# Detecting Cycle Failures at Signalized Intersections Using Video Image Processing

Jianyang Zheng
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 (Corresponding Author)
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
Professor

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

ABSTRACT:

Signal cycle failure (or overflow) is an interrupted traffic condition in which a number of queued vehicles are unable to depart due to insufficient capacity during a signal cycle. Cycle failure detection is essential for identifying signal control problems at intersections. However, typical traffic sensors do not have the capability of capturing cycle failures. In this paper, we introduce an algorithm for traffic signal cycle failure detection using video image processing. A cycle failure for a particular movement occurs when at least one vehicle must wait through more than one red light to complete the intended movement. The proposed cycle failure algorithm was implemented using Microsoft Visual C#. The system was tested with field data at different locations and time periods. The test results show that the algorithm works favorably: the system captured all the cycle failures and generated only three false alarms, which is approximately 0.9% of the total cycles tested.

# 1 INTRODUCTION

Performance evaluation for signal control plans at intersections is an important task for many transportation applications. According to the Highway Capacity Manual 2000 (HCM) (TRB, 2000), the measures of effectiveness for this task include Level of Service (LOS, or the control delay per vehicle), queue length, and occurrence of cycle failure (or overflow). Queue length refers to the maximum number of vehicles that are queued during a signal cycle. Cycle failure happens when one or more queued vehicles are unable to depart due to insufficient capacity during a signal cycle. The HCM provides methodologies to estimate LOS and queue length using roadway geometry, traffic volumes, and signal timing data. Another approach to estimate these measures of effectiveness is through simulation models that simulate the movement of each vehicle. These models require input data similar to that used by the HCM methodologies. Obviously, both types of methodologies depend on quality inputs to estimate the measures of effectiveness. If the inputs do not fully represent the actual diversity in traffic demand, the results from these methods may not properly reflect the performance of a signal control system.

Furthermore, approaches other than the aforementioned ones are needed for evaluating the performance of modern traffic control applications, which aim to adapt signal timing plans to changing traffic volumes and conditions. Control technologies, such as the Adaptive Signal Control (ASC) and the Transit Signal Priority (TSP) systems, require evaluations under actual traffic conditions because they are specifically designed to respond varying traffic conditions.

For example, because a TSP system impacts control delays for both transit vehicles and general purpose vehicles, several traffic parameters must be measured onsite in order to evaluate the benefit of the TSP system through comparisons of the before and after scenarios. The number of cycle failures is considered one of the most useful parameters for measuring drivers' frustration toward a signal control system. Real-time cycle failure data can also be used to

improve dynamic signal control. The occurrence of signal cycle failure on a phase indicates that the flow rate exceeds the capacity of the phase. If this information is available in real time in a traffic controller, the traffic signal control system may be able to optimize signal timing to provide more green time for the over-flowed phase. Another example concerns Advance Traveler Information Systems (ATIS). A key evaluation question is whether an ATIS is correctly describing the current performance of an arterial. The performance can be measured in terms of LOS, queue length, and/or cycle failures. Being able to detect and broadcast the occurrence of cycle failures (that implies "this intersection is currently congested") will help the pubic to perceive traveler information more specifically. Therefore, detection of cycle failures in real-time is of practical significance.

In recent years, Video Image Processors (VIPs) have been increasingly employed as traffic sensors for intersection signal control. Compared to loop detectors, VIPs can provide ground-truth video images in addition to vehicle count and presence detection. The video images can be used to extract further traffic information unavailable from loops and other traditional traffic detectors. Several studies have been conducted to collect intersection traffic data using video image processing. For example, Yin et al. (2004) used virtual loops to measure traffic parameters. The position and size of each loop can be adjusted by users for collecting volume, speed, occupancy, and vehicle classification data. A video image processing system, called SPatial Image processing Traffic flow Sensor (SPITS), was developed by Higashikubo et al. (1997) to detect traffic queue lengths. SPITS measures a queue length in meters, but cannot provide the number of vehicles in a queue. Fathy and Siyal (1995, 1998) also developed image processing systems to measure volume, speed, vehicle length, and queue length. The profiles used to detect queue length were divided into sub-profiles, each with approximately the same length per vehicle, thereby making it possible to measure the number of vehicles in the queue.

Gupte et al. (2002) proposed a method to track vehicles by matching regions with vehicles in the video stream. Vehicle parameters such as location, length, and speed can be extracted from images captured by a properly calibrated camera. They also proposed to use a dynamically updated threshold to separate vehicles from the background. In a study conducted by Saito et al. (2001), average stopped vehicle delays were estimated by image analysis. The total delay was calculated by adding all the stopped vehicle delays in a sampling time interval. The average stopped vehicle delay was then estimated by dividing the total delay by the total volume. Our review did not find studies specifically focused on cycle failure detections at signalized intersections, although cycle failure may be as important as other aforementioned parameters for intersection traffic control. From a motorist's point of view, cycle failure can be easily sensed compared to average control delay or queue length.

This paper introduces a video image processing algorithm developed by the Smart Transportation Applications and Research Laboratory (STAR Lab) at the University of Washington for cycle failure detection at signalized intersections. In this paper, cycle failure is defined from a drivers' perspective, i.e., cycle failure occurs when a driver joining a specific movement queue during the green time interval is forced to wait through more than one red light to complete the intended movement via the intersection. A computer system for cycle failure detection was developed based on this algorithm. The system was tested with field data collected from VIPs deployed for signal control at two intersections in the City of Lynnwood, Washington. Test results will be presented and discussed in detail in this paper.

## 2 METHODOLOGY

A VIP deployed for signal control at an intersection is typically mounted at a fixed location above the intersection facing down toward an intersection approach. In the camera's field of

view, everything is relatively stable except moving objects such as vehicles and pedestrians. In a video image, we regard moving objects as foreground and the rest as background. When no moving objects appear in an image, the image shows the complete background scene.

In ordinary full-motion video streams, the frame rate is from 24 frames per second (fps) to 30 fps. To increase the computational efficiency, the frame rate used in the proposed algorithm is four frames per second. Due to the relatively low speed at intersections, this frame rate is sufficient to capture the continuous movements of vehicles for our analysis.

In this study, we developed a motion detection algorithm for cycle failure detection. The algorithm contains four steps: (1) background extraction from prevailing traffic images; (2) dynamic threshold determination for segmenting foreground objects from background images; (3) locating the end-of-queue with motion images; and (4) determining whether a cycle failure occurred for each lane in each cycle. Details of the four steps are described in the following sections.

**Background Extraction**

A good quality background image is essential for our video image processing algorithm. It is used to identify moving objects for each video image. Since it is generally difficult to find a frame in a video stream without any moving objects, a background extraction method is needed to compose the background image for a video scene. Avery et al. (2004) presented a background image extraction algorithm based on the changes of pixel colors from frame to frame. If the absolute differences of a pixel's R (red), G (green), and B (blue) channel values are below the corresponding thresholds for any two consecutive frames, the pixel is identified as a background pixel and the color values are recorded. The algorithm scans all the unconfirmed pixels repeatedly until all pixels on the scene are confirmed as background pixels. These pixels form

the background of a camera scene. This algorithm worked well for freeway applications under non-forced-flow conditions at a frame rate of four frames per second.

Considering that vehicles may stop or move very slowly at intersections, we need a different method for background extraction. The method employed for this study assumes that each background pixel in a video scene is clearly visible (not occluded by moving objects) for at least 50% of the time in a given period. To apply the background extraction method, a number of consecutive frames need to be selected. For each pixel, the pixel's R (red), G (green), and B (blue) channel values in these frames are sorted and the *median values* for the R, G, and B channels are selected to be the background values for the pixel. Figure 1 shows how the value for a background pixel (i, j) is determined. As the figure shows, the location (i, j) was not occupied by foreground (or moving objects) in most of the frames captured over a period of time. In this case, the majority of the frames have the same R, G, and B channel values at pixel (i, j) when it is not occluded by any moving objects. The median of these values for pixel (i, j) will be the value of the background at this position. By repeating the same process for each pixel in a video scene, a background image of the scene can be composed. The efficiency and accuracy of this background extraction method is sensitive to the number of frames applied to the process. Figure 2 shows some examples of the background images obtained by this method. For the field data used in this study, image sets with 60 to 90 frames worked reasonably well. The reader may have noticed that the background image extracted from 30 frames does not look right, especially in the zone indicated by the arrow. This was due to the fact that, in the 30-frames data set, some background pixels were occluded by moving objects in more than 15 frames and, therefore, the median R, G, and B channel values selected did not represent the true values for the background pixels. Increasing the number of frames for analysis will reduce the probability of violating the method assumption, but will also reduce the method's computational efficiency. We recommend

using 60 to 90 frames for the median background extraction algorithm. Comparison of the background images extracted from using 60 frames and 90 frames did not find major differences. This indicates that, when the number of frames is large enough, further increasing the number of frames may not result in any significant improvements to the quality of the extracted background image. The appropriate number of frames for background extraction can also be determined by users, based on considerations of both efficiency and accuracy for specific applications. It may also be helpful if users manually find a period of video stream that maximally satisfies the background pixel assumption. All the example backgrounds were extracted when vehicles were moving most of the time. This avoids assigning vehicle pixels to the background images. However, if a user chooses only the red light time for background extraction, some stopped vehicles may become part of the background. The background extraction does not need to be performed frequently. Users can choose non-peak hours to extract the background to ensure a good quality background image.

The extracted background must be updated dynamically to adapt to changes of light and other environmental conditions in the scene. In this study, the method developed by Gupte et al. (2002) is applied to update the background. This method first finds the instantaneous background, which is defined as the background of the current frame. For pixels covered by the foreground, their color values are set to the same values as those in the background image. The background image is updated by the weighted summation of the instantaneous background and the original background. In our tests, the weighting factors used were 0.1 for the instantaneous background pixels and 0.9 for the original background pixels.

The grayscale value of each pixel in the frames and the background image were calculated from the R, G, and B channel values using a standard conversion formula: grayscale = 0.30*R + 0.59*G + 0.11*B (Steinmetz and Nahrstedt, 1995). Because the algorithms employed

for the following analyses worked more efficiently with grayscale values, only grayscale images were used.

**Dynamic Threshold**

The extracted background image is essential for vehicle detection. Each frame is compared with the background image. Non-background pixels are considered to be part of the moving objects such as an automobile, motor cycle, bicycle, or pedestrian. To determine whether a pixel is a background pixel, we need to establish a threshold. For a given pixel, if its value difference from the background value is larger than the threshold, it will be determined as a non-background pixel. Otherwise, it is confirmed as a background pixel. The threshold may be a constant value chosen by trial and error or by experience. However, with a dynamically updated background, a fixed threshold for each frame may not yield satisfying results. Therefore, a dynamic threshold was employed for this study.

A dynamic threshold value is calculated for each frame using a *difference image* that represents the differences between the background image and the current image. Each pixel value on the difference image is the absolute difference of pixel values between the current frame and the background. Therefore, the foreground pixels in the difference image will have higher values, and the background pixels will have lower values. Figure 3 (b) shows a difference image generated from the current image shown in Figure 3 (a). The grayscale histogram of all pixels in a typical difference image is shown in Figure 3 (c). Since most of the pixels in the difference image are part of the background, the histogram has a higher peak on the left. The other peak on the right of the histogram represents the grayscale distribution for foreground pixels. The threshold will probably be located at the bottom of the valley between the two peaks. Gupte et al. (2002) assumed that the threshold should be at the first point from the left with a

significant pixel value difference (90% of the peak value, for example) corresponding to that of the left peak. However, since the histogram of the difference image is bimodally distributed, we prefer using Otsu's method (Otsu, 2001) to find the threshold between the two peaks. Otsu's method has been widely adopted for threshold searching in image processing studies. The threshold determined by Otsu's method minimizes the intra-group variance and works well for many applications.

With a specifically identified threshold, the difference image can be transferred into a binary image, where the pixel value "0" means background and "1" means foreground. The following analysis is based on this binary image. Figure 3 (d) shows the binary image generated from the current image shown in Figure 3 (a).

**The End-of-Queue**

In order to detect cycle failure, one must keep tracking the end of queue when the traffic signal turns green. If the vehicle at the end of a queue clears the intersection, there is no cycle failure; otherwise, a cycle failure is recorded. In this program, the user needs to interactively input the position of the stop line and the longitudinal line of each lane. To determine the end of a queue, *motion images* were used. A motion image is the absolute difference of two contiguous frames in a video stream and shows only the moving objects between the two frames. A fully stopped vehicle in a queue cannot be reflected in the motion image, but can be shown in the difference image. By comparing a motion image with the corresponding difference image, a stopped vehicle can be confirmed. Figure 4 shows the flow chart for identifying stopped vehicles and finding the queue end for a lane. Since there is noise in both the motion image and difference image, the determined queue end sometimes changes noticeably, although not because of an approaching vehicle. To make the result more stable, a median filter was used for the position of

the queue end in the time domain. The positions of the queue end in the last several frames were recorded and their median was used as the queue end of the current frame. In this study, we chose to use seven frames for calculating the median. This number of frames produced a smooth performance of queue-end detection and a quick reaction to the change of queue length. Figure 5 shows a snapshot of the user interface with the identified end-of-queues. In this picture, the image box on the right shows the motion image. The reader can see that all stopped vehicles in the two left lanes are not shown in the motion image, but vehicles moving under the left-turn green signal in the third lane from the left are visible. The short horizontal bars at the end of the longitudinal lines were automatically drawn by the software to mark the detected positions of the end-of-queues.

Signal data at an intersection can be obtained from the signal controller when the system runs online. At the current offline development and test stage, however, signal controller data are unavailable. The status of signal lights is estimated from the movements of vehicles in the video stream. To detect vehicle movements, a virtual loop is created near the stop line for each lane as shown in Figure 5. We call these motion loops. Once the stopped vehicle closest to the stop line is detected as in-motion, it is assumed that the signal light for the lane has just turned green. Similarly, if the vehicle is detected stopping, the corresponding signal light is considered turning red.

When a long and tall vehicle such as a bus runs through the intersection from the cross streets, it may be projected to the virtual motion loops and trigger false alarms. To avoid this problem, we placed another virtual loop, also shown in Figure 5, in the core area of the intersection to detect long vehicles from cross streets. We call this a conflict detection loop. If this loop is occupied, all the motion loops are disabled because signals at conflict approaches

cannot be green simultaneously. Compared with the signal data directly from the signal controller, the signal timing estimated by this algorithm is less accurate.

### Determine the Cycle Failure

Once the signal light is determined to be changing from red to green, the current position of the queue end for a lane is stored. The queue length is expected to decrease under the green signal and the end-of-queue for each lane is tracked along the longitudinal line and updated from frame to frame. Figure 6 shows the flow chart for cycle failure detection for one lane in one frame of the video stream. The end-of-queue position is updated in the current frame and stored for usage in the next frame. If the end-of-queue reaches the stop line before the signal turns back to red, all vehicles accumulated in the queue during the previous cycle have been fully discharged and no cycle failure will occur during this signal cycle. If, when the signal light turns back to red, the queue end is still behind the stop line, the accumulated queue have not been fully cleared during the green interval and a cycle failure is detected for the lane during this signal cycle. During the red signal time, a queue length grows and the system sets the end of queue to the last vehicle in the queue and updates its position with new arrivals. At the end of each signal cycle, detection results are preserved in a data file.

## 3 SYSTEM TEST AND DISCUSSION

A video image processing-based cycle failure detection system implementing the proposed algorithm was developed using Microsoft Visual C#. Video data recorded by Trafficon VIPs were used to test the system. The VIPs are currently used as traffic detectors for signal control at two intersections in Lynnwood, Washington. At the intersection of SR-99 and 196$^{th}$ Street SW,

the cameras are mounted approximately 8.5 meters above the ground and aligned approximately 30 degrees below horizontal. The VIPs were configured for general-purpose detection rather than cycle failure detection. Based on the researchers' visual count, the maximum queue length visible in the VIPs' field of view was 18 vehicles under this configuration. Since the maximal number of queued vehicles visible in a camera's field of view is determined by the camera's lens, height, and posture, a VIP may be reconfigured to accommodate longer vehicle queues when there is a need to do so. At the intersection of 164$^{th}$ Street SW and 36$^{th}$ Ave W, the cameras' installation height and pointing angle are slightly different. The sample images captured from cameras at these two intersections are shown in Figure 7.

For the intersection of SR-99 and 196$^{th}$ Street SW, the test video data sets were recorded during the afternoon peak period on June 23, 2004 (Wednesday). The eastbound approach and the southbound approach at this intersection were selected for the test. Each approach has three lanes including one left-turn lane. Approximately 50 minutes of video data for each approach were tested. Since SR-99 is one of the most important and busiest highways in the Greater Seattle area and 196th Street SW is also a busy local arterial, the selected intersection is oversaturated during peak hours, especially for the left-turn movements. It was obvious that cycle failures occurred frequently at left-turn lanes during the test period. For the intersection of 164$^{th}$ Street SW and 36$^{th}$ Ave W, test video data sets were recorded during morning peak hours on April 2, 2005 (Wednesday). The eastbound approach and the westbound approach were selected for this test. The eastbound approach has two through lanes, one left-turn lane, and one right-turn lane (the traffic on the right-turn lane is not closely related to this research and was not studied). The westbound approach has two through lanes and one left-turn lane. Approximately 50 minutes of video data for each approach were tested in this study.

A total of approximately 200 minutes of video data were tested and the test results are encouraging. Of the 106 signal control cycles tested, 318 lane-based test results were recorded. All the test results were manually checked. These test results are summarized in Table 1.

For the intersection of SR-99 and 196$^{th}$ Street SW, there were 12 cycle failures occurred on the southbound approach during the test period. All of these cycle failures were on the left-turn lane. The left-turn signal had a 24.5-second protected green interval and a 3.5-second yellow interval. Through-lane traffic had a combined time of approximately 60 seconds for green and yellow intervals. The cycle failure detection system worked favorably for the southbound approach: it captured all 12 cycle failures and made no false dismissal (a false dismissal means a "yes" event is overlooked) or false alarm (a false alarm refers to the mistake of recording a "no" event as "yes") mistakes. The eastbound approach experienced eight cycle failures during the same period: seven on the left-turn lane and one on the through lane. The left-turn movement at this approach was protected by a 16.5-second green interval and a 3.5-second yellow interval. The combined green and yellow time for through-lane traffic was approximately 40 seconds. All eight cycle failures at this approach were successfully detected, including the only one on a through lane. However, the system generated two false alarms, one on the left lane and the other on a through lane.

For the intersection of 164$^{th}$ Street SW and 36$^{th}$ Ave W, the traffic volume was lower and fewer cycle failures occurred. At the eastbound approach, the green interval was 90 seconds and the yellow interval was 3.5 seconds. No cycle failure occurred on this approach during the test period. However, the system generated a false alarm on a through lane. At the westbound approach, the green and yellow interval lengths are exactly the same to the eastbound approach. One cycle failure occurred on the through lane during the test period. The system successfully

detected this cycle failure and did not generate any false alarms or false dismissals for this approach.

The three false alarms were further examined. Two false alarms were caused by the failure of the conflict detection loop. Since the conflict detection loop did not capture the long vehicle that triggered the motion loops, the red signal light was considered to be green and consequently resulted in a false alarm. Such a mistake can be eliminated when the system is operated online and signal control data are obtained from controllers rather than through estimation. The other false alarm occurred when the volume was very close to the critical volume for a cycle failure. Actually the last vehicle in the queue had just passed the intersection when the signal turned red, but the system mistakenly captured the vehicle behind it as the last vehicle in the queue. If there had been one more vehicle in the queue, this could have been a correct detection. The chances of such false alarms occurring can be lowered by using higher frame rates and real-time signal control information in that vehicle-queue status can be evaluated in a more accurate and timely manner.

In summary, all 21 cycle failures from the 318 test cycles were successfully captured by the system. No false dismissals occurred, but there were three false alarms, which is approximately 0.9% of the total cycles tested. The overall detection accuracy was 99.1%. However, all test data were collected in daytime under sunny conditions. Like most video detection algorithms, the performance of this algorithm is expected to degrade when applied to bad weather or lighting conditions, such as rain, fog, snow, nighttime, etc. Further research will be conducted to address the impacts from these unfavorable conditions.

## 4 CONCLUSIONS

In this study, an algorithm was proposed to detect cycle failures at signalized intersections using video data from traffic surveillance cameras. The algorithm includes four steps: (1) extracting background images from the prevailing traffic images by median algorithm; (2) determining dynamic threshold for segmenting foreground objects from background images; (3) locating the end-of-queue with the motion image; and (4) determining whether a cycle failure occurred in each lane in each cycle. This algorithm was implemented in the cycle failure detection system using Microsoft Visual C#. This system was tested with four sets of video data captured at two intersections.

The test results showed that the proposed algorithm for cycle failure detection is encouraging. During the nearly 200 minutes of test periods, the cycle failure detection system captured all 21 cycle failures, and detection accuracy was approximately 99.1%; the system generated only three false alarms, which is approximately 0.9% of the total cycles tested. This accuracy will probably be sufficient for most practical applications. Two of the three false alarms came from mistakes made by the system in estimating the status of signal lights. If the program can take signal status data directly from the signal controller, the accuracy level can be further improved. The algorithm, which extracts stopped vehicles from the video stream, tracks the end-of-queue, and determines if a cycle failure is occurring, has proven to be effective for the test locations over the test periods. The cycle failure detection system has the potential to provide reliable real-time cycle failure information to many traffic management and operation applications.

In the next phase of this research, we will focus on testing the algorithm under a variety of weather and lighting conditions to improve its robustness for future deployment. The future

cycle failure detection system will be operated online for real-time detection rather than processing archived video images.

## ACKNOWLEDGEMENTS

This research was funded by the Washington State Department of Transportation (WSDOT), Community Transit, and Transportation Northwest (TransNow), the US Department of Transportation University Center for Federal Region 10, at the University of Washington. Special thanks to Mr. Dick Adams and Mr. Paul Coffelt at the City of Lynnwood Transportation Division for providing the traffic video data for this study. The discussion with Patikhom Cheevarunothai and Sean Hoyt generated useful ideas for background extraction. All their help is greatly appreciated.

## REFERENCES:

Avery, Ryan P., Chris P. Thomas, Yinhai Wang, and G. Scott Rutherford. (2004), Development of a Length-based Vehicle Classification System Using Uncalibrated Video Cameras. *The 83rd Annual Meeting of the Transportation Research Board* (CD-ROM), National Research Council, Washington, D.C.

Fathy, M. and M. Y. Siyal. (1995), Real-time Image Processing Approach to Measure Traffic Queue Parameters. *IEE proceedings: Vision, image, and signal processing, Vol. 142, No. 5.*

Fathy, M. and M. Y. Siyal. (1998), A Window-Based Image Processing Technique for Quantitative and Qualitative Analysis of Road Traffic Parameters. *IEEE Transactions on Vehicular Technology, Vol. 47, No. 4*.

Gupte, Surendra, Osama Masoud, Robert F. K. Martin, and Nikolaos P. Papanikolopoulos. (2002), Detection and Classification of Vehicles. *IEEE Transactions on Intelligent Transportation Systems, Vol. 3, No. 1*.

Higashikubo, Masakatsu, Toshio Hinenoya, and Kouhei Takeuchi. (1997), Traffic Queue Length Measurement Using an Image Processing Sensor. *Sumitomo Electric Technical Review, No. 43, pp. 64-68*.

Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62-66.*

Saito, Mitsuru, Jaylen Walker, and Alan Zundel. (2001), Using Image Analysis to Estimate Average Stopped Delays per Vehicle at Signalized Intersections. *The 80th Annual Meeting of the Transportation Research Board (CD-ROM), National Research Council, Washington, D.C*.

Steinmetz, R. and Klara Nahrstedt. (1995), Multimedia: Computing, Communications & Applications. Prentice Hall.

TRB (Transportation Research Board). (2000), *Highway Capacity Manual*. TRB, National Research Council, Washington, D.C.

Yin, Zhaozheng, Fan Yang, Henry X. Liu, and Bin Ran. (2004), Using Image Sensors to Measure Real-time Traffic Flow Parameters. *The 83rd Annual Meeting of the Transportation Research Board (CD-ROM), National Research Council, Washington, D.C.*

## LIST OF FIGURES



## LIST OF TABLES

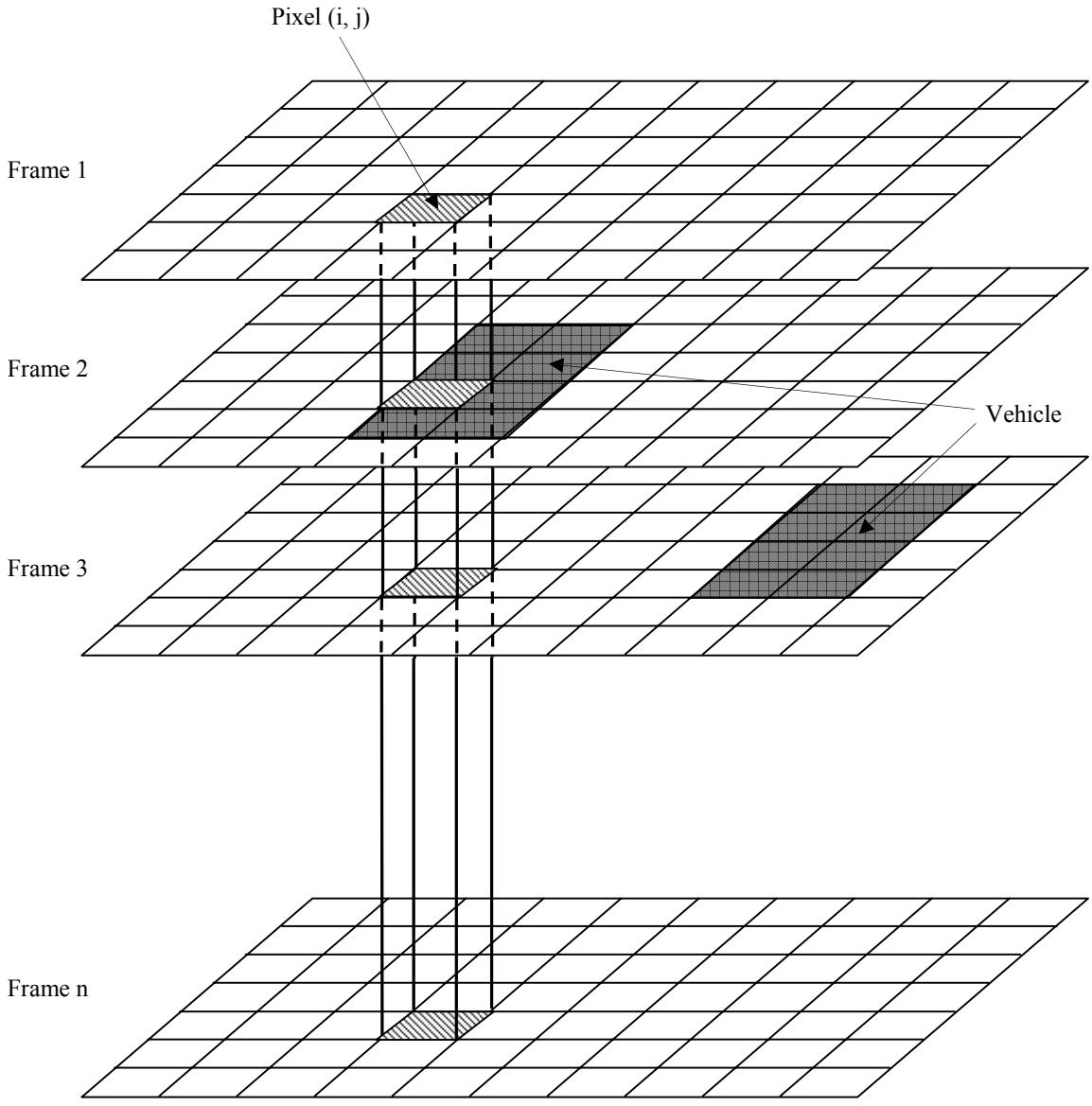
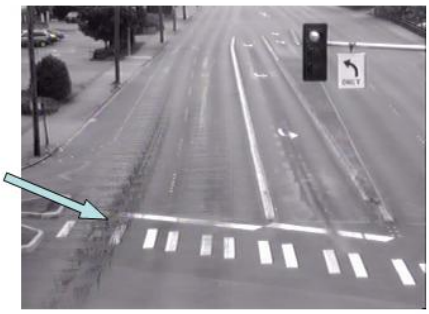



**FIGURE 1 Algorithm of background extraction**

(a) An Image from the Video Stream



(b) Extracted Background from 30 Frames

(c) Extracted Background from 60 Frames

(d) Extracted Background from 90 Frames

**FIGURE 2 Examples of extracted background**

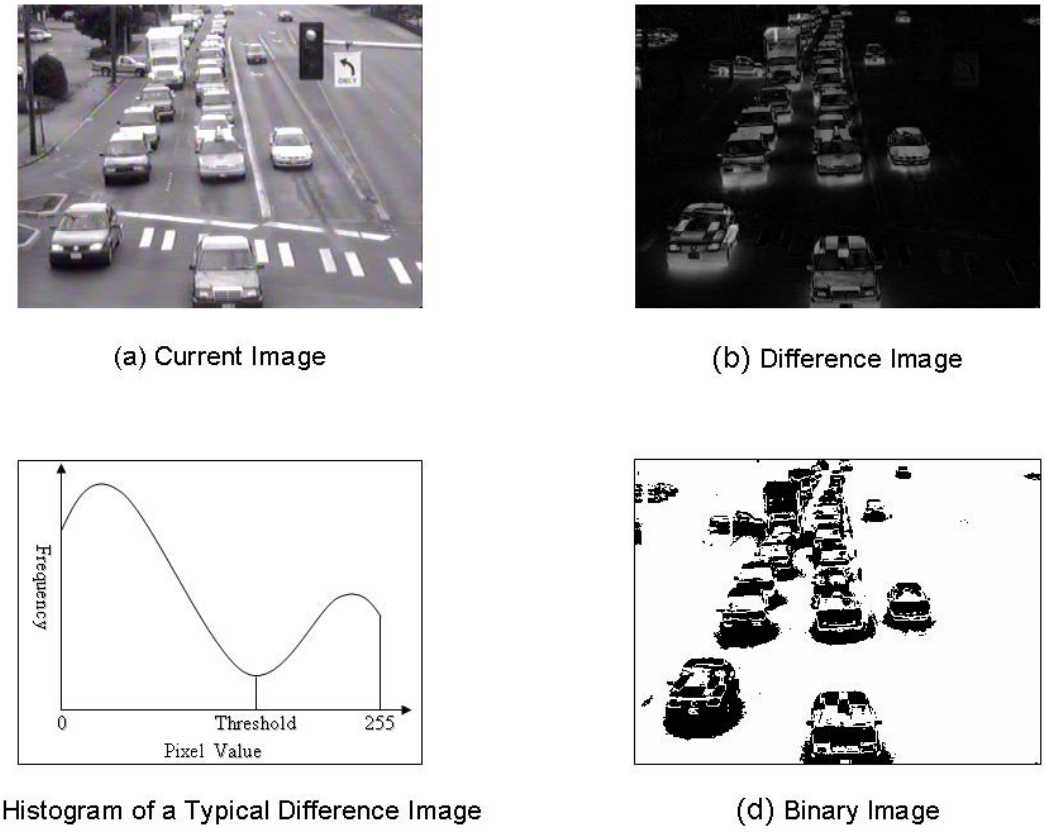



**FIGURE 3 Dynamic threshold**

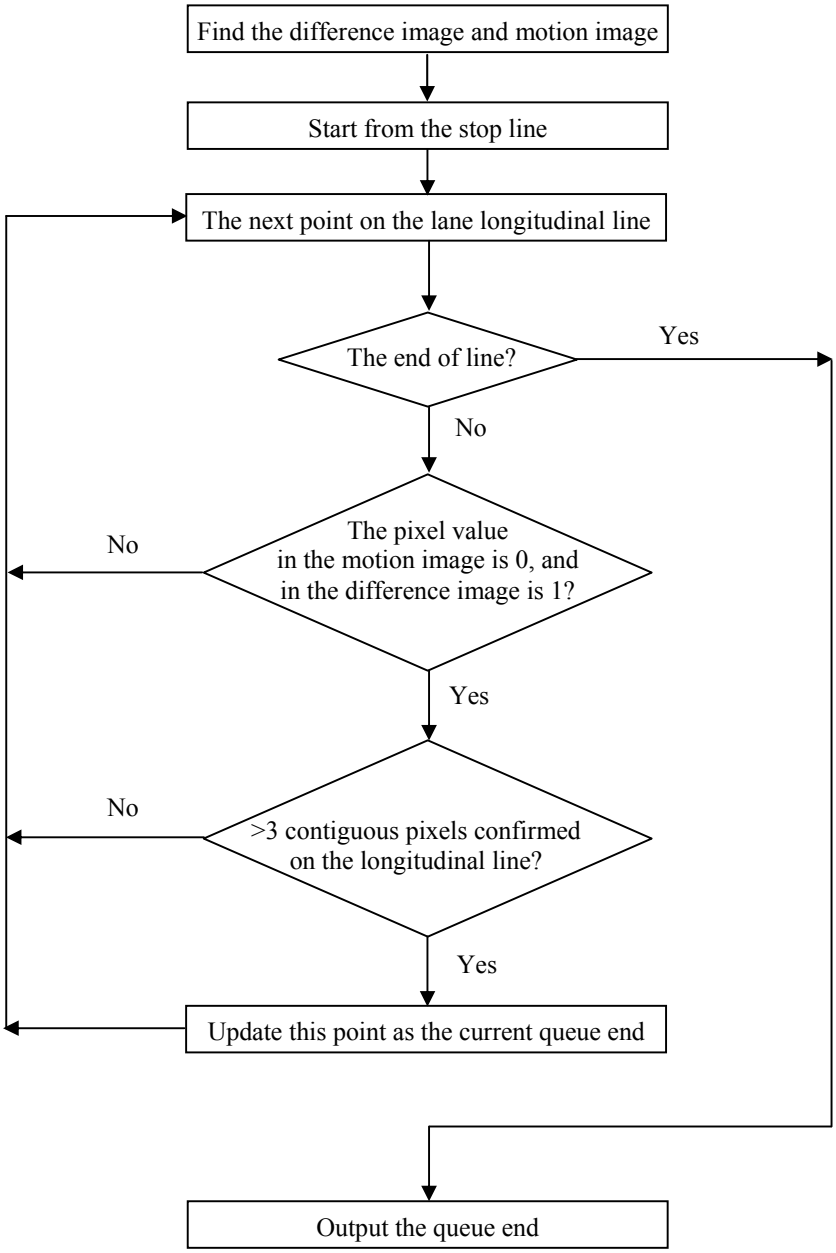



**FIGURE 4 Flow chart of the determination of the queue end**

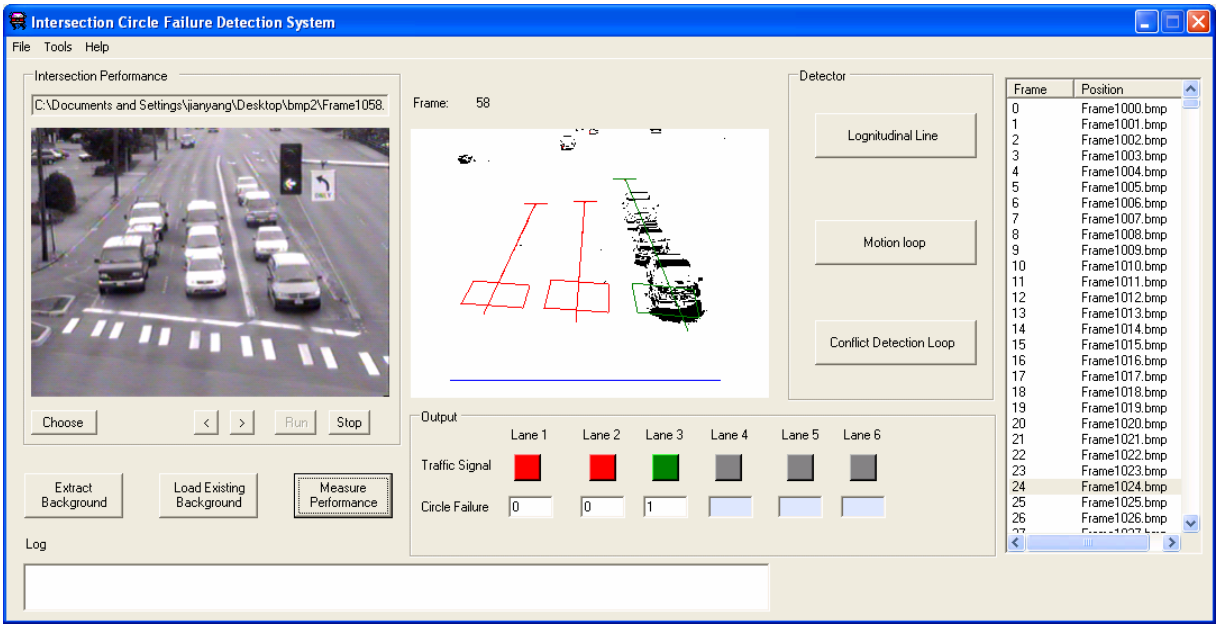



**FIGURE 5 A snapshot of the user interface**

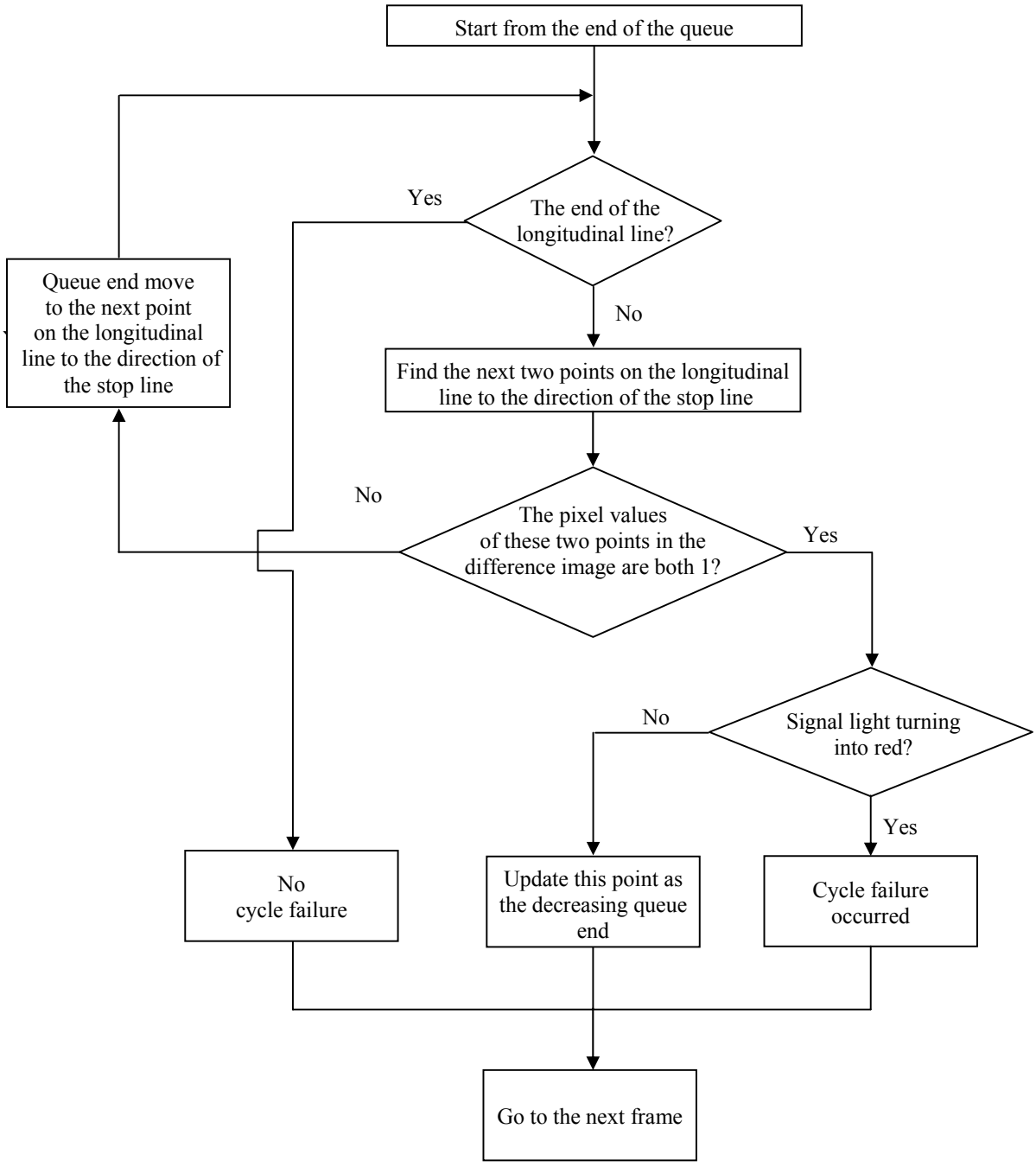



**FIGURE 6 Flow chart of the determination of cycle failure**

SR-99 and 196th Street, Eastbound


SR-99 and 196th Street, Southbound

164th Street and 36th Ave, Eastbound

164th Street and 36th Ave, Westbound

**FIGURE 7 Sample images captured at test intersections**

**TABLE 1 Test Results**

| Test Location | | Number of Test Cycles | Cycle Failures Occurred | Number of Correct Detections | Cycle Failures Detected | False Dismissals / Alarms |
|---|---|---|---|---|---|---|
| SR-99 and 196th ST, Eastbound | Lane 1 (through) | 21 | 1 | 21 | 1 | 0 / 0 |
| | Lane 2 (through) | 21 | 0 | 20 | 1 | 0 / 1 |
| | Lane 3 (left turn) | 21 | 7 | 20 | 8 | 0 / 1 |
| SR-99 and 196th ST, Southbound | Lane 1 (through) | 20 | 0 | 20 | 0 | 0 / 0 |
| | Lane 2 (through) | 20 | 0 | 20 | 0 | 0 / 0 |
| | Lane 3 (left turn) | 20 | 12 | 20 | 12 | 0 / 0 |
| 164th ST and 36th Ave, Eastbound | Lane 1 (through) | 33 | 0 | 33 | 0 | 0 / 0 |
| | Lane 2 (through) | 33 | 0 | 32 | 1 | 0 / 1 |
| | Lane 3 (left turn) | 33 | 0 | 33 | 0 | 0 / 0 |
| 164th ST and 36th Ave, Westbound | Lane 1 (through) | 32 | 1 | 32 | 1 | 0 / 0 |
| | Lane 2 (through) | 32 | 0 | 32 | 0 | 0 / 0 |
| | Lane 3 (left turn) | 32 | 0 | 32 | 0 | 0 / 0 |
| Summation | | 318 | 21 | 315 | 24 | 0 / 3 |