ON STORAGE, SEARCH AND OBJECT IDENTIFICATION OF VIDEO SEQUENCE

WU HAO

M.Phil

The Hong Kong Polytechnic University

2016

The Hong Kong Polytechnic University

Department of Electronic and Information Engineering

On Storage, Search and Object Identification of Video Sequence

WU HAO

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Philosophy

Aug 2015

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signed)

_____WU HAO________(Name of Student)

Abstract

Digital video is an important information carrier nowadays, as a representation of moving visual images in the form of encoded digital data. Video based applications are more and more popular, better knowledge of video coding, video retrieval, video surveillance, robot vision, etc are always required. In this thesis, we initially introduce some applications with existing approaches, including video recording for surveillance, smart robot vision and driver assistant system; followed by the improvement work we have done correspondingly.

In the video coding for surveillance video, the approach is very different from normal video coding because of high temporal redundancy. Conventionally, large Group of Pictures (GOP) and long term reference frame will be used to set up surveillance video encoder. In addition, some research works make use background extraction to help encoding process. Through these approaches, some bit-rate reduction can be achieved to encode surveillance videos. However, conventional methods are trapped into complying encoding standard with just one encoder. In this thesis, we propose a double encoder coding scheme. Foreground and background materials are firstly extracted and processed as masked foreground and condensed background sequences. Then separate encoders are used to encode foreground and background sequences. According to our experimental results, A large bit-rate reduction can be achieved through the proposed scheme. With defined side information, the video can be well reconstructed at decoder side without foreground distortion.

Moving object detection from a moving camera is a fundamental task in many applications, such as smart robot. The fundamental assumption of conventional moving object detection method is that the background is either static or moving as a 2D plane. However, for the moving robot car vision, the background movement is 3D motion structure in nature. In this

situation, the conventional moving object detection algorithm cannot handle the 3D background modeling effectively and efficiently. We have proposed a novel scheme by utilizing the motor control signal and depth map obtained from a stereo camera to model the perspective transform matrix between frames under a moving camera. Hence, the relationship between a static background pixel and the moving foreground corresponding to the camera motion can be related by a perspective matrix. The proposed scheme is able to detect moving objects in our moving robot car efficiently. Different from conventional approaches, our method can model the moving background in 3D structure, without online model training. More importantly, the computational complexity and memory requirement are low, making it possible to implement this scheme in real-time, which is even valuable for a robot vision system.

Rail extraction is a fundamental and important step in railway Driver Assistant System, which is now an important application of image processing. The task is challenging as the railway is exposed to different environments. In this research, we propose a railway extraction scheme, using a novel connectivity measure method named Angle Alignment Measure. The proposed scheme is robust to luminance and color variation, without explicit edge extraction process. Railways with different lengths and patterns can be extracted under various lighting and weather conditions. More importantly, the computational complexity of the proposed scheme is very low, requiring only on average 26ms to process a frame on a smart phone and 5.5ms on a desktop computer, which are significantly better than other algorithms in the literature.

Publications

Journal Paper

- Hao Wu and Wan-Chi Siu, “Robust Scheme for Real-Time Railway Extraction Using Angle Alignment Measure.” Submitted to *IEEE Transactions on Intelligent Transportation Systems*.

Conference Paper (Accepted or Published)

- Wu, Hao, and Wan-Chi Siu. "Real time moving object detection using motor signal and depth map for robot car." *IS&T/SPIE Electronic Imaging*. Intelligent Robots and Computer Vision XXXI: Algorithms and Techniques, 90250P. International Society for Optics and Photonics, February 3, 2014, San Francisco, USA.
- Hao Wu and Wan-Chi Siu, “Real-time Railway Extraction by Angle Alignment Measure”, Paper Accepted, to be published in Proceedings, IEEE International Conference on Image Processing, (ICIP’2015), 27-30, October 2015, Quebic City, Canada.

Acknowledgments

I would like to express my sincere thanks a group of people in Hong Kong Polytechnic University, where I get my bachelor degrees and pursue post graduate degree. I want to show my major appreciation to Prof. Wan-Chi Siu, who is my supervisor for both undergraduate final year project and post graduate programme. He gave me knowledge on digital signal processing and video technology. He made an example to me not only about how to be a good researcher, but also how to be a responsible people for the society and how to be a good family member. His devotion on career will always guide me for the rest of my life.

Special thanks to Mr. Hui Wai Lam and Dr. Calvin Cheung, for the time we work together on hitting deadlines on different projects. I am indebted to Dr. Wang Lili, Dr. Hung Kowkwai and Mr. Chen Zhang, who set me good example in my post graduate study. I also appreciate the companies from Zhou Huiling, Huang Junjie, Yao Meng and all other people in DSP lab.

Last but not the least, most thanks to my parents, who always stand behind me to give me selfless supporting for every decision I made and take away my pressures and worries.

Table of Content

List of Figures

List of Tables

Chapter 1. Introduction

Driven by highly integrated and reliable semiconductor technologies, fast development of multimedia and digital signal processing algorithm creative innovations, digital video has been widely used in many areas. It also stimulates a large amount of video based activities and studies, such as video recording, video broadcasting, video surveillance, smart robot vision, video aid system, etc. Under this big environment, tons of videos are processed and transmitted every day. In this thesis, two basic video technologies for surveillance are investigated, which are coding for video surveillance and scene picture extraction from vehicle vision.

1.1 Introduction of video surveillance

Surveillance is a process of monitoring behavior, activities, or other information, for the purpose of influencing, managing, directing, or protecting. It can include an observation from a distance by camera, which is the scope of video surveillance. Video surveillance is rather broad concept and it is also a quite popular topic for both research and commercial applications. Automatic means of video surveillance for public safety enhancement has existed for quite a long time but has gained significant popularity only in recent years [1]. As the increasing awareness of security, more and more high-definition surveillance cameras are installed in the public or private areas. According to recent research by International Data Corporation, there will be around 5,800 Exabyte of surveillance video to be stored, transmitted and analyzed [2]. In some research [2][3], it is emphasized that in the coming several years, the improvement in

compression rate can be achieved by H.264/AVC (Advanced Video Coding) or even HEVC (High Efficiency Video Coding) will be far behind the surveillance video data growth rate, which will challenge the data storage and transmission in foreseeable future. For video compression, HEVC excels in compression comparing with H.264/AVC with more than 35% bitrate reduction [4]. However, it also requires much more intensive computation and power consumption for encoding, which make the HEVC have not been well adopted into surveillance system.

Besides, information retrieval and mining is one key surveillance video application. Conventionally, manual screening is used to find out the target person or scene change, which is behind the times. They should be replaced by auto scene analysis, people detection and identification technique as an improvement of video processing technique. However, as what encountered in video compression, increasing video resolution, frame rate and video length will give great challenge video processing to future processing.

The main reason for problems discussed above is that a conventional video coding scheme, like H.264/AVC and HEVC, is not specifically designed for video surveillance. In other words, the conventional scheme cannot utilize the characteristic of surveillance video. In a surveillance video system, the content can be classified into two parts, which are relatively static background and active foreground, or region of non interest (RONI) and region of interest (ROI). Obviously, active foreground, which are usually people, vehicle, etc, are more important and more encoding resource should be allocated comparing with static background. Therefore, a surveillance based coding scheme is needed.

1.2 Introduction of vision on vehicle

As the development of technology, research into intelligent vehicle has expanded into smart control system that can control vehicle semi-automatically or automatically. In spite of Light Detection and Ranging (LIDAR), stereo imaging, Geographic Information System (GIS)/Global Positioning System(GPS)/ Inertial Measurement Unit(IMU) based methods, computer vision on vehicle plays an important role for smart control system[5][6]. Many techniques and applications are enabled for vehicles, such as robotic car auto navigation, automobile lane departure warning, train anti-clash warning, etc. In this thesis, we have made two typical studies on applying videos on vehicle applications, which are separately moving object detection on robotic car vision and railway extraction on train vision.

Moving object detection is useful in many automated video analysis including object detection, object tracking, and behavior analysis [7]. For robotic car vision application, moving object detection technology can be used for object detection and identification, and then to arrange a route or to avoid obstacle. Many schemes have been suggested corresponding to moving object detection, which can be categorized based on different types of background: static background, dynamic background and moving background. Static background means the background region in vision are silent, or only distorted by random noise. Dynamic background means some elements in background are comparatively active but not moving object, such as swinging tree and sea wave. There is a similarity between static background and dynamic background that is the camera is fixed hence the background can be regarded as quiet or with little variation. Different from the above two types, moving background means the entire background is moving,

which is usually brought by camera moving during the video recording. Moving object detection under robotic car vision is within the scope of moving background moving object detection, which is more challenging comparing with static background and dynamic background. In the vision of a robot car, the background movement brought by camera moving is usually 3D structure. As shown in Figure 1-1, the structure of the scene is complex. When the robot moves forward, movements of pixels on stools aside and pixels on the white board cannot be simply modeled by one homography matrix.

Figure 1-1 Scene of robot vision, background with 3D structure

Computer vision has been served for railway systems to ensure safety in several approaches, including moving pedestrian detection, train detection, railway crossing monitoring system, platform monitoring system, etc [8-11]. Furthermore, for railway based Driver Assistant (DA) system, computer vision based application has been studied as well, such as front safety distance estimation [12], vehicle detection [13][14], obstacle detection [15], etc. In railway extraction of the DA system, we have to find the position, shape and length of the railway in front to a train, which is an important and fundamental module of the DA system. It acts as a preprocessing step for other computer vision algorithms in the system. Moreover, in railway Driver Assistant system, there are three key concerns for railway extraction, which includes result

reliability, environment adaptability and computational complexity. Reliability of results is the fundamental requirement, which means a railway extraction scheme should provide accurate and stable extracted rail result. However, as the train is running, the railway pattern in front may vary both in shape, curvature and length, which challenge railway extraction. Environment adaptability refers to that the system should be adaptive to varying environment. The environment includes different lighting and weather situations, together with different ground conditions around railway, like sand road, rail sleeper, concrete road, etc. Computational complexity is in view of practical implementation. Time-efficient is an important factor for a real-time system. More importantly, railway extraction is served as a preprocessing step for other computer vision application, such as the detection of a train in the front. The railway extraction process should not be considered as the only item to be done in real-time. Instead, more time should be reserved for the follow up procedure, so that railway extraction module should occupy as small as possible the computation power.

1.3 Literature review of trends

1.3.1 Surveillance video coding

Numerous exciting researches put effort on improving the coding efficiency for surveillance video. They can be classified into four categories. The first category is model based coding, that uses explicit model to describe the scene, such as human face, background, etc. Only the corresponding model parameters are needed to be transmitted with video stream to represent the object at decoding end. For example, Aizawa and Harashima [16] produced a model-based analysis synthesis image coding (MBASIC) system which involves the construction of a 3D face model, and it can be

reconstructed without noise corruption. In [17], the background scene is parameterized by a 3-D model, which can be used to syntheses the background at decoder side. However, researches in this category cannot give effective object coding in the surveillance system. It is hard to use one or a set of models to describe a generalized object in surveillance scene. Then, object-oriented coding can be considered as the second category, as the development of MPEG-4 [18]. In [19], an efficient region-based video coding is proposed. To maintain both PSNR and bitrate, Babu et al.[20] proposed an object based coding scheme for video surveillance system. A separate coding method for segmented foreground and background scene can boost the coding efficiency. However, object segmentation is still an ill-posed problem even in the state-of-art computer vision area. If the object cannot be segmented preciously, the visual quality of the decoded frame will be downgraded significantly. Meanwhile, precious object segmentation result can also be a trade between computational complexity and quality. The third category is the block-based hybrid video coding scheme, which is increasingly popular, including H.264 and HEVC. A region based video coding scheme based on H.264 is proposed in [20], which can be used for surveillance video coding. Difference detection algorithm can automatically distribute the content to different encoding module according to their content features. In another direction, Wiegand et al. [21] proposed a long-term reference picture for motion compensation, which is accepted by H.264/AVC. Corresponding to long-term reference frame, many researchers have been working on the selection of optimal reference frame. In [22], simulated annealing is used in long term reference frame selection. PSNR is improved comparing with evenly assigned long term reference frames. This technique can be

naturally implemented in surveillance video coding since the background content is usually long lasting and can be efficiently used as a long term reference frame. In [23] Ding suggested a new background frame as reference frame to improve the motion estimation performance. Pushkar [24] used detected background to control skip selection block encoding and long term reference frame selection. Zhang et al. [25][43] also contribute to this area a lot. In [25], background model is used to classify each macroblock into three categories. They proposed two different prediction model, named background reference prediction and background difference prediction, for block inter prediction. For the last category, to meet special need of video surveillance, some new video coding standards were published. In 2010, the Surveillance Video and Audio Coding (SVAC)[26] issued some tailor designs for surveillance by providing friendly ROI coding, enabling greater interconnectivity and providing more intelligent data analysis. Many researchers have been working towards improving coding performance of SVAC [27][28]. A group leading by Gao Wen Group also proposed the IEEE 1857 video coding standard[29] [30], by using background modeling based prediction techniques for video coding, error resilience methods.

1.3.2 Moving object detection under moving background

Two categories can be defined for approaches in moving object detection under moving background, namely, the motion based method [38] and the affine transform based method [39][40]. The fundamental assumption for most approaches in both categories is that moving object occupies much smaller region comparing with moving background. Motion information is used to separate objects from background in motion based methods. Typically, Qiu Yue, et al. [38] used pixels on uniform grid as

sampling pixels. The optical flow is calculated through local variance and local entropy for each sampling pixel. An algorithm with hierarchical clustering and secondary clustering algorithm is used to cluster foreground motion and background motion. The moving object can then be detected. For affine based methods, the idea is to convert moving background modeling problem to static background modeling problem by affine transform. Affine transform represents the image geometric transformation, such as translational motion, shearing and rotation. In [39], Harris corner detector is used to extract feature points. A modified feature point matching is used to find the affine matrix between successive frames. Then frame difference is used to find the position of moving object. In [40], low-rank and sparse representation is used to find the affine matrix. Comparing with entire background frame, moving object is appeared to be sparse. Then by using low-rank analysis, the background modeling and foreground detection can be done together.

1.3.3 Railway extraction techniques

There are many similarities between lane mark and railway [31] [32]. First, both of them are on ground utilities in transportation system, used to guide the driving direction of vehicles. Therefore, they are important features in Driver Assistant System, to provide landmark for the follow up processing. Second, lane mark and railway are similar in appearance. Both of them appear as straight lines or parabolic curves, and normally well distinguishable with surrounding ground environment. For these reason, many similar techniques are used both in lane mark extraction and railway extraction, including feature extraction and curve modeling [6] [33-36]. In [36], image patches of lane marking and nonmarkings are studied. A cubic spline curve model can be used to

model the candidate lane mark, and the final lane results can be obtained after particle filtering and RANSAC. Liu, Wogotter and Markelic [37] used these ideas, and obtained an improved accuracy through using multiple kernel density to estimate the probability relationship between visual cues and the linear-parabolic lane model. In railway extraction research, ideas enlightened by land mark extraction have also been discussed and implemented [31][32].

However, crucial difference between lane mark and railway should be noticed. In many cases, lane mark appears in segments, therefore modeling the entire lane mark curve with stochastic estimation based method is popular for this scenario. But railway appears as continuous strip, which maybe not suitable for the estimation method mentioned above. Instead, line extrapolation based method can be more effective in railway extraction and many researchers handling railway extraction made use of extrapolation method. Nassu and Ukai [31] segmented a frame into short and long distances. They used dynamic hysteresis thresholded with nonmaxima suppression in edge extraction step to handle varying illumination. Distance transform was conducted iteratively and Chamfer distance was used as similarity metric to select the best fit railway model from several predefined patterns. In [32], Kaleli and Akgul proposed a pixel-wise extrapolation method was by utilizing Dynamic Programming. The cost of each pixel is defined by gradient magnitude and railway width. The scheme can extrapolate the railway iteratively. Similarly, recursive estimation [33] has been used to extrapolate railway based on pixel characteristics as well.

1.4 Objective

In this research, different applications based on video storage, search and object identification are investigated. The primary object of this research is to improve the performance of different applications through the explorative new algorithms.

First, huge temporal redundancy is existed in surveillance video. Conventional techniques are bonded by the framework of video coding standard and traditional quality assessment metric. To achieve a higher coding efficiency, surveillance video can be coded separately using video background and foreground coding. Therefore a coding scheme should be developed by integrating video sequence extraction, separation, coding and synthesis.

Second, moving object detection under robot vision is a challenging problem as the background movement in robotic vision cannot be modeled by simple affine transform. However, as an integrated robot, additional information can be utilized including depth map and motor control signal, which help to model the background movement more accurately. We focus on classifying pixels with different movement from background movement by using depth map and motor control signal.

Third, conventional railway extraction schemes depend highly on edge extraction, which is time consuming and lack of different environment adaptability. Especially in our scenario, both luminance and chrominance vary a lot on railway area. We want to develop a railway extraction scheme by new measurement metric, which is robust to environmental changes and fast for practical implementation.

1.5 Organization of Thesis

The rest of the thesis is organized as follows. In Chapter 2, we make a technique review on existing approaches on surveillance video recording, background modeling with moving object detection and railway extraction in driver assistant system. In Chapter 3, a double encoder scheme for surveillance video is introduced. At encoding side, masked foreground sequence and condensed background sequence are encoded separately. At the decoding side, the surveillance scene is recovered by foreground and background synthesis. In Chapter 4, a solution of moving object detection in robotic car vision is designed. Depth map and motor control signal are used to model background movement. In Chapter 5, we proposed a novel connectivity measure metric, noted as Angle Alignment Measure, to extract railway in city light railway scene. The proposed railway extraction scheme is robust with low computational complexity, which has been implemented in a real product. Chapter 6 concludes the research and discuses possible future research of these areas.

Chapter 2. Technical Review

2.1 Surveillance video coding

2.1.1 Supported in video coding standard

Since early 1980's, video compression performance was largely improved by utilizing intra-frame and inter-frame compensation [21], which is still a main technique in recent H.264/AVC[41] and HEVC[42] coding standards. In a hybrid coding scheme, a frame is divided into blocks. For each block in the current frame, the content inside block is compensated from the content of the previously decoded pictures, as shown in Figure 2-1. Thus, only the difference (residual frame), motion vectors and frame reference number are needed to be encoded. In this way, the temporal redundancy is largely decreased.

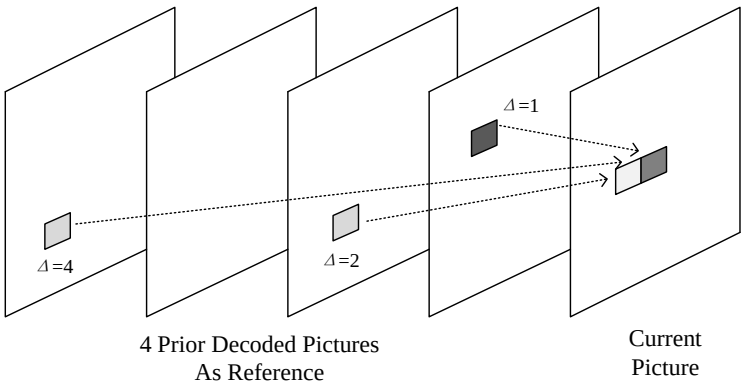


Figure 2-1 Motion compensation, with motion vectors and picture reference parameters (Δ)

In 1999, Wiegand [21] proposed a long-term memory motion compensated prediction, which is later adopted by H.264/AVC and HEVC. The proposed motion compensation scheme extends the temporal displacement vector used in block based motion compensation. For example, frames decoded 5 seconds ago can be used as the

reference frames of the current frame. Thus, extra memory is needed, both in encoder and decoder, to store previously decoded frames. As shown in Figure 2-2, the motion compensation for each block is conducted based on multiple frames' compensation.

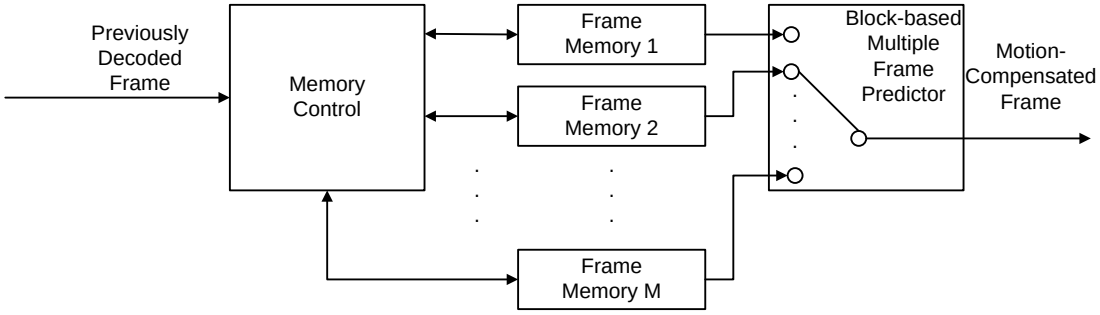


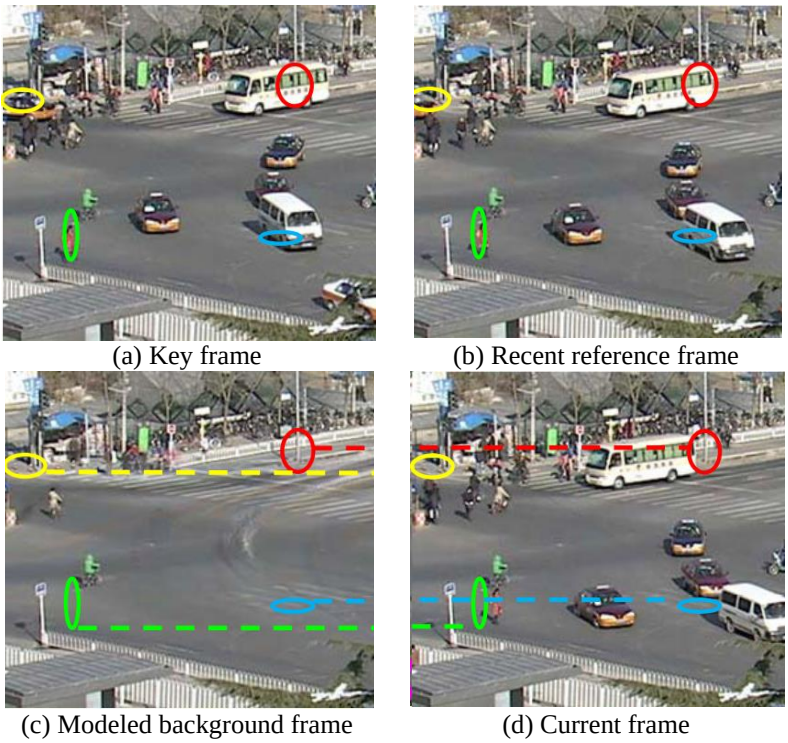
Figure 2-2 Long-term memory motion-compensation

A scheme with long-term reference frames can apparently increase the video temporal correlation, which can reduce the complexity of a surveillance video coding. Based on this feature, many researchers designed long-term reference frame (or referred to as key frames) selection and memory control scheme to further improve the surveillance coding efficiency [22] [24].

2.1.2 Background modeling based surveillance video coding

One issue which affects the efficiency of a surveillance video coding system is the as re-exposed background region problem [3]. Let us refer to Figure 2-3 (a) shows long term reference frame or key frame selected by a coding scheme. (b) shows a recent short term reference frame. (c) is a modeled background frame by Gaussian Mixture Model and (d) is the current frame, which is going to be motion compensated. Different shapes of circles are used to highlight different location in the same scene. If a foreground object left from original position, the background content is re-exposed.

However, in conventional reference selection scheme as (a) and (b), the re-exposed background content may not be encoded. In this situation, a scheme with background prediction can be involved in motion compensation, which is the idea of background modeling based surveillance video coding.



(a) Key frame (b) Recent reference frame

(c) Modeled background frame (d) Current frame

Figure 2-3 Re-exposed background content situation

Figure 2-4 is a typical background frame based surveillance video coding codec. Modeled backgrounds are firstly obtained for each surveillance video frame. As in a hybrid video coding scheme, background frames are encoded and decoded. Then in motion compensation stage for original surveillance frames, decoded background frame can then be involved.

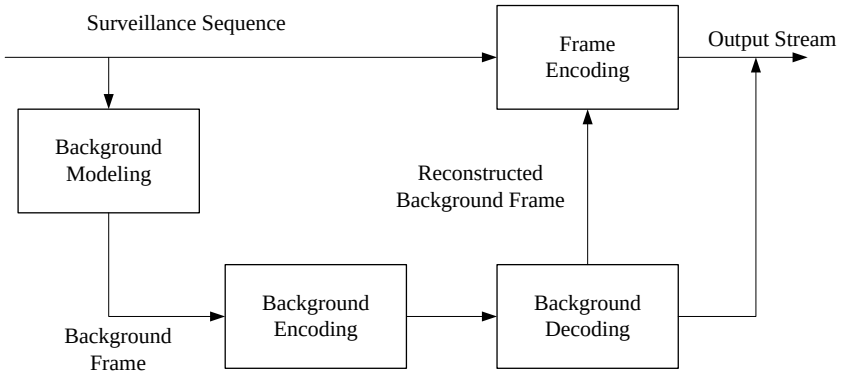


Figure 2-4 Background frame based surveillance codec

2.2Background Modeling with moving object detection

2.2.1 Gaussian Mixture Model based background modeling

Gaussian Mixture Model (GMM) background modeling [46] is a classic background modeling method. Normally, for a pixel on static background region, its intensity is regarded as static and it may appear differently due to noise. Therefore one particular type of distribution is explicitly used to model the pixel intensity. Pixel values that do not fit the background distribution value are regarded as foreground pixel. These kinds of schemes cannot fit well for online background updating due to content change or illumination change.

For GMM, multiple adaptive Gaussians are used in background modeling process. In this thesis, the pixel value over time is referred to as "pixel process", which is actually a time series pixel value. The pixel process for a particular pixel $\{x_0,y_0\}$ can be represented as

$$\{X_1,...,X_t\}=\{I(x_0,y_0,i):1\le i\le t\} \tag{2.1}$$

where *I* is the video sequence with length *t*. For each pixel, the pixel process is modeled by a mixture of K Gaussian distributions. The probability distribution of current pixel value can be constructed as

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \tag{2.2}$$

where *K* is the number of distributions, $\omega_{i,t}$ is the estimated weight of ith Gaussian distribution η at time t, which is actually the portion of data under that distribution. μ and Σ are the mean value and covariance matrix of Gaussian distribution.

Every new pixel value, X_t, is checked again the probability distribution to see whether the value can match a Gaussian distribution. If none of the existing *K* Gaussian distribution matches the new pixel value. A new Gaussian distribution is set up with the new pixel value as mean value. Then the Gaussian distribution weights $\omega_{k,t}$ is updated by Equation 2.3.

$$\omega_{k,t} = (1-\alpha)\omega_{k,t-1} + \alpha M_{k,t} \tag{2.3}$$

where α is the learning rate and $M_{k,t}$ is the matching flag, which equals 1 if the model matched and equals to 0 is the model fails in matching.

Then the background frame can be reconstructed by taking the mean value of Gaussian model with the highest estimated weight for each pixel position. Further, it is designed that the Gaussian with the largest *B* estimated weights are all possible background pixel value, due to the illumination change. If the current pixel value does not match these distributions, it will be regarded as a moving foreground pixel.

2.2.2 Histogram based background Modeling

Histogram based background modeling [45] is a fast static background modeling method, which is further simplified from the Gaussian Mixture Model background modeling method. The fundamental assumption is that, for each position, the occurrence of background color should occupy most of the time comparing with foreground moving object color.

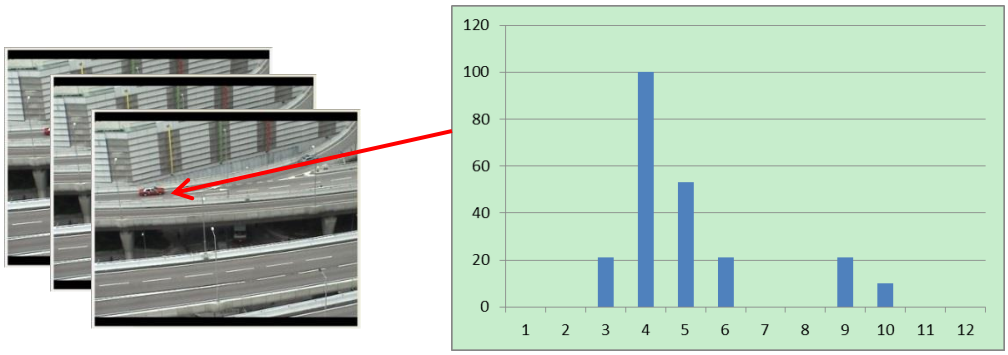


Figure 2-5 Histogram of intensity for one pixel position

For a pixel position, if intensity values are recorded within a time duration, a histogram of intensity can be constructed. According to the assumption, the bin with highest accumulation is more likely to be the color of background for this pixel. As one histogram is built for each pixel position of the background frame can be constructed by taking the intensity value of the bin with the highest accumulation within each histogram.

Histogram of each pixel position is updated after processing each pixel of this position. Different histogram temporal buffer sizes can be used during histogram updating. A small temporal buffer means short time period of frames are involved in background construction to model a short term background, which is more sensitive to background

updating. A large temporal buffer means long time period of frames are involved to model a long term background, which is more robust to the different color pattern brought by moving object. Short term background and long term background are usually employed together to deal with different application scenarios. As in [45], pixels are classified into background pixel, active pixel, static pixel and re-active pixel based on the relationship between current pixel value, pixel value on short term background and pixel on long term background. Then moving object can is detected based on active pixel evaluation.

2.2.3 Motion based moving object detection under moving background

The major difficulty for moving object detection under moving background is that both foreground and background are moving corresponding to camera. However, when an object is defined as moving object, the noted movement should be corresponding to background. Therefore, in camera vision, the moving characteristic of foreground should be different as shown in Equation 2.4.

$$MV_{foreground} = MV_{background} + MV_{relative} \tag{2.4}$$

It can be used to extract a moving object from moving background by identifying different moving pattern, which is the fundamental idea of motion based object detection. It is usually achieved by optical flow analysis.

Optical flow is pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and the scene [64] [65]. The concept was firstly introduced in the psychological area, later on this term was brought into computer vision in 1990s.

The basic model of the optical flow is very simple. It tries to calculate the motion between two image frames which are taken at times t and t + ∆t at every pixel position, with the assumption that the luminance is constant. Therefore, a pixel at (x,y) in frame t should come from the pixel at (x+ ∆x,y+ ∆y) at frame t+ ∆t, and we can get Equation 2.5.

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{2.5}$$

Assuming the movement is small. The image constraint at $I(x, y, t)$ with Tylor series can be deduced as Equation 2.6.

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + e \tag{2.6}$$

where *e* is high order terms. Assume *e* is 0 in this analysis, we get Equation 2.7

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0 \Rightarrow \frac{\partial I}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t}\frac{\Delta t}{\Delta t} = 0 \Rightarrow f_x u + f_y v + f_t = 0 \tag{2.7}$$

where $\Delta x = \frac{\Delta x}{\Delta t} = u$ and $\Delta y = \frac{\Delta y}{\Delta t} = v$ are the velocity or optical flow of $I(x, y, t)$. $\{f_x, f_y, f_t\} = \left\{\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial t}\right\}$, which are the partial differentiations of *I* corresponding to *x,y,t*.

There are many methods to calculate the optical flow, and these methods can be summarized in three major categories.

(1) Differential based methods: based on the partial derivatives of the image signal to calculate the optical flow pixel by pixel.

(2) Phase correlation methods: regard the phase information as the differentiation subject through the band pass spatial-temporal filter.

(3) Block-based methods: employed in video coding that calculates the minimizing sum of difference by motion estimation.

However, the computational complexity of the conventional differential based and phase correlation based optical flow methods are high. It is time consuming for real-time application. Down sample the pixels to be processed can reduce the computation. But if the down sampled candidate pixels are too sparse, the foreground and background may not be fully covered to model the movement. Therefore, feature point extraction method is usually associated with optical flow calculation.

The common assumption for motion based moving object detection method is that the background is 2D structure plane. Therefore the motion is identical for the entire background region. The rest motions with different patterns are regarded as foreground motion. Based on this assumption, after the optical flow calculation, the background motion will be identified. Note that, in the background motion identification, not only the quantity of feature points will be considered, but also the feature point spreadness. It means that the background motion vectors should occupy most of the space of the frame. As in [38], the spreadness of motion vector cluster is measured by modeling use of entropy. Higher entropy represents higher chaos order, which means the motion vector is widely spread in the frame.

Then the moving object can be identified for other motions which do not align with the motion of background.

2.2.4 Geometric transform and application in moving object detection under moving background

Geometric transform describes the image geometric transformation, including motion, shearing and rotation. [51] The geometric transform can be conventionally modeled as matrix, denoted as transform matrix.

If we only consider geometric transform on single plane, it can also be referred to as 2D affine transform. The affine transform can be modeled by Equation 2.8.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.8}$$

where (x',y') are the new pixel coordinate after the affine transformation of pixel (x,y). And parameter a,b,c,d,e,f is the addition of different transform parameters, including

Translation by (x_0,y_0) $$\mathbf{T} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

Scale by s_1(on x) and s_2 (on y) $$\mathbf{T} = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

Rotate by θ $$\mathbf{T} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.11}$$

To obtain the affine transformation matrix *T*, we can use the matrix operation to obtain the relationship between the two frames *Q* and *P*. We can find a number of points

$\{p_0,p_1,\ldots,p_{n-1}\}$ in image P that match points $\{q_0,q_1,\ldots,q_{n-1}\}$ in image *Q*. By homogeneous coordinate representation of each point as column in matrix **P** and **Q**:

$$\mathbf{P}=\begin{bmatrix} x_0 & x_1 & \ldots & x_{n-1} \\ y_0 & y_1 & \ldots & y_{n-1} \\ 1 & 1 & \ldots & 1 \end{bmatrix}=\begin{bmatrix} p_0 & p_1 & \ldots & p_{n-1} \end{bmatrix} \tag{2.12}$$

$$\mathbf{Q}=\begin{bmatrix} x_0 & x_1 & \ldots & x_{n-1} \\ y_0 & y_1 & \ldots & y_{n-1} \\ 1 & 1 & \ldots & 1 \end{bmatrix}=\begin{bmatrix} q_0 & q_1 & \ldots & q_{n-1} \end{bmatrix} \tag{2.13}$$

Then the pseudo-inverse of **P** will be used to find the affine transform matrix **T** with the least mean-squared error as shown in Equation 2.14.

$$\mathbf{T}=\mathbf{Q}\mathbf{P}^T\left(\mathbf{P}\mathbf{P}^T\right)^{-1}=\mathbf{Q}\mathbf{P}^{+} \tag{2.14}$$

To better explain a 3D geometric transform, we start from camera model. In a planar camera model as shown in Figure 2-6, a point in 3D space with coordinate (*X*,*Y*,*Z*) is projected on camera CCD plane on pixel (*x*,*y*).

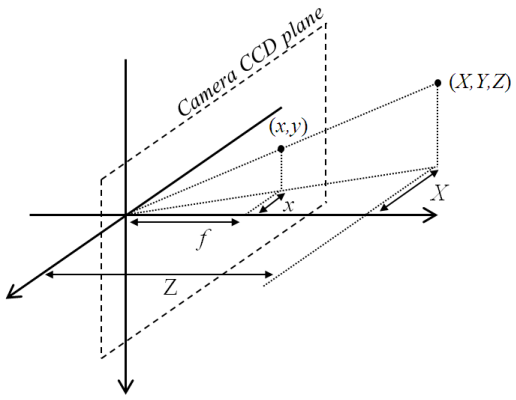


Figure 2-6 Planar camera model

Equation 2.15 is the matrix form of the projection relationship, where f is the focal length in a planar camera model and $\boldsymbol{V}$ is the projection matrix.

$$\begin{pmatrix} lx \\ ly \\ l \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} kX \\ kY \\ kZ \\ k \end{pmatrix} = \mathbf{VP} \tag{2.15}$$

Considering the movement in 3D space, similar with affine transform, the motion E of a point with coordinate (X, Y, Z) to the new coordinate (X', Y', Z') by Equation 2.16.

$$\mathbf{P'} = \begin{pmatrix} lX' \\ lY' \\ lZ' \\ l \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} kX \\ kY \\ kZ \\ k \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & T_1 \\ r_{21} & r_{22} & r_{23} & T_2 \\ r_{31} & r_{32} & r_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} kX \\ kY \\ kZ \\ k \end{pmatrix} = \mathbf{EP} \tag{2.16}$$

If we project both point $P(X,Y,Z)$ and $P'(X',Y',Z')$ on camera CCD plane by projection matrix **V**, we get

$$LHS = \begin{pmatrix} px' \\ py' \\ p \end{pmatrix} = \mathbf{VP'} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} lX' \\ lY' \\ lZ' \\ l \end{pmatrix} \tag{2.17}$$

$$RHS = \begin{pmatrix} qx \\ qy \\ q \end{pmatrix} = \mathbf{VEP} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & T_1 \\ r_{21} & r_{22} & r_{23} & T_2 \\ r_{31} & r_{32} & r_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} kX \\ kY \\ kZ \\ k \end{pmatrix}$$

$$= \begin{pmatrix} fr_{11} & fr_{12} & fr_{13} & fT_1 \\ fr_{21} & fr_{22} & fr_{23} & fT_2 \\ r_{31} & r_{32} & r_{33} & T_3 \end{pmatrix} \begin{pmatrix} kX \\ kY \\ kZ \\ k \end{pmatrix} = \begin{pmatrix} fr_{11} & fr_{12} & fr_{13} & fT_1 \\ fr_{21} & fr_{22} & fr_{23} & fT_2 \\ r_{31} & r_{32} & r_{33} & T_3 \end{pmatrix} \begin{pmatrix} k\frac{xZ}{f} \\ k\frac{yZ}{f} \\ kZ \\ k \end{pmatrix}$$

$$= k \begin{pmatrix} r_{11}xZ + r_{12}yZ + fr_{13}Z + fT_1 \\ r_{21}xZ + r_{22}yZ + fr_{23}Z + fT_2 \\ \frac{r_{31}xZ}{f} + \frac{r_{32}yZ}{f} + r_{33}Z + T_3 \end{pmatrix} = k \begin{pmatrix} r_{11} & r_{12} & fr_{13} + f\frac{T_1}{Z} \\ r_{21} & r_{22} & fr_{23} + f\frac{T_2}{Z} \\ \frac{r_{31}}{f} & \frac{r_{32}}{f} & r_{33} + \frac{T_3}{Z} \end{pmatrix} \begin{pmatrix} xZ \\ yZ \\ Z \end{pmatrix} \tag{2.18}$$

Finally the relation can be obtained as shown in Equation 2.19. To explain the equation, it means for a pixel on camera plane with coordinate (*x*,*y*) after movement of its coordinating point (X,Y,Z) in the 3D space, the new projected coordinate would be (*x'*,*y'*).

$$\begin{pmatrix} lx' \\ ly' \\ l \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & fr_{13} + f\frac{T_1}{Z} \\ r_{21} & r_{22} & fr_{23} + f\frac{T_2}{Z} \\ \frac{r_{31}}{f} & \frac{r_{32}}{f} & r_{33} + \frac{T_3}{Z} \end{pmatrix} \begin{pmatrix} xZ \\ yZ \\ Z \end{pmatrix} \tag{2.19}$$

The idea for geometric transform based moving object detection is to convert the moving background into a static background by compensating the background moving geometric transform matrix. Even though the 3D geometric transform is a general form, there is a problem that there is a depth term Z in the expression of transform matrix in Equation 2.19. It could be a problem to calculate the depth map using an image captured by a monocular camera. Meanwhile, if the distance between background

scene and camera is large enough, we can be assuming that the background contents are on same 2D plane so that the 2D affine transform is capable of modeling the background movement.

2.3Railway Extraction in Driver Assistant System

2.3.1 Hough transform based railway extraction

Hough transform is a classic straight line extraction method, which was firstly proposed in 1960s [66][67]. The purpose of the technique is to find instances of an object by a voting procedure, which is carried out in a parameter space (accumulator space or Hough space). The parameters are defined by the parametric representation of the line (or object) used to describe the lines (or other shapes) in the picture plane.

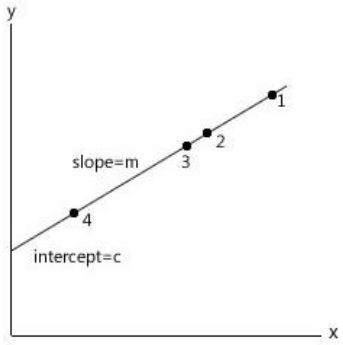


Figure 2-7 Points on straight line in x-y domain

As shown in Figure 2-7, a straight line across points 1-4 can be represented by a slope and intercept form as Equation 2.20.

$y = mx + c$, where m is the slope and c is the intercept (2.20)

At the same time, if we want to fine all possible straight lines across point 1 with coordinate (x_1,y_1). The collection of these can be expressed by Equation 2.21.

$$y_1 = mx_1 + c \tag{2.21}$$

In Equation 2.21, x_1,y_1 are known values and each pair of (m,c) fulfilled can represent a straight line pass through (x_1,y_1). If we convert the x-y domain to m-c domain, the collection of straight lines passing through (x_1,y_1) in the x-y domain is a straight line in the m-c domain. As shown in Figure 2-8. In the m-c domain, straight lines can be formed with (m,c) coordinates that model straight lines in the x-y domain with points 2,3 and 4. Their common line in the x-y domain is the line represented by crossing point amount lines 1,2,3,4 in the m-c domain. In this way, the straight line is modeled.

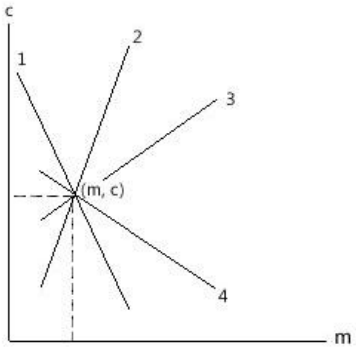


Figure 2-8 Transformed Hough space

To identify a straight line within an image, only pixels with high gradient or pixel on edge should be involved. Therefore edge extraction should be firstly conducted. Figure 2-9 consists an original frame containing railway and its extracted edge map by canny operator. Then for each pixel in edge map, a straight line can be drawn on Hough space. This process is usually described as Hough space cell voting. After all edge pixels in edge map are processed, in the Hough space, a larger voting represents a higher probability that there is a straight line.

Figure 2-9 Original image and extracted edge map

Straight line Hough transforms can be applied in railway extraction. The railway appears as two parallel strips with continuous changing curvature. If the image is segmented into several horizontal layers with proper layer height, the railway component inside each layer can be modeled by a straight line segment. Then the entire railway can be recovered by connecting the straight line segment result of each layer. Figure 2-10 shows an image with (N+1) processing layers, from the vision bottom to the top.

Full edge extraction will be conducted only for Layer 0, the straight lines with positive gradient angle and negative gradient angle are considered separately to find left railway result and right railway result. For each side, the straight lines with Hough voting higher than a predefined threshold will be recorded as candidate result. The distance range between two railways is empirically obtained and can be used to find a pair of railway.

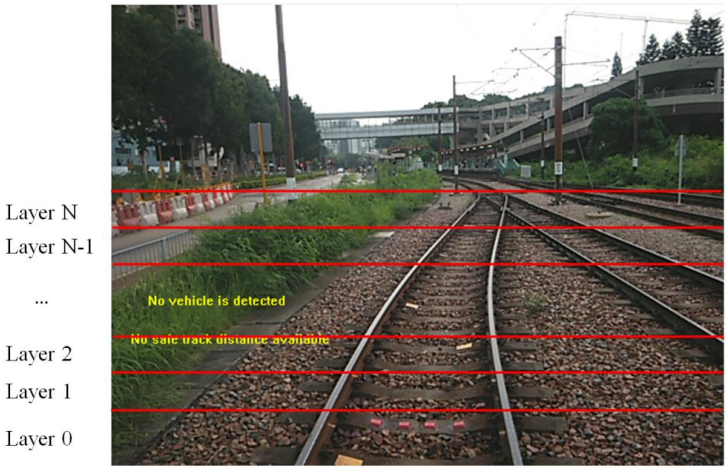


Figure 2-10 Railway Processing Layers Assignment

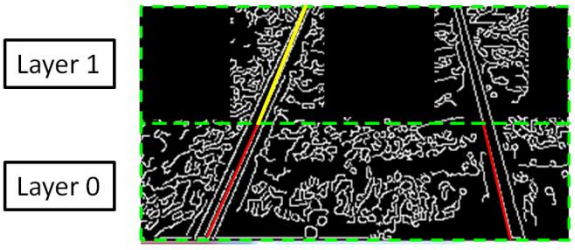


Figure 2-11 Region of interest based processing

For each upper layer, the position and angle of the best resultant straight line in one layer are used to anchor the search region of next layer. Edge extraction is only conducted within the defined search range. For Hough transform, only the straight lines start with the ending point with previous layer are considered. In this manner, the railway is extracted from bottom to top.

From experiments, it is found that the performance of this method depends heavily on edge extraction result. That is to say, if the edge pixels on railway can be extracted clearly, the railway can be well modeled. At the same time, even for the most favorable case, which means edge pixels can be well extracted, the Canny edge detector together with Hough transform modeling is time consuming that the entire

scheme requires around 250ms to process one frame on desktop PC, which is difficult to be implemented in real-time.

2.3.2 Color Histogram based railway extraction

Color cue is another metric to extract railway. For the Hough transform based railway extraction based method, edge extraction is heavily conducted both in bottom layer railway detection and upper layer railway extrapolation, which is time consuming.

Note that the color pattern should be similar for each single railway from bottom to top for each processing layer. For each candidate railway straight line, the color pattern is modeled by color histogram. As shown in Figure 2-12, the red line segment is a candidate result for current layer. The RGB value of all pixels in predefined region, which is usually a parallelogram, near candidate line segment is recorded and then formed a histogram as the color feature.

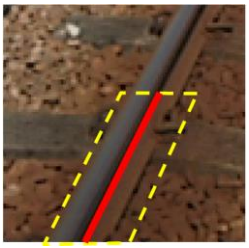


Figure 2-12 Candidate line segment and nearby color pattern

Histogram similarity measurement is needed to calculate distance between two histograms. For the sake of computational simplicity, histogram intersection can be used, which is calculated by Equation 2.20.

$$I(A,B)=\sum_{j=1}^{r}\min\left(A^{(j)},B^{(j)}\right) \tag{2.20}$$

where *A, B* are two histograms to be measured. $A^{(j)}$ is the histogram value of bin *J* in histogram *A*. *r* is the total bin number.

Therefore, the color histogram based railway extraction scheme is designed as discussed below. For Layer 0, the same method is used as discussed in section 2.3.1. Once the railway is detected, the color histograms of each side are constructed. For upper layers, the color histogram for each candidate straight line will be processed and the one gets smallest distance with color histogram template will be regarded as the best candidate line for this layer. The color histogram template will be updated after each layer's processing as shown in Equation 2.22, where α is updating rate.

$$H_{template,updated} = \sum_{j=1}^{r} \left(\alpha H_{current}^{(j)} + (1-\alpha) H_{template,previous}^{(j)} \right) \tag{2.21}$$

2.3.3 Distance transform based railway extraction method

Distance transform is also known as distance map or distance field, which is a term from morphology. Distance transform is classic shape based detection metric, covering the detection of arbitrary shaped object. It is an operator normally applied to binary image, including edge pixels, feature points, etc. The map labels each pixels of image with the distance to the nearest feature pixel. Different distance calculation methods can be used with different computational efficiency, including Euclidean distance, city block distance, chessboard distance, Chamfer distance, etc. Figure 2-13 is an illustration that the left hand side is an input image with labeled feature pixel, and right hand side the distance map calculated by chessboard distance.

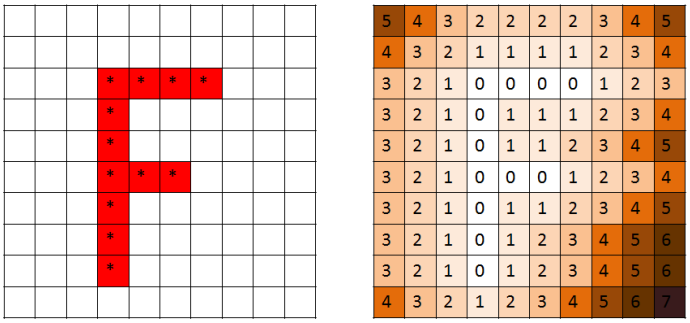


Figure 2-13 Feature pixel and distance map

Graphical example is shown in Figure 2-14 about how distance transform can be used to detect object. Image (a) is a testing image with target object. Image (b) is a template binary image, which is the silhouette of the target object. Image (c) is the edge extraction result of image (a) and image (d) is the distance map of edge extraction result, where a pixel with higher luminance intensity represents larger distance value. Then as shown in image (e), the silhouette template is covered over the distance map. The sum of the distance for the overlapped pixels is calculated as the total distance. For silhouette template position on distance map, the position with the lowest total distance is the position most likely to be the object.

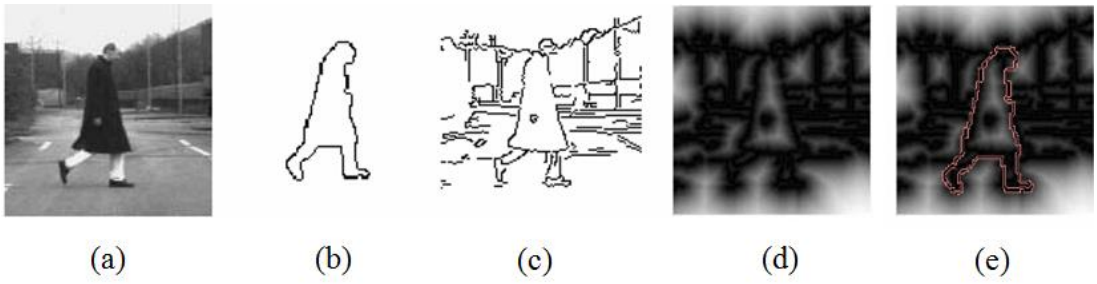


Figure 2-14 Distance map application illustration

As a similarity metric, the distanced transform can be used in railway extraction. In [47], the image with railway is segmented into short distance range and long distance range. Different railway segment lengths are defined for each range, as shown in Figure 2-15.

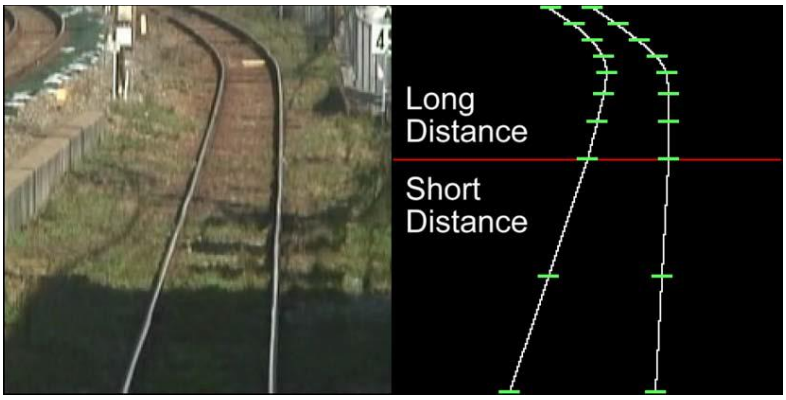


Figure 2-15 Railway segmented into short distance range and long distance range

Templates for different railway shape patterns are defined offline. For each segment, the shape pattern templates are applied on extracted edge map to find the position with smallest distance value or highest normalized confidence. Figure 2-16 is a brief illustration of the process that three templates are used to scan on an edge map. As shown in the example, on position column 242, pattern 0 gets the largest confidence level comparing with other positions and other templates. Therefore it is used as railway result for this segment.

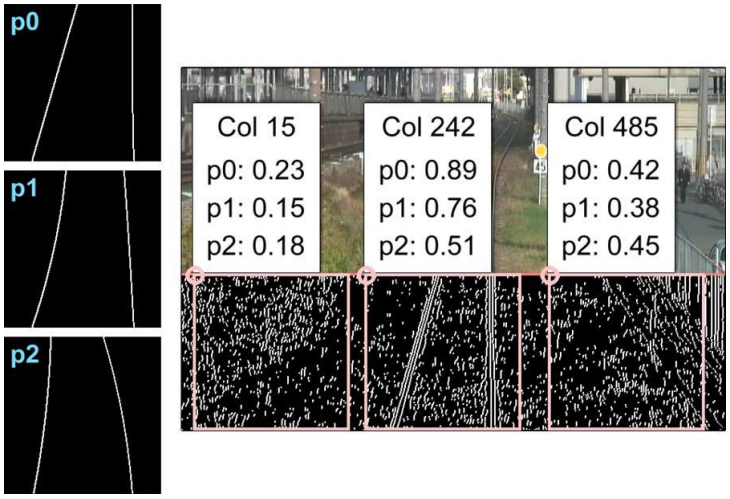


Figure 2-16 Distance value for different template and different position

Chapter 3. Double encoder Surveillance Video Encoding Scheme

3.1 Introduction

From Chapter 1, it is found that in a video surveillance system, the coding efficient improvement mainly comes from saving bits to be used to encode silent background. It is based on the fact that the background region contains a huge temporal redundancy, at the same time less important compared with foreground region with objects. For the background region, either skip mode is used or previous high quality background frame is used as reference frame in inter frame coding. It seems that this problem become a dilemma or paradox that on one hand the background region is regarded as less important, on the other hand these region is still fully or partially coded. Besides, even skip mode is used for all background regions, there is still much overhead in frame level and block level for existing coding standard.

One important characteristic that has not yet discussed in Chapter 2 is that we can wait at the time for rendering, the background region and foreground object can be combined to form complete video display. Based on this understanding, in this chapter we make an audacious investigation and try to explore more bits saving for surveillance video coding with separate background and foreground coding. A double encoder coding scheme is proposed that one encoder is used to encode extracted foreground moving object and another encoder is used to encode condensed background frame. Foreground object is extracted by a background modeling based method. At decoder side, full video can be

rendered by inserting foreground objects and background synthesis. The detailed algorithm and implementation are discussed in the following section.

3.2 Methodology

In this section, our proposed surveillance video double encoder coding scheme is introduced. The fundamental basis for the proposed scheme is that in surveillance video, background region could be less important but make the scene visually completely compared with the precious coding required. It is tolerable to have the background region with slightly lower quality. Therefore, the scheme is designed in this way.

For the sake of discussion, the surveillance video is classified into active segment and silent segment. Active segment means there could be foreground object in scene and silent segment means the scene to have no foreground object.

First, static background is modeled by our previous proposed method and object detector is used to extract the position of moving object in scene. In this way, if a moving object is contained in current frame, the extracted object and corresponding modeled background can be obtained. The modeled background is not coded for every frame. Instead, we used a dynamic skip method to down sample the background frame temporally to obtain a condensed background frame sequence. Then two encoders are used to encode the extracted foreground and condensed background separately. The details about the proposed double encoder schemes are introduced in the remaining part of this section.

3.2.1 Background Modeling and object extraction

In the initial stage of this research, an experiment was conducted to encode video sequence with pure static background content. We found that the coding performance varies a lot between different sequences. If there is a certain degree of Gaussian noise in a video frame, which is a normal situation for some aging surveillance camera or even normal camera under low luminance, the coding overhead gets increased substantially comparing with the scene with less noise. In our simulation, a static scene is captured by a relatively old camera. Then both the original sequence and denoised sequence are encoded with the same configuration. Figure 3-1 is the inter frame coding mode decision illustration for each block for the same frame. The left hand side is the original sequence, and the right hand side is denoised frame. We can see that the mode decision in original frame is more complicated comparing with the denoised frame and more overheads are needed to encode different modes and noise residual. Therefore, a denoised background frame is preferred to be encoded.

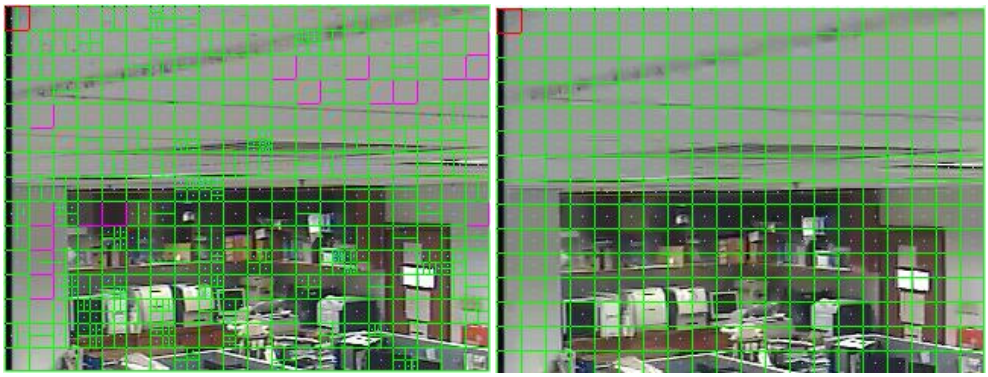


Figure 3-1 Mode selection for static scene with different QP (left QP=22, right QP=32)

We use histogram based background modeling, which is previously proposed by our group in [45], to extract background. In this background modeling scheme, as other background extraction method, a training stage is required in the beginning of a video

sequence. For each pixel position, the pixel intensity in different frames will be recorded and an N-bins histogram H(x,y) will be constructed. For example, we generally use histogram of 16 bins and each bin contains 16 pixel intensity values (i.e. {[0:16], [17:33]…[239:255}). The bin with highest accumulation is most likely to the intensity value for the background, for the reason that background should occupy more time comparing with foreground pixels. Therefore, the latest value in the highest bin is used as the intensity value in the background model of that pixel position.

After training, the intensity histogram for every pixel position is obtained. During processing, the background frame can always be generated by taking the latest pixel value in the histogram bin with highest accumulation for each pixel position. Meanwhile, the histogram for each position is kept updating during the process. Therefore, a longer history in background region can be tracked as well.

The foreground object can be obtained through subtraction between the current frame and background frame, i.e. the difference frame. A robust object detection method in [45] is used as follows. A threshold is set to binarilize the difference frame. Then a 3x3 detector is used to filter out salt and pepper noise and locate the object initial position. This is followed by using an expanding method to find a minimum size rectangle to cover entire object, as shown below in Figure 3-2.

Figure 3-2 Detected object indicated by rectangle box

3.2.2 Foreground Masking

By background modeling and foreground detecting, if there is a foreground object, the foreground object and modeled background frame can be obtained. In our design, separated encoding schemes will be used to encode the foreground frame and background frame.

Figure 3-3 Object with different size under same sequence

There is a natural problem for foreground object encoding as shown in Figure 3-3. Due to the perspective relation, same objects may look small when it is far away and appear

big when they are closed to camera. Object occlusion may also change the object size. For this reason, the size of foreground object to be encoded may vary substantially. However, for all existing video encoding schemes, it is very hard to support encoding video with variable frame sizes. More than that, there may be more than one foreground object in the scene. It is unlikely to encode each foreground object individually by a single encoder.

To deal with this problem, a masked foreground frame (F_m) is generated and is encoded instead of coding the whole picture. The position and size of foreground object is obtained in the previous background modeling step. Then a masked foreground frame is generated by assigning all pixel value outside the object rectangle to 0. As shown in Figure 3-4, the left is the original frame with detected foreground object inside the red rectangle and the right is the masked foreground which is to be encoded by the foreground encoder. The area with artificial unique pattern will not bring much coding overhead. In inter frame coding, the region out of foreground object will be coded straightly by skip mode or mode with no residual.

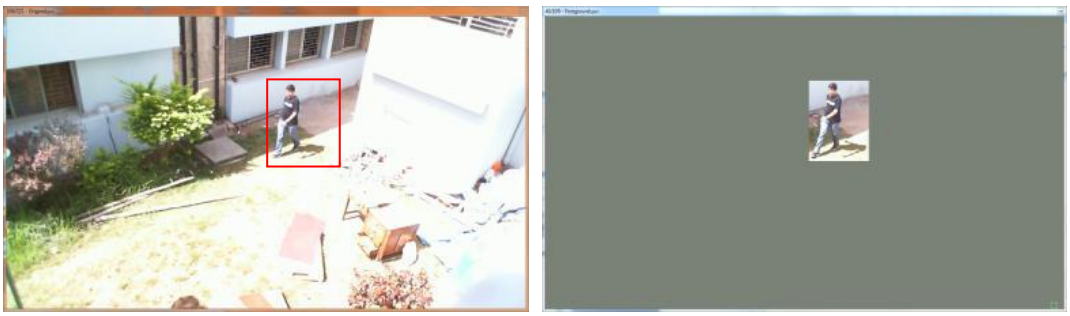


Figure 3-4 Original frame and masked foreground frame

3.2.3 Background condensing

Repeated static background region carries little information, as discussed in the previous section. Thus, it does not need to be encoded frame by frame. The modeled background sequence can be down-sampled at encoder side by skipping. Then at the decoding and rendering side, the condensed background is up-sampled to make up a long sequence. At the same time, a background frame update mechanism should be enabled in case there is changing in background. Based on the above discussion, for background sequence, only the background frames contain information to be updated will be encoded and other frames are skipped. A dynamic background condensing method is designed.

The modeled background at time t is denoted as Bt. The mean absolute difference D_t between two consecutive background Bt and Bt-1 is calculated by Equation 3.1.

$$D_t = \frac{\sum_{P(x,y)\in B} \left| P_t(x,y) - P_{t-1}(x,y) \right|}{N} \tag{3.1}$$

where P(x,y) is the pixel intensity and N is the total pixel number on one frame. Below in Figure 3-5 is difference D_t for 100 consecutive modeled background frames with a scene update at frame index 35. As discussed above, even though the background is supposed to be static, there is still difference between two background frames, which is mainly due to noise and is generated by the CCD camera during capturing. However, these kinds of noises are usually in Gaussian pattern that normally the difference tends to vary in small range as show in Figure 3-5. Meanwhile, when there is a background change the difference will increase instantly. Therefore, the background frame with large difference comparing with normal background should be encoded.

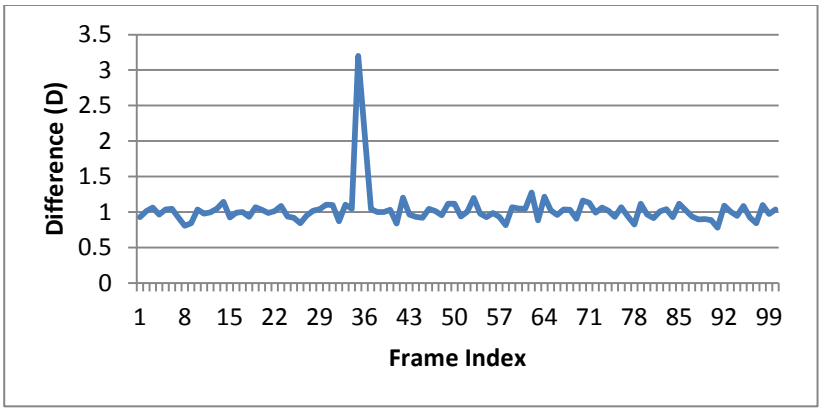


Figure 3-5 MAD between successive frames for background with background changing

The Gaussian like noise pattern is changing under different surveillance cameras and scenes. An absolute difference threshold is not suitable to detect background changing. Therefore, moving average M_t is calculated to track the normal varying range of background difference by Equation 3.2.

$$M_t = \frac{\sum_{k=0...K} D_{t-k}}{K} \tag{3.2}$$

where K is the number of frames past since the last encoded background frame. For the current background frame at time *t*, the ratio between the current difference and the previous moving average will be calculated as Equation 3.3.

$$\delta_t = \frac{D_t}{M_{t-1}} \tag{3.3}$$

If δ_t is larger than predefined threshold 1.5, it indicates the current background frame is significant different from the previous pattern. Then current frame should be coded. Otherwise, this frame should be skipped.

Figure 3-6 below is an illustration of the frame operation for one surveillance video sequence. The original sequence can be divided into silent scene and active scene according to the result of background modeling and foreground detection module. The frames containing foreground object are active frames and the frames only containing background are silent frames. For active frames, the foreground objects are detected and area out of foreground objects is masked to generate a masked foreground sequence. For modeled background sequence, dynamic skip is used to down sample the background to obtain condensed background sequences. The masked foreground sequence and condensed background sequence are subsequently encoded.

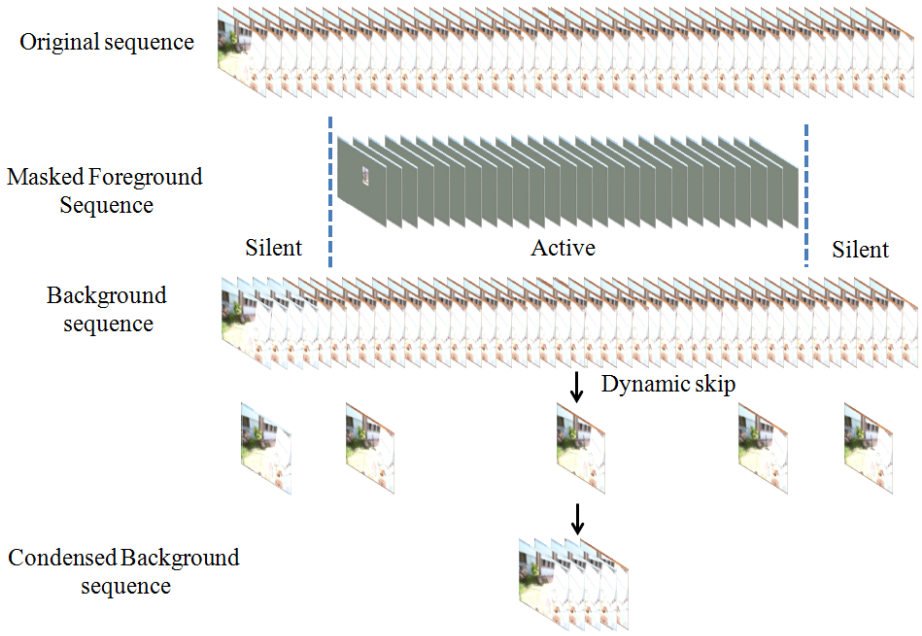


Figure 3-6 Illustration of masked foreground frame and condensed background frame

3.2.4 Double encoder scheme

The proposed double encoder scheme is shown in Figure 3-7. Two encoders are included in the scheme separately, for masked foreground sequence and condensed

background sequence. For both encoding processes, a conventional encoder, such as H.264 or HEVC can be used.

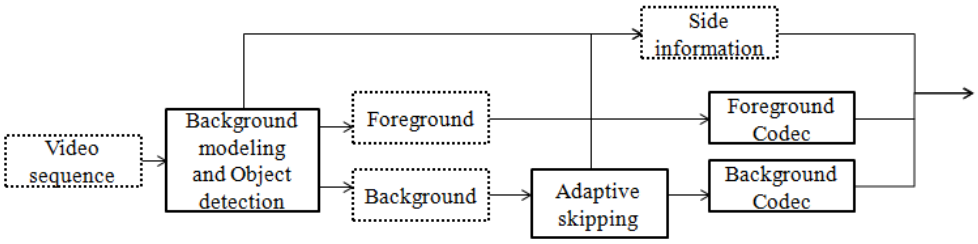


Figure 3-7 Double encoder scheme

It is impossible to recover the entire sequence at the decoder side if only two video streams are provided without extra information. Firstly, during the rendering, the renderer should know whether the modeled background is encoded or skipped at encoder side. Second, the renderer should know how many objects to be are encoded and the position of the object at the masked foreground sequence, which will be used in foreground synthesis. Therefore, after background modeling and object detection module and adaptive skipping module, side information should be recorded and transmitted with foreground and background video stream. The side information syntax is designed as shown in Table 3-1, including (1) whether the background encoded, (2) number of object detected and if there is more than one object detected, (3) position of object.

Table 3-1 Side information syntax

syntax	**# of bits**	**type**
background_encoded	1	Boolean
object_num	8	unsigned(8)
object_info()	variable	
for i = 0 to object_num		
object_left	16	unsigned(16)
object_top	16	unsigned(16)
object_right	16	unsigned(16)
object_bottom	16	unsigned(16)

At decoding and rendering side, side information will be checked for current time t. If the background_encoded is true, a frame will be decoded from the condensed background stream. Otherwise, a repetition of previous decoded background frame will be used as the background of current time index. Then, object_num will be checked to determine whether one or more objects are encoded in this frame. If object_num is not zero, a frame will be decoded from foreground stream. The content within each object region, which is decoded from side information, will overlay background frame to obtain a synthesized frame.

3.3 Experiment result and discussion

Surveillance video sequences are captured in our laboratory with Vivotek PZ6214 PTZ surveillance camera. Some other testing surveillance video is downloaded from internet, which is used in [48]. AVC/H.264 is used as video encoder. The codec used is JM18.6 reference model. To conduct a basic comparison, video fully encoded by a single encoder is used as reference with QP = 22 and 24. For the proposed double encoder scheme, the same encoding setting is used for both foreground and background sequence. Note that the first frame is encoded as I frame and all the rest frames are encoded as P frame. To simulate double encoder scheme, the background modeling, object detection, background frame condensing and foreground masking are processed offline on each surveillance video. Then the two video sequences obtained are fed into the two encoders separately.

Table 3-2 Total bits and saving between conventional codec and proposed codec

		Total bits					
		QP=22			QP=24		
DE503 (3000p)	Foreground(104 Frames)	769929	72.63%		619846	74.60%	
	Background(7 Frames)	290104	27.37%		211008	25.40%	
	Total	1060033		2.48%	830855		9.88%
	Single encoder	42710900		100.00%	8411500		100.00%
	Bit Saving			**97.52%**			**90.12%**
DE503 (221p)	Foreground(104 Frames)	769929	76.54%		619846	77.16%	
	Background(4 Frames)	235960	23.76%		183448	22.84%	
	Total	1005889		26.06%	4117025		60.63%
	Single encoder	3859484		100.00%	1324992		100.00%
	Bit Saving			**73.94%**			**39.37%**
Backyard (221p)	Foreground(109 frames)	3842791	76.26%		3057897	74.27%	
	Background(3 Frames)	1196336	23.74%		1059128	25.73%	
	Total	5039127		36.76%	4117025		62.77%
	Single encoder	13709919		100.00%	6558536		100.00%
	Bit Saving			**63.24%**			**37.23%**

The total number of bits used in each stream from the sample surveillance sequences is shown in Table 3-2. Sequence DE503 (3000p) was captured in our laboratory, with 104 frames containing moving foreground. A shrunk version of this sequence is also used with less background frame duration. Sequence backyard (221p) is the sequence used in [48]. The encoded number of background frames after condensing is shown in the figure as well. Comparing with the conventional single encoder scheme, the bitrate can be significantly saved. For example in sequence DE503(3000p), only 2.48% of bits used for proposed double encoder scheme comparing with conventional coding scheme. Besides, the bit saving percentage for DE503(221p) under QP=22 is 73.94%, and a further saving to 97.52% for sequence DE503(3000p). It indicates that the bitrate saving is positively correlated with the background frame duration. The larger background frame portion, more frames can be condensed. Thus more coding bits can be saved. As result for DE503(3000p) and DE503(221p), only 3 additional background frames are needed to be code to represent the extra 2779 frames.

Another observation can be found from experiment result is that, the bitrate saving is more obvious for high quality encoding setting. For sequence backyard(221p), the bitrate saving increases from 37.23% to 63.24% as the QP changes from 24 to 22. This is because when QP is high, the quantization process will act as a low pass filter to filter out the Gaussian noise in static background frame with less residual remained after inter frame coding. However, theoretically even for the most extreme case that all block in the background frame are coded as skip mode, the proposed scheme is still excel at saving block header by using frame level skip.

In Figure 3-8, the left hand side is the original frame and the right hand side is the synthesized result. There is little difference between the original and synthesized images. For each pair of synthesized output, the content used in the background region is actually the modeled background frame encoded in certain time duration before. It is because real background in current frame is similar to previously encoded one and skipped during background frame condensing. Major information in background region is reserved and error or mismatch within certain degree is tolerable. More importantly, foreground object regions are well recovered by the proposed scheme.

More than that, for video storage and transmission, the proposed scheme is fully supported by current MPEG transport stream standard. The two elementary streams, which are foreground video stream and background video stream, can be multiplexed as one Multi Program Transport Stream (MPTS) with side information packetized and packaged as private data in transport stream. Therefore, the proposed scheme can be easily encoded by current integrated broadcasting hardware.

Figure 3-8 Left: Original frame, Right: decoded and synthesized frame

3.4 Chapter summary

In this research, a double encoder scheme is proposed for surveillance video coding. The fundamental philosophy of this research is that, in surveillance video static background carries less information and the major significance for background region is to constitute a completed frame with foreground object. Therefore, in proposed scheme, background modeling and object detection are firstly conducted to obtained masked foreground sequence and background sequence. Then an adaptive skipping method is used to get a condensed background sequence. Two encoders are used to coding two sequences separately. The bitrate can be significantly saved without affecting on the foreground area and entire frame visual quality.

Chapter 4. Real time moving object detection using motor signal and depth map for robot car

4.1 Introduction

Moving object detection under moving background is more challenging compared with static and dynamic background for the reason that there is intrinsic movement for the entire scene brought by camera moving during the video recording. This situation widely exists and is very useful for robot vision application like obstacle tracking and avoidance [49]. Many vision based schemes have been developed for moving object detection under moving background. For example, the motion clustering based scheme [50] [38] assumes a pixel on moving object moves differently with background, therefore the motion of object and background can be classified based on different motion patterns. Background motion compensation method models the background movement to convert the moving background problem into static background problem.

Some common assumptions are widely used in the current literature that moving object should be small in size compared with the global scene and the movement of the background can be modeled by 2D natural homography matrix, which are usually true for many cases, such as a surveillance site with a long distance pan-tile-zoom camera. However, the situation is different for the robot car scenario. In the vision of a robot car, the background movement brought by camera moving is usually 3D structure [51]. As shown in Figure 4-1, the structure of the scene is complex. When the robot moves

forward, movements of pixels on the stools aside and pixels on the white board cannot be modeled by one homography matrix.

Furthermore, the advantage of vision in robot can be utilized that for robot vision, the movement of background is brought by the movement of camera or movement of robot itself. More fundamentally, the movement is driven by the motor control signal, which is usually available in an integrated robot. In [52-53], the motor signal is used to predict the camera motion represented by a homography matrix. The feature points between successive frames are matched and can be classified according to the predicted background movement homography matrix from motor signal. The method works well when the background structure can be simulated by a plane, not so robust when the background movement is in 3D nature, which is a normal situation when the robot is moving forward/backward.

Figure 4-1 Scene of robot vision, background with 3D structure

To model the 3D background movement for robot car vision, a 3D transform matrix should be used to transform the motion in 3D space to the motion on 2D camera frame. In this thesis, we propose to build a 3D motion projection matrix by making use of both

motor signal and depth map to predict the static motion pattern between successive frames. Then the moving object can be extracted if a different motion relationship is found. We further propose a multi reference frame scheme to increase the robustness and get avoid the "ghost effect", which will explained in the methodology part.

4.2 Methodology

4.2.1 3D Motion projection

Recall in chapter 2, we have discussed the camera model, camera projection and 3D movement project. As shown in figure 2, by using a homogeneous projection matrix in Equation 4.1, a position in 3D space $P(X,Y,Z)$ can be projected on 2D camera CCD plane with coordinate $P(x,y)$, where f is the camera focal length in pixel unit.

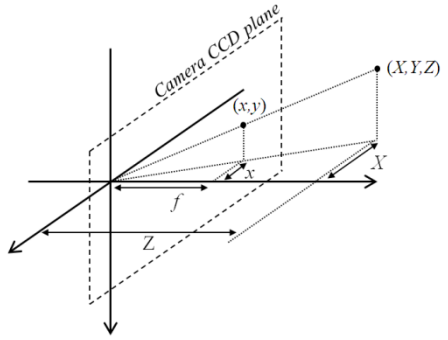


Figure 4-2 3D prospective projection model

The motion in 3D space for the scope of this thesis can be summarized as translational and rotational movements. In matrix form, these movements can be modeled as **P'=RP+T**, where *P(X,Y,Z), P'(X',Y'Z')* are the coordinates before and after the movement. $\mathbf{T}=[T_1\ T_2\ T_3]^T$ represents the translation moving for the object corresponding to the 3D coordinate center on X, Y and Z direction and **R** is a 3×3 matrix defined in

Equation 4.2, φ, θ and ψ represent the rotations on axis-Z, X and Y respectively, as shown in Figure 4-3.

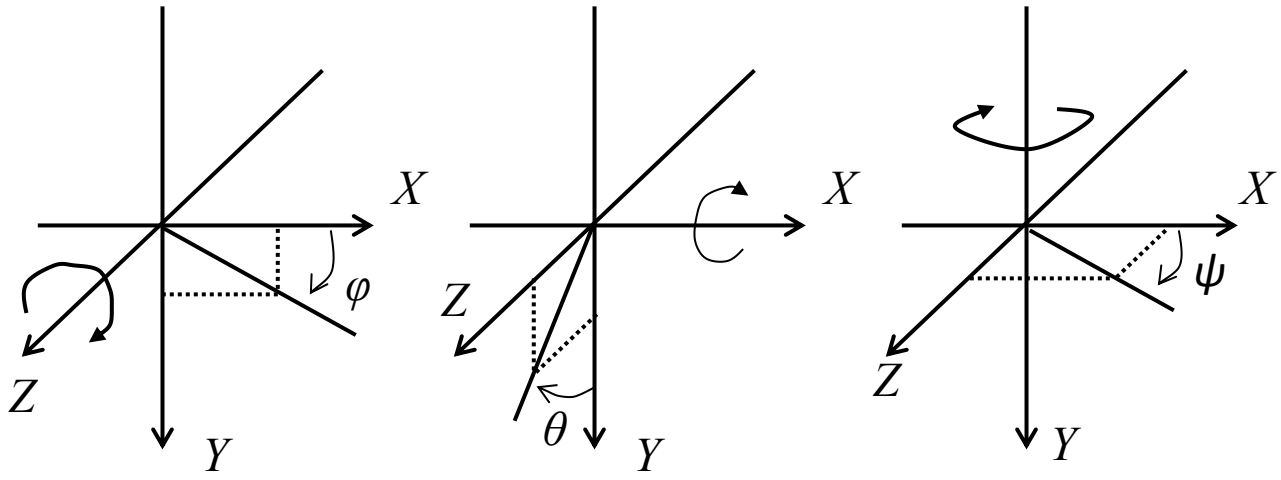


Figure 4-3Rotations on axis-Z, X and Y

$$\begin{pmatrix} lx \\ ly \\ l \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} kX \\ kY \\ kZ \\ k \end{pmatrix} \tag{4.1}$$

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & 0 & \sin\psi \\ 0 & 1 & 0 \\ -\sin\psi & 0 & \cos\psi \end{bmatrix} \tag{4.2}$$

After projecting the movement relationship **P'=RP+T** to the camera CCD plane through the 3D prospective matrix, the formulation in Equation 4.3 can be obtained. It implies that, for pixel with coordinate (*x*,*y*), noted as **Q** in matrix form, on the 2D frame and after the camera movement in 3D space, a new coordinate (x',y'), noted as **Q'** in matrix form, can be obtained through the 3D motion projection matrix, designated as **Q'=MQ**, where **M** is a simplified vector representation of the product of a matrix and a vector as

shown in Equation 4.3. In other words, in the robot vision, the 3D motion projection matrix is the movement of the background affected from the camera/robot moving. Different with the conventional homography matrix that models the background into a 2D plane, the 3D motion projection matrix represents the motion pattern in 3D structure that objects with different distances to the camera will be modeled by different motion vectors.

$$\mathbf{Q'}=\begin{bmatrix} lx' \\ ly' \\ l \end{bmatrix}=\begin{bmatrix} r_{11} & r_{12} & fr_{13}+f\dfrac{T_1}{Z} \\ r_{21} & r_{22} & fr_{23}+f\dfrac{T_2}{Z} \\ \dfrac{r_{31}}{f} & \dfrac{r_{32}}{f} & r_{33}+\dfrac{T_3}{Z} \end{bmatrix}\begin{bmatrix} xZ \\ yZ \\ Z \end{bmatrix}=\mathbf{MQ} \tag{4.3}$$

The movement characteristic of the ground robot car used in this research can be used to further simplify the 3D project matrix, as the motion can be decomposed into forward/backward moving and rotation on the ground. Therefore, the simplified motion projection matrix can be obtained as shown in Equation 4.4.

$$\mathbf{Q'}=\begin{pmatrix} lx' \\ ly' \\ l \end{pmatrix}=\begin{bmatrix} \cos\theta & 0 & f\sin\theta \\ 0 & 1 & 0 \\ -\dfrac{\sin\theta}{f} & 0 & \cos\theta+\dfrac{T_3}{Z} \end{bmatrix}\begin{bmatrix} xZ \\ yZ \\ Z \end{bmatrix}=\mathbf{MQ} \tag{4.4}$$

4.2.2 3D Motion projection construction with motor signal and depth map

Many existing moving object detection under moving background methods seek to find the background motion pattern through the feature point matching. However, for a robot car case, the situation is different. Fundamentally, the movement of the background relative to the camera is the motor signal, which is available in an integrated robot

system. In other words, the required motor signal for constructing the motion projection matrix can be obtained by offline training and calibration.

The term Z in the 3D motion projection matrix refers to the distance between object and camera. In computer vision, the depth value can be obtained through the stereo view matching, which is widely studied and available in the software development kit of many stereo cameras. However, a common drawback for the stereo matching based depth map is that the depth value is often not available or incorrect for flat areas. Therefore a pre-processing on the depth map is necessary. During our research, we found that the incorrect depth value in the depth map is either extreme high or low and the depth value within a certain range is quite reliable. Based on this, a binary mask, which indicates the availability of the depth map, can be obtained according to a lower threshold T_{low} and an upper threshold T_{high} as shown in Equation 4.5.

$$D(x,y)=\begin{cases}1 & T_{low}<Depth(x,y)<T_{high}\\ 0 & otherwise\end{cases} \tag{4.5}$$

For the reason that the configuration of robot and camera is relative stable, the motor signal, depth value and the unknowns in the 3D motion matrix can be trained and calibrated by offline video with corresponding motor signal and depth value. The training video sequences were captured with markers in the background. Motion matrix, depth value and motor signal are recorded and the Least-Square algorithm can be employed to adjust the scaling factors of motor signal and depth value to convert them into θ, T_3 and Z terms in Equation 4.4.

4.2.3 Backward matching

The trained 3D motion project matrix with motor signal and depth value indicates the coordinate relationship for the static background pixel between frames under the known depth value and motor signal. This coordinate relationship can be used to compensate the camera motion. In other words, for each pixel on the current frame, if this pixel belongs to the static background, there should be a similar pixel located at the position obtained from the 3D motion project matrix on the reference frame. Otherwise it should be a pixel on moving object. In the implementation, there is likely random error during the 3D motion project matrix construction. For this reason, we use a 3×3 search window on the reference frame to compensate the system errors. The process of backward matching is shown in Figure 4-4. Each pixel with available depth value will be backward mapped to a position on the reference frame. The yellow rectangle in figure 3 indicates the search window corresponding to the mapped positions.

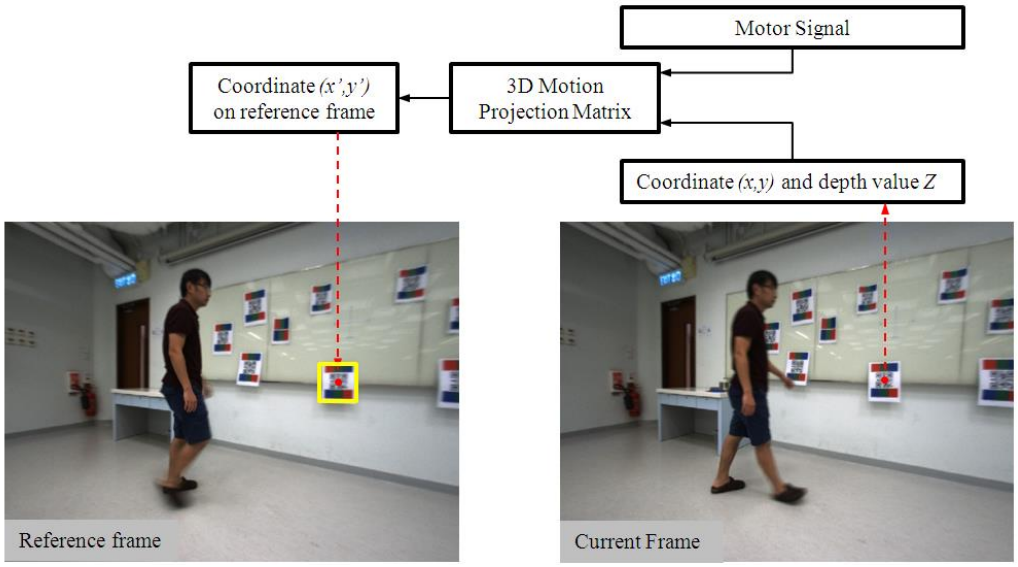


Figure 4-4 Backward matching

The color cue can be used for matching the current pixel and candidate pixel for color change would not be significant in a short time period. However, shadows of the moving object on the floor or background can also affect the matching performance if pure RGB is used to evaluate the similarity. Therefore, color angle [54] is employed that converts the RGB color space of the frame into $c_1c_2c_3$ color space by Equation 4.6. Then the Euclidian distance is used to measure the pixel similarity that two pixels are matched if the Euclidian distance between two pixels is lower than threshold T_{color}.

$$\begin{cases} c_1 = \arctan\left(\dfrac{R}{\max\{G,B\}}\right) \\ c_2 = \arctan\left(\dfrac{G}{\max\{B,R\}}\right) \\ c_3 = \arctan\left(\dfrac{B}{\max\{G,R\}}\right) \end{cases} \tag{4.6}$$

A binary object plane $P_{object,n}$ can then be obtained after the backward mapping as presented in Eq.7. In Eq.7, n refers to the time distance between the current frame and reference frame, *D{p(x,y),p'(x',y')}*represents the Euclidian distance between pixel *p(x,y)* on the current frame and location *p'(x',y')* on reference frame. *Ω* indicates the search window on reference frame, whose center is the coordinate obtained after corresponding 3D motion projection matrix is obtained.

$$P_{object,n}(x,y) = \begin{cases} 1 & \text{if any } D\{p(x,y), p'_{\Omega}(x',y')\} < T_{color} \\ 0 & \text{else} \end{cases} \tag{4.7}$$

4.2.4 Multi reference frame matching

Primary pixel classification result can be obtained after the backward matching. During our implementation, we found that "ghost effect" exists for single reference frame

backward mapping, as shown in Figure 4-5. In Figure 4-5, the background pixels are labeled in white. Pixels in the green rectangle are the correct moving object pixels, and the pixels in the red rectangle, with a similar boundary with the moving object, are the false alarms and we call it "ghost effect". The reason for such effect is that, during the backward mapping, the pixel belonging to background in the current frame may map to the position on moving object in reference frame. Therefore, these pixels will be classified as moving foreground pixels.

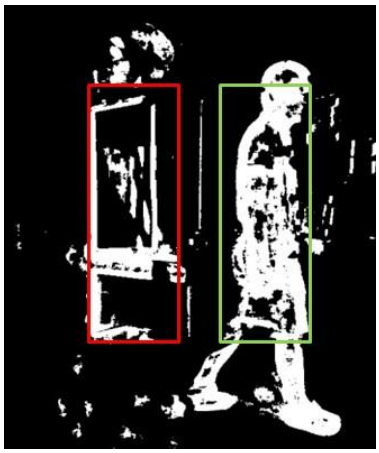


Figure 4-5 Ghost effect under single reference scheme

As the absolute coordinates of the moving object in 3D space change with time, which means that the position of the "ghost effect" depends on the time distance between current frame and reference frame. Therefore a multi reference frame scheme is suggested to moderate the "ghost effect". The final object plane P_{object} is obtained through logic AND operation applied on the backward matching results with different reference frames, which is indicated in Equation 4.8, where N is the total number of reference frames.

$$P_{object}(x,y) = \prod_{n=1}^{N} P_{object,n}(x,y) \tag{4.8}$$

4.2.5 Motor signal and depth map training

The relationship between the motor signal, depth map and 3D motion projection matrix is trained offline. In our implementations, two motor signals for the left wheel m_l and the right wheel m_r are obtained from motor codec and the depth map raw value d is provided by stereo camera software development kit. There are three unknowns (θ, T_3, Z) in the projection matrix in Equation 4.4. We assume that the relationship between (m_l ,m_r , d) and (θ, T_3, Z) can be modeled by Equation 4.9.

$$\begin{cases} \theta = a_1 m_1^2 + b_1 m_2^2 + c_1 m_1 m_2 + d_1 m_1 + e_1 m_2 + f_1 \\ T_3 = a_2 m_1^2 + b_2 m_2^2 + c_2 m_1 m_2 + d_2 m_1 + e_2 m_2 + f_2 \\ Z = a_3 d \end{cases} \tag{4.9}$$

In Equation 4.9, the rotation θ and translation T_3 are modeled by the second order polynomial of motor signal (m_l ,m_r). Z and depth map d are linear relationship, which is sufficient according to our experiment. (a,b,c,d,e,f) are undetermined coefficients.

To solve this equation, ground truth results of pixel movement between frames are obtained by manually adding anchor labels in static background scene. As shown in Figure 4-6, there are labels with designed color patterns stuck on the wall. Between two consecutive frames, for each pattern in the current frame there is a corresponding pattern in previous frame and we note it as a pair. Transform motion matrix of this plane can be obtained by four pairs matching. At the same time, motor signals and depth map on each point are recorded. Therefore one set of solution for Equation 4.9 is obtained. The undetermined coefficients (a,b,c,d,e,f) can be calculated in least mean square manner

when a large number of solution sets are obtained. And in this way, the relationship between motor signal, depth map and 3D motion projection matrix is obtained.

Figure 4-6 Anchor labels in static background scene

4.2.6 Scheme implementation

The multi reference frame backward matching can be conducted exhaustively on every pixel position on current frame. However, an exhaustively search is time consuming and many regions are redundant and less representative within the frame. Therefore, a FAST feature detector [55] is used to select the candidate feature point because FAST can provide enough and representative corner points with favorable processing speed. Then the backward mapping is conducted based on the feature point position.

4.3 Experiment and Result

The scheme is implemented in our robot car to evaluate the proposed algorithm. The motor signal is obtained from the motor encoder directly during the frame capture. A PointGrey Bumblebee2 stereo camera was installed on the robot car. We captured 3 sequences of video in 1024×768 resolution with moving human and different global background motion. Some random markers are placed in the background to increase the available candidate pixels in the background area with depth value. Our experiments are

conducted on a desktop PC with Intel Core2 2.83GHz CPU and only single core is used in the simulation.

For quantitative evaluation, the detection accuracy is defined by $A=A_{object}/A_{total}$, where A_{total} is the total number of feature points classified as moving object points, and A_{object} is the number of feature points classified as moving object pixels, which match the corresponding position on the moving object correctly. The accuracy can be calculated through Equation 4.10, where $P_{object,true}$ is the ground truth mask (in binary form) for the object.

$$A=\frac{A_{object}}{A_{total}}=\frac{\sum P_{object}(x,y)P_{object,true}(x,y)}{\sum P_{object}(x,y)} \qquad (4.10)$$

$$\text{where } P_{object,true}(x,y)=\begin{cases}1 & \text{pixel (x, y) is on moving object}\\ 0 & \text{otherwise}\end{cases}$$

Single reference frame and multi reference frame backward mapping scheme were evaluated based on the accuracy. In multi reference frame scheme, 3 reference frames (short, median and long time interval) were used to conduct the backward mapping. The evaluation results are shown in table 1 below.

Table 4-1 Pixel backward mapping accuracy, single reference frame and multi reference frames

Video Sequence	**Single reference frame backward mapping**	**Multi reference frame backward mapping**
Rotation1	0.587	0.956
Rotation2	0.686	0.936
ForwardMoving1	0.826	0.913
Average	0.697	0.936

From the results in Table 4-1, it is clear that the multi reference frame scheme can boost the accuracy a lot compared with the single reference frame. On average 34.3% accuracy increase can be achieved by involving two more reference frames. To evaluate among the motion characteristics, we can find the accuracy is significant higher for robot car rotation moving, as compared to that of the forward moving. It indicates that the "ghost effect" affects less in the forward/backward robot motion. Lastly, overall 93.6% accuracy for pixels classified as moving object pixels after backward mapping can be achieved with the multi reference frame backward mapping scheme. The accuracy of each frame is shown in Figure 4-7, and the result of which are in line with what we said.

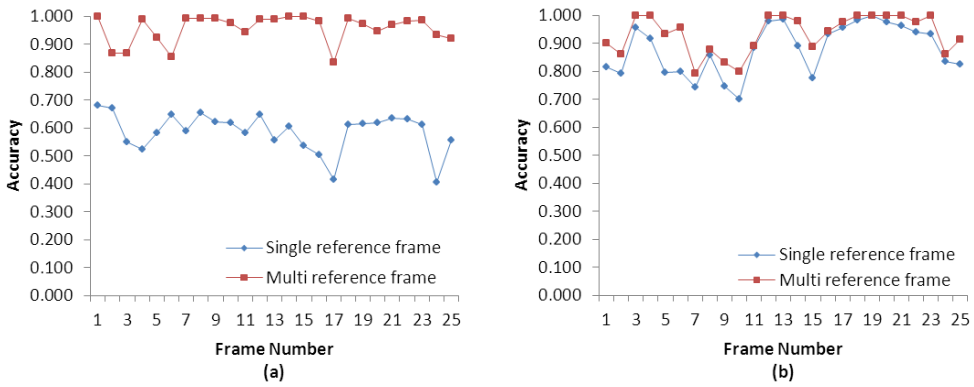


Figure 4-7 Quantitative evaluation result (a) accuracy of sequence with Rotation1 (b) accuracy of sequence with ForwardMoving1

Some of the feature point classification results after backward mapping by two schemes are shown in Figure 4-8. The red dots on the frame are the classified moving object pixels. In the result, we can see that the proposed 3D motion projection method can detect the moving object in a moving background environment. The detection performance is robust, even when the background is on a complex 3D structure, which is seldom studied before, as the result as shown in the third row and the fourth row in

Figure 4-8. Furthermore, the proposed multi reference frame can remove not only the "ghost effect", but also filters out random noise in the background.

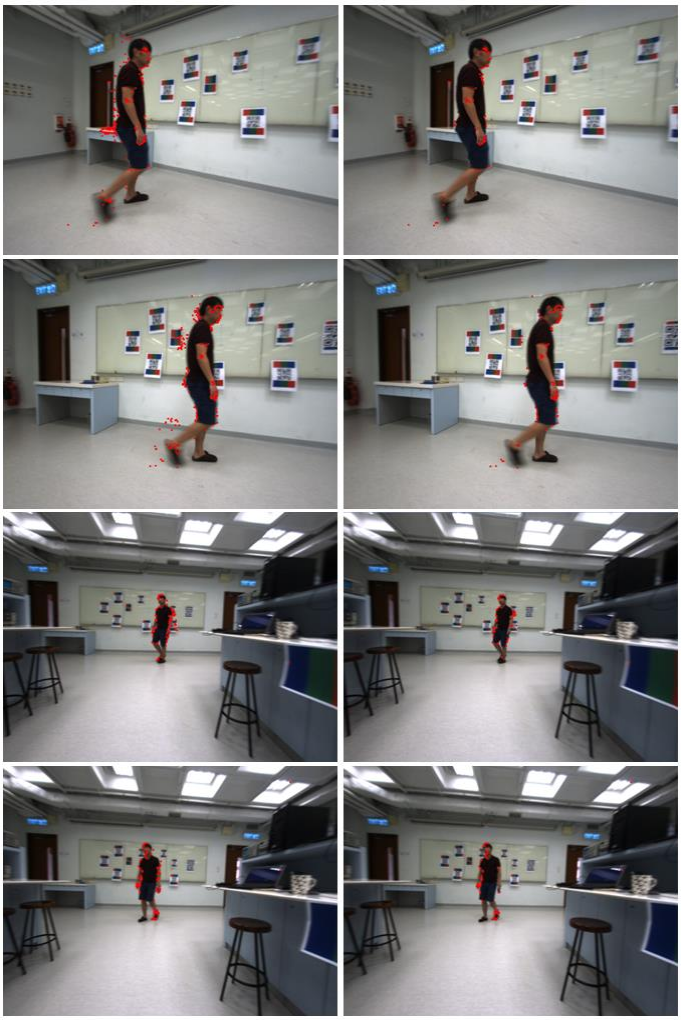


Figure 4-8 Backward mapping result (left) signal reference scheme (right) multi-reference scheme

The average processing time is 54ms/frame, which includes feature point extraction, 3D motion projection matrix construction and multi reference frame backward matching. It indicates that around 18 frame per second frame rate can be achieved, which is sufficient for a real time implementation. Besides, the frame size used in our experiment is relatively large. A down sampling of the frame, such as using Gaussian pyramid, can further boost the processing speed.

4.4 Chapter summary

In this chapter, a novel moving object detection algorithm is proposed to model the background motion in robotic vision by utilizing the motor signal and depth map. It is verified that multi reference frame scheme can further improve the performance. Compared with other vision based algorithms, the proposed algorithm find a practical way on moving background modeling with 3D structure, especially for a moving background with zoom-in and zoom-out action, which has not been studied sufficiently in the literature. Besides, the 3D motion projection matrix is obtained through offline training and calibration, instead of online modeling, therefore, the computation complexity and memory requirement is low, which makes it possible to implement this scheme in real-time.

Chapter 5. Real time railway extraction scheme

5.1 Introduction and discussion

In this chapter, we propose a novel robust connectivity measure method based on gradient angle, for which we are going to define it as Angle Alignment measure. As extrapolation method is suitable for railway extraction, we use Angle Alignment measure as extrapolation metric to evaluate the connectivity of candidate point iteratively. It can be implemented in mobile phone together with other techniques, including front train detection, distance estimation and road sign recognition.

In our application, the extracted railway is used to calculate the region of interest (ROI) to detect the front train. And the system to be implemented is used in a city light rail transportation system, for which a precise camera calibration cannot be guaranteed due to massive installation and daily train operation. Only a rough camera configuration can be used by railway extraction. Besides, different from a high speed railway system, the city light rail gets more complex scenario, with more sharp turns, various road situations, different weather and lighting conditions. At the same time, the railway was also built on different city ground environments, including concrete road, sand/rock road, road with sleeper, etc. And the maintenance of railway in these areas are quite different. Considering our observation above, the following assumptions are made for our system:

-The target railway consists of two parallel strips in front of the train head/camera, with continuous changing curvature;

-The precise camera calibration cannot be used and the appearance of the rails immediately in front of the train may vary a lot with different positions;

-The railway may not have the strong amplitude response to gradient calculation based filter;

-If railway is segmented in layers and the size of each layer is suitable, the railway in each layer (segment) can be modeled by a pair of straight lines, and pixel-wise operation or curve based modeling is not necessary;

-There is no sudden change for rail position and curvature between consecutive frames.

As edge extraction based method cannot achieve a favorable performance, in both extracted railway result and computation complexity. And conventional features, such as gradient magnitude, color patch and contrast, can vary a lot under different scenarios. A further pre-processing or post-processing of these features will only lead to an unaffordable computation increase. During our investigation and enlightened by our previous research [56], we found that gradient angle is a quite robust feature. As shown in Figure 5-1, images on the left hand side are the frames under different environments within one route. The right hand side images are gradient angle maps, where the plotted pixel intensity is the degree of gradient angle. Even though the luminance, color and contrast vary a lot between two images, the gradient angle map gets similar appearance and characteristics. The gradient value is similar in straight railway regions and gradually changing in curved railway regions. In non-railway area, gradient angle pattern tends to be random.

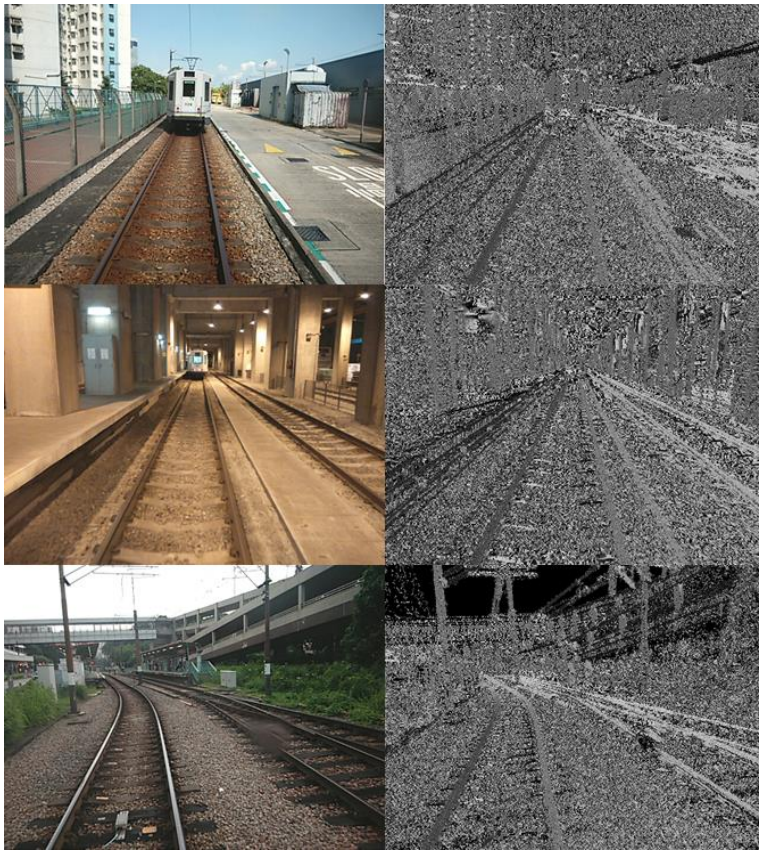


Figure 5-1 Scene under different environment and corresponding gradient angle map

Let us have a careful investigation, the original image and gradient angle map of one image patch cropped from the third scene in Figure 5-1 are zoomed and displayed in Figure 5-2 below. Different from conventional perception that the color is uniform on the railway surface, the color intensity is gradually changing, with direction perpendicular to railway. Therefore, on the surface of railway, there are normally uniform gradient angle values, with a direction the same as the railway direction.

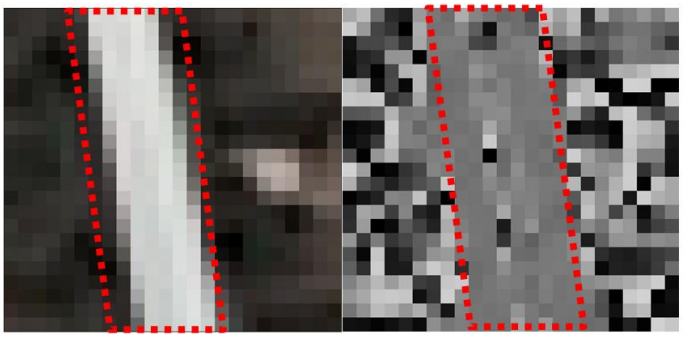


Figure 5-2 Zoomed railway area block and corresponding gradient angle map

Therefore, as the gradient angle map is relative invariant under different environment, railway extrapolation can be conducted in gradient angle map instead of conventional color image domain or gradient intensity domain. Based on the assumptions above, a two-stage railway extraction scheme is proposed. We divide the lower half of the frame into multiple layers, indexed as Layer 0, Layer 1… Layer N, from the bottom to the top as discussed in Chapter 2. In the first step, railway detection is conducted in Layer 0 to find the bottom position of the target railway. For the second step, if a railway can be found, the position and direction of the detected railway from Layer 0 will be used as parameters to extrapolate the railway from lower to higher layers. In both steps, the pixel gradient angle α(i,j) is mainly used, which is calculated by: $\alpha(i,j) = \arctan\left(\frac{G_y(i,j)}{G_x(i,j)}\right)$, and $G_x(i,j)$ and $G_y(i,j)$ are the horizontal gradient and vertical gradient for pixel P(i,j), which is calculated by Equatio 5.1 and Equation 5.2.

$$G_x(i,j) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} p(i-1,j-1) & p(i,j-1) & p(i+1,j-1) \\ p(i-1,j) & p(i,j) & p(i+1,j) \\ p(i-1,j+1) & p(i,j+1) & p(i+1,j+1) \end{bmatrix} \quad (5.1)$$

$$G_y(i,j)=\begin{bmatrix}-1 & -2 & -1\\ 0 & 0 & 0\\ 1 & 2 & 1\end{bmatrix}\bullet\begin{bmatrix}p(i-1,j-1) & p(i,j-1) & p(i+1,j-1)\\ p(i-1,j) & p(i,j) & p(i+1,j)\\ p(i-1,j+1) & p(i,j+1) & p(i+1,j+1)\end{bmatrix} \quad (5.2)$$

5.2 Bottom layer railway detection

The objective of this step is to detect the most possible pair of straight lines, as the possible railway in the bottom layer. We suggest a two-step method to detect the pair of tracks of a railway. The first step is to divide the bottom layer into different blocks, and classify the blocks into rail area blocks and non-railway area blocks. This is followed by obtaining the straight lines to represent the railway in the identified block of the previous phase. The second step is to pair up the straight line results as line pairs, then select the best line pair through projection and color coherence analysis.

5.2.1 Candidate straight line detection

Samples of gradient angle maps under different environments are shown in Figure 5-1. Figure 5-2 shows the zoomed image patches and the corresponding gradient angle maps. We can observe that if a block is in the railway area, which we named as railway block, there is a major gradient angle value distributed inside the block. For a block in non-railway area, named as non-railway block, the gradient angle values are usually randomly distributed and no major gradient angle can be found. This is also clearly illustrated in Figure 5-3. One non-railway block and one railway block are highlighted in yellow square and red square separately. The gradient angle of each pixel is calculated. Then, the corresponding gradient angle histograms (GAH) are built as shown in Figure 5-4, for which the horizontal axis is the gradient angle value quantized by 10

degree. In Figure 5-4, (1) and (2) are GAH that the numbers of pixel are accumulated. It can be seen that the gradient angle distribution is relative flat among all bins in non-railway block GAH Figure 5-4 (1). And in railway block GAH, Figure 5-4 (2), there is a peak, or refer it as a dominant bin, around bin 7, which means there is a majority number of pixels with gradient angles between 60 degree and 70 degree. The dominant bin is even more distinct if the accumulated value is gradient magnitude instead of number of pixel, as shown in Figure 5-4 (3) and (4).

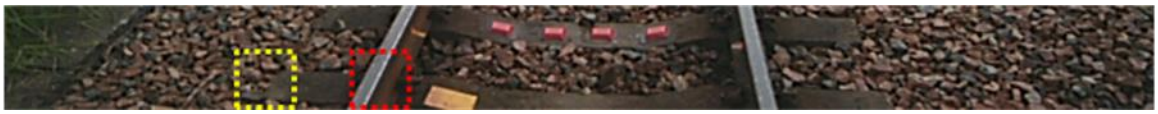


Figure 5-3 Non railway area block and railway area block

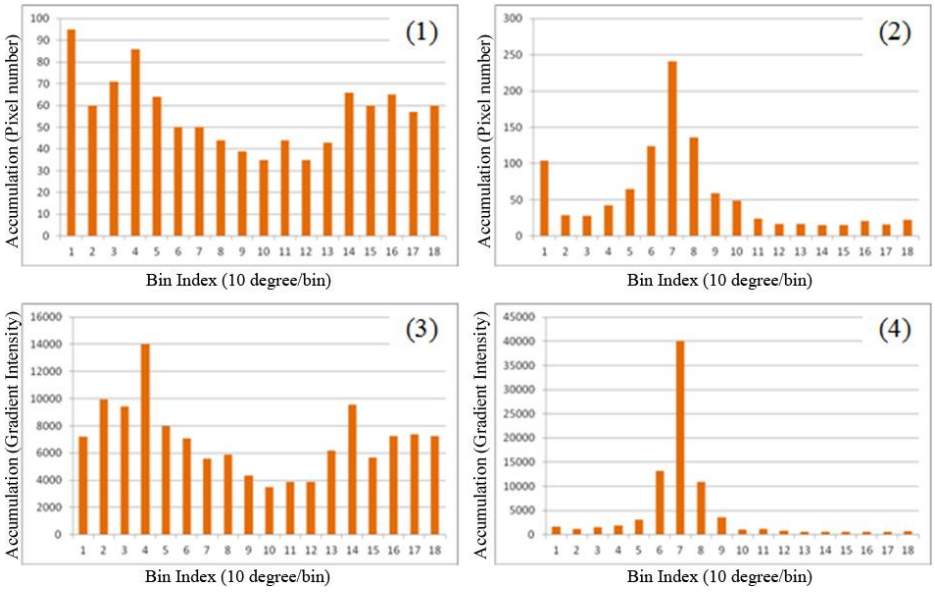


Figure 5-4 Gradient Angle Histogram for Non railway area block and railway area block

Based on the above observation, a block based processing is designed for the bottom layer processing. For each pixel, the Sobel kernel is used to get the horizontal gradient

and vertical gradient. Similar to [57] [58], gradient angle histogram of each block is constructed. In the histogram, the gradient magnitude of each pixel is accumulated. At the same time, during the construction of gradient angle histogram, the total bin accumulation A_{Total} is recorded, as a benchmark to evaluate the gradient response within one block. As a priori knowledge about track direction, the direction of the rail should be within a certain range, which is between 60 degree and 140 degree. The camera is mounted on the front right side. Therefore the range setting is nonsymmetrical. Hence only the peak within this range will be considered. To check whether there is a strong peak in the GAH, consecutive m bins with largest accumulation summation is found, and denoted as A_{max}, where m is an important parameter, which will be explained in later part of this section.

Then, if the ratio between the peak accumulation A_{max} and histogram total accumulation A_{Total} is larger than the predefined threshold α, this block will be regarded as an ***initial rail area block***. The threshold α depends on the size L of a block. If L is too small, the number of gradient angle samples within the block would not be sufficient, such that the GAH pattern of a block in non-railway block may be similar to the pattern of a railway block. If L is too large, the rail will occupy too small region within block that the block between railway block and non-railway is less distinguishable based on GAH. According to our experiment, we found that, if the width of a single rail is w, the best performance can be achieved when block size is around $3w$. Under this block size, the threshold α we used is 0.3.

For an initial rail block, pixels belong to the m bins with peak accumulation, referred to as dominant m bins, will be plotted back as feature pixel, and referred to as dominant

gradient angle map *DG(m)* of this block. Figure 5-5 is an example of 1-bin dominant gradient angle map *DG(1)* of the bottom layer. The blocks with white pixels inside are the initial rail area blocks and a white pixel is a pixel belonging to dominant 1 bin. Other blocks are non-rail area blocks.

Then, a forward-mapping Hough transform [59] is conducted on DG and the Hough space cell with the largest accumulation (C_{hough}) in a block is found. Only straight lines with close angle range from the most dominant bin (10 degrees) will be processed. Therefore, there are very few candidates in Hough space to be processed so that the computational complexity in forward-mapping Hough transform is decreased. For each block, the maximum Hough accumulation within each block is the block size *L*, i.e. one pixel per row. Therefore, *L* can be used to evaluate the accumulation. Within each block, the cell with the highest accumulation is *L* in the Hough domain. In this step, only blocks with accumulation larger than *L/2* will be further classified as possible rail area blocks, and the corresponding straight line will be recorded.

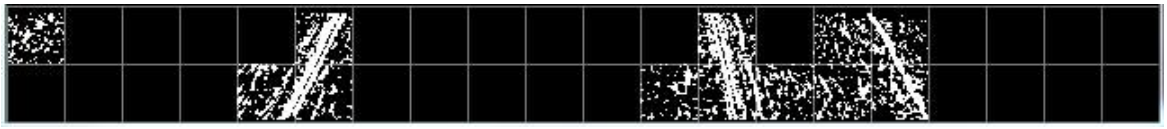


Figure 5-5 Dominant gradient angle map (block size =32 pixel)

During implementation, different scenarios are found, including the train with high speed with lots of directional motion blur in the camera vision and night situation with very low luminance. The dominant gradient angle map *DG(m)* is not always favorable if the same continuous bin number *m* is used. As shown in Figure 5-6, if m=3 is used in the initial railway block classification step, for outdoor fast moving situation, there is

too much noise in the *DG*. The screening performs weakly because most pixels in bottom layer are under motion blur. A favorable DG can be obtained by decreasing the consecutive bin number used. However, for night situation, if m is set to be too small, there will be too few pixels to form the dominant bins. The consecutive bin number should be increased to deal with this scenario. Therefore, an adaptive method is designed for this candidate straight line detection process.

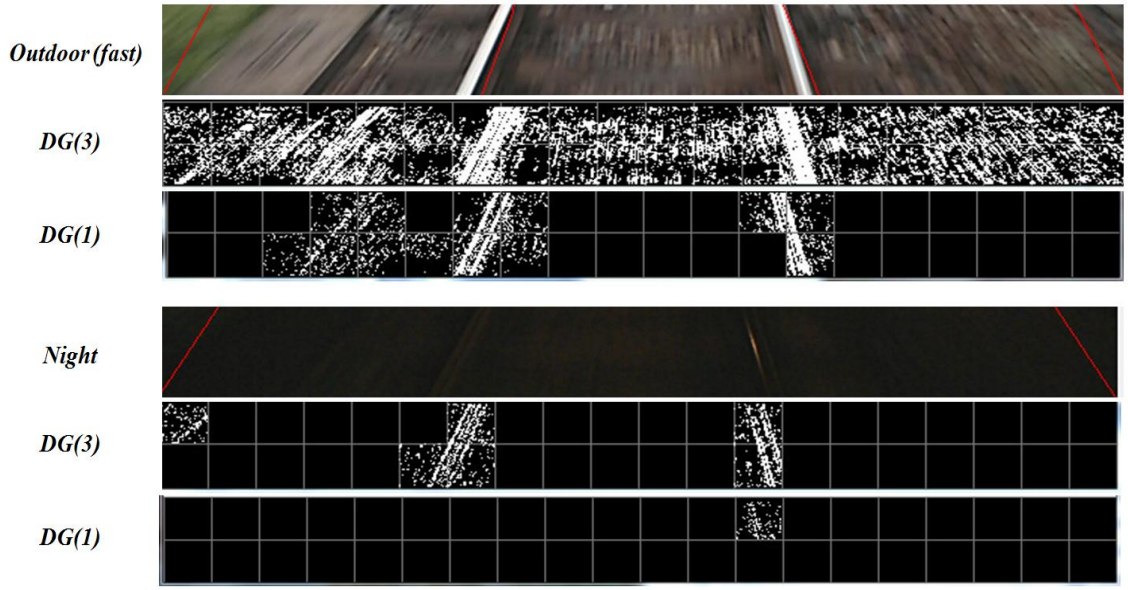


Figure 5-6 Dominant gradient angle map (DG) with different bin number considered under difference situation

As shown in the flowchart in Figure 5-7, a two steps classification and verification for candidate straight line is designed. The consecutive bin number m=1, is used in the first round of initial railway block classification. After the Hough transform verification, the number of railway blocks will be checked. If there is not sufficient number of rail area blocks verified (less than 5 railway blocks found), which means the scene probably lack of luminance, the target bin is not well distinguishable in the gradient angle histogram.

The scheme will re-investigate the gradient angle histogram with continuous bin number *m*=3. In this manner, both scene with directional blur and scene lack of luminance can be handled.

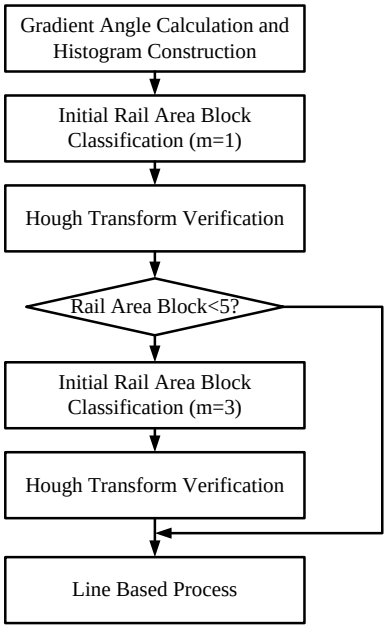


Figure 5-7 Workflow of adaptive candidate straight line detection

At the end of this step, the Hough cell with the highest accumulation in the verified railway block will be recorded as candidate railway straight line for following processing.

5.2.2 Line pair selection

The resultant lines from previous step include not only desired straight lines belongs to the target railway, but also unrelated straight line simulated by other environment elements. As railway is a pair of straight lines with certain relationship, straight lines from previous step are used to form line pairs. All possible line pairs are recorded and in this step, the most suitable line pair will be regarded as the bottom rail result. Two

simple and robust verifications are suggested to find out the target straight lines/rails, and our next process involves projection relationship and color coherence.

(1) Projection Relationship. As stated in the introduction that, the camera cannot be precisely calibrated due to mass production. Therefore, the width between two rails and the angle of rail are not constants. That is to say these two parameters cannot be utilized directly. Instead, the relation between camera and rail can be found based on statistical method from plenty of training video sequences, which can be used to filter out less possible line pair candidates. As shown in Figure 5-8, a point in 3D space with coordinate (*X,Y,Z*) is projected on camera CCD plane with coordinate (*x,y*). The perspective transform is conducted according to Equation 5.3, where *f* is the focal length of the camera [60]. Thus, in Equation 5.3, focal length (*f*) and camera height(*Y*) can be roughly obtained. For each pixel on the frame, the corresponding coordinate in top view (X-Z domain) can be obtained.

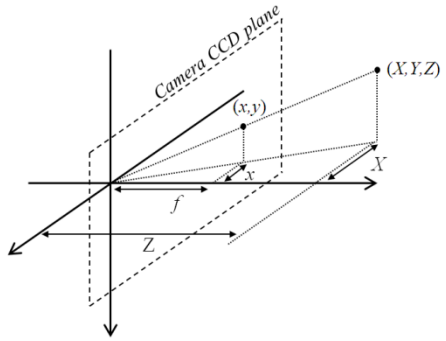


Figure 5-8 3D space projection on camera plane

$$\begin{pmatrix} lx \\ ly \\ l \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} kX \\ kY \\ kZ \\ k \end{pmatrix} \tag{5.3}$$

The upper image in Figure 5-9 shows one sample detected candidate line pair. End points of each line in bottom layer are labeled as $P_{left,top}$, $P_{left,bottom}$, $P_{right,top}$ and $P_{right,bottom}$. Then the coordinates are transformed from camera planar plane to the top view (X-Z domain) in 3D space where X axis is horizontal axis of the camera and Z axis representing depth/distance. The transform line pair are plotted in lower image of Figure 5-9, with notation $P'_{left,top}$, $P'_{left,bottom}$, $P'_{right,top}$ and $P'_{right,bottom}$. The original point in the X-Z domain is camera projection position and the oblique dash lines are the view range of the camera. Then the projected left line and right line are further extended to intersect with X-axis and mark the intersection point as $P'_{left,extend}$ and $P'_{right,extend}$. We use $|P_1,P_2|$ to indicate the distance between two positions P_1 and P_2.

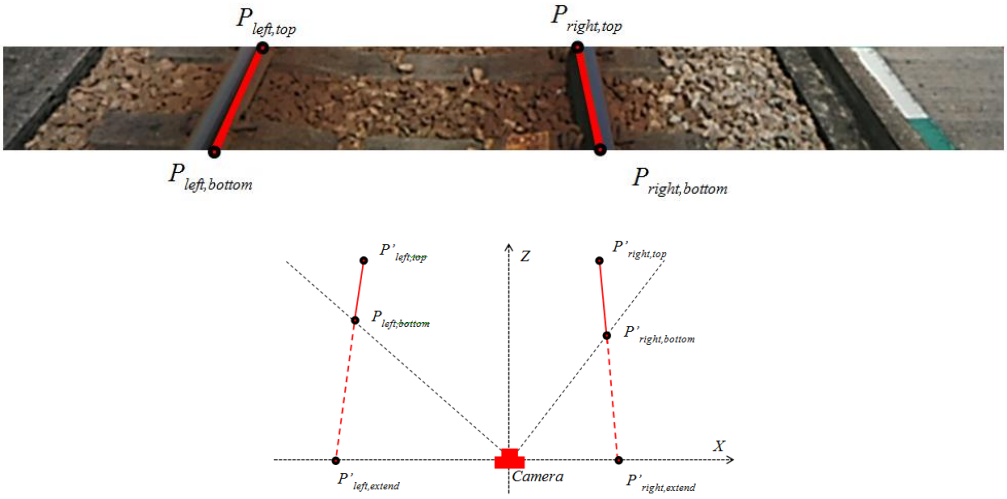


Figure 5-9 Detected line pair and Projected line pair on top view

The relationship between these two points and the camera is empirically obtained through offline training in Equation 5.4 that only a line pair satisfied the relation below will be further processed.

$$\begin{cases} 0.22 < \left| P'_{\text{left,extend}}, Camera \right| < 1.65 \\ 1.25 < \left| P'_{\text{left,extend}}, P'_{\text{right,extend}} \right| < 1.65 \end{cases} \tag{5.4}$$

where $|P_1,P_2|$ is used to indicate the distance in meter between two positions P_1 and P_2 and the distance is measured in meter.

(2) Color Coherence and confidence measure. Railway color varies a lot due to luminance changing, different maintenance condition, etc. Therefore, only using color cue is not reliable to detect the railway. However, color tends to be similar within a small region on railway, and let us refer it to as color coherence. The color pattern can be used to provide further verification and confidence measure. As shown in the samples, the red straight line is the straight line resulting from the previous step. A parallelogram can be constructed based on the result. The width of the parallelogram should be wide enough to cover the railway. We further split the parallelogram into upper region and lower region, as shown in Figure 5-10. It can be seen that the color pattern is coherent within railway candidate track but may not be coherent for non railway candidate lines.

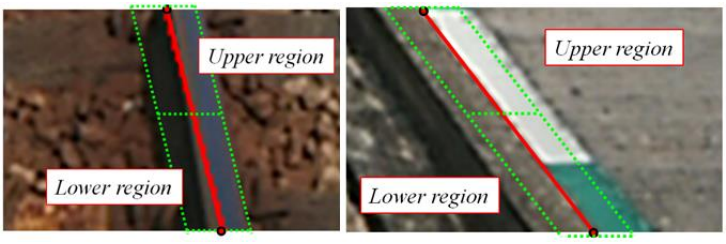


Figure 5-10 Lower region for railway candidate line and non railway candidate line

To measure the color coherence, color histograms of the upper region and lower region corresponding to detected candidate line will be constructed, denoted as H_{upper} and H_{lower}.

For the sake of computation complexity, Histogram intersection $\tilde{I}_{color}(H_{upper}, H_{lower})$ is used to measure the similarity between two histograms [61], as shown in Equation 5.5.

$$I(A,B) = \sum_{j=1}^{r} \min\left(A^{(j)}, B^{(j)}\right) \tag{5.5}$$

where *A, B* are two input histogram with same bin number, $A^{(j)}$ is the histogram value of bin *J* in histogram *A*. *r* is the total bin number. *I* is the similarity. In our implementation, two color histograms are normalized with 100 unit total accumulation in each histogram, and normalized histogram similarity $\tilde{I}_{color}$ is calculated through Equation 5.6:

$$\tilde{I}_{color}(A,B) = \sum_{j=1}^{r} \min\left(\tilde{H}_{upper}^{(j)}, \tilde{H}_{lower}^{(j)}\right) \tag{5.6}$$

where $\tilde{H}$ is the normalized color histograms. 16 bins are used for each color channel therefore the total bin number *r* is equal to 48. In this way, the range for color coherence $\tilde{I}_{color}$ is (0 – 100). In Fig.13(1), it can be observed that the color pattern between upper region and lower region is similar and a higher similarity will be obtained. And for Fig.13(2), the bottom region get quite different color pattern with upper region. Therefore, the color histogram similarity $\tilde{I}_{color}$ can be regarded as the confidence level in color coherence.

In block based operation, the Hough space cell with highest accumulation C_{hough} is recorded. Normalized accumulation $\tilde{C}_{hough}$ can be regarded as the confidence level in straight line aspect. Then the two confidence measures are further fused into one confidence level C_{line} by Equation 5.7.

$$C_{line} = \alpha \times \tilde{I}_{color} + \beta \times \tilde{C}_{hough}, \tag{5.7}$$

where $\tilde{C}_{hough}$ is normalized Hough transform accumulation. α and β are the corresponding weight, and $\alpha + \beta = 1$ to normalize the result straight line confidence *C*. A higher weight is recommended to assign to α for the reason that once there is color histogram mismatch, a relatively larger penalty should be given. Besides, the Hough cell accumulation has already been partially used in previous block verification step. In our implementation, we used a setting $\alpha = 0.7$ and $\beta = 0.3$, which is obtained experimentally.

In this manner, the confidence level for each single straight line is well defined. In our earlier investigation, we tried to match the color histogram between left rail and right rail within a line pair to define the line pair confidence level. However, we found that in many situations of the scenarios, the color patterns are quite different between the real left rails and right rails. Therefore, we define the line pair confidence C_{pair} in Equation 5.8 by taking the lower value between the left rail confidence and right rail confidence.

$$C_{pair} = \min\left(C_{left}, C_{right}\right) \tag{5.8}$$

If C_{pair} is smaller than a predefined threshold T_{pair}, it is very likely that there is no railway in current frame or the detected result is not reliable. Otherwise, the line pair with the highest confidence level will be regarded as the rail in the bottom layer.

5.3 Upper layer railway extrapolation

The first assumption of this research is that target railway is two parallel strips in front of the train head/camera, with continuous changing curvature. Besides, we found that the conventional edge extraction based method and color segmentation method is not stable and reliable both in detection result quality and computational complexity. Therefore, we use extrapolation method in this research. Some of previous research [31][32][62] has already implemented extrapolation based method by detecting the track layer/segments based. We propose a new extrapolation method, by formulating a novel metric entitled as Angle Alignment Measurement, to measure the connectivity for candidates in each layer. The whole extrapolation can be processed iteratively. In this section, we will first introduce Angle Alignment Measurement, then how Angle Alignment Measure can help to extrapolate iteratively will be provided in the second part.

5.3.1 Angle Alignment Measure

In geometry, as shown in Figure 5-11, for a point on a continuous curve, the gradient of the curve on that point is the slop θ of the tangent line on that point. At the same time, the gradient can be calculated by differentiation method, that for a small segment, when the change in x direction (dx) and change in y direction (dy) are small enough, then the gradient of points on this small segment should be equal to dy/dx. Therefore, it can be expressed in this way, that if a curve is continues and smooth (derivable), for each segment that is small enough, the gradient of the points on that segment should be equal to the change in y direction divided by change in x direction.

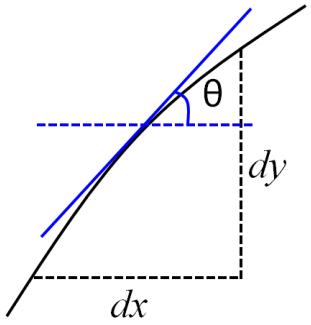


Figure 5-11 Tangent line on continuous curve

The statement above can be converted into image processing and utilized through gradient angle map in our railway layer based extrapolation. Figure 5-12 shows the original image (left) and the gradient angle map (right), on which the intensity is the gradient angle value. As discussed in previous section, the gradient angle value in a rail region is similar. In this scenario, for each pixel, the slop θ of the tangent line can be represented by the gradient angle value. At the same time, extrapolation can be conducted based on the previous layer. Therefore, if the layer size is properly assigned, as the discussion in previous paragraph, the technique can be implemented in following way.

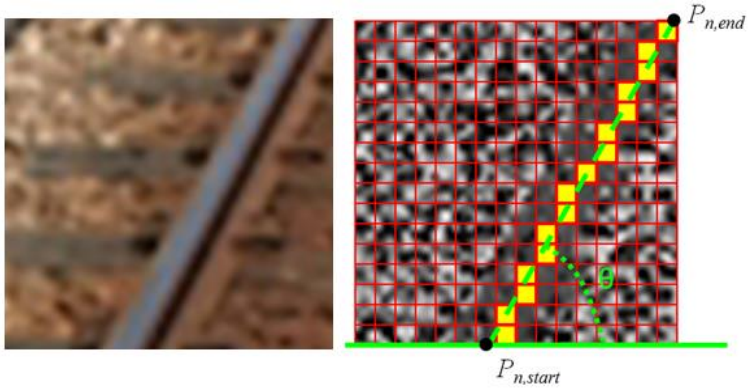


Figure 5-12 Railway and its gradient angle map

As shown in Figure 5-12 right, for the current operating layer n, assume that the extrapolation end in Layer n-1 is $P_{n-1,end}$. The extrapolation starting point $P_{n,start}$ should be the ending point of previous layer $P_{n-1,end}$. Then on the top of the current layer n, for each candidate ending point $P_{n,end}$, a straight *line i* can be draw to connect $P_{n,start}$ and $P_{n-1,end}$, indicated by the green dash line in Fig.15. The slope θ_i of *line i* can be calculated by Equation 5.9.

$$\theta_i = \arctan\left(\frac{P_{n,end}(y) - P_{n,start}(y)}{P_{n,end}(x) - P_{n,start}(x)}\right) \tag{5.9}$$

where *P(x)* and *P(y)* are the coordinate of point *P*. Therefore, according to what was discussed at the beginning of this part, if $P_{n,start}$ and $P_{n-1,end}$ are well connected, for each pixel p(x,y) on candidate *line i*, the gradient angle *α(x,y)* of pixel *p(x,y)* should be equal to θ_i. Otherwise, the connection between $P_{n,start}$ and $P_{n,end}$ is less reliable. We name this feature as Angle Alignment. To measure the Angle Alignment, we define a metric as Angle Alignment Score (*AAS*) for candidate *line i* as below in Equation 5.10 and Equation 5.11.

$$AAS = \frac{\sum_{p \in Line\,i} \mathrm{D}(\mathrm{p}, \theta_{\mathrm{i}})}{\mathrm{N_i}} \tag{5.10}$$

$$D(P(x,y),\theta) = \begin{cases} 1 & |\alpha(x,y) - \theta| < t \\ 0 & else \end{cases} \tag{5.11}$$

where $\boldsymbol{N_i}$ is the total number of pixel on candidate *line i*. A predefined threshold *t* is used here to provide some tolerance for the matching between pixel gradient angle and line slope.

In this manner, if *AAS* of *line i* is high, the starting point and ending point are very likely to be well connected. And if the AAS is low, the current ending point is not well connected with starting point.

5.3.2 Iterative railway extrapolation

In our scenario, the length of railway can vary a lot. When another train is in the front of this train (hence in front of the camera), the railway will be very short in vision. Therefore, a proper termination method should be designed at the same time. In the rest of this section, how the Angle Alignment measurement will be used to extrapolate the railway for one layer will be discussed.

The proposed Angle Alignment Measure is featured at using an interactive procedure. Recall the representation of the current *layer i*. The ending points $P_{i-1,left,end}$ and $P_{i-1,right,end}$ together with their slops $\theta_{i-1,left}$ and $\theta_{i-1,right}$ from the previous layer are known, which will be used as the starting points $P_{i,left,start}$ and $P_{i,right,start}$ of current layer.

***Step 1: Search ROI Definition and gradient angle calculation*.** In our layer assignment, the pixel shift between two consecutive layers is restricted to no larger than 50 pixels or the slop difference is no higher than 50 degree. Therefore, for each side, the region of interest can be defined accordingly below:

$$ROI_{i,side,left} = min(P_{(\theta i-1,side-50)}, P_{i-1,side,end} -50),$$

$$ROI_{i,side,right} = max(P_{(\theta i-1,side+50)}, P_{i-1,side,end} +50),$$

where $ROI_{i,side,left}$ and $ROI_{i,side,left}$ are the left and right of one *side*(*left/right*) for *layer i*. $P_{(\theta i-1,side-50)}$ is the point on top of *layer i* that the candidate line linking between $P_{i,side,start}$

between P $_{(\theta i\text{-}1,side\text{-}50)}$ has slope 50 degree smaller than $\theta_{i\text{-}1,side}$. And $P_{i\text{-}1,side,end}$ -50 is the point on top of layer i that shift left by 50 pixel corresponding to the ending point of previous layer of this side.

As the region of interest has been defined according to the extrapolation result from previous layer, to save the computation time, the gradient angle calculation is conducted only within the region. We can use Sobel operator to calculate the gradient magnitude and gradient angle as discussed previously. As shown in Figure 5-13(1), the blue dash rectangle is the candidate line windows, which is used as region of interest for gradient angle calculation. The intensity plotted within blue dash rectangle is gradient angle degree.

Step 2: Direct extension verification*.** In most scenarios, the railway is a pair of long straight lines in front with possible turning at the tail. The extrapolation result is very likely to have the same slope with result from previous layer. Therefore, in this step, the ending points on top of the layer, which are directly extended from the result slope of previous layer, are denoted as $P_{i,left,extend}$ and $P_{i,right,extend}$. The connectivity between the extended point and starting point will be checked by Angle Alignment measure. For each side, if the corresponding AAS is higher than a predefined threshold T_{verify}, the result will be recorded and a flag of this side will be marked as *verified*. If both sides are verified, no other candidate will be processed further and the scheme will directly jump to ***step 4.

Step 3: Other candidate processing

(1) One side verified. The verified result will be used as an anchor to find the other side of the pair. Assume that the left side extension is verified in previous step to illustrate the process here. Then the right side candidate lines are found according to left verified result. First, we define the distance between $P_{i,left,start}$ and $P_{i,right,start}$ as the bottom distance *d*, which is used to define the range of other side ending point. Because in bird view/top view, two railways are parallel. In camera vision, there is a good relationship between the starting points and the ending points in each layer. Empirically, the difference between the top distance and bottom distance should be within 10 pixels to 16 pixels. Therefore, the right ending point is within the range $(P_{i,left,extend}+(d-16), P_{i,left,extend}+(d-10))$. To provide further tolerance, the corresponding starting point will have a few more tests by making one pixel shift ($P_{i-1,right,end}$-1,$P_{i,right,end}$+1). In this manner, a point at the lower part of *Layer i* may connect to 7 points of the upper part of this layer as shown in Figure 5-13(3). Note that there are 3 starting points at the lower part of *Layer i*. In this stage, the Angle Alignment measure will be conducted on 20 straight lines, except the one verified in the previous step. The line with the highest AAS is regarded as the best right railway for this layer. However, if the best AAS result for this side is lower than threshold $T_{terminate}$, it is considered that the extrapolation starts the lack of reliability. Therefore the terminate flag will set and extrapolation will terminated after this layer. The process is similar if the right side line is verified in the beginning of this step.

(2) Both side failed in verification. If both sides are failed in the verification step, a local range search will be initiated. In our scheme, the search starts from left railway.

All positions within the left side ROI are considered as possible ending point candidates. Similarly, more tests with one pixel shift will be given to the starting point as well. The line with the highest AAS result will be regarded as the best extrapolation for the left side rail. In this step, if the highest left side AAS is lower than threshold $T_{terminate}$, the extrapolation will be terminated instantly. Otherwise, the scheme will carry on finding the best right side rail by the same method in *Step 4(1)* based on the left side result.

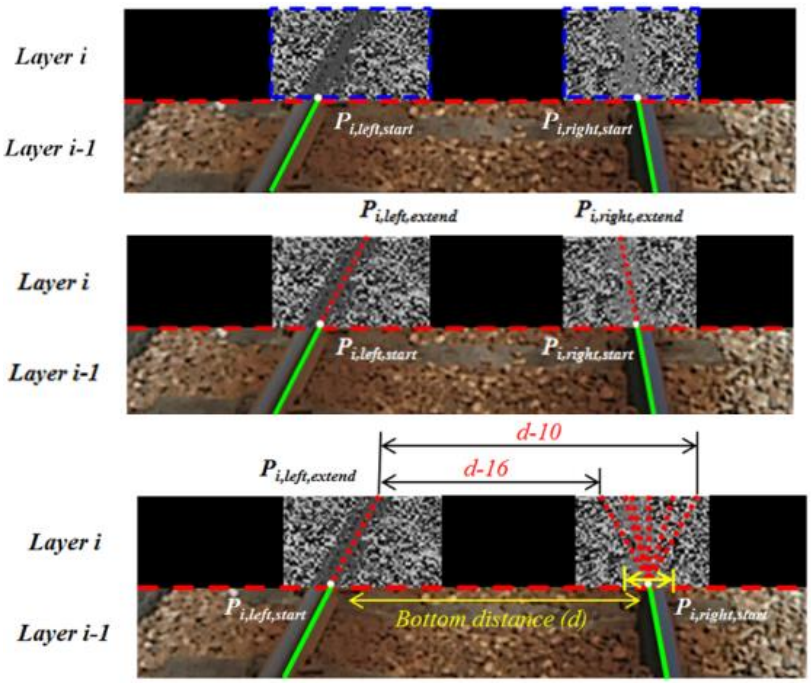


Figure 5-13 Railway extrapolation: (1) Processing Region of interest (2) Directly verification (3) Case only one side verified

Step 4: Termination Decision. The maximum number of extrapolation layers ***N*** is defined before the start of extrapolation, such that the iteration will stop when current layer is layer *N*. This is useful. Otherwise, whether the extrapolation should continue to the next layer or not is decided in this step. During the extrapolation in previous layers, the accumulated color histograms $\tilde{H}_{acc,side}$ of each side railway result are constructed and

accumulated. If the railway is detected in current layer, the color histogram $\tilde{H}_{i,side}$ in this layer should then be constructed as well. Then histogram intersection $\tilde{I}(\tilde{H}_{i,side}, \tilde{H}_{acc,side})$ will be calculated if $\tilde{I}$ is smaller than the threshold T_{color}. It indicates that the color histogram near railway of current layer is very different from the accumulated result of previous layers, which often happens when the extrapolation reaches the train in the front. Then the extrapolation is terminated in this layer. Otherwise, the color histogram $\tilde{H}_{i,side}$ will be used to update the accumulated histogram as below in Equation 5.12.

$$\textit{For all bin in } \tilde{H}_{acc,side},\ \tilde{H}^{(j)}_{acc,side} = \beta \times \tilde{H}^{(j)}_{i,side} + (1-\beta) \times \tilde{H}^{(j)}_{acc,side} \tag{5.12}$$

where $\tilde{H}^{(j)}$ is the value of bin j in histogram $\tilde{H}$, β is the updating rate, which is 0.5 in our implementation. The ending point and slope of each side in current layer are then saved. If the iteration is not terminated, these values are used as the input to the next iteration, otherwise, the track extrapolation is done.

5.3.3 Using Temporal Information

One important assumption of this research is that there is no sudden change for rail position and curvature between consecutive frames. Furthermore, between the consecutive frames, the length of detected railway should be similar. Based on the information above, the temporal correlation can be used to enhance the extrapolation performance. First, for the same layer, results of the previous frame can be used for reference in current frame. Second, if the current layer is not the last layer in previous frame, then in current frame, we must make exception process if no rail is found in

current layer. Therefore, the extrapolation scheme in previous section is modified as below.

(1) Results on both sides of the railway in each layer in previous frame are recorded. In the extrapolation of current layer in current frame, if one of the railway fails in direct straight verification in Step 2, result of this layer in the previous frame will be involved for Angle Alignment measurement, to find the line with the highest AAS.

(2) If the current layer is not the last layer in previous the frame, but the iteration is terminated in Step 4, results in the previous frame will be directly used and extrapolation will resume and continue to next layer based on this result.

5.4 Experiment Result

In this section, firstly, experiment setup will be introduced. Then the performance of proposed railway extraction scheme will be assessed and discussed. Objective evaluation will also be provided in terms of reliability error analysis and computational complexity.

5.4.1 Experiment setup

The research approach was tested on the Hong Kong light railway. The transportation system is running both in urban and suburb areas, composing of different environments, including sheltered region with low luminance, railway built on concrete road, etc. At the same time, the running speed of train changes frequently. The primary task of the entire developed driver assistant system is to detect train in front to avoid clash. Railway extraction scheme is used as a fundamental module, and the extracted railway can be used to provide ROI for possible train position. The DA system is installed in a smart

phone mounted in driving cab to capture frames through the smart phone camera. Then the frames will be processed by the phone processor. Processed result and warning will be provided through phone interface.

All algorithm tests were performed on a desktop with a 3.50GHz CPU and 16GB memory. Implementation is in C++ programming language using standard windows library without parallel processing. For mobile platform, the software was installed on Xiaomi III Android phone with 2.30GHz CPU and 2GB RAM. The testing videos were captured by mobile phone in 640×480 resolution with RGB format. For this resolution, the frame rate was 25 frames per second. We specifically tuned parameters for implementation in our case. Our target is to detect front train within 40 meters. In the camera vision system under our camera installation, the distance is around 290 pixels' height from frame bottom. Hence a total of 13 layers were assigned, with 1 bottom layer and 12 upper layers. The bottom layer is 60 pixels height and each upper layer gets 20 pixels height. In the bottom layer processing, the size of each block is 32×32 pixels. For the upper layer extrapolation iteration, the setting thresholds are: T_{pair}=0.5, T_{verify}=0.9, $T_{terminate}$=0.5 and T_{color} =0.5.

Table 5-1 shows the testing video information, with video index, frame number and video content. Most of challenging scenarios previously discussed are included. The algorithm is designed to extract railway that can be distinguished visually, even the luminance is not sufficient. However, when the train is running during nighttime, the railway area in video captured by the camera using the smart phone is completely dark. This kind of scenarios is not considered in this research. Besides, in this research, we are focusing on extracting single pair of track. Therefore, when there is a branch ahead of

the train, the extrapolation may goes to either side. The evaluation for railway branch case is not considered in this phase.

Table 5-1 Testing sequences information

Index	**Frame number**	**Content**
Factory 1	650	Factory Outdoor, camera approaching front train
Factory 2	900	Factory Indoor, camera approaching front train
Urban 1	1043	Urban view, concrete road, cloudy
Urban 2	2514	Urban view, camera approaching front train
Urban 3	687	Urban view, concrete road, night time
Urban 4	668	Urban view, rain blur, camera approaching front train
Suburban 1	274	Suburban view, fast motion blur

5.4.2 Railway extraction result evaluation

5.4.2.1 Visual based result evaluation

Some of extracted railway results are shown in Figure 5-14, including some challenging situations. The red curves plotted on the graph are the extracted railway results of our proposed scheme.

We can see that, the color of railway in Figure 5-14 (1) appears as brown color due to rusting. In scenario Figure 5-14 (2), the railway is in gray color. An even worst situation shown in situation (3), which is beyond the project scope, that the illumination condition is not sufficient. However, our scheme can extract target railway successfully under these scenario, which indicates the proposed scheme is robust to different color and illumination condition. Some other patterns with great challenges can also be seen on Figure 5-14 (2)(4)(5). In (2), the train is running in relatively high speed that there is directional motion blur, which induces directional gradient information at the bottom part of the image. In (4), the raining blur happens that some parts of railway are

distorted by raindrop on windshield in front of the camera. Situation (5) is a complex urban scenario, in that there are directional lane marks mixed with railway. The scheme can extract railway in these situations as well, which proves the robustness of our connection analysis. Figure 5-14 (6) is a common situation for light railway system and there are lots of sharp railway turn, which do not often happen for high speed railway system. This situation can be well handled by our scheme as well.

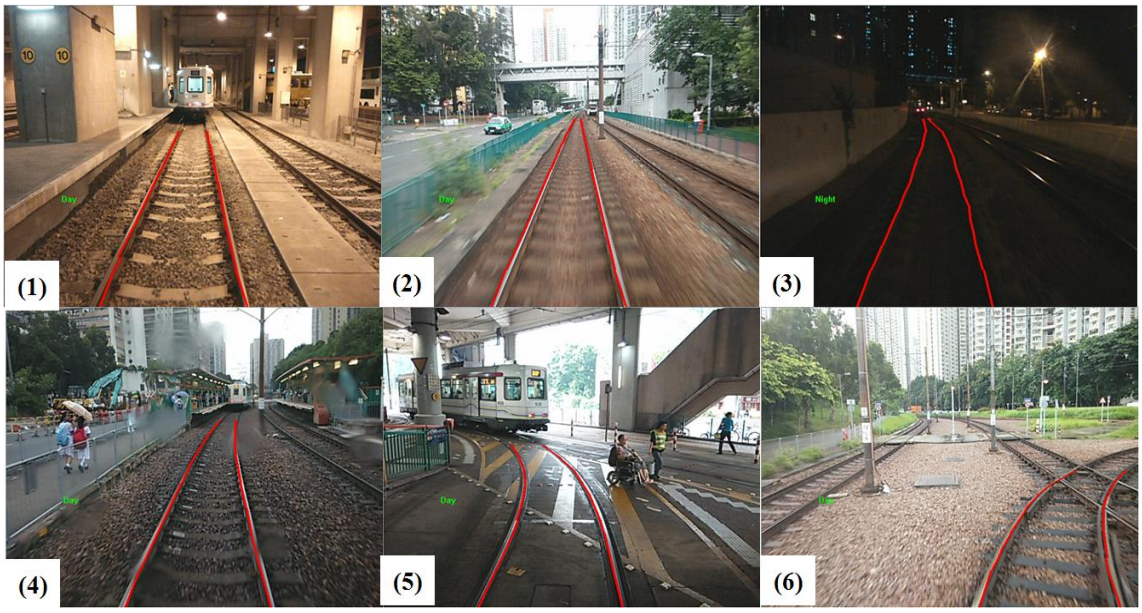


Figure 5-14 Sample railway extraction result under different environment

5.4.2.2 Objective result evaluation

To evaluate the performance of proposed scheme objectively, we further define the evaluation metric. The extracted railway is used to provide region of interest for front train search. Therefore, the correctness in both railway alignment and railway length are important. The railway alignment discussed here refers to whether the extracted curve is on the target rail, or whether the extracted railway extrapolated out of railway region.

Railway length is discussed especially there is a target train in front of the camera that a favorable extrapolation should be terminated just under the back of front train. Based on above aspects, the following error types are defined:

1. No result: There is target railway that can be inspected by human eye, but no railway result return after railway extraction.
2. Inaccuracy: The tail of extracted rail deviated from real railway for more than 20 pixels.
3. Extrapolation overshoot: Extracted railway is longer than real railway length.
4. Extrapolation insufficient: When the railway length is shorter than 40 meters, extracted railway is shorter than real railway length for more than 2 layers (40 pixels).

It can be found that the order of error severity is in descending order. If there is no railway detected in current frame. It means a global search will be used for following front train detection, which is extremely time consuming. For extracted railway with error type 2, the following train detection ROI would have a wrong horizontal position. If this kind of errors occurs frequently, the ROI is less meaningful due to extra processing is needed in the train detection step to consider both horizontal and scale tolerance. Error types 3 and 4 are more tolerable compared with error type 2 that only the train scale needs to be considered.

The evaluation result is shown in Table 5-2. A remark is that in sequence Urban 4, there are windshield wipers appearing in vision as shown in Figure 5-15 left. The corresponding extrapolation error is not counted. It can be calculated that, in terms of error frame defined earlier, the proposed algorithm can achieve more than 99.3%

successful detection rate. This figure is convincing even though the algorithm research scenarios are different for different researches in this area and the detection rate depends highly on the selected video. Therefore, on the other hand, error analysis could be more important.

Table 5-2 Different Error types occurrence in testing sequences

Index	Frame number	Error type 2	Error type 2	Error type 3	Error type 4
Factory 1	650	1	1	0	0
Factory 2	900	0	1	0	0
Urban 1	1043	0	0	0	15
Urban 2	2514	0	0	0	13
Urban 3	687	3	0	0	8
Urban 4	668	0	0	0	0
Suburban 1	274	0	0	0	0

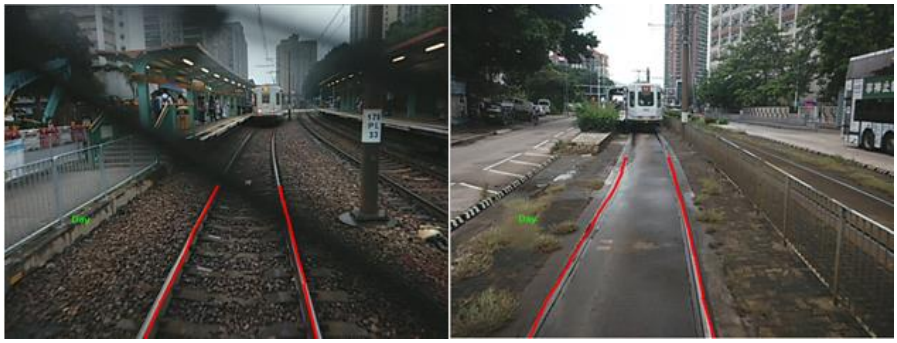


Figure 5-15 Sample railway extraction fail results

It can be seen that most errors occurring in insufficient extrapolation. We further investigated that once an error occurred, the error is very likely to propagate to following frames. For the reason that, an error is caused by unexpected element in frame, and the unexpected elements will not change a lot between consecutive frames. For example, the insufficient extrapolation error in video sequence Urban 1 is shown in Figure 5-15 right. There is a sudden change caused by reflection on left rail, the extrapolation terminated after color coherence verification, which lasts for several

frames. Therefore there are 15 frames in error type 4 for this sequence. Furthermore, in our system implementation, low pass filter technique is used in the generation of ROI. Therefore, short consecutive error frames will not affect ROI position largely.

Table 5-3 Time performance for each testing sequence

Index	Frame number	Average processing time per frame(ms)	Max processing time per frame(ms)	Variance(ms)
Factory 1	650	4.47	6.45	0.58
Factory 2	900	5.02	7.47	0.45
Urban 1	1043	5.89	7.09	0.09
Urban 2	2514	5.58	7.22	0.31
Urban 3	687	5.93	8.06	0.34
Urban 4	668	5.81	7.90	0.25
Suburban 1	274	6.42	7.78	0.17
Total	6730	5.54		

The computational complexity is more important in this project, as we discussed earlier. In the current phase, we are focusing on improving the performance of the algorithm. Therefore, multi-thread processing or using other optimization tools have not been implemented. The processing time for one frame depends on how many layers are processed. If the sequence contains more cases approaching to the front train, fewer layers need to be extrapolated and the average processing time per frame is relatively lower. For this reason, the average time per frame, maximum time per frame and variance to process one frame on desktop are provided for the testing sequences, as shown in Table 5-3. We can see that generally the average time to process one frame is around 5.5ms, which is the faster than all other research with similar conditions [63] to the best of our knowledge. At the same time, the computation complexity varies little,

which can be seen from the calculated variances. Even for the worst case in sequence Urban 3, the maximum processing time on that particular frame is 8.06ms.

We further implemented the algorithm into Xiaomi III android phone and conducted several online testing, using the same setting with experiments conducted on desktop. On smart phone platform, the processing time per frame is around 26.49ms, which occupies 66% processing time for a 25 frame per second driver assistant system. To the best of our knowledge, it is the fastest railway extraction method both on desktop and smart phone platform.

5.5 Chapter summary

Railway extraction is very useful for a driver assistant system. Various outdoor and indoor environments give us great challenges on railway extraction. In this research, we have proposed a railway extraction scheme, with gradient angle being the key feature used in the scheme. Enlightened by gradient angle pattern on railway area, we have proposed a novel Angle Alignment Measure to measure the connectivity between two points, which performs extremely well in railway extrapolation. Moreover, we have further developed an adaptive bottom layer railway block classification to handle the directional motion blur problem. Besides, we further utilize temporal correlation to increase the stability of extracted railway results.

Experimental results show that the proposed scheme is robust against color and luminance variation, that the railway can be extracted in various challenging scenarios. More preciously, the computation complexity of proposed scheme is much lower

comparing with other existing railway extraction approaches, which makes it possible to develop a real-time DA system on smart phone platform in our project.

Chapter 6. Conclusion and Future Work

6.1 Conclusion

In this thesis, our major works are on efficient on storage, searching and object identification. The discussions are based on video applications in video surveillance and vehicle vision. Three major researches are made which are surveillance video coding scheme, moving object detection in robot car vision and railway extraction scheme.

A double encoder coding scheme is proposed for surveillance coding. Input video sequence is pre-processed to obtain masked foreground sequence and condensed background sequence. Then a separate encoder is used for each sequence. The information carried by input surveillance is well transmitted and recovered with the help of side information. The bitrate can be significantly saved without affecting foreground area and visual quality of the entire frame.

Then a novel moving object detection algorithm is proposed to detect moving object under moving environment in robot car vision. Motor signal and depth map are used to model the motion relationship in 3D space. The proposed algorithm can successfully detect moving object, especially for vision zoom-in and zoom-out situations, which has not be sufficiently studied in the literature. At the same, the low computational complexity makes the algorithm be implemented in real-time application.

At last, not the least, a railway extraction scheme is proposed for challenging outdoor and indoor environments. In this research, we have proposed a novel connectivity

measurement metric, Angle Alignment Measure by utilizing the gradient angle pattern. The metric perform very well for railway extrapolation. The experiment results show the proposed scheme is robust against color and luminance variation. More importantly, the computational complexity of the proposed scheme is lower comparing with other existing approaches. And we have implemented the scheme on a smart phone platform for a real product in our project.

6.2 Future work

For the proposed surveillance encoding scheme, based on current research, there are still issues to be resolved. The foreground masking is conducted in object level. An improvement can be made by integrating foreground masking to block and sub-block level used in the encoding to achieve higher coding efficiency. Due to the fact that bitrate saving is affected by multi factors, the current double codec is not optimal, which can be investigated further by model the relation between background region ratio, noise model, quantization parameter, etc.

For moving object detection on robot car vision, the moving object can be detected based on the matched feature points remained. In the future, the following module can be designed to provide more information for smart robot system, such as to indentify moving objects and analyses the movement of the objects.

In railway extraction scheme, we have proposed the Angle Alignment Measure, which is a simple and robust connectivity measure. The application of this metric can further be explored. It can possibly be used very effectively for lane mark detection, roadway detection in aerial view graph, object silhouette construction, and other applications.

References

[1] Ahmad, I., He, Z., Liao, M., Pereira, F., & Sun, M.-T.. “Special Issue on Video Surveillance”. *IEEE Transactions on Circuits and Systems for Video Technology*, *18*(8), 2008.

[2] Gao, W., Tian, Y., Huang, T., Ma, S., & Zhang, X. (2013). IEEE 1857 Standard Empowering Smart Video Surveillance Systems. *IEEE Intelligent Systems*, 1–1. doi:10.1109/MIS.2013.101

[3] Zhang, X., Huang, T., & Member, S. (2014). Background-Modeling-Based Adaptive Prediction for Surveillance Video Coding, *23*(2), 769–784.

[4] Ohm, J-R., et al. "Comparison of the coding efficiency of video coding standards—including high efficiency video coding (HEVC)." *Circuits and Systems for Video Technology, IEEE Transactions on* 22.12 (2012): 1669-1684.

[5] A.B. Hillel, R. Lerner, D. Levi, and G. Raz. “Recent progress in road and lane detection: a survey.” *Machine Vision and Applications* 25, no. 3 (2014): 727-745.

[6] A. Borkar, M. Hayes, and M.T. Smith. “A novel lane detection system with efficient ground truth generation”. *IEEE Transactions on Intelligent Transportation Systems*, 13, no. 1 (2012): 365-374.

[7] Li, Haojie, et al. "Automatic detection and analysis of player action in moving background sports video sequences." Circuits and Systems for Video Technology, IEEE Transactions on 20.3 (2010): 351-364.

[8] T.S. Chan and K.S. Chung. “Applications and selections of intelligent surveillance system in railway industry” *Railway Engineering - Challenges for Railway Transportation in Information Age, 2008. ICRE 2008. International Conference on* , vol., no., pp.1,6, 25-28 March 2008

[9] Z, Xie, et al. "Research on moving object detection method of high-speed railway transport hub video surveillance." *Information Science and Engineering (ISISE), 2012 International Symposium on*. IEEE, 2012.

[10] J. Xue, et al. "Visual monitoring-based railway grade crossing surveillance system." *Image and Signal Processing, 2008. CISP'08. Congress on*. Vol. 2. IEEE, 2008.

[11] O. Sehchan, S. Park, and C. Lee. "A platform surveillance monitoring system using image processing for passenger safety in railway station." *Control, Automation and Systems, 2007. ICCAS'07. International Conference on*. IEEE, 2007

[12] H.K. Cheung, W.C. Siu, C.S. Ng, S. Lee and L. Poon, "Accurate Distance Estimation Using Camera Orientation Compensation Technique for Vehicle Driver Assistance System", Proceedings, pp.231-2, *IEEE International Conference on Consumer Electronics (ICCE'2011)*, 13-16 January 2012, Las Vegas, USA.

[13] Z. Sun, G. Bebis and R. Miller, "Monocular precrash vehicle detection: features and classifiers," *IEEE Transactions on Image Processin* , vol.15, no.7, pp.2019,2034, July 2006

[14] J.Arrospide, L. Salgado. "Log-Gabor Filters for Image-Based Vehicle Verification," *IEEE Transactions on Image Processing,* vol.22, no.6, pp.2286,2295, June 2013

[15] M. Rüder, N. Mohler, and F. Ahmed. "An obstacle detection system for automated trains." *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*. IEEE, 2003.

[16] Aizawa, Kiyoharu, Hiroshi Harashima, and Takahiro Saito. "Model-based analysis synthesis image coding (MBASIC) system for a person's face." *Signal Processing: Image Communication* 1.2 (1989): 139-152.

[17] Martins, Isabel, and Luis Corte-Real. "A video coder using 3-D model based background for video surveillance applications." *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*. Vol. 2. IEEE, 1998.

[18] Richardson, Iain E. *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons, 2004.

[19] Francois, Edouard, Jean-François Vial, and Bertrand Chupeau. "Coding algorithm with region-based motion compensation." *Circuits and Systems for Video Technology, IEEE Transactions on* 7.1 (1997): 97-108.

[20] Jin, Xin, and Satoshi Goto. "Encoder adaptable difference detection for low power video compression in surveillance system." *Signal Processing: Image Communication* 26.3 (2011): 130-142.

[21] Wiegand, Thomas, Xiaozheng Zhang, and Bernd Girod. "Long-term memory motion-compensated prediction." *Circuits and Systems for Video Technology, IEEE Transactions on* 9.1 (1999): 70-84.

[22] Tiwari, Mayank, and Pamela C. Cosman. "Selection of long-term reference frames in dual-frame video coding using simulated annealing." *Signal Processing Letters, IEEE* 15 (2008): 249-252.

[23] Ding, Rong, et al. "Background-frame based motion compensation for video compression." *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*. Vol. 2. IEEE, 2004.

[24] Gorur, P.; Amrutur, B., "Skip Decision and Reference Frame Selection for Low-Complexity H.264/AVC Surveillance Video Coding," *Circuits and Systems for Video Technology, IEEE Transactions on* , vol.24, no.7, pp.1156,1169, July 2014

[25] Zhang, Xianguo, et al. "Background-modeling-based adaptive prediction for surveillance video coding." *Image Processing, IEEE Transactions on* 23.2 (2014): 769-784.

[26] GB/T 25724-2010, "Technical Specification of Surveillance Video and Audio Coding", Beijing: Standardization Administration of the People's republic of China (SAC), 2010.

[27] Shu, Ruo, et al. "A novel scheme for SVAC audio encoder." *Communications and Information Technologies (ISCIT), 2014 14th International Symposium on*. IEEE, 2014.

[28] Shu, Ruo, Shibao Li, and Xin Pan. "An Optimization Scheme for SVAC Audio Encoder." *Intelligent Information Processing VII*. Springer Berlin Heidelberg, 2014. 221-229.

[29] Xianguo Zhang; Tiejun Huang; Yonghong Tian; Wen Gao, "Overview of the IEEE 1857 surveillance groups," *Image Processing (ICIP), 2013 20th IEEE International Conference on* , vol., no., pp.1505,1509, 15-18 Sept. 2013

[30] Ma, Siwei, Shiqi Wang, and Wen Gao. "Overview of IEEE 1857 video coding standard." *Image Processing (ICIP), 2013 20th IEEE International Conference on*. IEEE, 2013.

[31]B.T. Nassu, M. Ukai, "A Vision-Based Approach for Rail Extraction and its Application in a Camera Pan–Tilt Control System", *IEEE Transactions on Intelligent Transportation Systems*, vol.13, no.4, pp.1763,1771, Dec. 2012

[32]F. Kaleli, Y.S. Akgul, "Vision-based railroad track extraction using dynamic programming", *12th International IEEE Conference on Intelligent Transportation Systems*, 2009. pp.1,6, 4-7 Oct. 2009

[33]A. Borkar, M. Hayes and M.T. Smith, "Robust lane detection and tracking with ransac and Kalman filter", *16th IEEE International Conference on Image Processing (ICIP)*, pp.3261, 3264, 7-10 Nov. 2009

[34]Z.W. Kim, "Robust Lane Detection and Tracking in Challenging Scenarios", *IEEE Transactions on Intelligent Transportation Systems*, vol.9, no.1, pp.16,26, March 2008

[35]D.J. Kang, J.W. Choi, and I.S. Kweon. "Finding and tracking road lanes using "line-snakes"." *In Intelligent Vehicles Symposium*, 1996., Proceedings of the 1996 IEEE, pp. 189-194. IEEE, 1996.

[36]Z.W. Kim. "Robust lane detection and tracking in challenging scenarios."*Intelligent Transportation Systems, IEEE Transactions on* 9.1 (2008): 16-26.

[37]G. Liu, F. Worgotter, and I. Markelic. "Stochastic lane shape estimation using local image descriptors." *Intelligent Transportation Systems, IEEE Transactions on* 14.1 (2013): 13-21.

[38] Q. Yue, M. Zheng, Y. J. Jian, and Q. J. Song, “Real-Time Moving Target Detection in the Dynamic Background,” no. 2, pp. 439–443.

[39] T. Jin, F. Zhou, and X. Bai, “Moving Vehicles Detection in Airborne Video,” *2008 International Symposium on Information Science and Engineering*, pp. 697–701, Dec. 2008.

[40] X. Zhou, C. Yang, and W. Yu, “Moving object detection by detecting contiguous outliers in the low-Rank representation,” vol. 35, no. 3, pp. 597–610, 2013.

[41] Wiegand, T.; Sullivan, G.J.; Bjontegaard, G.; Luthra, A., "Overview of the H.264/AVC video coding standard," *Circuits and Systems for Video Technology, IEEE Transactions on* , vol.13, no.7, pp.560,576, July 2003 doi: 10.1109/TCSVT.2003.815165

[42] Sullivan, G.J.; Ohm, J.; Woo-Jin Han; Wiegand, T., "Overview of the High Efficiency Video Coding (HEVC) Standard," *Circuits and Systems for Video Technology, IEEE Transactions on* , vol.22, no.12, pp.1649,1668, Dec. 2012

[43] Xianguo Zhang; Yonghong Tian; Tiejun Huang; Wen Gao, "Low-complexity and high-efficiency background modeling for surveillance video coding," *Visual Communications and Image Processing (VCIP), 2012 IEEE* , vol., no., pp.1,6, 27-30 Nov. 2012

[44] Zhang, Xianguo, et al. "An efficient coding scheme for surveillance videos captured by stationary cameras." *Visual Communications and Image Processing 2010*. International Society for Optics and Photonics, 2010.

[45] Kin-Yi Yam, Wan-Chi Siu, Ngai-Fong Law and Chok-Ki Chan, ‘Fast Video Object Detection via Multiple Background Modeling’, Proceedings, pp.729-732, IEEE International Conference on Image Processing, (ICIP’2010), 26-29 September, 2010, Hong Kong.

[46] Stauffer, Chris, and W. Eric L. Grimson. "Adaptive background mixture models for real-time tracking." *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*. Vol. 2. IEEE, 1999.

[47] Nassu, Bogdan Tomoyuki, and Masato Ukai. "A Vision-Based Approach for Rail Extraction and its Application in a Camera Pan–Tilt Control System."*Intelligent Transportation Systems, IEEE Transactions on* 13.4 (2012): 1763-1771.

[48] Gorur, P.; Amrutur, B., "Skip Decision and Reference Frame Selection for Low-Complexity H.264/AVC Surveillance Video Coding," *Circuits and Systems for Video Technology, IEEE Transactions on* , vol.24, no.7, pp.1156,1169, July 2014

[49]Lee, Seungwon, et al. "Moving object detection using unstable camera for consumer surveillance systems." Consumer Electronics (ICCE), 2013 IEEE International Conference on. IEEE, 2013.

[50]Sheikh, Yaser, Omar Javed, and Takeo Kanade. "Background subtraction for freely moving cameras." Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009.

[51] Szeliski, Richard. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[52]Changhai Xu; Jingen Liu; Kuipers, B., "Motion Segmentation by Learning Homography Matrices from Motor Signals," Computer and Robot Vision (CRV), 2011 Canadian Conference on , vol., no., pp.316,323, 25-27 May 2011

[53]Changhai Xu, Jingen Liu, and Kuipers, B. "Moving object segmentation using motor signals." Computer Vision–ECCV 2012. Springer Berlin Heidelberg, 2012. 676-689.

[54]T. Gevers and A. W. M. Smeulders, "Color based object recognition", Pattern Recognit., vol. 32, pp.453 -465 1999

[55]Rosten, Edward; Drummond, Tom, "Fusing points and lines for high performance tracking," Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on , vol.2, no., pp.1508,1515 Vol. 2, 17-21 Oct. 2005

[56]Hao Wu and Wan-Chi Siu, "Real-time Railway Extraction by Angle Alignment Measure", *Paper Accepted, to be published in Proceedings, IEEE International Conference on Image Processing, (ICIP'2015)*, 27-30, October 2015, Quebic City, Canada

[57] R.K. Satzoda, S. Suchitra, and T. Srikanthan. "Robust extraction of lane markings using gradient angle histograms and directional signed edges", *2012 IEEE Intelligent Vehicles Symposium (IV)*, pp. 754-759, IEEE, 2012.

[58] R.K. Satzoda, S. Suchitra, and T. Srikanthan. "Gradient angle histograms for efficient linear hough transform." *2009 16th IEEE International Conference on Image Processing (ICIP)*, pp. 3273-3276. IEEE, 2009.

[59] S.E. Shih, W.H. Tsai, "Hough transform with dynamic thresholding for robust and real-time detection of complex curves in images," *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on* , vol., no., pp.1454,1458, 26-31 May 2013.

[60] H.Wu and W.C. Siu, "Real-time Moving Object Detection using Motor signal and Depth map for Robot Car", *Proceedings, Intelligent Robots and Computer Vision XXXI:*

Algorithms and Techniques of the IS&T/SPIE Electronic Imaging, 2-6 February 2014, San Francisco, California, USA.

[61] A. Barla, F. Odone, and A. Verri. “Histogram intersection kernel for image classification.”, *2003 International Conference on Image Processing (ICIP)*, vol. 3, pp. III-513. IEEE, 2003.

[62] J.C. Espino, B. Stanciulescu, “Rail extraction technique using gradient information and a priori shape model”, *15th International IEEE Conference on Intelligent Transportation Systems*, pp.1132, 1136, 16-19 Sept. 2012.

[63] Tsung-Yu, Chien, and Chung Sheng-Luen. "Android-based driving assistant for lane detection and departure warning." *Control Conference (CCC), 2014 33rd Chinese*. IEEE, 2014.

[64] Horn, Berthold K., and Brian G. Schunck. "Determining optical flow." *1981 Technical symposium east*. International Society for Optics and Photonics, 1981.

[65] Barron, John L., David J. Fleet, and Steven S. Beauchemin. "Performance of optical flow techniques." *International journal of computer vision* 12.1 (1994): 43-77.

[66] P.V.C. Hough, *Machine Analysis of Bubble Chamber Pictures*, Proc. Int. Conf. High Energy Accelerators and Instrumentation, 1959

[67] Duda, R. O. and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Comm. ACM, Vol. 15*, pp. 11–15 (January, 1972)