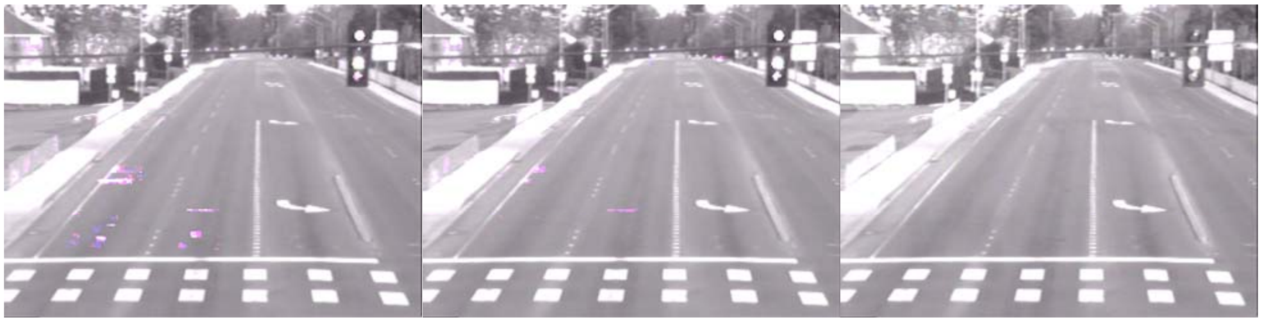
FIGURE 3 Background extraction on congested freeway.



(a) Traffic Image

(b) Background extracted in one minutes

(c) Background extracted in two minutes



(d) Background extracted in five minutes

(e) Background extracted in 10 minutes

(f) Background extracted in one minutes with green lights

FIGURE 4 Background extraction for a signalized intersection.

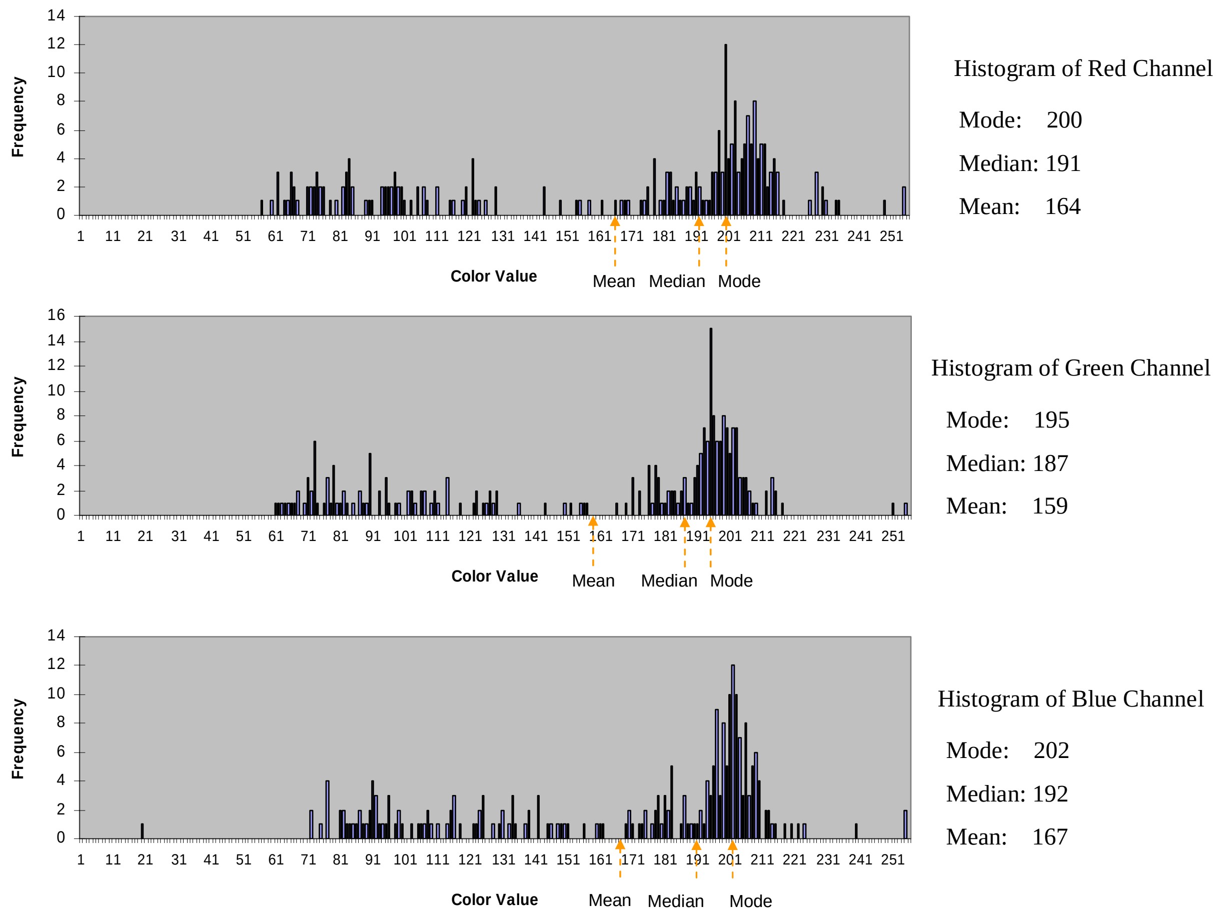


FIGURE 5 Illustration of background extraction using mode.

TABLE 1 Comparison of Computing Time

Test Location	Number of Frames Processed	Computing Time (seconds)	
		Mode Algorithm	Median Algorithm
Freeway with free flow traffic	6	2	2
	60	20	22
	600	219	284
Congested freeway	6	2	2
	60	20	21
	600	222	251
Signalized intersection	6	2	2
	60	20	22
	600	231	253