5-9-2003

Vehicle detection and tracking by computer vision for intelligent transportation applications.

Peijun Shi

VEHICLE DETECTION AND TRACKING BY COMPUTER VISION FOR INTELLIGENT TRANSPORTATION APPLICATIONS

A Thesis

Presented to the

Department of Computer Science

And the

Faculty of the Graduate College

University of Nebraska

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

University of Nebraska at Omaha

By

Peijun Shi

May 9th, 2003

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

THESIS ACCEPTANCE

Acceptance for the faculty of the Graduate College, University of Nebraska, in partial fulfillment of the requirements for the degree (name the degree), University of Nebraska at Omaha.

Committee

Chairperson

Date 4/23/2013

Vehicle Detection and Tracking By Computer Vision for Intelligent Transportation Applications

Peijun Shi, M.S

University of Nebraska, 2003

Advisor: Dr. Qiuming Zhu

The research of this thesis is generally divided into two phases: the first phase deals with background image generation and vehicle detection, the second phase deals with vehicle tracking and video handoff. We propose median model in this thesis for background generation in vehicle detection. This median model is adaptive to lighting change and weather condition change in that it only samples a small size of recent images for background generation. An improved median model is also developed to handle slow moving vehicles and stationary. A robotic detector (without human supervision) is proposed by assembling a series of image process operators together and a satisfactory result is obtained. In the second phase, the weighted intensity information and shape information for each vehicle is scored and minimum-distance classification method is used for vehicle match. More than 400 vehicles are tested. An overall detection rate of 97.6% and tracking rate of 74.45% are obtained in this system.

Keywords:

Background subtraction, vehicle detection, vehicle tracking, object classification

Acknowledgments

I would like to thank my supervisor, Dr.Qiuming Zhu. His expertise in many fields such as image processing and artificial intelligent guide me through many difficulties. I appreciate not only his knowledge and skill, but also his understanding and patience. Without his inspiration and direction, this work is impossible.

I would like to thank Dr. Elizabeth G. Jones. I appreciate her help during my one-year's experience working in her I^3 lab. Dr. Jones gave me some precious advice on my thesis writing and computational complexity analysis. The very good facilities in her I^3 lab have been a helpful resource for this thesis.

Special thanks go to my family and friends. My mother Miaozhen Liao, my wife Yufang Zhang and my daughter Jenny, they are my moral support. My parents-in-law, Xuewen Han and Defu Zhang have been helping us take care of Jenny for 3 years. My friend Peter helped me finishing my thesis draft. I'm grateful for their support and encouragement.

I also would like to thank National Science Foundation for their funding on this thesis.

Contents

Figures

1. Thesis background and introduction

This research project provides computer vision support for vehicle detection and tracking in a framework of Information Management Support (IMS) for Intelligent Transportation Systems (ITS). All required research facilities for ITS is located at the Intelligent Transportation Systems Information and Infrastructure Laboratory (I^3 Lab) in the Peter Kiewit Institute, University of Nebraska at Omaha, which has the capability to act as a Traffic Management Center (TMC). Two intersections (67^{th} ST and Pacific ST, 69^{th} ST and Pacific) are signalized with video camera detection. Video detection is also available at two mid-block locations between these two intersections. The real-time traffic video and data streams are delivered through fiber optic cable to the I^3 lab.

1.1 Traffic detectors

One of the important goals of ITS is to reduce traffic congestion [1]. To fulfill this goal, vehicle detection and accident detection techniques are necessary. The current techniques for vehicle detection are [2]:

1) Inductive loop detectors (sensors).
2) Video Detection and Closed-Circuit Television (CCTV).
3) Microwave detector.
4) Ultrasonic detector.
5) Magnetic detector.
6) Infrared sensors.

Inductive loop and CCTV are the predominant surveillance technologies in use today. Inductive loop detector includes an insulated wire loop buried in the roadway, a controller and an electronic unit. The inductance of the wire loop changes when a vehicle passes the wire loop. This inductance change stimulates the electronic unit to send a pulse to the controller, indicating a detection of a vehicle. Inductive loop is usually installed at 0.5mile intervals on roadways or on approaches to signalized intersections in existing urban traffic systems. Each detector is controlled by a local processor or a distributed processor to provide an interface to TMC[3].

CCTV detectors are used in both intersection traffic detection and freeway surveillance systems. They are typically installed at 1-3 mile intervals on freeways and on approaches to signalized intersections. Video images are transmitted to the TMC, displayed on video monitors and further processed on PC for detection and tracking. CCTV usually involves two parts: an above-the-road camera and a video image processor (VIP). VIP provides software and hardware support for image processing and provides an interface to TMC. The real-time traffic data streams are digitalized and analyzed in VIP. The VIP can be located in the camera assembly or in a location separate from the camera. Recently, more intelligent software was introduced in VIP, so the VIP can automatically extract the traffic features and communicate directly with traffic control devices. VIP system is not only cost-effective in term of system maintenance, but also outputs a much larger set of traffic parameters such as vehicle detection, classification, tracking, count, speed measurement, lane specific

information and driving pattern as well as live image[4].

There are also some other detectors currently available. They have their own advantages and disadvantages. The ultrasonic detector is easy to install and use but may be sensitive to temperature and air turbulence. The microwave detector performs well in inclement weather and could directly measure vehicle speed, but can not detect stopped vehicles or vehicles moving less than approximately 5 mph. Active (transmit and receive) infrared detector can directly measure vehicle speed but has potential degradation by obscurants in the atmosphere and by inclement weather.

1.2 Traffic analysis and traffic control

Traditional traffic control focuses on reducing traffic congestion and detecting accidents. For traffic congestion control, some of the current ITS work effectively. But for accident detection, most current systems fail at to perform properly. For example;

(1) Only a small fraction of accidents are detected by using the current automated detection systems. Most accidents are declared to the TMC by passing motorists, traffic advisory aircraft, or highway patrol.

(2) Current detection system depends highly on experience and personal judgment of the TMC operator. This is labor intensive and not efficient for handling the dramatically increasing traffic situations [5].

Our Information Management Support (IMS) has the potential to significantly improve the performance of the current TMC in traffic control in the ways such that [6]:

(1) The IMS makes intelligent decisions. There are some traditional vehicle detectors

such as ultrasonic detectors, sound detectors, inductive loops and pneumatic sensors. These detectors are used for vehicle detection and vehicle number counting. They can only make a binary decision: if there is a vehicle on a certain spot. They are not intelligent enough to distinguish individual vehicles according to their specific image information. For more intelligent transportation system, such as vehicle tracking, traffic accident analysis and driving pattern analysis, computer vision technique is imperative. (2) In addition to the accident detection routine, our IMS also tries to target dangerous driving patterns and catch dangerous drivers, which could prevent a traffic accident before it takes place.

1.3 Thesis objective

A general IMS for Intelligent Transportation System involves several different scientific disciplines such as transportation engineering, computer science and systems engineering. It includes different kinds of tasks such as traffic data collection and pre-processing, object detection and video handoff, and traffic pattern analysis. The objectives of my research in the IMS development are:

(1) Video handoff and vehicle tracking.

(2) Traffic flow information extraction.

(3) Driving pattern analysis.

To achieve these goals, the research is divided into three phases:

(1) the image acquisition and pre-processing phase,

(2) the vehicle tracking and hand-off processing phase, and

(3) the traffic video understanding (driving pattern recognition) phase [6].

The first task phase involves hardware installation, data transmission, data storage and data pre-processing. It is known that video images are 2D projections of 3D objects. When projecting from 3D to 2D in an imaging process, the information of one of the dimensions is lost. Such loss inevitably causes some perception errors. Another issue in tasks of this phase is the handling of signal noises. When data is transmitted via cable, considerable noise is added. The first task phase needs to take care of correcting these errors and filtering the signal noises [6].

The second task phase (ie, vehicle identification and video handoff) is more specifically related to providing a computer vision support for the transportation system so that the system could interact with real time traffic and make intelligent decisions based on the traffic pattern analysis results [6].

After noise-filtered and quality-enhanced image frames are obtained from the first phase, the second phase process attempts to identify the individual vehicles from the consecutive views, and mark the individual vehicle with a unique and traceable label. The majority of the thesis deals with these issues. In our experimental setting the camera locations are fixed. Thus the field of view for each camera remains constant. This setting allows us to use "background-subtraction" method for vehicle identification. The tasks of this phase can be further divided into following five steps:

(1) Generate the background image.

(2) Subtract the background from target image.

(3) Remove noises that come from the previous step.

(4) Connect the pixels of individual vehicle into one object and label each vehicle respectively.

(5) Video handoff by using object-match technique.

Major challenges for these tasks are:

(1) Environment factors such as weather condition and illumination condition change by the time of day or night. The vehicle detecting and tracking system must be adaptive to such condition changes and develop a set of effective image processing algorithms to handle these situations.

(2) Images delivered to TMC are real time so the algorithms must be capable of producing real-time outputs, and pass the outputs to the next phase promptly. This needs to be done to allow the IMS to make real-time intelligent decisions for TMC.

Figure 1 shows a flow chart of tasks to be performed:

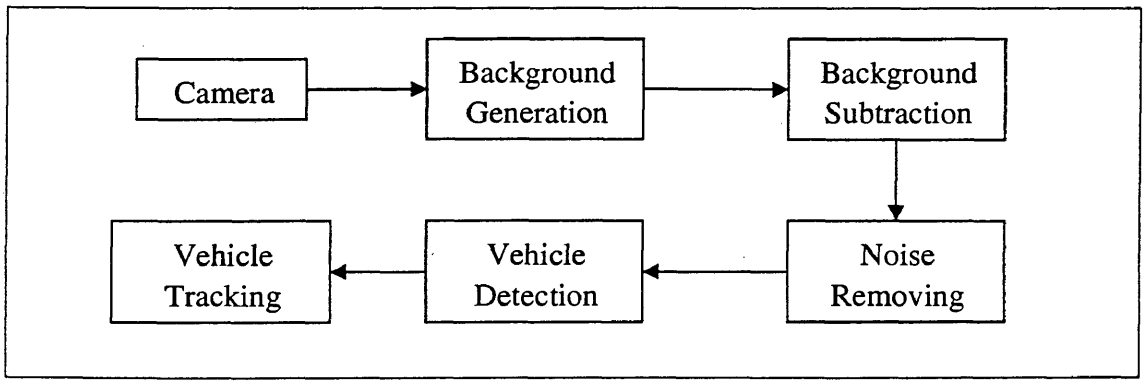

Figure 1: Flow chart of task phases of the project

2. Review of Previous Related Research

2.1 Summary of previous works on background subtraction

Object detecting using computer vision technique in ITS can be divided into two main categories: (1) Motion analysis (such as optical flow) concentrating on the analysis of moving objects, and (2) background subtraction concentrating on stationary background generation.

Motion analysis uses the frame-difference approach to extract the motion features from image sequences. It includes three different methods [7]:

- Difference with background
- Two frame difference
- Three frame difference

The difference image D_m is defined as:

$$D_m(i,j) = | I_m(i,j) - I_{m-1}(i,j) |$$

The basic idea is to extract motion features by using the difference image:

- Stationary region: $S_n = \{P(i,j) \in I_n \mid D_n(i,j) = 0\}$
- Moving region: $M_n = \{P(i,j) \in I_n \mid D_n(i,j) \neq 0\}$

The motion analysis technique requires various parameters and is expensive in terms of computational time compared with the background subtraction approach [8]. Background subtraction is straightforward and fast, which is important for real time traffic analysis. For motion analysis method, static background information is of little use. This is a waste of information since background can provide information about

illumination changes as well as stable information of the scene. Moving object detection is appropriate if we only interested in moving objects and want to detect, track and extract specific features from these moving objects. Static background, however is another way which can make things easy in ITS. Once a background reference image is built, the moving objects could be easily separated from images by simply making a comparison between the current image and the background reference image.

The basic idea to build the background reference image is quite simple: sample the multiple intensity values on each pixel by time, and find the most stable values over images. Based on this idea, several different strategies are proposed such as time average, exponential forgetting, classified updating and mixture of the Gaussian model.

Background subtraction has always been an interesting field in computer vision. Since video camera is been widely used in traffic control, shopping mall, stadium, security building and anywhere its usage is warranted, the potential market is understandable.

2.2 Time average technique

Time average or sequence average is a method that was proposed several decades ago. It uses a long-term time average of an image sequence to produce the reference background image. This strategy assumes that if we expose our vision system to a background much longer time than to moving objects, then what we get is

roughly a background image. The background will eventually dominate the sample values and the average value will give us a good estimation of the background image:

$$B(x, y, t) = \frac{1}{t}\sum_{t'=1}^{t} I(x, y, t')$$

The underlying principle is still being used today with a model of Gaussian distribution. In Gaussian distribution, the mean value of sampled values has highest probability density and could be used to represent the background value. The shortcoming of time average technique is that: the pixel is not classified, and therefore all information (including non-background noises) is used for background generation. Noise may be significant when objects are slowly moving or occupying a large portion of the image region. Another disadvantage of this technique is that each image has the same weight in background generation, so the effect of illumination change is a problem for this strategy [9].

2.3 Edge detection method

It is observed that when a pixel represents a background, its intensity value tends to be relatively more stable than the non-background pixel. When a pixel is related to a moving object, its intensity value randomly changes in a larger range. Therefore if we record and analyze the pixel intensity values over time, pixels represent the background and the moving object can be distinguished.

The boundary between the relatively stable pixel regions and the significant changing regions can be detected by finding some sharp intensity changing image

features called edges. The average value of pixel intensity for pixels not near the edge in a time sequence can be calculated and used for background estimation.

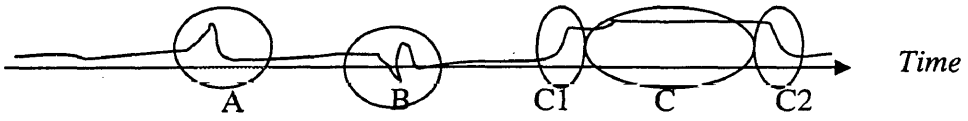

Figure 2: Edge detect background

As shown in Figure 2 above, values in oval shape "A", "B", "C1" and "C2" are near edge values and will be deducted from background calculation.

Edge information could be used to improve the general time average method. In this method some possible vehicle pixel values are deducted, and only possible background pixel values are used for background calculation. The idea is generally correct but it still has a problem: when a vehicle is slow moving, like "C" in figure 2, a large number of vehicle pixels will be considered as background pixel values even though they are far away from edges [10].

2.4 Smoothness detection method

As contrast to edge detection, smooth detect is used to find a smooth region in a trace of pixel intensity values. It is based on the same idea as the edge detection: background intensity is relative stable. In this method, a slide window is used to move along (in time) in the trace of pixel intensity. When all pixel values within a slide window are in a certain range, it's considered as a smooth region. By repeatedly moving the slide window to the next pixel until all pixel values are

touched, the average intensity of all smooth regions is calculated and used as background pixel intensity estimation. This method is illustrated as in Figure 3:

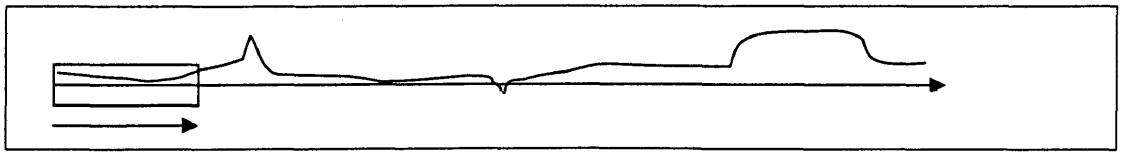

Figure 3: Smooth detect background

The size of the slide window is a very important parameter in the above process. If the size is too small, then it may classify some slow moving vehicle pixels to background. Otherwise it would be hard to find a positive match of background pixel intensity. Different smooth regions may need different sizes of the slide window otherwise slow moving pixels can also be classified as background.

Another parameter in the above process is the threshold of the intensity value resulting from the slide window. The threshold is used to determine if the region is smooth enough. A different region may need a different threshold. It is somehow difficult to find both a reasonable different size and a threshold value for a different region so that the system can make the right decision about the background identification [11].

2.5 Exponential forgetting

Exponential forgetting is a background updating method based on time average. This method was proposed for trying to make the background model more adaptive to most recent illumination changes. In this method, the intensity values are

weighted, and the most recent value has the most significant weight. It counts the contribution of an image to the background image according to time, illustrated by equation [11]:

$$B(x,y,t) = (1-\alpha)B(x,y,t-1) + \alpha I(x,y,t)$$

This equation shows that the most recent images and past images have different contributions to the background intensity determination. When recursively using this formula (suppose α=0.5), what we get is an exponential equation:

$$B(x,y,t) = 0.5I(x,y,t) + 0.5^2 I(x,y,t-1) + ... + 0.5^{n+1} I(x,y,t-n)$$

Past images have exponentially decreased contributions to the generated background image [12]. By doing this one can make the calculated background image represent the most recent illumination change.

Another method in background pixel identification makes use of the Kalman-filtering strategy. It is different in that a prediction is produced as a reference image. The machine needs to learn the difference between the prediction and the actual image, and use this learning ability to improve the next prediction [12].

2.6 Mask Classified Updating

Classified updating is a different method from exponential forgetting in that the pixel sequences are classified into background pixel class, shadow pixel class and object pixel class, respectively. This method is intended to improve the background updating performance.

In Surendra's works [13] [14], an object mask was used to classify pixels into

background and vehicle pixel. This allows us to update the background only using the classified background pixel. However, this method has a problem with slow moving vehicles that will cause error classification of the pixel classes.

2.7 Gaussian distribution model

Gaussian distribution is a widely used model to describe the distribution of random variables, such as the background intensity value in this research. Gaussian distributions have many convenient properties, so random variables with unknown distributions are often assumed to be in Gaussian distribution according to the central limit theorem. This theorem tells us that the sums of random variables tend to be approximately normally distributed under certain conditions. Because of its convenience in modeling and analysis, even if a distribution is not truly normal, often it is still approximated as a normal distribution in many applications, as long as such approximation is within the certain limit of error estimation. In this case, we can describe the entire distribution by simply a mean and a variance parameter.

The probability of a certain pixel intensity occurring depends on whether this certain intensity value is close to the mean value of the samples. For example, in the traffic scene imaging process, if we sample a group of pixels in a certain period of time then most values of the pixel in a background region would be in a narrow range that is close to the mean value of the Gaussian distribution. Only a few values drift out of this range because of object intruders and shadow. This observation could be quantified by probability density function (PDF) [15]:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2\sigma^2}(x-\mu)^2 \right],$$

where $-\infty < x < \infty$ and $\sigma > 0$ for some parameters μ and σ.

The PDF of Gaussian distribution could be plotted as figure 4:

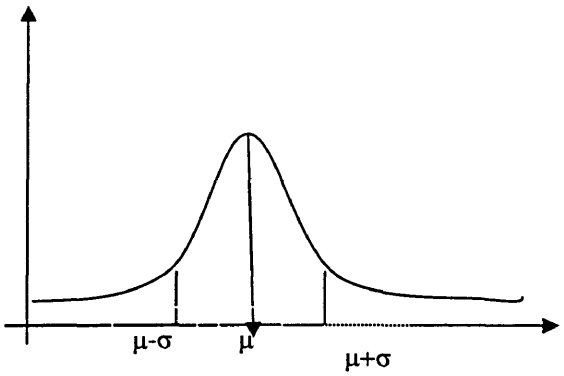

Figure 4: Gaussian distribution

PDF of Gaussian distribution is plotted as a bell-shaped curve, with the most frequently value occurring at μ. The curve is symmetric around μ. μ and σ^2 are the expected value and variance of this pixel sample respectively[16]. A more specific mixture of three Gaussian model was proposed by N.Friedman[17] to model image pixel intensities for vehicle shadow and slow moving objects. When a vehicle is slow moving, the object intensities may dominant the pixel samples. The single Gaussian model thus tends to make a wrong decision that considers vehicle pixels as background pixels and updates background using the wrong pixel information. Single Gaussian model may also consider shadow as moving vehicles. In the mixture of three Gaussian

model, distribution of a intensity value at pixel (x,y) is calculated as weighted sum of three distributions $r_{x,y}$(road),$s_{x,y}$(shadow) and $v_{x,y}$(vehicle):

$$i_{x,y} = w_{x,y}.(r_{x,y}, S_{x,y}, V_{x,y})$$

In this research, the mixture of Gaussian model was chosen to represent multiple intensity distributions where each distribution is believed to have its own texture or structure that provides different ranges of intensity values in the same image [18]. The mixture of Gaussian probabilistic approach classifies pixels into shadow, vehicle and background (road) classes. The method makes background calculation using only those pixels that are classified as background pixels. We assume that each of the ranges of background intensity values is approximately in Gaussian distribution. Figure 5 shows that pixel samples on a pixel by time could be fitted in a mixture of 3 Gaussian models.

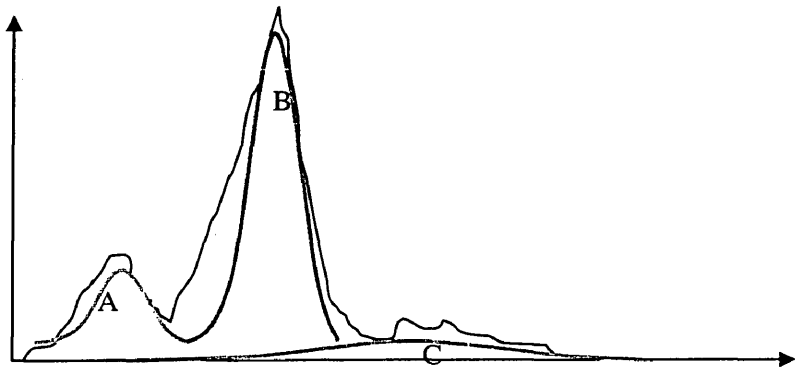

Figure 5: Mixture of Gaussian model

"A" represent shadow, "B" represents background and "C" represents vehicle. "A"

and "B" could be easily fitted to a Gaussian curve, but "C" is not so easy.

The mixture Gaussian model not only allows us to correctly update background without noise interference from vehicles, but also handles shadows properly.

Associated with the mixture of Gaussian model is the mask updating. Mixture of Gaussian and mask updating has the same idea: before updating the background pixel must be classified. Otherwise, the vehicle pixel will cause noise.
Doing so leads to an improvement in the time average and exponential forgetting methods. However, the mixture of Gaussian has its own problems. The Gaussian distribution may be used to describe the distribution of vehicle pixels where the actual distribution may far from Gaussian distribution.

Overall, the above mixture 3-Gaussian model is efficient for background modeling in case the view of camera is static, such as in an office environment and roadway. Some outdoor scenes such as forest-like environments contain lots of leaves and brushes. Outdoor wind may make these leaves and brushes move around in high speed, thus causing a non-static background. Background change depends on the strength of the wind. For example, outdoor wind may make a pixel change from sky to leaf, or from leaf to brush very quickly causing a pixel to be a mixture of different things with different texture and color.

Elgammal [19] uses a generalized K Gaussian model to handle this non-static background in outdoor forest-like environment. The generalized mixture of K Gaussian model calculates the PDF of each pixel intensity value X_t at time t by a

kernel function k [20]:

$$P_r(X_t) = \frac{1}{n}\sum_{i=1}^{N} k(X_t - X_i)$$

This PDF can estimate the probability density by using the most recent information so it is adaptive to quick small background changes in a forest-like scene. Since we mainly deal with city traffic scenes in this research, the forest model is of less interest.

3.Vehicle detection (1): Median model

3.1 Median model

Background image generation is the first and the most important step for applying the background subtraction method to detect vehicles. There are different ways we can use background image generation as shown in the previous section. In this section, we discuss a median model that is developed in this research for background generation and application of the background for the vehicle detection.

As we have discussed in last section, a critical issue of background generation is handling the background changes over time. It is important that background generation model use features that either adapt to these changes or are invariant to these changes. Another issue when vehicles move very slowly or stand still for a while is how to distinguish them from background. In our project most image sequences are taken at the intersection of the road so that vehicles move relatively slow and possibly stationary. Exponential forgetting or time average will fail in this case because both of them will not avoid treating some stationary vehicles as background thus producing a corrupted background image (Figure 6).

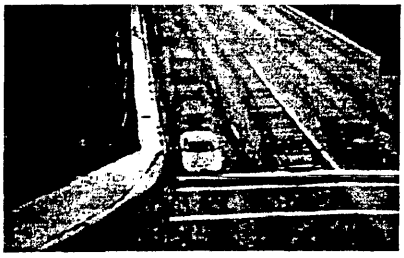

Figure 6: Corrupted Background image when vehicles are stationary

Since classified the updating technique uses only background information to update background representation, error classification is a very common problem when vehicles are stopped at an intersection. In this case, stationary vehicles will be classified as background thus being used for background updating. For example, Surendra[13] used a binary object mask to distinguish the foreground pixels and background pixels. The object mask is obtained by subtracting the current image from current background. In a stationary situation the current background will be noise corrupted by stationary vehicles and thus produce an incorrect object mask and pass this incorrect classification down to next step.

When using a mixture of Gaussian model, sometimes the vehicle pixel samples are far away from Gaussian distribution, thus causing an error classification.

For this project, the most critical considerations are: be adaptive to illumination change; handle slow moving or stationary vehicles and be applicable in a real-time image processing system. The median model is proposed based on these considerations.

3.2 Introduction of the median model

In this project we focus on dealing with slow moving or stationary vehicles, illumination adaptation and time complexity in generating background images for vehicle detection. We ignore the effect of shadows in this project.

We model the median of a pixel in an image sequence as in Gaussian distribution N (μ, σ^2) when the sample size is large enough. This means that in a

certain period of time a pixel is classified as either belonging to the background or to objects. We assume the combination of the object sets and background sets provides a complete picture of an image.

If we sort the pixel sequence, we could get one background set and two object sets. The background set is in the middle of two object sets and represents the flat region. The reason is that in a certain period of time the background intensity does not change too much and it is in normal distribution. On the other hand, if the objects are totally random intruders, then their intensity values are also random. They may either lessen the background intensity value or increase the background intensity value. When the sample size is large enough, two objects sets are statistically equal. This makes the background set locate exactly in the middle of the pixel sequence. We call it the balanced median model. (Figure 7):

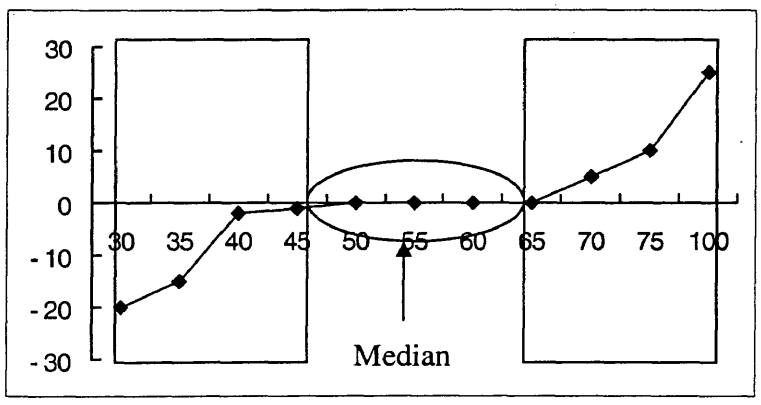

Figure 7: Balanced median model

In the balanced model the median is within the background set. We can use the median to estimate the background intensity in this case. Sometimes the object

sets may occupy a large portion of the sequence and the background set interval may be very small. In this case as long as the sample size is large enough and the pixel sequence is in the balanced median model, the median could still be used for background estimation. This is why the median model is capable of dealing with the situations where the objects are slow moving or the traffic is busy and background is not visible most of the time.

When the sample size is not large enough, two object sets may have different distribution probabilities. In this case the object set will shift to either the right or left side and the median of the sequence will locate in one of the object sets. We call it a shifted median model (Figure 8). In shifted median model we could not use the median for background estimation because the median is within an object set.

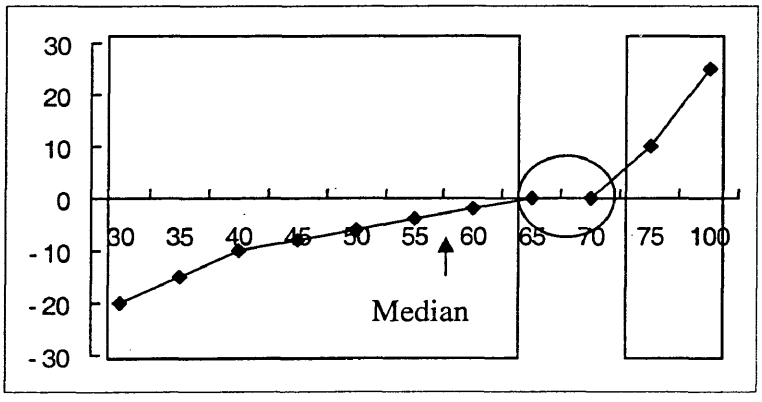

Figure 8: Shifted median model

Therefore there are two cases: the first one is the balanced median model, and the second one is the shifted median model. We can use median for background

estimation only when it is in the first case, the balanced median model. For each pixel sequence we must make the sample size large enough so that we can use the balanced model to locate the median for background estimation.

The problem is for each pixel, the required sample size could be quite different. For the regions where the intruders seldom entered, the sample size could be very small. For other regions for example surrounding the intersection, the probability for each pixel belonging to objects is large. Therefore, the required sample size is also large.

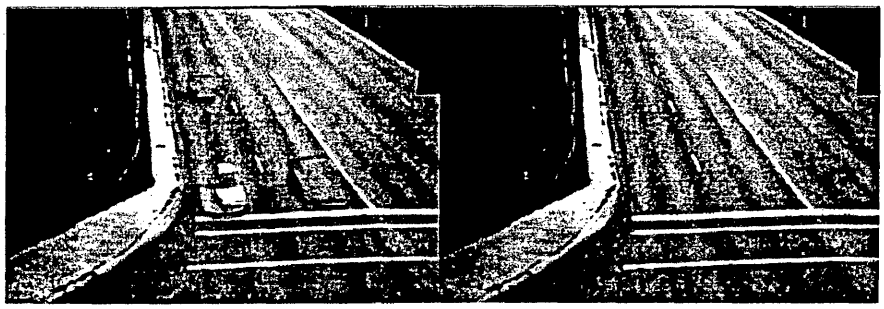

Figure 9: Background generation with different sample size

Left image size= 5, right image size=13

Figure 9 shows that in different regions of the image, the sample size should be different. The question is how could we find a different sample size for a different region?

We observe that within the background pixel set the deviation of pixel intensity is relatively small, while within the objects pixel set, the deviation is much larger than that of the background set. This is because objects are completely random intruders (Figure 10) with respect to the background.

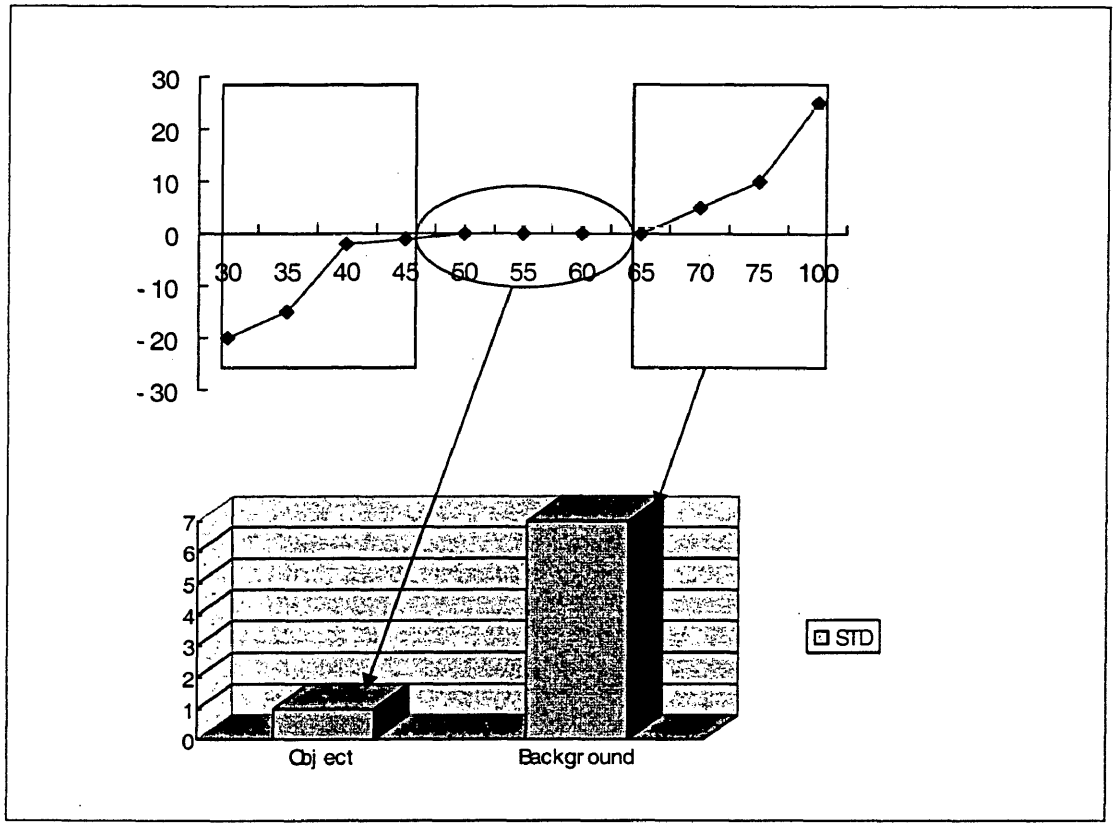

Figure 10: Deviation Comparison

We thus took advantage of this property and adopted a heuristic approach. The strategy is to compare the median value with its two neighbors. The differences are d1 and d2 respectively. If the median has no significant difference from its two neighbors, then we consider it is in a balanced median model and the median is within the background set. In this situation the median is accepted for background estimation. If the difference is significant it suggests that the sample size must be very small and it is in the shifted median model. Since the median is within the object set it could not be used for background estimation.

The median model background generation algorithm then will repeat the above process until all pixels are touched and then the background image will be generated from the image sequence. The algorithm could be roughly described in this pseudo code:

```
For (each pixel)
{
   size=s   Continue = true
   while (size < max and continue = true)
   {
      A[size]= sample pixel set for current pixel
      sort (A)
       m=median (A)
      diff1 = abs(m – neighbor1)
      diff2 = abs(m - neighbor2)
     if (diff1 < d and diff2 < d)
      accept m and continue = false
     else size ++
  }
}
```

Figure 11 shows that when using this median model algorithm, the generated background image is very successful:

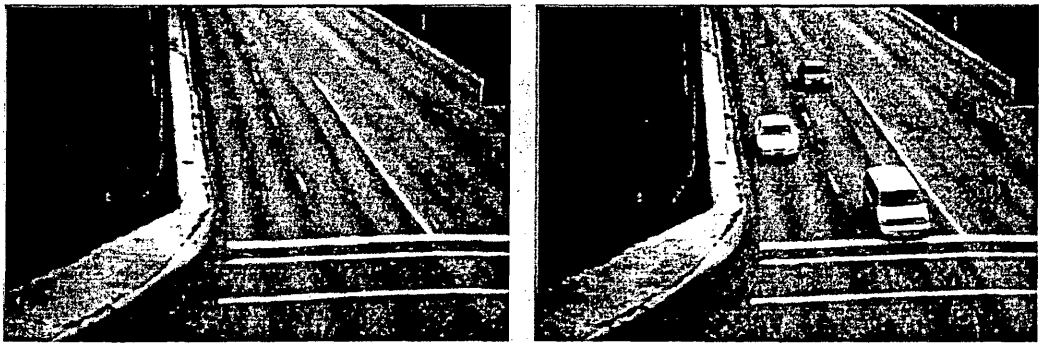

Figure 11: Generated background using Median model

Left: real background image, Right: Current image

The computational complexity for this algorithm is dependent on the sorting

process. Suppose we use a sorting algorithm with a time complexity of O(nlogn) and have a sequence of image (each with size of P pixels) with sequence size equal to S. The computational complexity for this algorithm is:

$$\sum_{1}^{P} K_i S_i \lg S_i ,$$

where S_i is randomly from 1 to S. K_i is the number of loops needed to find the right median. For background pixels, S_i and K_i could be very small. For object pixels, S_i and K_i could be as large as S.

3.3 Improved median model

Two common requirements of using background subtraction method for object detection are: (1) adapting to illumination change (2) and handling slow moving or even stationary objects in image frames. The above Median model uses the image sequence to calculate background. Theoretically it is capable of dealing with slow moving vehicles, however the sample size must be large enough and a different pixel may require a different sample size so that each pixel sequence is in the balanced median model. This sometimes will sacrifice speed and affect the illumination adaptability. For real time image processing we make use of the following improved median model.

Instead of samples of different sizes for each pixel, in the improved median model samples are a constant small size for each pixel and make most of the samples out of stationary sequence. We do this by sampling intensity values in a longer time interval.

If interval is long, it will sacrifice illumination adaptability but be more effective in handling stationary vehicles since more images are selected from the set out of the range of stationary images. If interval is short, it will sample only within the stationary images, therefore be better in illumination adaptability but poorer at handling stationary vehicles. This is a trade-off.

For effectively handling stationary vehicles at the intersection of the road some parameters are necessary. The most important parameter is how long vehicles stop at the intersection. This stop duration decides the sample size and interval of time for sampling. We test this model by using different sample sizes and different interval times in the intersection of our video camera setting – the street intersection of 67ST and Pacific ST where the stop duration for East-West direction is about 40 seconds. We set the sample size from 5 to 20. The interval between two sample images is about 15-20 seconds. The total sample time should be 5 times more than the stop duration, which is about 200 seconds. The time complexity for this improved median model is:

$$\sum_{1}^{P} S \lg S = pk$$

Here p is the pixel resolution of an image. S is a constant value, usually $5 < S < 20$. Figure 12 is an image sequence in the worst situation: vehicles stop for 40 seconds in the intersection.

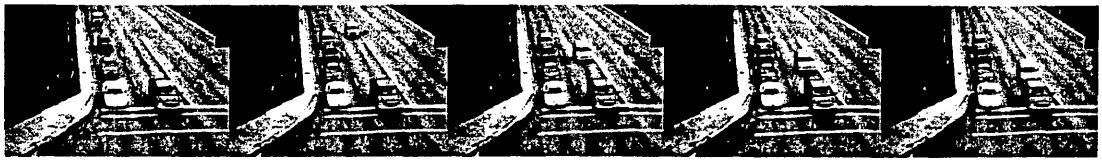

Figure 12: stationary image sequence for background image generation

The improved median model is illumination adaptive when illumination changes slowly. This is true in most situations on outdoor roadways where illumination doesn't change much in several minutes. When the illumination changes very quickly and frequently, this model will not hold. For fast changing illumination weather conditions, we need to update the background using the most recent image, for example the current image. The current image contains information of both background and vehicles. We need to classify the current image, classify the background pixel and the vehicle pixel, and only use the background pixel to update the background image. This is called classified updating. When this improved median model combined with classified updating, it is capable of handling not only slow moving objects but also quick illumination changes, and yet it has an acceptable time complexity of computation.

3.4 Background updating of median model

As we pointed out, the improved median model is a long-term background model and is adaptive to slow illumination changes. There is another short-term background updating method called mask classified updating that uses the most recent object mask to selectively updating the background. The mask classified updating uses the most recent image for background updating so it readily adapts to fast illumination changes.

The drawback of mask classified updating is: The incorrect detection result could be used to generate an incorrect object mask and the later detection using this mask will produce an incorrect result, therefore passing another incorrect object mask for later detection. This is a deadlock situation common for short-term updating.

In order to build a detection system that is not only capable of handling slow moving and stationary vehicles but is also adaptive to fast illumination changes, we combine the long-term and short-term background together.

The long-term background generated from the improved median model is subtracted from the current image to produce an object mask. This object mask is used to classify the current image into object regions and background regions, then updating the object region by sampling pixels from the current background image, updating the background region by sampling from the current image. By doing this, the short-term background image is generated.

Based on the long-term background and short-term background, the final background is produced by:

$$UB = \alpha \bullet SB + (1-\alpha) \bullet LB$$

Where SB stands for short-term background and LB stands for long-term background. The final updated background image (UB) is the weighted sum of SB and LB. The value of α has range from 0 to 1. When α is too small, the short-term background will dominant the background updating. The detection system will be more adaptive to fast illumination changes but more likely to get into a deadlock

situation. When α is too large, the detection system is more capable of handling slow moving vehicles but less adaptive to fast illumination changes.

3.5 Performance Evaluation

We evaluated some of major methods for background generation through experimentation on real street traffic images. The results are discussed in the sections below.

3.5.1 Time average

Time average has inherent drawbacks. The most critical one is that it uses both background pixels and vehicle pixels for background generation. The result of time average is highly dependent on the specific image sequence and the size of the image sequence. In situations where vehicles kept moving, time average works only when the background is visible in most cases in the image sequence:

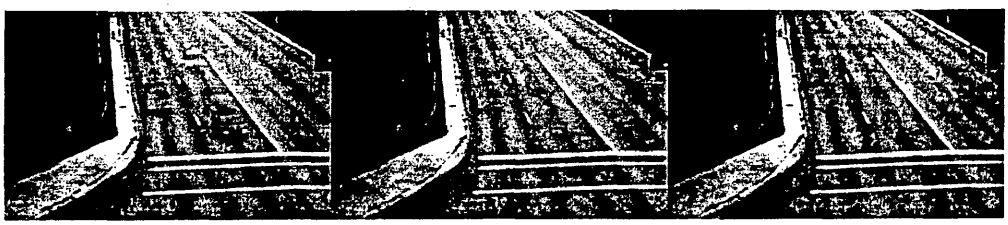

Figure 13: Time average result on moving vehicles
Sequence size is: 5, 13 and 21 respectively

When the background is blocked by moving vehicles and is invisible, the result is still not good cnough cven the sequence size is very big. Figure 14 shows this situation:

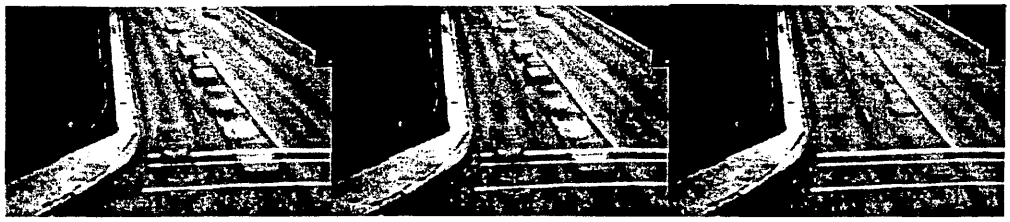

Figure 14: Time average result on blocking vehicles
Sequence size is 5, 13 and 21 respectively

When vehicles are stationary, time average will fail because stationary vehicles will block some of the background:

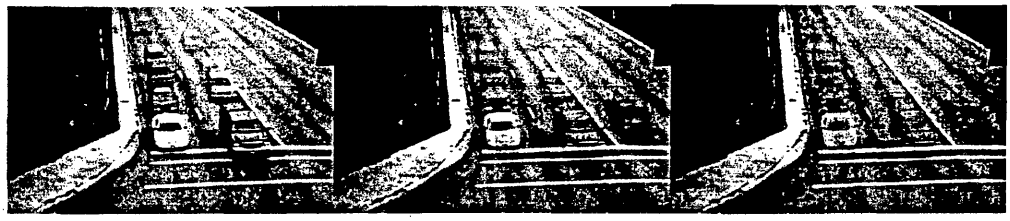

Figure 15: Time average result on stationary vehicles
Sequence size is 5, 13 and 21 respectively

3.5.2 On Exponential forgetting

Exponential forgetting has a common problem on background generation: If the α value is too large in the formula:

$$B(x,y,t) = (1-\alpha)B(x,y,t-1) + \alpha I(x,y,t)$$

The current image will seriously affect the background. Figures 16, 17 and 18 show that when α value is 0.5, the current image dominates the background when the sequence size varies from 5 to 21:

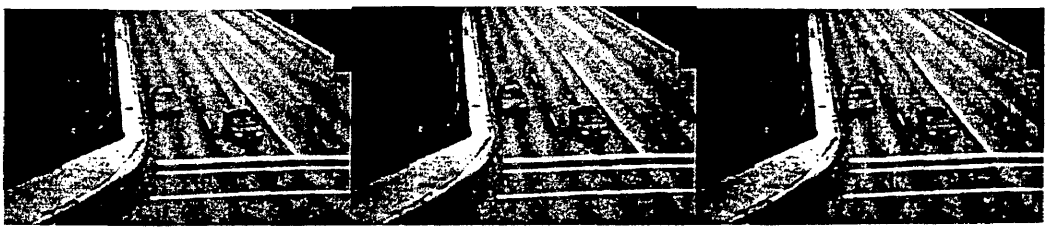

Figure 16: Exponential forgetting result on moving vehicles
Sequence size is 5, 13 and 21 respectively

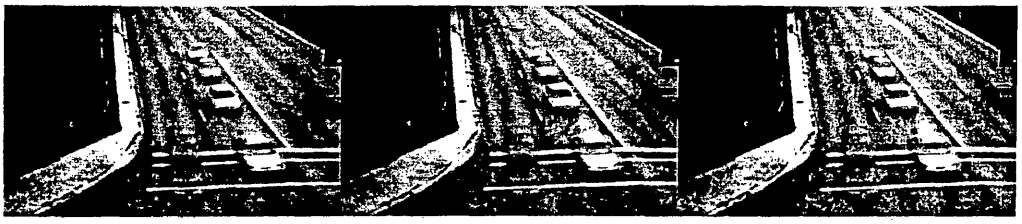

Figure 17: Exponential forgetting result on blocking vehicles
Sequence size is 5, 13 and 21 respectively

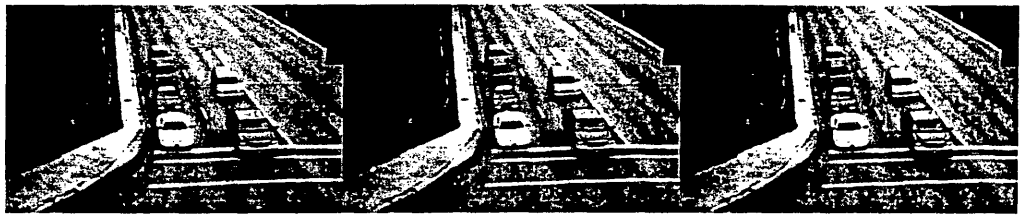

Figure 18: Exponential forgetting result on stationary vehicles
Sequence size is 5, 13 and 21 respectively

When α is small, a better result may be obtained on moving vehicles and blocking vehicles, but it still cannot handle stationary vehicles. Figure 19 shows this situation:

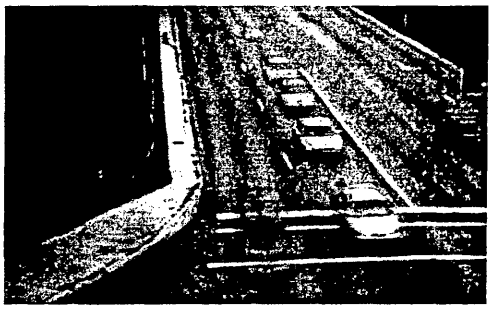

Figure 19: Exponential forgetting result

3.5.3 On Median model and background updating

We take image sequence from the intersection of our experimental setting with the stop duration about 40 seconds on the east-west direction. The time interval of each sample is about 20 seconds. For moving vehicles, sample size of 5 is large enough to build the current background image.

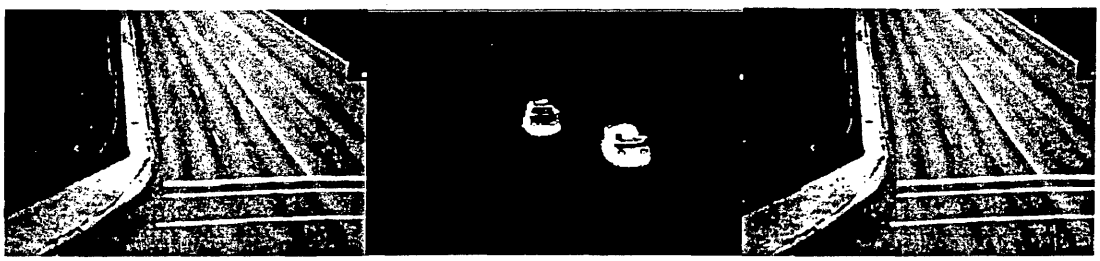

Figure 20: Median model result on moving vehicles
From left to right: current background, object mask and updated background

The current background is XORed with current image to produce an object mask. Object mask is binary image contains two regions: vehicle region (value is 1) and background region (value is 0). Current background pixels will be sampled where pixel values are 1 in mask. Current image will be sampled where pixels values are 0 in mask. Figure 20 shows the result.

For blocking vehicles that other algorithms cannot handle, the improved median model handles it very well. Figure 21 shows the result:

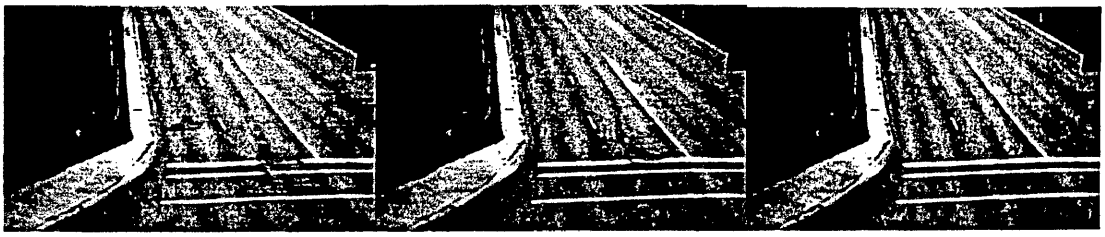

Figure 21: Median model result on blocking vehicles
Sequence size is 5, 13 and 21 respectively

When sample size is 5 to 13, generated background image contains a little noise, but better than the result of time average and exponential forgetting. When sample size is 14 to 21, the generated background image is very good. Figure 22 shows the object mask and updated background image for the blocking vehicles:

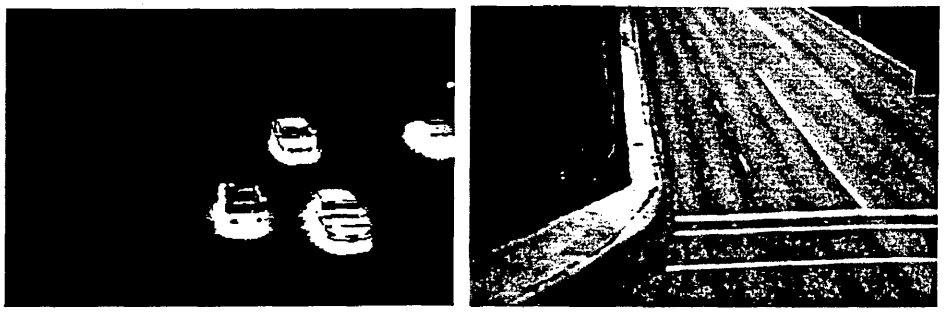

Figure 22: Median model result on blocking vehicles
Left: object mask, right: updated background

Stationary vehicles are a big problem for both the time average and exponential forgetting method. When we use the improved median model with sample size of 13 or larger, a very successful result is reached. Figure 23 and 24 show these results, respectively.

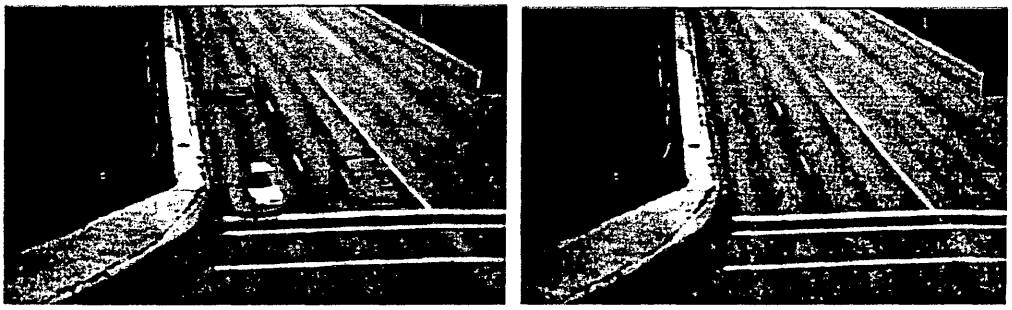

Figure 23: Median model result on stationary vehicles
Sequence size is 5 and 13 respectively

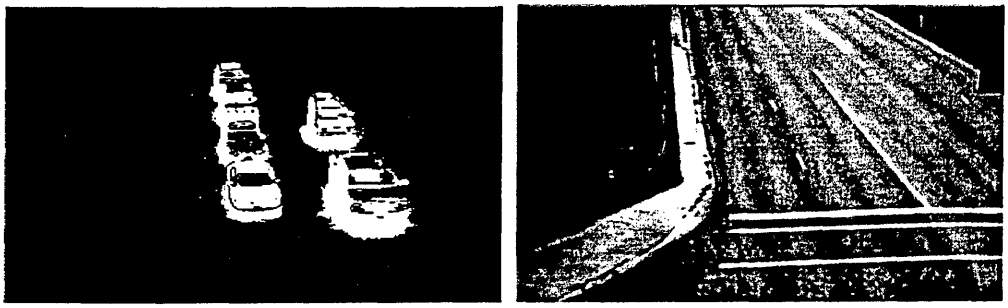

Figure 24: Median model result on stationary vehicles
Left: object mask, right: updated background

4. Image Segmentation and Noise Removal

4.1 Background subtraction

After the background reference image is built by using the above median model and improved median model, arithmetic image operations are applied to subtract the background from the current image. The final result of this arithmetic operation would be a binary image with value 0 represents background and value 1 represents objects.

4.1.1 Pixel level background subtraction

Pixel level image subtraction is the simplest arithmetic operation. In this operation pixel intensity value of current image at position (x,y), represented as C(x,y), will be compared with corresponding pixel intensity value in the background reference image R(x,y). The subtraction operation could be represented as:

$$S(C(x,y),R(x,y)) = \begin{cases} 0, C(x,y) = R(x,y) \\ 1, C(x,y) \neq R(x,y) \end{cases}$$

In a real implementation, the corresponding pixel value pair R(x, y) and C(x, y) is probably not exactly equal even though they represent the same vehicle pixel. We need a range value *"g"* to make it flexible: The above expression thus is changed to

$$S(C(x,y),R(x,y)) = \begin{cases} 0, \mid C(x,y) - R(x,y) \mid < g \\ 1, otherwise \end{cases}$$

Pixel level subtraction is straightforward and effective in most situations. When the current image pixel cannot be precisely mapped to background reference image, pixel level subtraction may produce too much noise.

4.1.2 Block level background subtraction

Block level image subtraction is a more robust method that performs subtraction over a small neighborhood of a pixel. A k*k slide window is used. The center pixel of the slide window moves along pixels of the background reference image and the current image. The mean value of the slide window instead of the single pixel is used for comparison: Let M_{ij} and M'_{ij} represent the mean values of the slide window for the current image and the background image respectively, we have

$$M_{ij} = \frac{1}{k^2} \sum_{x,y \in W_{ij}} K(x, y)$$

$$M'_{ij} = \frac{1}{k^2} \sum_{x',y' \in W'_{ij}} K(x', y')$$

The subtraction function is very similar to the pixel level subtraction function:

$$S(C(x, y), R(x, y)) = \begin{cases} 0, |M_{ij} - M'_{ij}| < g \\ 1, otherwise \end{cases}$$

This block level subtraction method has the computational complexity of O(nk) where k is the size of the window. Usually k is a very small constant, so the computational cost is actually O(n), which is the same as the pixel level subtraction algorithm [21].

4.2 Noise removal

4.2.1 Threshold

After using the background subtraction method a difference image is produced. If we sort and count the difference image pixel according to intensity values a histogram of the difference image is obtained as shown in figure 25.

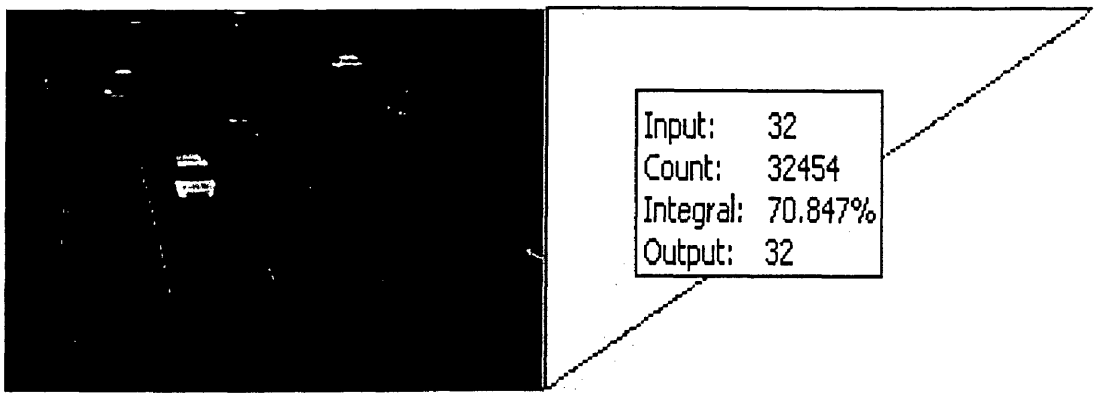

Figure 25: Difference image and histogram

In the current image vehicles usually occupy a smaller part of the image. The background covers a larger part. After subtraction, the background pixel pairs will output a very small intensity value while object pixels will produce a much larger value. The intensity value of the background pixel pairs is represented by the first peak in the histogram of the difference image as shown in the above figure. At the value of 32, a large count of 32454 is obtained.

To distinguish the background pixels and object pixels in the subtracted (difference) image, a threshold value should be selected to locate on the right side of the first peak. That is, it should be greater than 32 in the above example. Figure 26 shows when we use different threshold values on the difference image, we could get quite different results.

After threshold on the difference image, vehicles are separated from the background and there is only a little noise staying with the vehicle pixels.

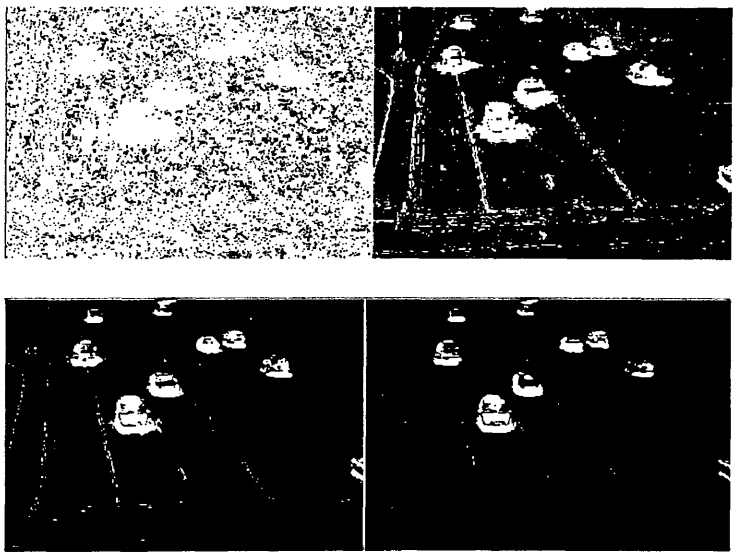

Figure 26: From up-left to bottom-right with threshold value Of 0, 10, 20 and 33 respectively

A more accurate threshold algorithm is available, such as [22] a procedure applying the following sequence of operations:

- smooth the image,
- calculate the gradients,
- calculate gradient histogram,
- calculate gradient threshold as a percentile,
- separate pixels with large gradients,
- calculate image threshold as average intensity of high-gradient pixels,
- ensure validity of threshold by comparing it to average image intensity, and
- apply threshold.

The algorithm performs a smoothing operation on the image first. Then it tries to find edge pixels with very large gradient value. The average value of the high gradient edge pixels is used as the threshold value. This does make sense because edge pixels will occur mainly in the neighborhood of the boundaries between vehicles and the background. If the image is properly smoothed, approximately half of the selected edge pixels are on the background side and half belong to the object side. Thus the average intensity of the edge pixel values could separate the objects from background.

This algorithm still could not guarantee to completely avoid noise. Actually, it is impossible to completely remove noises by using threshold alone for vehicle detection at the roadway. Therefore we combine threshold, erosion and dilation operations together for noise removal in this thesis.

4.2.2 Eroding and Dilating [23]

The image morphology is based on set operation. Consider 2 sets, A and B, for which each element is a 2-D point (vector).

- The intersection of the sets is defined as $A \cap B = \{P \mid P \in A \wedge P \in B\}$.
- The union of the sets is defined as $A \cup B = \{P \mid P \in A \vee P \in B\}$.
- The complement of a set is defined as $\overline{A} = \{P \mid P \notin A\}$.
- The translation of a set by vector q is defined as $A_q = \{P + q \mid P \in A\}$.

(I) Erosion

Erosion is the opposite of dilation. The basic effect of the operator on a binary image is to erode away the boundaries of regions of foreground pixels. Thus

areas of foreground pixels will shrink. Holes within those areas become larger. Erosion can remove a layer of pixels all the way around an object. If the object is not large enough, some or all of it will disappear completely. This is useful for separating closely spaced particles which have come out as connected after segmentation, for removing small spots of noise which are not features. The erosion operator takes two pieces of data as inputs. The first is the image to be eroded, such as the one shown in figure 27.

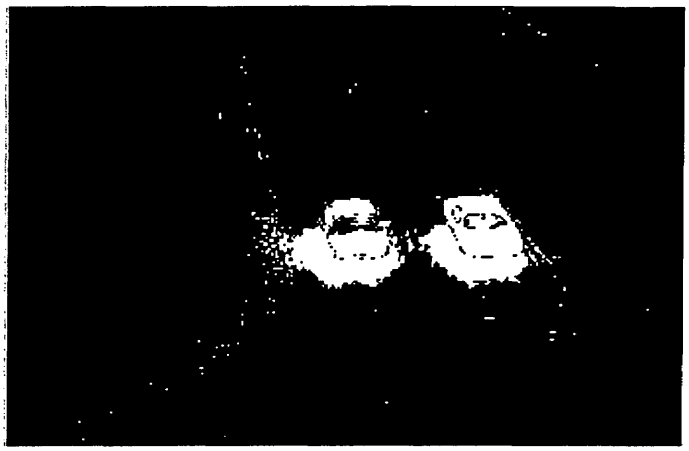

Figure 27: Difference image after threshold, before erode and dilate

The second image is a (usually small) set of coordinate points known as a kernel. The kernel determines the precise effect of the erosion on the input image. An erosion operator (sometimes called "Minkowsky subtraction", illustrated by symbol θ) is defined as follows [9].

$$A \ominus B = \{p \mid B_p \subseteq A\}$$

Here B is a kernel moving through A. This means a pixel belongs to $A \ominus B$ only when translation of B by this pixel belongs to A. For an example, let B={(0,0),(1,0),(0,1)} and A={(1,2),(1,3),(1,4),(2,2),(2,3),(3,2),(4,2)}.

The erosion result is $A \ominus B=\{(1,2),(1,3),(2,2)\}$. This could be illustrated in figure 28.

We can think of this as to take copies of A and translate them by movement of kernel B over the copies. If we intersect these copies together, we can get $A \ominus B$. This can be illustrated in figure 28:

00	01	02	03	04
10	11	12	13	14
20	21	22	23	24
30	31	32	33	34
40	41	42	43	44

↓

00	01	02	03	04
10	11	12	13	14
20	21	22	23	24
30	31	32	33	34
40	41	42	43	44

Figure 28: Erosion

Using erosion alone for removing noise will cause some problems. Erosion not only can erase the noise but also corrupt the image. For example, a single vehicle in an image could be corrupted into several small pieces which is not acceptable for vehicle identification and labeling. To solve this problem, we repeatedly use dilation and erosion to remove noise as well as keep vehicles in good shape. This strategy has

the same effect as an opening operation. Figure 29 shows the eroding result after the operator is applied to the image in figure 27.

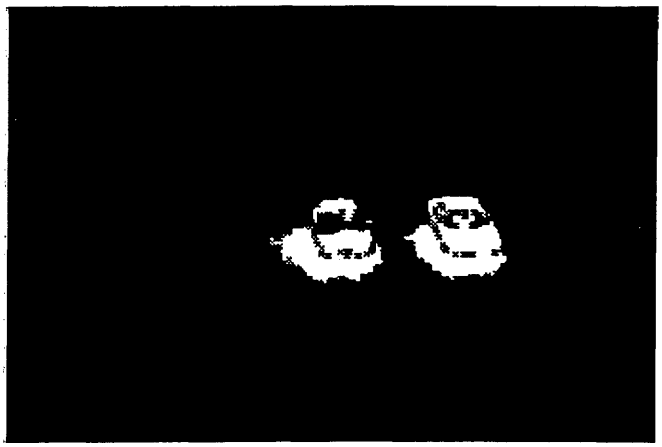

Figure 29: Erode result

(II) Dilation

The basic effect of a dilation operator on a binary image is to gradually enlarge the boundaries of regions of foreground pixels. In this operation a pixel with a value 0 having pixels of value 1 as its nearest neighbor will be changed to the value 1. Thus areas of foreground pixels grow in size while holes within those regions become smaller. This is particularly useful for enhancing thin, faint objects. If this procedure is not used the wrong number of particles (or other features) will be counted and the average size of the objects will be completely wrong. Dilation is defined as [10]:

$$A \oplus B = \{a + b \mid a \in A \text{ and } b \in B\}$$

We can think of this as to take copies of A and translate them by kernel B. If we union these copies together, we get $A \oplus B$. This can be written as:

$$A \oplus B = \bigcup_{b \in B} A_b$$

Alternatively, we can take copies of B and translate them by A. An interpretation of this latter way of thinking is to put a copy of B at each pixel in A. If we "stamp" a copy of B at each pixel in A and union all of the copies, we get $A \oplus B$.

This can be written as: $A \oplus B = \bigcup_{t \in A} B_t$. In this way, dilation works like slide a kernel to each position in the image and at each position "apply" (union) the kernel. For an example, let B={(0,0),(1,0),(0,1)} and A={(1,2),(1,3),(2,2),(2,3),(3,2)}. The erosion result is $A \oplus B$={(1,2), (1,3), (2,2), (2,3), (3,2), (1,4), (2,4), (3,3), (4,2)}. This could be illustrated as following:

00	01	02	03	04
10	11	12	13	14
20	21	22	23	24
30	31	32	33	34
40	41	42	43	44

00	01	02	03	04
10	11	12	13	14
20	21	22	23	24
30	31	32	33	34
40	41	42	43	44

Figure 30: Dilation

A dilation result of figure 29 is shown in figure 31.

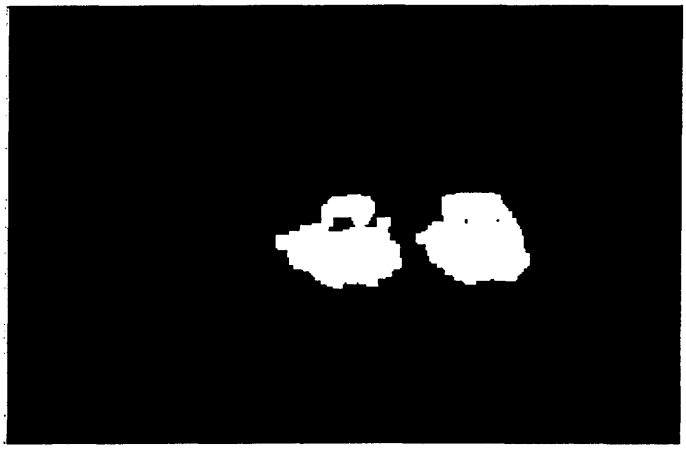

Figure 31: Dilate result

4.3 Connected components labeling [23]

Connected component labeling method works on binary image or grayscale image. Different measures of connectivity can be applied. In this thesis we assume binary input images and *8-connectivity*. The input image for labeling comes from performing the background subtraction operation. The images are in binary scale. Figure 32 shows an input image for labeling.

Figure 32: Input image for labeling

The connected components labeling operator scans the image by moving along a image from left to right then up to down. In 8-connected binary image when a pixel with a value of 1 is met, four of its neighbors that have already been encountered in the scan (the four neighbors to be checked are pixels to the left, above, upper-left, and upper-right) will be checked.

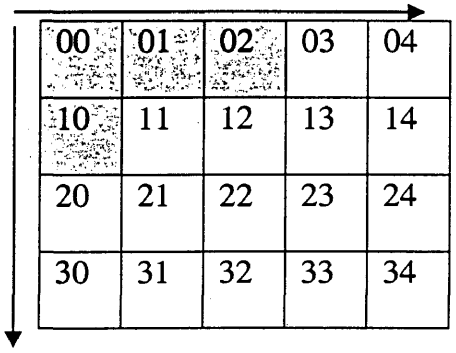

Figure 33: Connected components labeling

Based on information of its neighbors, the labeling of this pixel will consider following cases [14]:

- If all four neighbors are 0, assign a new label to this pixel.
- If only one neighbor is 1, assign its label to this pixel.
- If one or more of the neighbors are 1, assign one of the labels to this pixel and make its neighbors have an equivalent label.

After completing the scan, the equivalent labels are sorted into classes and a unique label is assigned to each class. Then another scan is made through the image so that each label is replaced with the label assigned to its equivalence classes. Figure 34 shows the labeling result with the input image from figure 12: totally 12 objects

are labeled with different color. These labeled objects include two vehicles, others are noises from background subtraction.

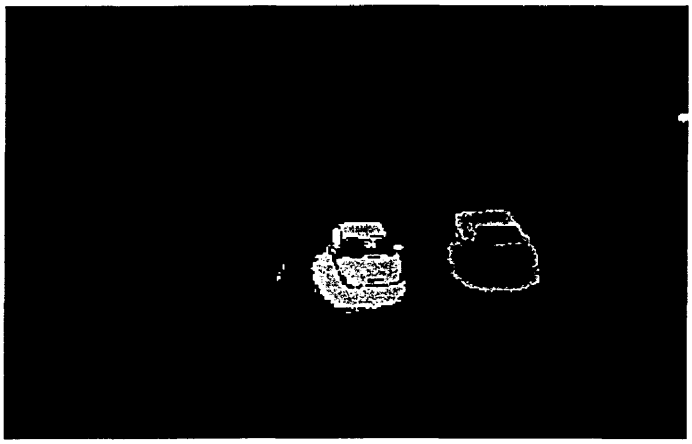

Figure 34: Labeling result (12 objects are labeled)

4.4 Automatic vehicle detection

The automatic vehicle detection is a robotic and efficient way to deal with vehicle detection. No manual operations are involved. We do this by combining background subtraction, threshold, eroding, dilating, connected components labeling and detection together to form an assemble line. We call it an auto-detect process. The sequence of operations is:

Input (Current image and background image) →Background-subtraction

→Threshold →Erode →Dilate →labeling →detect →output.

When the above operations are done with human intervention, noise removal can be helped by manual inspection and selection of proper operators and operator parameters. However, when they are assembled together and do automatically, some noises may be inevitably passed down to the vehicle detection phase.

Figure 35: Output image with noises

Figure 35 shows an output image with some noises. In this image, components of vehicles are not fully connected with each other. As a result, the labeling process will count one object as more than two objects as shown in figure 36.

Figure 36: Labeling of noisy image

To solve this problem, we introduce a threshold operation before the detection operation, so the revised operator flow is:

Input (Current image and background image) →Background-subtraction

*→Threshold →Erode →Dilate →labeling →**Threshold** →detect →output*

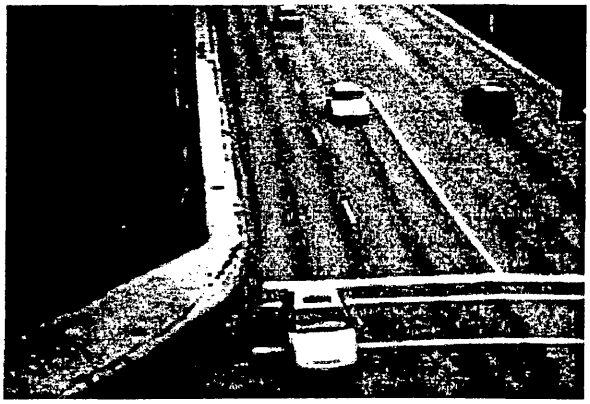

Figure 37: Input image for revised operator flow

Noises passed down to the vehicle detection process usually have very small size while on the contrary; vehicles have usually much larger size. Threshold on detection considers an object as a vehicle only when its size is large enough. For each labeled object, detection threshold counts its pixel number and calculate its size at pixel level. If the size is too small, the object will be discarded. This allows us to save some time on difference threshold, erode and dilate. For an example we can provide a universal small constant value for threshold.

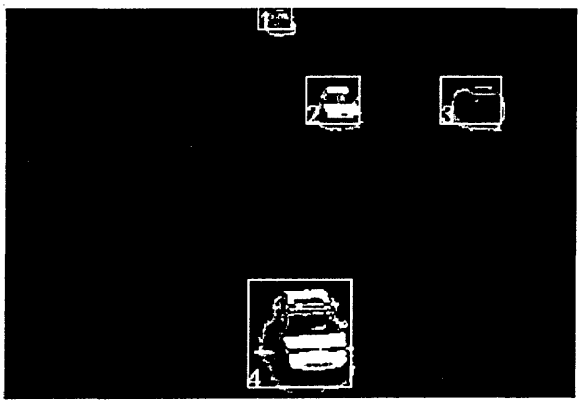

Figure 38: Auto-detect result of the revised operator flow

We also only go through erode and dilate twice, instead of several times until all noises are removed. Figure 38 shows the experimental result of the revised operator flow.

4.5 Detector performance

4.5.1 Detection performance with human supervision

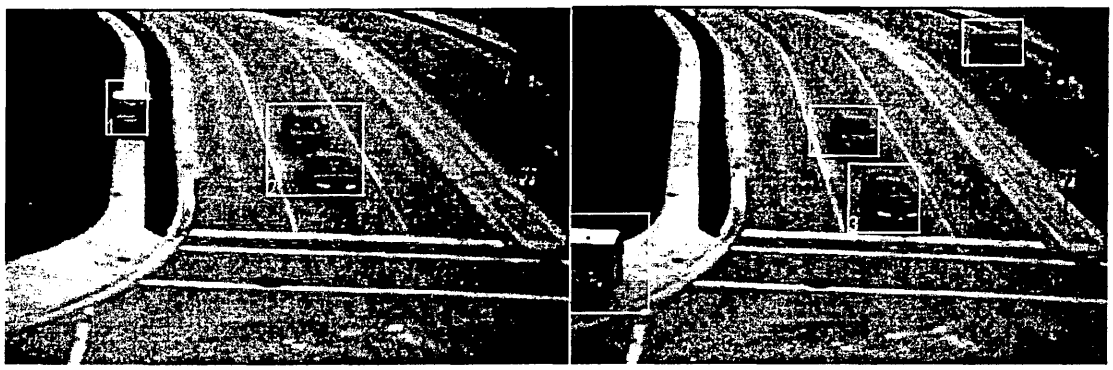

Figure 39: shadow- connected vehicles

Vehicle detection with human supervision can generally result in a 100% correct rate. The only problem is when two vehicles are very close, a shadow between them connects them together and these two vehicles will be detected as one vehicle, as shown in figure 39.

4.5.2 Auto-Detect performance

Problems to be solved in the auto-detect process include the following.

(1) The noise will not be completely removed. Thus some false positive pixels will be produced. Such false positive pixels do not happen very often. We tested several hundreds of vehicles and this kind of false positive only happened in about 2.4%. Further analysis gives us a deeper understanding about how this false positive could happen. We found that all false positives

occurred in the shoulder of the roadway when a vehicle passed by and projected its shadow on the shoulders. The projected shadow on shoulder causes a big intensity value change and thus being detected as a vehicle. Those noises caused by shadow are usually large in size so they cannot be completely removed by threshold. A mixture of Gaussian model could be used to handle this situation.

Figure 40: Shadow noise in detection

Figure 40 shows when the left white color car is very close to shoulder it projects shadow on shoulder of the roadway. This noise cannot be removed and produce a false positive detection.

(2) A vehicle may have several quite different components. Each component may be detected as a standalone vehicle. As in figure 41, the left side image has a white colored car that would be considered as two objects by the auto-detect process because its rear window divides the image region into two components. Auto-detection cannot connect these two components together. In the right side image there is a car containing two distinctive parts. The

front part of the car is black in color while its rear part is white in color. The detector cannot connect these two parts together and would output it as two separate vehicles.

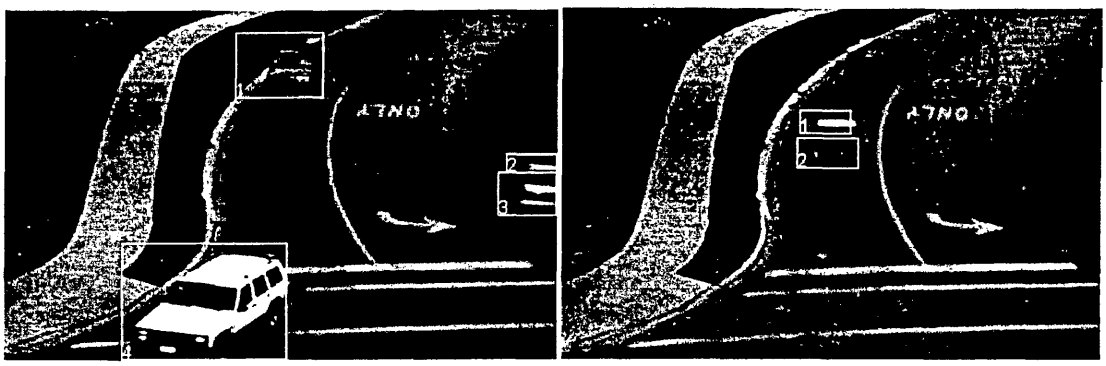

Figure 41: Auto-detection result

(3) Very small vehicles will be considered as noise. The advantage of this is: when vehicle is very small in size, they are usually far away from camera, thus their size, shape and intensity are very obscure. These kinds of vehicles are very difficult for tracking. By ruling out such vehicles we have more qualified objects for tracking.

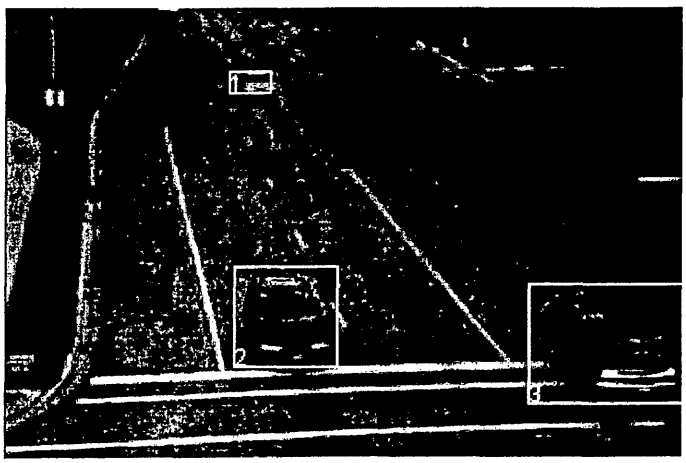

Figure 42: Threshold on object size

Figure 42 shows a vehicle at the up-left corner. It is very small and only appears partially in view. Its shape and intensity are very obscure. It is not a qualified object that we should detect and pass down to the tracking system. On the other hand, partially disappearing vehicles are usually very close to camera and have a larger size. Their intensity values are valuable for the tracking system.

We experimented with both the processes of auto-detect and detection with human supervision. Even though the auto-detect process encountered more problems in isolating and identifying individual vehicles, the overall performance is good in comparison with the human supervised detections. The following table shows the results of detection on 8 image sequences (166 image pairs) containing a total of 407 vehicles.

	Detected Vehicles	Vehicle Number	Detect Rate	False Positive
Supervised Detection	407	407	100%	0.00%
Auto- detect	417	407	97.6%	2.4%

5. Tracking system and experimental performance

5.1 Introduction

Video handoff needs the matching an image frame taken by camera i with another image frame taken by camera i+1. Vehicles inside these two images are compared to find their matches. Once all vehicles in image i+1 either are found with their matches in image i, or are identified as newly entered vehicles in image i+1, video handoff is completed. By doing this, sequentially deployed cameras can unite together to surveillance the whole roadway as well as communicate with each other.

As we have seen from the last section, in a vehicle detection stage different components of a vehicle are connected together and a unique color was assigned to the vehicle by using connected-components labeling technique. We assume in each of such labeling space there is only one vehicle though in reality more than one vehicle could be detected as a single vehicle.

There are some data that the detect stage could pass to tracking stage such as: detected image, labeled image and difference image. Figure 43 shows these data respectively. From up-left to bottom-right, these images in figure 43 represent the detected image, labeled image and difference image respectively.

The tracking stage takes these data from image i and image i+1 as input and outputs a match result between these two image frames. In the following we present techniques applied in this stage.

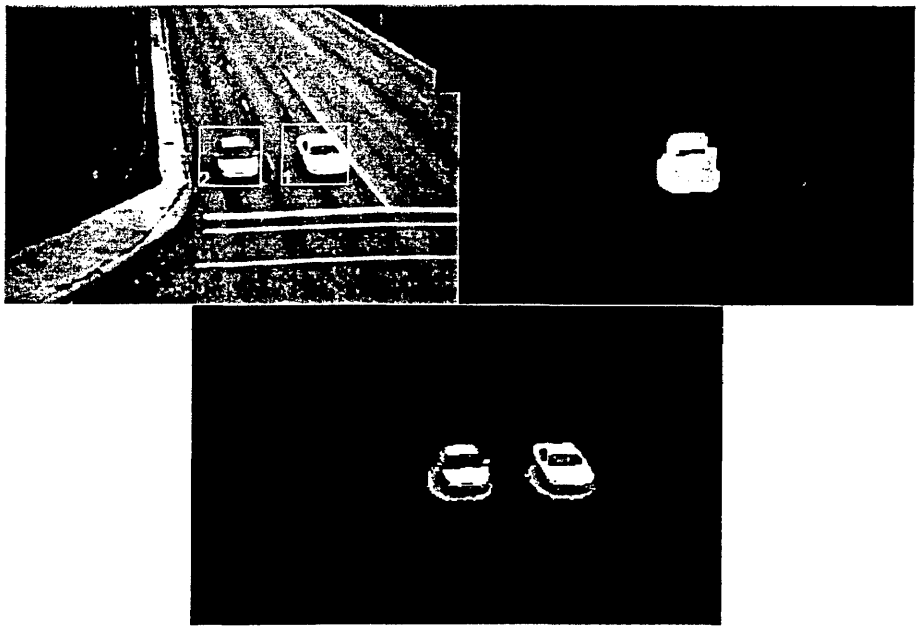

Figure 43: Output from detector

5.2 Minimum distance classifier

The classification method used in this thesis is a minimum distance classifier. Each vehicle in the previous image has a group of features, which includes

(1) Ratio of width and height: could be obtained from detected image data. For example in following image, the ratio of width and height of the left car is the ratio of X and Y.

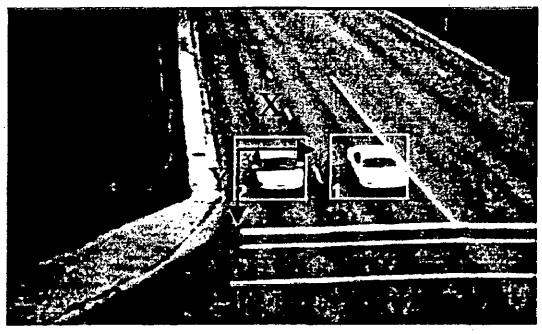

Figure 44: Ratio of width and height

(2) Shape information and average intensity value: This information could be obtained from difference image data. We scan each detected object according to the data of the labeled image and calculate shape and intensity of individual object in the difference image.

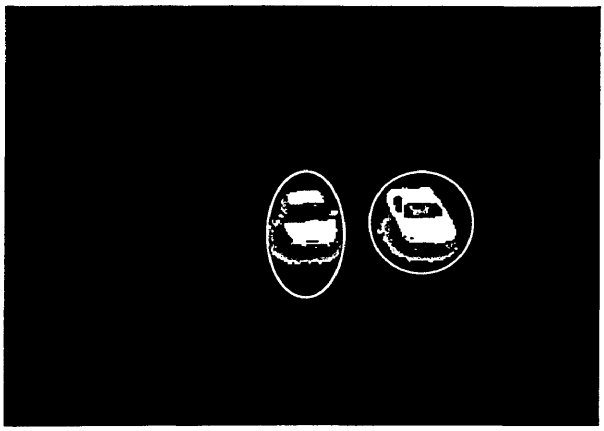

Figure 45: Shape and intensity information

The difference image is gray level image. If we add all pixel intensity values and divide by the number of pixels, we can get average intensity information of an individual object. We use the triple line to represent the shape of vehicle as shown in figure 46.

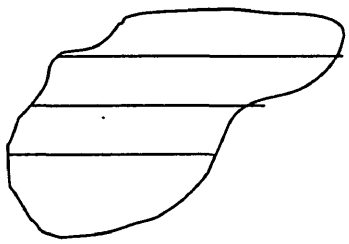

Figure 46: Triple-line

(3) Position and size information could be easily obtained from the labeled image, detect image and difference image.

These features can be generally modeled as an N-dimensional feature vector. The feature vector of each vehicle in the current image is compared with each feature vector of the previous image. The comparison with the minimum distance above a certain threshold will be considered as a match. Suppose each feature vector is expressed as:

$$X = \begin{bmatrix} X1 \\ X2 \\ . \\ Xn \end{bmatrix}$$

According to decision theory, if a vehicle X belongs to class i, then for decision function d(x), $d_i(x) > d_j(x)$; j=1,2,3...m, $j \neq i$. With a minimum distance classifier the value of the decision function is the maximum when the Euclidean distance is the minimum value.

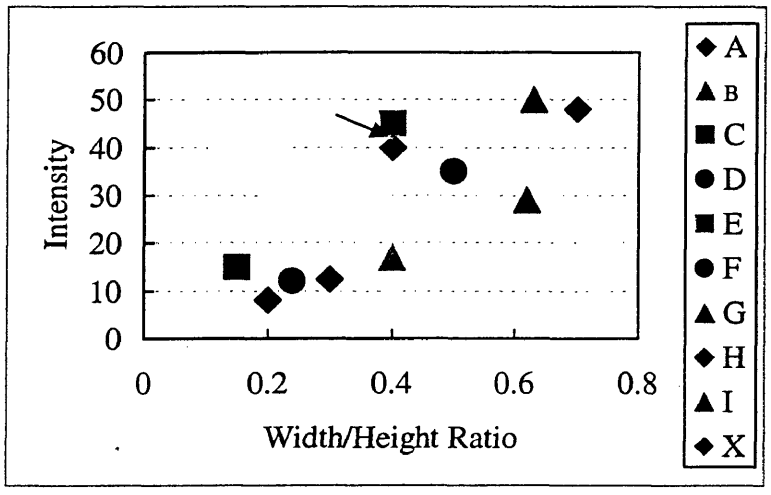

Figure 47: Classification

In the above figure 47, vehicle X will be considered to match with vehicle E because there is a minimum distance between X and E and the distance is above a certain threshold.

5.3 Feature Registers

As described above, each vehicle object holds a vector of features such as shape information, intensity value and size. The register extracts these features from input data and stores these feature vectors in a list. Two such lists are generated for two comparable image frames.

For example, suppose $p_{dif}(x,y)$ represents a pixel in position (x,y) in a difference image and $p_{lab}(x,y)$ represents a pixel of position (x,y) in a labeling image. A detected image provides a vehicle space for each vehicle in form of a rectangle frame. Basically, the register scans the difference image within a certain rectangle space pixel by pixel. If $p_{dif}(x,y)$ is greater than 0 and the label of $p_{lab}(x,y)$ is k, then the size of vehicle k is increased by 1. After all pixels within the rectangle are touched, the next rectangle space is processed. The total intensity of a vehicle is calculated in the same manner. Then the average intensity value is obtained by dividing total intensity by the vehicles size. Shape information such as width/height ratio is obtained the same way.

After scanning the entire image, a set of features is extracted for each vehicle. They are:

- Vehicle size: Total number of non-zero pixels within a rectangle space.
- Vehicle average intensity: Intensity/vehicle size.

- Width/height ratio: width/height.
- Other information such as shape and position.

Figure 48 shows two images with the vehicle features identified and marked in the images.

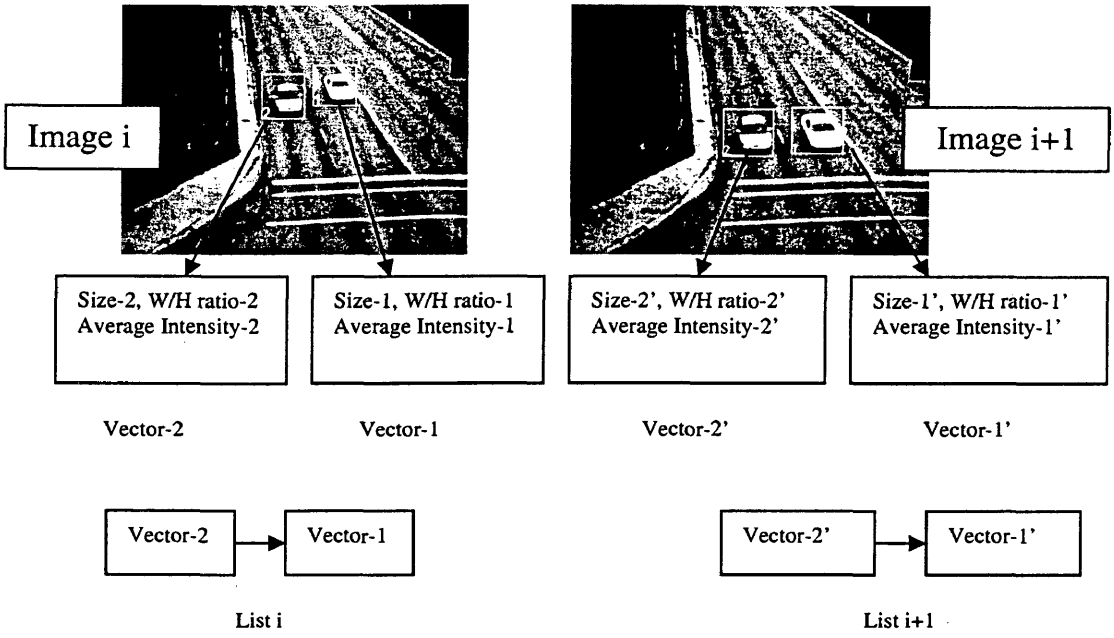

Figure 48: Tracking system

Features are packed into vectors by the register and then the feature vectors of an image are connected together to form a list of vectors.

5.4 Video handoff and vehicle tracking

Feature vectors in list i are to be classified in terms of a set of classes:

$$S= \{C_1, C_2, \ldots C_i\}.$$

Each feature vector of list i+1 is compared with C_i of list i. The similarity between feature vector V_i of image i+1 and C_i of image i is evaluated by a weighted distance.

As a measure of distance in feature space, a natural choice is the usual *Euclidian* distance: $d_e(\vec{x},\vec{y}) = \sqrt{\sum_i (x_i - y_i)^2}$, or the *weighted Euclidian* distance:

$$d_{we}(\vec{x},\vec{y}) = \sqrt{\sum_i \lambda_i^{-1}(x_i - y_i)^2}$$

In our tracking system, average intensity value is given a heavier weight since average intensity is more adaptive to changes of size, distance and orientation. Shape has a smaller weight because when a big orientation happens width/height ratio will also change. Height value and width value we used in this thesis are 2D values that are not exactly the same as in 3D real world. Each comparison gives a score to indicate the similarity between them. The match score must be greater than a certain threshold value because otherwise the minimum distance algorithm will return a pair of vector with minimum distance no matter how far they are. When the score is less than the threshold it will return zero, otherwise it will return the score value. A matrix of scores is built after all comparisons are completed.

$\rightarrow V_i$ $\downarrow C_i$

S_{00}	S_{01}	S_{02}	...	S_{0i}
S_{10}	...	...	...	...
S_{j0}	S_{j1}	S_{j2}	...	S_{ji}

Figure 49: Score matrix

A maximum score of a column indicates that a minimum distance is found between C_i and V_i, thus V_i is associated with C_i, which means they match. V_i in this

case will be assigned the same label as C_i. If there is no such match (for example, all score in column are less than threshold value therefore are all zero), this indicates that V_i is a newly entered vehicle. A new label will be assigned to this vehicle in this case.

5.5 Tracking system evaluation

In this section, we present the results and discussions on the experiments with real images to evaluate the performance of vehicle tracking and handoff.

5.5.1 How to count correctly tracked vehicles

We compare two successive image frames, the current image and the past image. Suppose there are n vehicles in the current image which includes x newly entered vehicles and y vehicles inherited from the past image. Then we have: n = x + y. If there are x' vehicles are correctly tracked as new entered vehicles and y' vehicles are correctly matched with vehicles in the past image, then the tracking rate between these two successive images is: (x'+y')/(x+y). For an example in figure 50, there are 2 vehicles in the current image (right image).

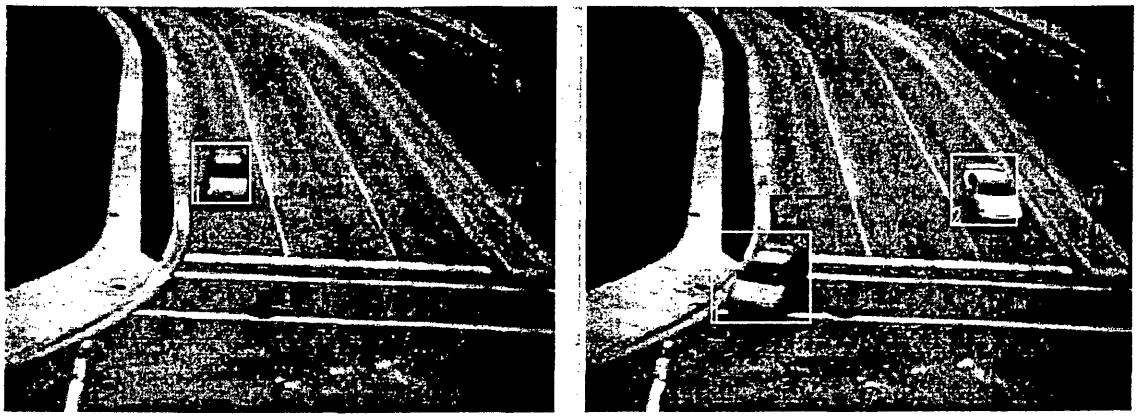

Figure 50: tracking performance

The white vehicle is new entered vehicle. The gray one is an inherited vehicle. The white vehicle is correctly tracked as a new car, labeled with "2", a number which is larger than any number in the past image (the left image). The gray car is correctly matched with an inherited car from the past image, and is labeled with number "1", which is the same number as the car in the past image. The total tracking rate for these two images is: 2/2=100%. Both two cars in the second image are correctly tracked.

5.5.2 How to match vehicles using minimum-distance classifier

For each vehicles in the vehicle set of past image $V_p\{P_1, P_2, ...P_m\}$ and the vehicle set in current image V_c $\{C_1, C_2,...C_n\}$, the average intensity value and shape information are registered as F $[W_1F_1, W_2F_2]$. F_1 is intensity value and F_2 is shape information. W1 is a weight of intensity value in the tracking system. W_2 is a weight of shape information in the tracking system. The algorithm calculates the difference of F between each vehicle of V_c and each vehicle of V_p. The vehicle pair with smallest F difference above a certain threshold is considered as a match. In our tracking system, the weight of intensity value W_1 is assigned a value 0.9 which is heavier than the weight of shape. All other shape features share the remaining weight of 0.1. This is because the intensity value is more adaptive to orientation and position change than shape.

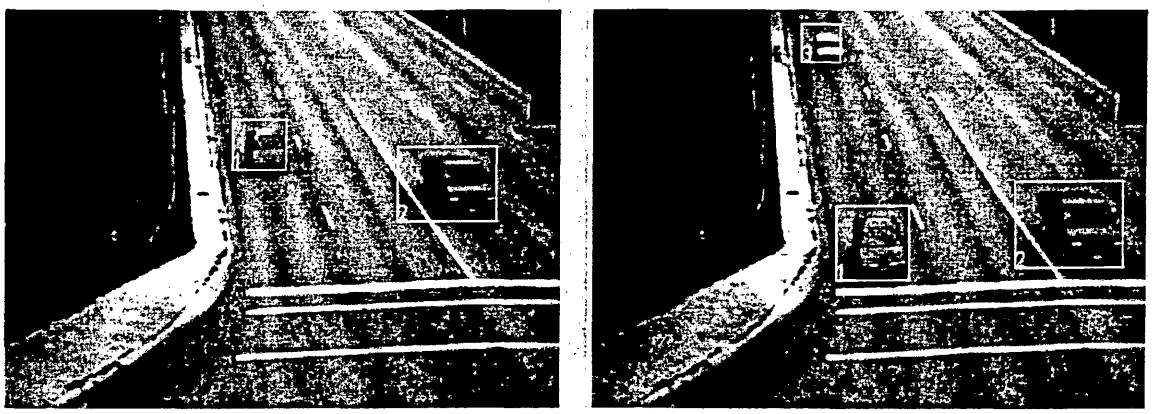

Figure 51: tracking strategy

This tracking system is theoretically able to handle orientation changes and position changes, such as the gray color car with orientation change in figure 49 and gray color cars with position change in figure 51.

5.5.3 Some tracking strategies

As shown is figure 52: The very small car in right image is ignored.

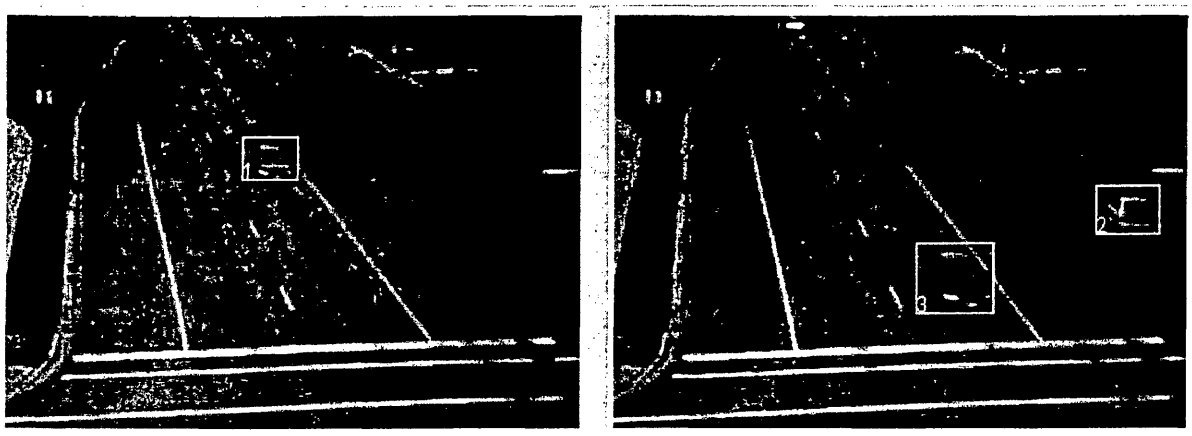

Figure 52: Tracking threshold

In tracking, if two vehicles are connected together, then they may be mistakenly considered as one vehicle. If one vehicle is divided as several parts, as long as part of it is recognized, consider it as a correct tracking as shown in figure 53.

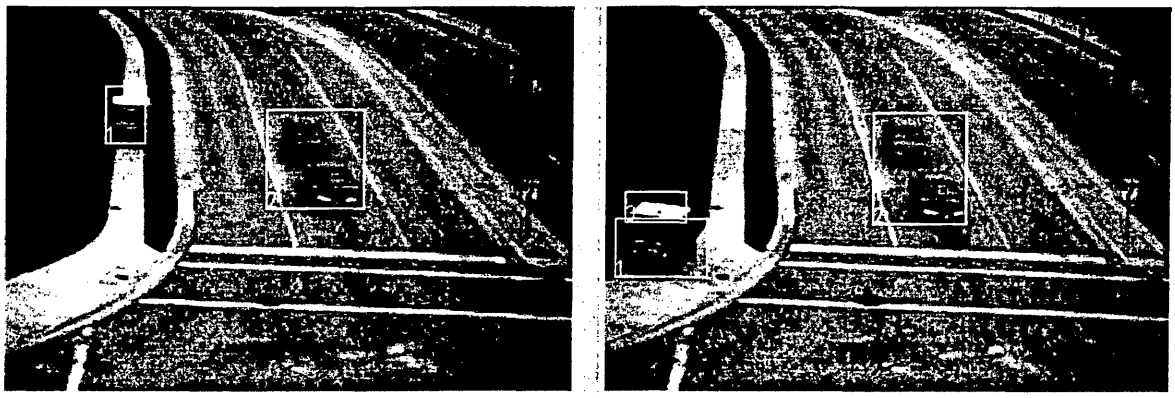

Figure 53: Connected vehicles

Noise coming from false positive vehicle detection will not be considered in tracking. As shown in figure 54: Noise labeled as "4" in the right image is due to false positive detection.

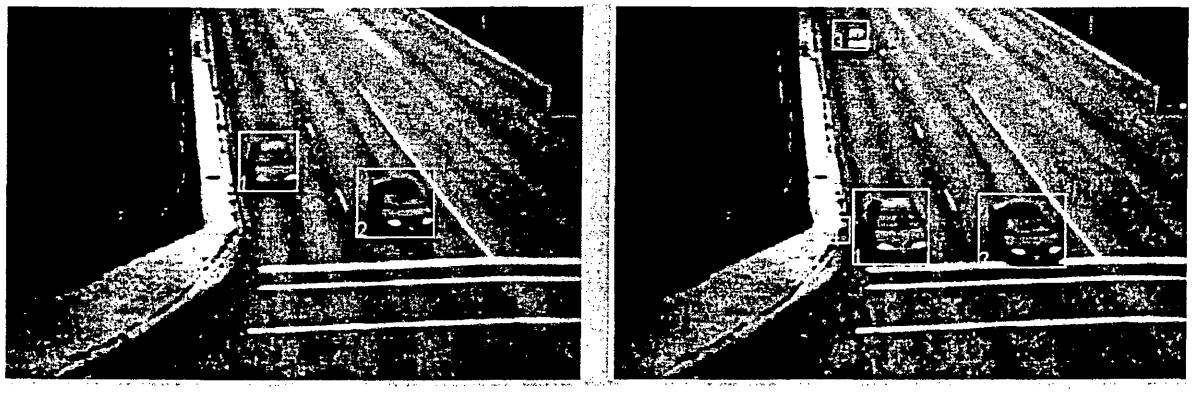

Figure 54: Noise from detection

5.6 Performance and Discussion

This tracking system is good though not perfect. It has an overall tracking rate of 74.45%. It correctly tracked 303 vehicles out of 407 vehicles but fails in 25.55% cases. The following analysis will illustrate why it fails and in which situation it most likely will fail.

1. Scenario 1. Sometimes, vehicles switch lights. Like the gray color car in figure 55.

Figure 55: Scenario 1

This will affect tracking. Also, when vehicles partially disappear from view, like the white car in figure 55 the tracking system cannot handle it. This occurs because in the first situation, the intensity value is totally changed, while in the second situation both intensity value and shape are totally changed.

2. Scenario 2. Sometimes the tracking is confused by some very similar vehicles as shown in figure 56:

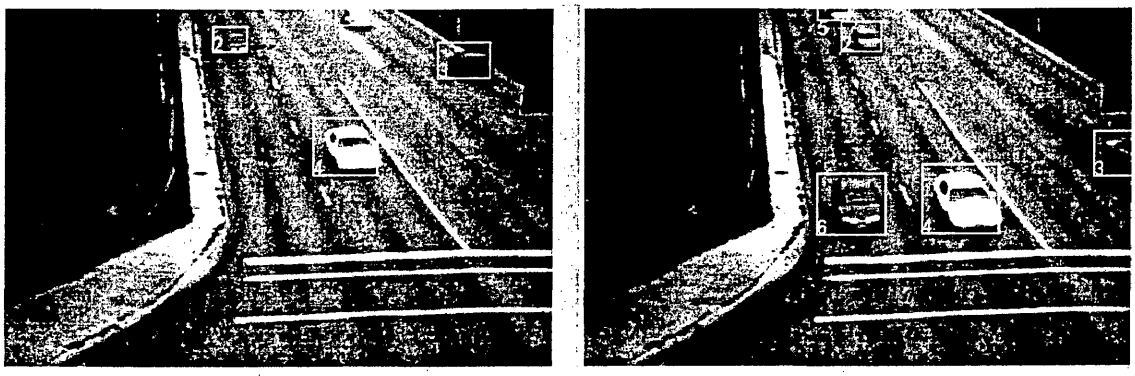

Figure 56: Scenario 2

Vehicle 6 in right is actually vehicle 2 in left, but mismatched as a new entered vehicle because the vehicle 2 in right is very similar with vehicle 2 in left.

3. Scenario 3. When vehicle is small and not clear, as shown in Figure 57.

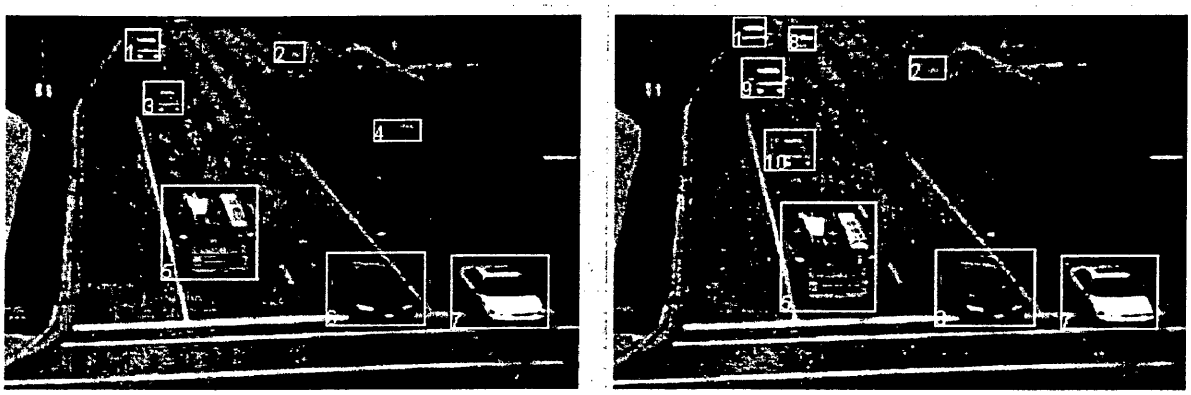

Figure 57: Obscure situation

The information about intensity value and shape are very obscure. In these situations, the tracking result is not good.

4. Scenario 4. Orientation change cannot be handled perfectly especially when the orientation change causes big intensity value change and shape change. As shown in figure 58. Vehicle "2" in the left image is mismatched as "4" due to orientation change while other vehicles are correctly tracked.

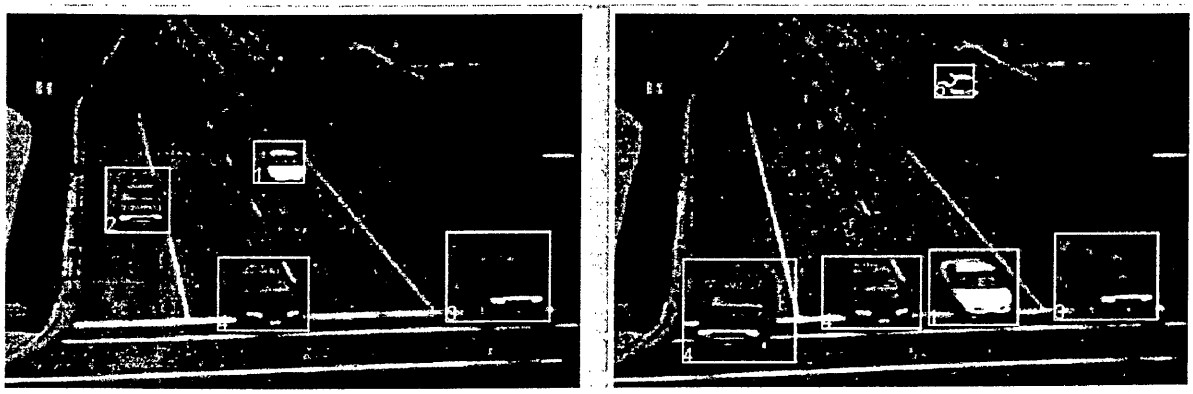

Figure 58: Orientation change

6. Conclusion

6.1 Development of the median models

In this thesis, we proposed both a median model and an improved median model to generate background reference images in order to handle slow moving and stationary vehicles in real time system. The median model for background generation is an idea inherited some of Y.H.Yang's previous work [10] and current researches in mix-Gaussian model.

In Y.H.Yang's research (1992) [10] [11], he tried to find a smooth region in a pixel sequence and consider this smooth region as background pixel. Compared with Y.H.Yang's method, median model proposed here is not only faster in terms of computational time but also more adaptive to slow moving vehicles.

Currently mix-Gaussian is the dominant method used in background subtraction. The advantage of mix-Gaussian is that it could potentially identify shadows. Our median model is not an alternate to mix-Gaussian. On the contrary its mathematical fundamental is based on a single Gaussian distribution assumption and we'd rather view it as a complement to the mix-Gaussian model. The median model is a cheaper way to implement the Gaussian model. Because of this, it may be used in mix-Gaussian in the future to reduce the time complexity. This is essential to real time image processing.

In this thesis we proposed a shifted-median model and a balanced-median model. Based on these two median models we keep track deviation of median and its

neighbors in a reordered pixel sequence. When sample size is large enough, the reordered pixel sequence is in balanced-median model. This model is indicated by a very small deviation value of the median and its neighbors. In this case the median of the pixel sequence falls in background set and could be used for background estimation. When the sample size is not large enough, the reordered pixel sequence is in shifted-median model. This model is indicated by a much bigger deviation value of median and its neighbors. In this case the median falls out of the background set and cannot be used for background estimation.

This median model has an impressively better performance to handle slow moving or even stationary vehicles than traditional time average or exponential forgetting methods. Besides, time complexity is acceptable for real time image processing in improved median model. The improved median model is proposed to reduce the time complexity to a reasonable level. In improved median model, we take samples in a longer time interval to make it capable of dealing with slow moving and stationary vehicles. The sample size is a small constant value between 5 and 20. This small sample constant size could dramatically reduce the time complexity from $\sum_{1}^{P} K_i S_i \lg S_i$ to $\sum_{1}^{P} S \lg S = pk$.

As a complement to this improved median model, a mask-classified updating method is introduced to update the background image in a short term and only classified background pixels are being used for updating. The combination of the improved median model and mask-classified updating has several advantages:

(1) Adaptive to fast or slow illumination change and weather condition change without deadlock situation in mask updating.

(2) Can handle slow moving or even stationary vehicles.

(3) Time complexity is better so it could be used in real time image processing.

(4) Less false positive, less false negative alarm.

6.2 Vehicle detection performance improvements

Background generation is the first stage for vehicle detection and tracking. The generated background reference image is output to the next stage for vehicle detection. The biggest challenge for detection is noise. Several techniques such as threshold, dilate and erode are used for noise removal. When performing these techniques with manual supervision, we can adjust the operator parameters to the best performance state and obtain a 100% detect rate and 0% false positive result. All noises could be removed by using these techniques; however, it is not a robotic solution. An automatic detector is developed to handle noise and detection in a robotic way. When these operations are put in an assembly line and the detector allowed to go through the assembly line automatically no human supervision activity or decisions are involved.

Automatic detector cannot completely remove noises because the assemble line are a series of fixed operations done without effective check and feedback. It cannot adjust its parameter to get the best performance. The automatic detector will pass noises down to next stage anyway. If there is no consideration about these noises, the subsequent tracking system will encounter big difficulty. In order to reduce such

noise to a reasonable level we apply threshold on objects detection according to their sizes. When making a detection decision, an object will be considered as a vehicle and will be labeled only when its size is large enough. If its size is too small, it will be considered as noise and will not be labeled. Threshold on size not only allows us to ignore noises even though they are not removed thus, but also benefit the tracking system. When vehicles are far away from cameras, they usually have a very small size and their shape and intensity are very obscure. These obscure vehicles may confuse the tracking system and cause it make a wrong decision (such as error match). By ignoring these obscure vehicles the tracking system performance will be boosted. The overall performance of automatic detector is satisfactory. A 97.6% detect rate and 2.4% false positive rate of the automatic detector are observed in experimentation with real time traffic images.

The detector outputs individual vehicle information to the tracking stage for further vehicle tracking and video handoff. The tracking stage basically needs to match vehicles between two images. For vehicle tracking these two images may be successive images while for video handoff they may have a longer time interval since different cameras take them. The information passed from detector to tracking system includes: (1) a pair of difference images; (2) a pair of object masks; and (3) a pair of detected images with vehicles being labeled. The tracking system first extracts features such as vehicle size, vehicle intensity and vehicle shape from these input images and builds up a feature vector for each vehicle. Then vehicle match problem is modeled as a minimum distance classifier problem. Each feature vector in image i

is a unique class. What we need to do is to assign such a class C_i in image i to a vehicle V_i in image $i+1$ according to the similarity between them. A score matrix is used to display such similarity. Each pair of C_i and V_i gives us a similarity score by calculating the normalized Euclidian distance between them. In the Euclidian distance each feature has a different weight. We assign a heavier weight to the vehicle average intensity since we think it is most likely invariant to distance change and orientation change. Since a minimum distance classifier will return a class anyway no matter how far C_i and V_i are, we need to threshold the similarity score before going through minimum distance classifier. All similarity scores less than a certain threshold value will be assigned a value of zero. The classifier then scans the revalued score matrix, associates C_i with V_i if their similarity score is the maximum score in the column of C_i. In this case V_i will be considered as the same vehicle as C_i. If there is no C_i found for V_i, V_i will be considered as a new entered vehicle. The performance of this tracking system is satisfactory, a tracking rate of 74% is observed.

6.3 Summary and directions of future research

In conclusion, the median model and improved median model we proposed in this thesis has been successful in handling slow moving or even stationary vehicles. The improved median model combined with mask-classified updating could boost the performance speed dramatically as well as maintain the same advantages of the standard median model. Both methods provide satisfactory experimental results and therefore could be used as alternative methods in addition to current background

subtraction methods. For vehicle tracking and video handoff, we modeled the vehicle-tracking problem as a minimum distance classifier problem and obtained a satisfactory result.

We observed that vehicle shadow is a problem in both detector and tracking system. Shadows can change the shape of a vehicle, or can connect two vehicles together thus cause both detector and tracking system hardly to make a right decision. We suggest a 3-Gaussian model or HMM-Based segmentation method [26] to handle this situation in the future works.

7. References

1. 2002 Intelligent Transportation Systems (ITS) Projects Book, HTML Version, www.its.dot.org
2. Masataka Kagesawa, shinichi Ueno and Katsushi. "Recognizing vehicles in infrared images using IMAP parallel vision board." IEEE transactions on intelligent transportation systems, vol 2, No 1, March 2001
3. Ivan, John N and Shyuan-Ren Chen. "Incident detection using Vehicle-Based and fixed-location surveillance." Journal of Transportation Engineering May/June 1997:200-215
4. Gangisetty, Ramesh and Douglas W. May. "Traffic and Incident Management System for I-95 in the Philadelphia Area" ITE Journal February 1995: 37-44
5. Sobhi, Nazemeh and Micheal J.Kelly. "ATMS Human factors experiments produce design guidelines." Public roads spring 1997:7-9.
6. E. Jones and Q.Zhu. *Development of computer vision system for vehicle recognition and tracking, Technique report, Department of Civil Engineering, University of Nebraska at Omaha, 2000.*
7. Giadiv, Determining 3-dimentional motion and structure from optical flow generated by several moving objects. IEEE Trans.Pattern.Anal.Mach.Intell, 7. 384-401, 1985

8. Hans-Hellmut Nagel. *On the estimation of optical flow: Relations between different approaches and some new results.* Artificial Intelligence 33, pp. 299-324, 1987.

9. Alan C. Bovik, D. Gibson, *Handbook of Image and Video Processing,* Academic Press, Inc, March 2000

10. W.Long and Y.H.Yang. "Stationary background generation: an alternative to the difference of two images." Pattern recognition, 23:1351-1359, 1990

11. Y.H.Yang and M.D.Levine. "The background primal sketch: An approach for tracking moving objects." Machine vision applic, 5:17-34, 1992.

12. K.P.Karmann and A.Von Brandt. "Moving object recognition using an adaptive background memory." Proc.Time-varying Image processing and moving object recognition, vol 2 1990: 289-296

13. Surendra Gupte, Osama Masoud and Robert F.K. "Detection and Classification of vehicles." IEEE transactions on intelligent transportation systems, vol 3 no1, March 2002

14. P.L.Rosin and T.Ellis. "Detecting and classifying intruders in image sequences." British machine vision conf, 1991: 2057-2060

15. W.E.L.Grimson and etc, "*Adaptive background mixture models for real-time tracking*", in CVPR, 1999.

16. J.Rittscher, J.Kato, S.Joga and A.Blake. "A probabilistic background model for tracking." 6th Eur. Conf. Computer Vision, Vol 2, 2002:338-350

17. N.Friedman and S.Russell. "Image segmentation in video sequences: a probabilistic approach." 13th Conf. Uncertainty in artificial intelligence, RI, 1997

18. Y.Raja, S.J.Mckenna and S.Gong. "Tracking color objects using adaptive mixture modles." Image vision computer. No 17: 225-231, 1999

19. Ahmed Elgammal, Ramani Duraiswami, David Harwood and Larry S. Davis. "Background and forground modeling using nonparametric kernal density estimation for visual surveillance." Proceedings of the IEEE, vol 90, no 7, Jul 2002:1151-1163

20. A.Elgammal, D.Harwood and L.S.Davis. "Nonparametric background subtraction." 6th Eur,Conf.Computer vision. Vol 2,2002, 751-767

21. Rafael C.Gonzalez, Addision-Wesley, *Digital image processing*, p42, 1992

22. A. K. Jain, *Fundamentals of Digital Image Processing,* vol. IP. Englewood Cliffs: Prentice-Hall, 1989.

23. Michael Seul, Lawrence O' Gorman, Michael J. Sammon, *Practical Algorithms for Image*, Cambridge University Press, August 1999

24. Liang Zhao and Charles E.Thorpe. "Stereo- and neural network based pedestrian detection." IEEE transactions on intelligent transportation systems, Vol 1, No 3, September 2000:148-154

25. P.L.Rosin and T . Ellis. *Detecting and classifying intruders in image sequence.* In British Machine Vision Conf, pages 293-300, 1991.

26. Jien Kato. An HMM-Based segmentation method for traffic monitoring movies. IEEE transactions on pattern analysis and machine intelligence, pages 1291-1296, Vol 24, No 9. September 2002.