http://dx.doi.org/10.5755/j01.eee.18.8.2634

Optimization of Image Processing in Video-based Traffic Monitoring

Fei Zhu[1], Jiamin Ning[1], Yong Ren[2], Jingyu Peng[2]
[1]School of Computer Science and Technology, Soochow University,
Suzhou, Jiangsu, 215006, P.R.China
[2]Applied Technology College of Soochow University,
Suzhou, Jiangsu, 215325, P.R.China, phone: 86 521 65113205
zhufei@suda.edu.cn

Abstract—The video-based traffic monitoring systems have been widely used. The system usually reads real time monitoring video and converts them into images for processing. However, such systems are often limited by image processing algorithms and don't behavior as expected. We hereby study optimization approaches for different image processing stages. As getting a binary image is important and fundamental to image processing, we first present an adaptive thresholds approach for binary conversion. The approach chooses thresholds according to each pixel and its space information. Then we introduce a three-dimension Gaussian filter for denoising. We also propose a Gaussian model based background generation approach so that we can update background at any time, thus getting rid of a major limitation of background subtraction approach. Moving objects shadow detection and removing mechanism is also added in. We put forward a burst noise eliminating algorithm that uses several continuous frames to wipe off sudden noise in real world monitoring. Finally, we assert that removal of the reflection light band is to alleviate negative effect from high-energy reflection light. We conclude that adding in white light will reduce color saturation. In the area affected by reflection light band, there are much more white light involved than other area. Based on the conclusion, we use Gaussian model and an improved four direction Sobel operator to achieve reflection light band elimination from image.

***Index Terms*—Digital images, image processing, image quality, image color analysis.**

I. INTRODUCTION

The rapid development of image processing facilitates the application of video-based monitoring systems in different domains and areas. Video-based traffic monitoring system, a typical kind of widely used monitoring system, can achieve automatic surveillance by video obtained from monitoring system [1]. However, many video-based traffic monitoring systems, although their convenience and low cost, don't perform as well as expected.

Most video-based traffic monitoring systems involve image processing, including image graying, binary conversion, and denoising [2]. Image processing is an essential part to the monitoring system. Optimizing image processing is able to improve monitoring effect. We hereby focus on optimization of the video-based traffic monitoring system, mainly in image processing optimization. We will talk about optimization for image processing in different phrases of processing routine. In the first section, we present an adaptive thresholds approach for binary conversion. Next, we will introduce a three-dimension Gaussian filter to remove noise in the image. As a most popular approach, background subtraction, although being effective, is usually limited by background real time detection. We take advantage of a Gaussian model to generate background with which we can refresh the background dynamically at any time. We also put forward a burst noise eliminating algorithm to remove its interference. Finally, we use a novel approach to erase reflection light band which tends to disturb monitoring badly.

II. BINARY CONVERSION WITH ADAPTIVE THRESHOLDS

As color information of an image is not so much of use, images are usually converted to binary ones. Image binary conversion is so important that sometimes it largely determines affect final result. In a binary image, typically black and white are used for each pixel. White is often used to represent foreground objects and black is for background ones.

The simplest method of binary conversion is by threshold [3]. Each pixel is set to 1 if its gray value is not less than some threshold or otherwise set to 0, as (1)

$$f(x,y)=\begin{cases}1, & g(x,y)\geq threshold,\\ 0, & otherwise,\end{cases} \tag{1}$$

where $g(x, y)$ is the gray value of pixel (x, y).

We can see that threshold is a key to carry on image segmentation. An ideal threshold is the one that remains as much feature information as possible [4]. Adaptive threshold, which take the position information of the pixel into consideration, is a better solution than a single predefined threshold. We carry out image binary conversion with adaptive thresholds method, as follows [5].

Step 1: Segment the image into foreground part and background part with (2). Set each pixel of which grey value is less than $threshold_1$ to 0; otherwise it will be set to 1

Manuscript received March 19, 2012; accepted May 7, 2012.
This work was supported by the natural science foundation of Jiangsu Province under Grant No. BK2011375.

$$threshold_1=\sum_{i=1}^{N}L_iP\left(L_i\right)=\frac{\sum_{i=1}^{N}L_ih\left(L_i\right)}{\sum_{i=1}^{N}h\left(L_i\right)}, \qquad (2)$$

where $h(L_i)$ represents the occurrence of the gray value L_i, and $P(L_i)$ is its occurrence probability.

Step 2: Get *threshold₂* by (3). Set each pixel of which gray value is less than *threshold₂* to 0; otherwise set it to 1:

$$\begin{cases} u= \begin{array}{l} f^2\left(x-1,y\right)+f^2\left(x-1,y-1\right)+f^2\left(x,y-1\right)+ \\ +f^2\left(x,y\right)+f^2\left(x+1,y-1\right)+f^2\left(x+1,y\right)+ \\ +f^2\left(x-1,y\right)+f^2\left(x+1,y+1\right)+f^2\left(x,y+1\right), \end{array} \\ v= \begin{array}{l} f\left(x-1,y\right)+f\left(x-1,y-1\right)+f\left(x,y-1\right)+ \\ +f\left(x+1,y-1\right)+f\left(x+1,y\right)+f\left(x,y\right)+ \\ +f\left(x-1,y\right)+f\left(x+1,y+1\right)+f\left(x,y+1\right), \end{array} \\ threshold_2=\dfrac{\sqrt{u}}{v+\theta}, \end{cases} \qquad (3)$$

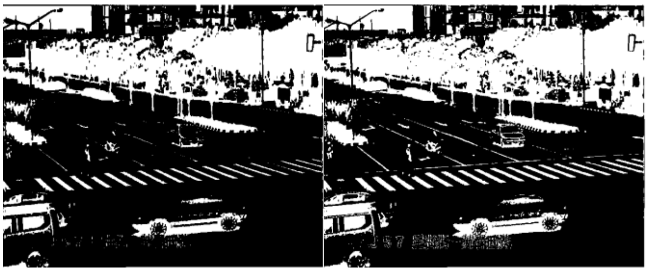
where $f(x,y)$ is the gray value of pixel (x,y) and θ is minor sign less integral.

Step 3: Get an adaptive window A of pixel (x,y). Get a local adaptive *threshold₃*. Set each pixel of which gray value is less than *threshold₃* to 0; otherwise it will be set to 1

$$threshold_3=\frac{\sum f\left(x,y\right)}{M+\theta}, \qquad (4)$$

where M is the total pixel number of A.

The following two figures are converted by threshold. The left one is processed by a single predefined threshold method and the right one is processed by our adaptive thresholds method. We can see the segmented objects in the right image are distinct, but they are fuzzy in the left one, especially for the vehicle in the center of the image.



a) b)

Fig.1. The left one is processed by conversional single threshold method and the right one is processed by adaptive thresholds method. Obviously the right one is more distinct than the left one.

III. DENOISING WITH THREE-DIMENSION GAUSSIAN FILTERING

Binary converted images usually have lots of noise. Hence we need image denoising to reduce negative effect caused by noise. There are many commonly used algorithms, such as neighborhood averaging, median filtering and Gaussian filtering.

A. Neighborhood Averaging Filtering

Neighborhood averaging filtering (NAF) [6] assigns average gray value of a pixel and its neighborhood pixels to the pixel, as (5)

$$g\left(x,y\right)=\frac{1}{M}\sum_{x,y\in S}f\left(x,y\right), \qquad (5)$$

where S is coordinate set of neighborhood pixels, and M is total number of pixels in S. NAF approach works good in most cases but it tends to make image fuzzy.

B. Median Filtering

With median filtering (MF) [7], gray value of each original pixel will be replaced with the median value of a predefined window W. (6) shows how to get median value. MF approach can keep details of image and protect edge from being fuzzy. However the image is still possible to get fuzzy.

$$g(x,y)=median\{f(x-k,y-l),k,l\in W\}. \qquad (6)$$

C. Gaussian Filtering

Gaussian Filter [8] is a linear smoothing filter whose weight relies on shape of Gaussian function. Gaussian filter works especially efficient in removing the noise with normal distribution (Gaussian distribution) [8].One-dimension Gaussian function is as (7)

$$g\left(x\right)=e^{-x^2/2sigma^2}, \qquad (7)$$

where *sigma* determines width of Gaussian function.

D. Three-dimension Gaussian Filtering

In our experiment [9], we use a three-dimension Gaussian filter (3DGF), with the kernel function:

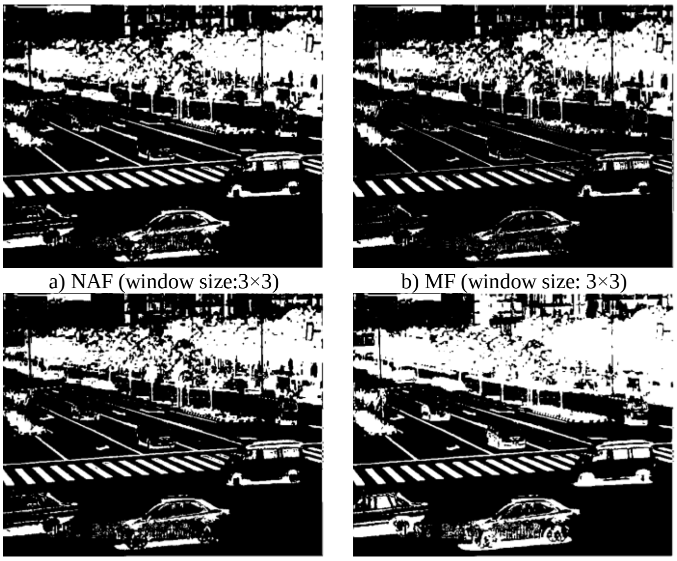
$$\begin{cases} W=1+2\times\lceil 3\times sigma\rceil, \\ f\left(x\right)=\dfrac{1}{\sqrt{2\pi sigma}}e^{-\frac{1}{2}\left(x-W/2\right)^2/sigma^2}, \\ g\left(x\right)=\dfrac{f\left(x\right)}{\sum_{i\in W}f\left(i\right)}, \end{cases} \qquad (8)$$

where W is the width of Gaussian function. Data context window [-3×sigma, 3×sigma] covers most filter coefficient.

We tested filtering using the three approaches with different parameters. Table I shows the processing time, and Fig. 2 shows filtering effects.

TABLE I. PROCESSING TIME OF THREE APPROACHES WITH DIFFERENT PARAMETERS. SIZE OF TESTING IMAGE:704×576. CPU:INTEL CORE I5-2300 2.8GHZ. MEMORY: 8GB.

Approach	**Time(s)**
NAF(3×3)	0.125
MF(3×3)	0.226
GF(sigma=0.0625)	0.075
3DGF(sigma=0.0625)	0.076



a) NAF (window size:3×3) b) MF (window size: 3×3)

c) GF (sigma=0.0625) d) 3DGF (sigma=0.0625)

Fig. 2. Denoising effects of three approaches with different parameters.

As we could see, GF had least processing time and 3DGF took second place. As for the effect, 3DGF, the best one, identified the cars as well as lines of traffic lanes in the center of images; while two cars and some part of the traffic lanes were so indistinct in resulting image by GF. In general, 3DGF had the best cost efficiency.

IV. Dynamic Background Refreshing

Background subtraction is a practical approach to detect moving object in video monitoring [10]. However, in practice, environmental factors will disturb monitoring. For example, sun light illumination variation and different weather conditions make environmental backgrounds so different. As a result, conventional background subtraction without background refreshing periodically and dynamically usually does not work well in real world. We here introduce a dynamic background refreshing method so as to alleviate impact from environment.

A. Background Detection

We find the background of monitoring video, taken as a whole, tends to change gradually within a certain time period rather than suddenly and sharply. The changed pixels generally pertain to a Gaussian distribution with mean value μ and variance σ. We hereby utilize Gaussian function to model different states of a pixel throughout all time periods. Consequently, given time t, the states of pixel *(x, y)* are a time sequence vector. Each pixel is subject to Gaussian distribution

$$P(z_t)=\frac{1}{\sigma\sqrt{2\pi}}e^{\frac{-(x-u)^2}{2\sigma^2}},$$
$$z\in\{I(x,y,k),x=1,\cdots,M;y=1,\cdots,N;k=1,\cdots,t\},\tag{8}$$

where *I* is an image sequence and (*x*, *y*, *k*) represents the pixel (*x*, *y*) of frame *k*.

We use the model to fit the new coming frame of pixel. Given a pixel, if the values of H (hue), S (saturation), and V (brightness) components [11] are all between μ and σ, it will be set as a background pixel; otherwise it is a foreground one. However, in HSV [11] model, if the value of S component is less than some threshold, the image will be regarded as a monochromic one. As a result, H component becomes not so useful and color precision of the image reflected by component of S drops with decreasing of V component value. Hence, we can see that H component is only involved in some dedicated cases, S component is used to exam whether the image is chromatic or monochromic, and V component is essential all through the processing.

Let H_1, S_1, V_1 be the H, S, and V component value of a pixel respectively; let H_2, S_2, V_2 be the H, S, and V component value of a new coming pixel respectively; let S_T be a threshold of saturation; let σ be the standard deviation. We can use the following criterion to examine whether the coming pixel is a background one or a foreground one. There are altogether four kinds of cases:

Case 1: if both S_1 and S_2 are less than S_T, we will check whether variation of *V* component is not more than variance σ. If true, the pixel is regarded as a back one.

The condition and criterion are showed as follows:

Condition: $(S_1 < S_T)\quad AND\quad (S_2 < S_T)$;

Criterion: $|V_2 - V_1| \le \sigma$.

Case 2: if S_1 is less than S_T and S_2 is greater than S_T, we will check whether both variation of *V* component and variation of *S* component are not more than variance σ. If true, the pixel is regarded as a back one.

The condition and criterion are showed as follows:

Condition: $(S_1 < S_T)\quad AND\quad (S_2 > S_T)$;

Criterion: $\left(|V_2 - V_1| \le \sigma\right) AND \left(|S_2 - S_1| \le \sigma\right)$.

Case 3: if S_1 is less than S_T and S_2 is less than S_T, we will check whether both variation of *V* component and variation of *S* component are not more than variance σ. If true, the pixel is regarded as a back one.

The condition and criterion are showed as follows:

Condition: $(S_1 > S_T)\quad AND\quad (S_2 < S_T)$;

Criterion: $\left(|V_2 - V_1| \le \sigma\right) AND \left(|S_2 - S_1| \le \sigma\right)$.

Case 4: if both S_1 and S_2 are greater than S_T, we will check whether variation of *V* component, variation of *S* component, and value of S_2 multiply cosine the variation of *H* component are not more than variance σ. If true, the pixel is regarded as a back one.

The condition and criterion are showed as follows:

Condition: $(S_1 > S_T)\quad AND\quad (S_2 > S_T)$;

Criterion:

$\left(|V_2 - V_1| \le \sigma\right) AND \left(|S_2 - S_1| \le \sigma\right) AND \left(|S_2 \times \cos(H_2 - H_1)| \le \sigma\right)$.

After background detection, those unmatched pixels will be kept unchanged and those matched ones will be updated by the (9):

$$\begin{cases}\mu=(1-\alpha)\mu+x,\\ \sigma^2=\max\left(\sigma_{\min}^2,(1-\alpha)\sigma^2+\alpha(x-\mu)^2\right),\end{cases}\tag{9}$$

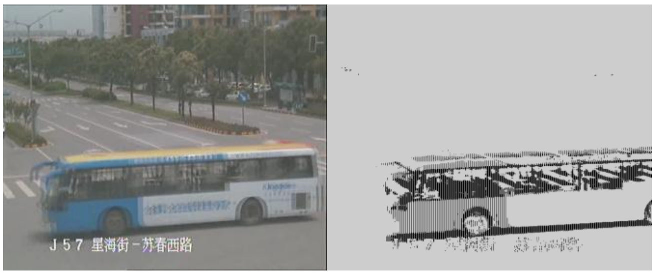
where σ_{min} is a threshold of noise, α represents the model learning rate, and *x* denotes the components of H, S and V.

As the initial distribution of pixels is unknown, we set the component values of the pixels in first frame to be the mean value of Gaussian distribution and variance to be zero. After that we save values of H, S and V components, as well as corresponding mean values and variance values. Then we will compare the values with the saved old values. Finally we will update the ones which have the same values as the saved old values, and leave alone others.

B. shadow detection

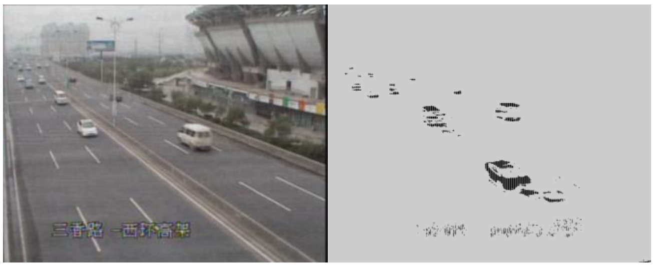
The shadow of the moving objects will make object separation component incorrectly detect object along with its shadow as moving object. If the subsequent processing is based on detected moving object, it will result in mistake [12]. Hence, we need to detect and remove the shadow of moving objects. We find that compared with the object part, the shadow part is darker and has less color. By this criterion, we put forward an approach to detect and remove the shadow of moving objects. We compare the value of the background pixel with that of shadow pixel. If the variation of H component value and the variation of S component value are both less than some given threshold, it can be determined as a shadow pixel. The criterion is showed as follows.

That is, give a pixel (*x*, *y*), if $\beta \le \dfrac{V_2}{V_1} \le \gamma$ and $\left|S_2 - S_1\right| \le T_s$, and $\left|H_2 - H_1\right| \le T_H$, we will set S component value of (*x*, *y*) to 1; otherwise we will set it to 0. Here T_s and T_h denote the thresholds of S component and H component respectively. As the V component value of the pixel in shadowed area is generally less than that of the pixel in unshadowed area, γ and β are less than 1. Meanwhile β is used to measure the strength of brightness. The greater β shows the brighter the light the pixel has. By this, we can remove the shadow of moving objects. The following figures show us the effect of getting separate moving objects with shadow removing.



a) b)

Fig. 3. The left figure is snapshot from monitoring system in urban road. The right figure is the greyed image after separating moving objects and removing it shadow.



a) b)

Fig. 4. The left figure is snapshot monitoring system in highway. The right figure is the greyed image after separating moving objects and removing it shadow.

The left two figures are snapshots from monitoring system in urban road and in highway. The right figures are corresponding greyed image after separating moving objects and removing it shadow. We can see that the moving objects are effetely identified and separated; the shadows are also removed.

V. BURST NOISE ELIMINATING

Burst noise, such as flying bird and falling leaves, is an unexpected and sudden interference from outside and usually exists for a very short time span [10]. Due to the characteristic that most of the burst noise merely exists in several continuous frames, we will take advantage of several anterior and posterior frames to remove the burst noise. First, we get subtraction of current frame F_n and previous frame Fn-1, namely Dif_t. Then we get subtraction of current frame and current background, namely Dif_b. We can see that Dif_b includes current moving objects and noise. As the time span between two continuous frames is very short, the burst noise would generally exist in both frames. Therefore we can remove the burst noise by their subtraction. Thus, *Dift* only has relative moving information. After that, we get moving objects and background frames Dif_t and Dif_b. The algorithm that segments moving objects from image with burst noise removing is shown as follows.

Step 1: Get frame subtraction Dif_t

$$Dif_t\left(x,y\right) = \left|F_n\left(x,y\right) - F_{n-1}\left(x,y\right)\right|. \quad (10)$$

Step 2: Get subtraction of current frame and background $Dif_{b.}$

$$Dif_b\left(x,y\right) = \left|F_n\left(x,y\right) - B\left(x,y\right)\right|. \quad (11)$$

Step 3: Given pixel (*x*, *y*), if Dif_b (*x*, *y*) and Dif_t (*x*, *y*) are both 1, it is regarding as a moving pixel and will be set to 1;otherwise it will be set to 0.

Step 4: Repeatedly deal with each pixel in the frame with step 1 to step 3.

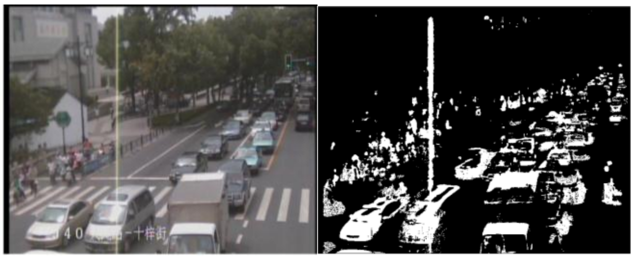
Step 5: Get total movement value by summing up the results returned from step 4.

Step 6: If the total movement is greater than the predetermined threshold, it will be treated as a moving object; otherwise it is a non-moving one.

VI. REFLECTION LIGHT BAND ERASING

Many objects, such as ground and water, can reflect light. Yet these reflections can be ignored as they perturb monitoring so slightly. However, light band, which is caused by highly reflective objects such as glass, is a kind of clustered reflection light with high energy. It will disturb monitoring very much [2], [5], [9], as it is often incorrectly regarded as moving objects. Here we will introduce an approach to remove the reflection of highly reflective objects.

We have found that a light band has an obvious edge with symmetrical brightness, which is the part with highest energy in reflection light. After binary conversion, the variation of light around light band will become more distinct. We can see the distinguishing change of brightness in Fig. 5. Indeed, it is neither environment change of light, nor shade, and nor burst noise.



a) b)

Fig. 5. The image with reflection light band.

Few of existing background updating algorithms and shadow eliminating algorithms can deal with the reflection light band [1],[2],[10]. In our approach, we regard removal of the reflection light band as eliminating high-energy part of image. We have found that the low-beam of car headlamp is asymmetric on both sides of the reference axis while the full-beam of car headlamp is symmetric. In fact, the brightness of reflected light is with a gradient distribution. In particular, the gradient distribution of the part that is not affected by the reflection is uniform and similar to the gradient distribution of the background.

Detecting the edge of light band is an important and fundamental step of light band eliminating. There are many operators for edge detection algorithm, such as Canny operator [12], Sobel operator [13], and Prewitt operator [13]. As the Canny edge detector uses a multi-stage algorithm, although usually returning good results, it is more time consuming than some other ones [14]. Both Sobel operator and Prewitt operator are discrete differentiation operators. The Prewitt operator uses averaging filtering [13] to compute an approximation of the gradient of the image intensity function; while the Sobel operator uses weighted filtering [13] to get an approximation of the opposite of the gradient of the image intensity function.

Taking into account of brightness variation and real-time processing requirement, we here use improved four direction Sobel operator [15] to add weight to the pixels near the center and the pixels near the edge highlight so as to get the gradient approximation of brightness function, thus improving accuracy and fastening the operating speed. The improved four direction Sobel operator for convolution operation are given in Table II.

TABLE II. IMPROVED FOUR-DIRECTION SOBEL OPERATOR FOR CONVOLUTION OPERATION.

3	10	3
0	0	0
-3	-10	-3

3	0	-3
10	0	-10
3	0	-3

10	3	0
3	0	-3
0	-3	-10

0	-3	-10
3	0	-3
10	3	0

We utilize the above Sobel operators to get each gradient component of V components of background image and pending image, as (12):

$$\begin{cases} g_x = \left(3*V(x-1,y-1)+10*V(x-1,y)+3*V(x-1,y+1)\right)- \\ \quad -\left(3*V(x+1,y-1)+10*V(x+1,y)+3*V(x+1,y+1)\right), \\ g_y = \left(3*V(x-1,y-1)+10*V(x,y-1)+3*V(x+1,y-1)\right)- \\ \quad -\left(3*V(x-1,y+1)+10*V(x,y+1)+3*V(x+1,y+1)\right), \\ g_l = \left(3*V(x,y-1)+10*V(x-1,y-1)+3*V(x-1,y)\right)- \\ \quad -\left(3*V(x+1,y)+10*V(x+1,y+1)+3*V(x+1,y)\right), \\ g_r = \left(3*V(x,y-1)+10*V(x+1,y-1)+3*V(x+1,y)\right)- \\ \quad -\left(3*V(x-1,y-1)+10*V(x-1,y+1)+3*V(x,y+1)\right), \end{cases} \quad (12)$$

where $g_{x,}$ g_y, g_l, and g_r are gradient components for x-direction, y-direction, left diagonal direction and right diagonal direction. We use $g_{x,}$ g_y, g_l, and g_r to get x and y direction gradient and diagonal direction gradient, as (13):

$$\begin{cases} G_c = \sqrt{{g_x}^2 + {g_y}^2}, \\ G_d = \sqrt{{g_l}^2 + {g_r}^2}. \end{cases} \quad (13)$$

As brightness gradient distribution of the light caused by diffused reflection is different from the one caused by full reflection, we can then we can test whether a given region is affected by full reflection light, as (14)

$$\left\| G_c^t - G_d^t \right| - \left| G_c^b - G_d^b \right\| \geq threshold, \quad (14)$$

where G^t denotes the testing image, and G^b represents the background. When a full reflection light comes, we examine the variation between V component gradient of the region and that of the background. As the energy distribution of the full reflection light rapid declines with the expansion of range of irradiation, while at the same time the diffuse reflection light declines in a slower speed rate, along with directional characteristics of the full reflection light, it is more effective to separate background and foreground by using gradient difference of the two directions than by using gradient of single direction.

As we have known it will cause a large variation of the background saturation if the target light is projected to the background. We conclude that the saturation of the testing image, S^t, is greater than that of the background one, S^b. Adding in white light will reduce color saturation. When using background subtraction to extract moving objects, we take advantage of the above judgment rule. If the following conditions are satisfied, we can consider the image is affected by full reflection light rather than moving object

$$\left\| G_c^t - G_d^t \right| - \left| G_c^b - G_d^b \right\| \geq threshold. \quad (15)$$

Fig. 6 shows us the effect of reflection light band erasing. The left one is image of Fig. 6 is the binary converted image and the right one is the corresponding generated greyed image. We can see that most of disturbing reflection light band is eliminated.

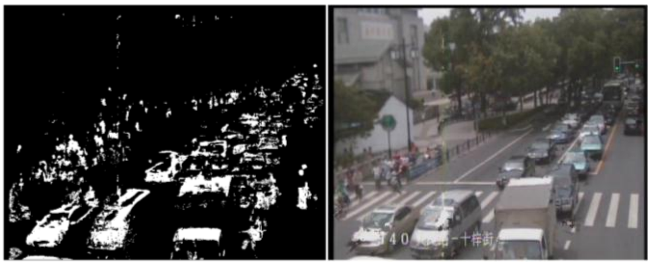


Fig. 6. The image that is after reflection light band erasing.

However, we can see from Fig. 6 that although most of reflection light band is removed, the edge pixels that are near the light band are also corroded. To solve the problem, we use Canny algorithm [12] to detect the edge of the reflection light band. The left figure of Fig. 7 is the edge detected by Canny before reflection light band erasing; the right figure of Fig. 7 is the edge detected by Canny after reflection light band erasing. Fig. 7 shows us that Canny algorithm can deal with edge corrosion.

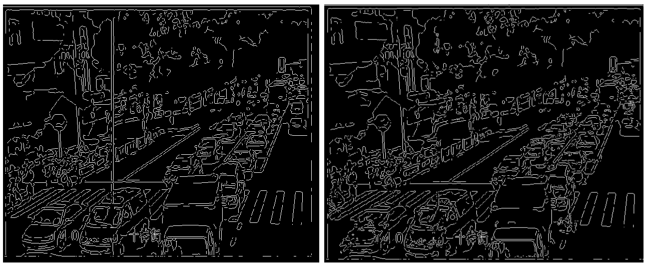


Fig. 7. The left figure is the edge detected by Canny algorithm before reflection light band erasing; the right figure is the edge detected by Canny algorithm after reflection light band erasing.

VII. Conclusions

A video-based traffic monitoring system must be capable of working in various weather and illumination conditions. However, it has a lot of prerequisites and constraints. In this paper, we focus on optimizing image processing in video-based traffic monitoring.

We first improve the threshold-based image binary conversion approach. Different from conventional approach, out adaptive threshold take into space information as well and proves to be a better one. Then we make a tradeoff between time cost and effect, introducing a three-dimension Gaussian filtering denoising approach. We use a Gaussian model to generate background, so that we can refresh background any time, thus throwing away a major limitation of background subtraction approach. At same time, we also introduce method to detect and remove shadow of moving objects so that the separated moving objects will not be disturbed by their shadows.

As burst noise will often interfere monitoring, we, taking advantage of the characteristic that most of the burst noise merely exists in several continuous frames, put forward an algorithm to remove burst noise by several anterior and posterior frames. Aiming at eliminating reflection light band, we utilize the characteristics of H, S and V component of reflection light band and conclude that removal of the reflection light band is eliminating high-energy part of image. Based on the rule, we propose an approach to remove reflection light band.

The experiments show that the optimization approaches and algorithms prove to improve effeteness and efficiency of system.

References

[1] P. Kumar, S. Ranganath, W. M. Huang, K. Sengupta, “Framework for real-time behavior interpretation from traffic video”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 1, pp. 43–53, 2005. [Online]. Available: http://dx.doi.org/10.1109/TITS.2004.838219

[2] J. Zhou, D. Gao, D. Zhang, “Moving Vehicle Detection for Automatic Traffic Monitoring”, *IEEE Transactions on Vehicular Technology*, vol. 56, no. 1. pp. 51–59, 2007. [Online]. Available: http://dx.doi.org/10.1109/TVT.2006.883735

[3] H. Zhang, J. E. Fritts, S. A. Goldman, “Image segmentation evaluation: A survey of unsupervised methods”, *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 260–280, 2008. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2007.08.003

[4] O. Miller, A. Averbuch, “Color image segmentation based on adaptive local thresholds”, *Image and Vision Computing*, vol. 23, no. 1, pp. 69–85, 2005. [Online]. Available: http://dx.doi.org/10.1016/j.imavis.2004.05.011

[5] F. Zhu, “A Video-Based Traffic Congestion Monitoring System Using Adaptive Background Subtraction”, in *Proc. of the ISECS*, IEEE, 2009, pp.73–77.

[6] D. Barash, “Nonlinear diffusion filtering on extended neighborhood”, *Applied Numerical Mathematics*, vol. 52, no 1, pp. 1–11, 2005. [Online]. Available: http://dx.doi.org/10.1016/j.apnum.2004.07.002

[7] W. W. Boles, M. Kanefsky, M. Simaan, “A reduced edge distortion median filtering algorithm for binary images”, *Signal Processing*, vol. 21, no. 1, pp. 37–47, 1990. [Online]. Available: http://dx.doi.org/10.1016/0165-1684(90)90025-T

[8] P. F. Johannes, D. Haeyer, “Gaussian filtering of images: A regularization approach”, *Signal Processing*, vol. 18, no 2, pp. 169–181, 1989. [Online]. Available: http://dx.doi.org/10.1016/0165-1684(89)90048-0

[9] F. Zhu, L.Y. Li, “An Optimized Video-Based Traffic Congestion Monitoring System”, in *Proc. of the Knowledge Discovery and Data Mining*, IEEE, 2010, pp.150–153.

[10] C. C. Chiu, M. Y. Ku, L. Y. Liang, “A Robust Object Segmentation System Using a Probability-Based Background Extraction Algorithm”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no 4, pp. 518–528, 2010. [Online]. Available: http://dx.doi.org/10.1109/TCSVT.2009.2035843

[11] T. F. Chan, S. H. Kang, J. H. Shen, “Total Variation Denoising and Enhancement of Color Images Based on the CB and HSV Color Models”, *Journal of Visual Communication and Image Representation*, vol. 12, no. 4, pp. 422–435, 2001. [Online]. Available: http://dx.doi.org/10.1006/jvci.2001.0491

[12] J. Canny, “A Computational Approach To Edge Detection”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698,1986,

[13] I. E. Abdou, W.K. Pratt, "Quantitative design and evaluation of enhancement/thresholding edge detectors”, *IEEE Transactions on Image Processing*, Vol. 67. No. 5, pp. 753–766, 1979.

[14] L. J. Ding, A. Goshtasby, “On the Canny edge detector”, *Pattern Recognition*, vol. 34, no. 3, pp. 721–725, 2011.

[15] H. Farid, E. P. Simoncelli, “Differentiation of discrete multi-dimensional signals”, *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp.496–508, 2001. [Online]. Available: http://dx.doi.org/10.1016/S0031-3203(00)00023-6