

2011

A statistical approach for shadow detection using spatio-temporal contexts

Yiyang Liu
West Virginia University

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Liu, Yiyang, "A statistical approach for shadow detection using spatio-temporal contexts" (2011).
Graduate Theses, Dissertations, and Problem Reports. 152.
<https://researchrepository.wvu.edu/etd/152>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

**A STATISTICAL APPROACH FOR SHADOW DETECTION
USING SPATIO-TEMPORAL CONTEXTS**

by

Yiyang Liu

**Thesis submitted to the College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of**

**Master of Science in
Computer Science**

Approved by

Dr. Donald Adjeroh, Committee Chairperson

Dr. Arun Ross

Dr. Xin Li

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia

2011

**Keywords: Shadow detection, background segmentation, chromaticity, texture,
visual surveillance, spatio-temporal contexts**

Copyright 2011 Yiyang Liu

Abstract

SHADOW DETECTION USING SPATIO-TEMPORAL CONTEXTS

by Yiyang Liu

Background subtraction is an important step used to segment moving regions in surveillance videos. However, cast shadows are often falsely labeled as foreground objects, which may severely degrade the accuracy of object localization and detection. Effective shadow detection is necessary for accurate foreground segmentation, especially for outdoor scenes. Based on the characteristics of shadows, such as luminance reduction, chromaticity consistency and texture consistency, we introduce a nonparametric framework for modeling surface behavior under cast shadows. To each pixel, we assign a potential shadow value with a confidence weight, indicating the probability that the pixel location is an actual shadow point. Given an observed RGB value for a pixel in a new frame, we use its recent spatio-temporal context to compute an expected shadow RGB value. The similarity between the observed and the expected shadow RGB values determines whether a pixel position is a true shadow. Experimental results show the performance of the proposed method on a suite of standard indoor and outdoor video sequences.

DEDICATION

I dedicate my thesis to my parents

Ziyi Liu

Xiujun Cheng

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor, Dr. Donald Adjero, a respectable, responsible and resourceful professor who has provided me with valuable direction, inspiration, suggestions and criticisms at every stage of my graduate studies. Without his instruction, kindness and patience, this thesis could not have reached its present form. His timely guidance and careful observations have helped shape my thinking and approach to problem solving.

I would also like to express my deepest gratitude to Dr. Arun Ross and Dr. Xin Li. I cannot forget the knowledge and brilliant ideas they imparted to me, the talented way they solved those tricky problems, as well as their kindness and help. It also provides me a good foundation to complete this thesis.

Last but not least, I would like to thank my parents and all my friends for their encouragement and support.

TABLE OF CONTENTS

Abstract	ii
Dedication	iii
Acknowledgement	iv
Table of Contents	v
List of Figures	vii
List of Tables	xiii
Chapter 1: Introduction	1
1.1 Overview	1
1.2 Motivation and Problem	2
1.3 General Approach	3
1.4 Main Contribution of the Thesis	3
1.5 Thesis Organization	4
Chapter 2: Related Work	5
2.1 Background Subtraction.....	5
2.2 Shadow Detection	8
2.3 Our Contribution.....	9
Chapter 3: Proposed Method	11
3.1 Shadow Model	11
3.2 Learning Moving Cast Shadow	16
3.2.1 Shadow Luminance Classifier	16
3.2.2 Updating the Shadow Model	19
3.3 Detecting Moving Cast Shadows.....	23
3.3.1 Detecting Shadow Regions with Temporal Information.....	23
3.3.2 Spatial Information for Shadow Correction.....	25
Chapter 4: Results & Discussion	28
4.1 Qualitative Results	28

4.2 Quantitative Results	31
4.3 Comparative Result.....	32
4.3.1 Validation with SNP	32
4.3.2 Comparison with Other Methods.....	36
4.4 Discussion on Parameters	37
Chapter 5: Conclusion & Future Work	39
5.1 Conclusion	39
5.2 Future Work	39
References	41

LIST OF FIGURES

Figure 2-1.....	10
Figure 3-1.....	14
Figure 3-2.....	15
Figure 3-3.....	17
Figure 3-4.....	21
Figure 3-5.....	26
Figure 3-6.....	27
Figure 4-1.....	29
Figure 4-2.....	29
Figure 4-3.....	29
Figure 4-4.....	30
Figure 4-5.....	30
Figure 4-6.....	30
Figure 4-7.....	32
Figure 4-8.....	34
Figure 4-9.....	35
Figure 4-10.....	38

LIST OF TABLES

Table 4-1	32
Table 4-2	36
Table 4-3	37

Chapter 1: Introduction

1.1 Overview

Object detection is an important and challenging problem in many video processing applications such as video surveillance, traffic monitoring and human detection. Usually, the most common schemes for object segmentation are based on inter-frame difference or background suppression. However, all the moving points of both objects and shadows are detected at the same time.

Shadows result from the occlusion of light sources by objects in the scene. The part of an object that is not illuminated is called self-shadow, and the area projected on the scene by the object in the direction of direct light is called cast shadow, which can be further classified into umbra and penumbra. The umbra corresponds to the area where direct lights are totally blocked by the object, whereas in the penumbra area they are partially blocked. If an object is moving, its cast shadows are more properly called moving cast shadows. In this thesis, our main work is to remove the moving cast shadow in a video sequence. The shadow here is just the dark part in a scene due to the occlusion of light. That is not the mirror refraction of the object. Sometimes, the refraction and the shadow are mixed up. The shadow can be removed but the reflection cannot.

In dynamic scenes, moving cast shadows often result in apparent merging of objects, distortion of shapes, and objects loss (due to the shadow cast over another object). Shadows may be either attached to or disconnected from detected objects. In the first case, shadows distort the object shapes, such that traditional shape recognition methods become unreliable (shadows and objects are merged in a single blob, thus geometrical properties are affected). In the second case, shadows may be

classified as an erroneous object in the scene. This may severely degrade the accuracy of object localization and detection. Therefore, an effective shadow detection method is important for accurate object detection.

1.2 Motivation and the Problem

The difficulties associated with shadow detection arise since shadows and objects share two important visual features. First, shadow points are detectable as foreground points since they typically differ significantly from the background. Secondly, shadows have the same motion as the true objects casting them. Further, it is hard to remove shadows using common segmentation techniques, because shadows are adjacent to foreground object points in most cases.

Moving cast shadows are caused by the occlusion of light sources. Shadows reduce the total energy incident at the background surfaces where the light sources are partially or totally blocked by foreground objects. Typically, shadow points have lower luminance values but similar chromaticity values. Also, texture characteristics around shadow points remain very similar since shadows do not alter the background surface.

Luminance, chromaticity and texture are the basic features of shadows. Therefore, an important issue is how to define and express these features accurately. The definition of similarity is a key point in the classification of surface points, especial for the chromaticity feature. Across different scenes, chromaticity distortions could vary over a large range. These features are defined in different color spaces and different forms. Most of shadow removal algorithms use one or two of these features as their basis for analysis of shadows. In this thesis, we propose a shadow removal algorithm that uses all the features in the RGB color space.

1.3 General Approach

As mentioned above, the three features of shadows are widely used in the shadow removal algorithms. Most current approaches to shadow detection and removal can be classified as either deterministic or statistical approaches. The deterministic approaches assume an a priori knowledge of scene geometry, foreground objects or light sources. They use thresholds for the classification of surface points, whereas statistical approaches use probabilistic functions.

1.4 Main Contribution of the Thesis

The key contributions of this thesis are:

(1) A novel adaptive shadow model has been proposed to describe the behaviors of an image surface under cast shadows. We make no assumptions on the number of illumination sources, view geometries, surface texture (e.g. grass, road, etc.), types or shapes of shadows, foreground objects or the background. Our model can deal with multiple sources with different spectral power distributions (SPD).

(2) We introduce a novel framework to describe cast shadows using spatio-temporal contexts. This provides effective descriptors of the principal characteristics of shadows.

(3) Qualitative and quantitative evaluation of our approach validate that our new shadow model is more effective in characterizing background surface distortion under moving cast shadows.

1.5 Thesis Organization

In Chapter 2, we will introduce the Gauss Mixture Model (GMM) which is used to segment the foreground. The GMM can detect moving pixel regions including both objects and shadows. The chapter also introduces shadow removal approaches in recent literature.

In chapter 3, we present our new shadow model and explain in detail the proposed shadow removal algorithm. The algorithm includes stages for shadow luminance classifier, learning shadow model, and for detecting the shadow region.

In chapter 4, the experimental results on benchmark video sequences will be represented, including indoor and outdoor scenes. We will provide qualitative results, quantitative results and comparative result with other methods. A discussion on the parameters in our algorithm is also given in this part.

In chapter 5, we conclude the thesis and provide directions for future research.

Chapter 2: Related Work

A color model is an abstract mathematical model describing the way that colors can be represented as tuples of numbers, typically as three or four values or color components, such as RGB, HSV or CMYK. When this model is associated with a precise description of how the components are to be interpreted, the resulting set of colors is called color space. The color model is important in computer graphics and video surveillance. For some special purpose applications, we can design our own color model to describe an image. The image model is a computational model describing an image to stress some special characteristics of the image, using a special color model. In this section, we will introduce a background model to describe the reference background image and some proposed approaches for shadow detection in the literature.

2.1 Background Subtraction

Our approach use the well-known Gaussian Mixture Model (GMM) described in [1] and modified for online implementation in [2] to detect foreground objects. GMM can effectively detect moving pixels, including objects, shadows, and some erroneous pixels, in the image. It has strong adaptability to changing conditions in the image.

GMM assumes that each pixel is subject to a mixture of Gaussian distributions. For a given pixel, its observed RGB value X_t in the video sequence is a sample of a random variable X . The observed value set $\{X_1, X_2, \dots, X_t\}$ is modeled by a mixture of K Gaussian distributions. The probability of the observed pixel in the current image is

$$P(X_t) = \sum_{k=1}^K \omega_{k,t} * f(X_t, \theta_{k,t}) \quad (2.1)$$

where K is the number of Gaussian distributions, typically between 3 and 5. Some of these distributions correspond to background and the rest are deemed to be foreground. GMM assumes that each pixel views background distributions more often than foreground ones. The parameter $\omega_{k,t}$ is the a priori probability that the pixel is associated to the k^{th} Gaussian in the image at time t and $\sum_{k=1}^K \omega_{k,t} = 1$. A Gaussian probability density function f with parameters $\theta_{k,t} = \{\mu_{k,t}, \Sigma_{k,t}\}$ describes the color components of a surface point,

$$f(X_t, \theta_{k,t}) = \frac{1}{(2\pi)^{n/2} |\Sigma_{k,t}|^{1/2}} \exp\left(-\frac{1}{2} (X - \mu_{k,t})^T \Sigma_{k,t}^{-1} (X - \mu_{k,t})\right) \quad (2.2)$$

where $\mu_{k,t} = (\mu_{k,t,r}, \mu_{k,t,g}, \mu_{k,t,b})^T$ is the mean value vector and $\Sigma_{k,t}$ is the covariance matrix of the k^{th} Gaussian in the mixture at time t . $n = 3$ is the dimension in RGB color space. For computational reasons, a further assumption is usually made that the three components of the pixel RGB value are independent so that $\Sigma_{k,t}$ is diagonal and may be presented by the 3-dimensional variance, $\Sigma_{k,t} = \text{diag}(\sigma_{k,t,r}^2, \sigma_{k,t,g}^2, \sigma_{k,t,b}^2)$.

The K distributions are sorted in decreasing order using the ratio $\omega_{k,t} / \|\Sigma_{k,t}\|$. The larger the ratio, the more stable the distribution. This order indicates that the most likely background distributions remain on the top and the less likely ones gravitate towards the bottom and may be replaced by a new distribution. The first B distributions whose total a priori probability is greater than a threshold T are considered as the background models.

$$B = \arg \min_b (\sum_{k=1}^b \omega_{k,t} > T) \quad (2.3)$$

Each pixel is labeled as background or foreground according to whether the pixel's RGB value is associated with the first B distributions. The pixel value is associated to the k^{th} distribution if the value falls within $\lambda = 2.5$ standard deviations of the mean of k^{th} Gaussian distribution. That is

$$d_{k,t}^T d_{k,t} < \lambda^2 \quad (2.4)$$

where

$$d_{k,t} = (\text{diag}(\sigma_{k,t,r}, \sigma_{k,t,g}, \sigma_{k,t,b}))^{-1} (X_t - \mu_{k,t}) \quad (2.5)$$

If we cannot associate a pixel value to an existing distribution, a new distribution k is created with μ , the pixel current value, \sum_{init} and a priori probability ω_{init} . The last probable distribution is dropped. For each frame, the a priori probability ω_k and the distribution parameters of each state are updated. The GMM is initialized by the first received image.

We define:

$$M_1 = \{X \mid X \in GMM_F\} \quad (2.6)$$

where GMM_F is the set of foreground pixels detected by GMM.

In practices, some of the foreground objects that we get from GMM are not true foreground. The foreground objects contain the true foreground, their shadows and

noisy isolated pixels. Not only is the true foreground different from the background, the shadows are also different from the background. The GMM however does not consider the change in the background that arises due to shadows. Other background models also have the same problem. The noisy isolated pixels may result from salt and pepper noise or the shaking of the camera, which can be ignored or removed using morphological methods. Therefore, shadow removal is a necessary step for improved and accurate detection of moving objects.

2.2 Shadow Detection

In recent years, many approaches have been proposed to remove cast shadows in a scene. Horprasert et al. [3] used a computational color model which defined brightness and chromaticity distortion, and a pixel is classified as shaded background or shadow if it presents similar chromaticity but lower brightness compared with the pixel's corresponding background. Cucchiara et al. [4] detected shadows by exploiting the color differences between shadow and background in the HSV color space. Schreer et al. [5] adopted the YUV color space to avoid using the time consuming HSV color transformation. They segmented shadows from foreground objects based on linear intensity scaling and the observation that shadows reduce the YUV pixel value linearly. Savador et al. [6] used the fact that shadows darken surfaces and removed them using color invariance and geometric properties of shadows. Joshi et al. [7] and Nghiem et al. [8] utilized the chromaticity, edge information/texture and intensity with different thresholds to detect shadows one level at a time. Stauder et al. [9] used edge width information to differentiate penumbra regions from background. A linear luminance edge model is applied to detect shadow boundaries. Leone et al. [10] used Gabor features extracted from a textured patch to detect shadows since shadow regions present the same textural characteristics as the corresponding background. Wei et al. [11] showed that ratio edges are illumination invariant and that

the distribution of normalized ratio edge difference follows a chi-square distribution. A significance test was then used to detect shadows.

Recently, the statistical approach has gained popularity in detecting cast shadows. Liu et al. [12] proposed to remove shadow using local regions and global level information in the HSV color space. Martel-Brisson et al. [13] used the Gaussian mixture model (GMM) to model moving cast shadows with nonuniform and varying intensity. The shadow model merged into background model is called GSM. Huang et al. [14] built a global shadow model and a local shadow model with the GMM to learn the behavior of cast shadows. Using graph cut algorithm, they viewed the problem as minimizing a Markov random field energy function, composed of the background, shadow and foreground models. Martel-Brisson et al. [15] proposed a new physical cast shadow model, by estimating the direction of cast shadows in RGB color space. They used a nonparametric density estimation (Kernel Density Estimators, KDE) to estimate many parameters in an unsupervised manner. In [15], the SPDs of all light sources are assumed to have the same profile but different power factor. Huang et al. [16] extended the model by releasing this assumption.

2.3 Our Contribution

The deterministic methods based on a priori knowledge have achieved some good results, but their applicability is limited to some particular environments. Also, the statistical methods may suffer from insufficient training samples. Unlike in background modeling, where we can obtain samples in every frame, shadows may not appear at the same pixel in each frame. A single pixel needs to be shaded many times before its estimated parameters converge, that is, assuming the illumination conditions are stable. Therefore, this kind of pixel-based shadow models require a longer period of training time when foreground activities are not frequent.

In this thesis, we propose a new statistical approach to detect cast shadows based on the RGB color space. It uses a novel characterization of shadow features and statistical strategies to learn the behavior of shadows. We can consider it as a combined approach. From the physical illumination model and the characteristics of cast shadows, we know that the pixel under a shadow has a lower luminance compared to its corresponding background. Therefore, we consider a surface point with lower luminance as potential shadow point. Applying chromaticity consistency and texture consistency based on local gradient, we give a confidence (weight) to the potential shadow point, and then record its weight and current RGB value. To determine whether a potential shadow point is an actual shadow point, we compare it with its expected shadow value, which will be computed from recent spatio-temporal contexts. The flow diagram of our approach is shown in Figure 2-1.

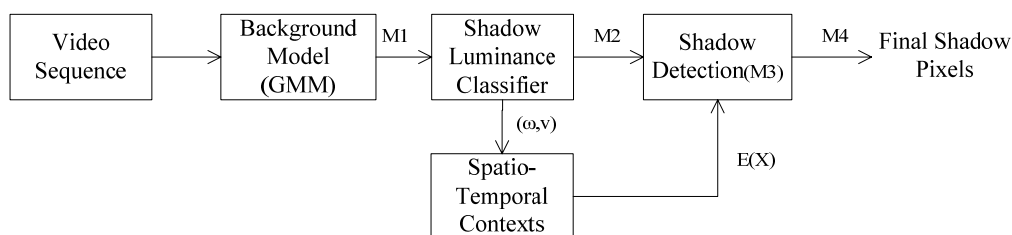


Figure 2-1: The flow diagram of our approach.

Chapter 3: Proposed Method

In this section, we propose a new shadow model and describe our approach on learning and detecting cast shadows. We first apply the shadow luminance classifier to the foreground pixels detected by GMM to select the possible shadow samples. The probability of a possible shadow pixel is determined by its chromaticity feature and spatial context. Local gradient is used to describe spatial context. We learn the features of possible shadow pixels by recording the probability and RGB value of the pixel. We call this sequence of historical records for each pixel in an image the temporal context. Then, we can detect shadows by using spatio-temporal context. Note that our approach is pixel-based, models the behavior of pixels in shadows, and adapts fast to the environment with complex illumination.

3.1 Shadow Model

A surface's appearance depends on its reflectivity properties and the total energy incident at the surface. The basis of our approach is the Phong illumination reflection model [17], an empirical model used to describe local illumination. It describes the way a surface is lit by reflecting the lights that come from the environment, which is the combination of ambient light, diffuse light and specular light. Ambient light is the amount of light scattered around the entire scene. Diffuse light is the reflected light on rough surfaces. Specular light is the reflected light on smooth surfaces, which is usually small and can be neglected when surfaces are not very shiny. By this model, we can determine how shadows appear on surfaces.

For each surface point $P(x,y)$, the Phong model describes the illuminance $I(x,y)$ of this point as follows:

$$\begin{aligned}
I(x, y) &= \text{ambient light} + \sum_{LS} (\text{diffuse light} + \text{specular light}) \\
&= k_a i_a + \sum_{j=1}^{LS} (k_d (\vec{L}_j(x, y) \cdot \vec{N}(x, y)) i_d + k_s (\vec{R}_j(x, y) \cdot \vec{V}(x, y))^\alpha i_s)
\end{aligned} \tag{3.1}$$

where i_a controls the ambient light, which usually can be considered constant; i_d and i_s are respectively defined as the intensity of diffuse and specular component of light sources; k_a is ambient reflection constant, the ratio of reflection of reflected ambient light to total ambient light present in all points in the scene rendered; k_d is diffuse reflection constant, the ratio of reflection of reflected diffuse light to total diffuse light of incoming light (Lambertian reflectance); k_s is specular reflection constant, the ratio of reflection of reflected specular light to total specular light of incoming light; LS is the number of all light sources. $\vec{L}_j(x, y)$ is the unit direction vector from the point P on the surface toward the j^{th} light source, $\vec{N}(x, y)$ is the normal at this point on the surface, $\vec{R}_j(x, y)$ is the direction that a perfectly reflected ray of j^{th} light source would take from this point on the surface, and $\vec{V}(x, y)$ is the direction pointing towards the viewer (such as the camera); α is a shininess constant, which is larger for smoother surfaces. When we have color representations as RGB values, this equation will typically be calculated separately for R, G and B intensities. That is,

$$\begin{aligned}
I_c(x, y) &= k_{a,c} i_{a,c} + \\
&\sum_{j=1}^{LS} (k_{d,c} (\vec{L}_j(x, y) \cdot \vec{N}(x, y)) i_{d,c} + k_{s,c} (\vec{R}_j(x, y) \cdot \vec{V}(x, y))^\alpha i_{s,c}), \quad c \in \{r, g, b\}
\end{aligned} \tag{3.2}$$

Ambient light refers to the illumination from the surrounding objects. This is different from the direct illumination from light sources. We can assume that it is invariable in shadow region. When a foreground object casts shadow on a surface, it deprives the surface of some direct illumination from light sources, and thus induces a variation of the surface's appearance. Therefore, some of the light sources are partially or completely blocked. Then, Eq. 3.2 becomes:

$$I_{c,shadow}(x, y) = k_{a,c} i_{a,c} + \sum_{j=1}^{LS} \beta_{j,c} (k_{d,c} (\vec{L}_j(x, y) \cdot \vec{N}(x, y)) i_{d,c} + k_{s,c} (\vec{R}_j(x, y) \cdot \vec{V}(x, y))^\alpha i_{s,c}), \quad c \in \{r, g, b\} \quad (3.3)$$

where $\beta_{j,c} \in [0,1]$ is the attenuation factor, which represents to what extent the component c of the j^{th} light source is blocked. Since we could have different lines of light from different light sources to a given object, the attenuation factors could be different for different light sources. Further, given that diffraction is different at different wavelengths, the attenuation factors are different for the different components of light. Under cast shadows, the energy incident on the surface point is reduced, and thus each of the three components of the RGB value on the surface points will decrease. This reduction in the RGB value is captured in Figure 3-1.

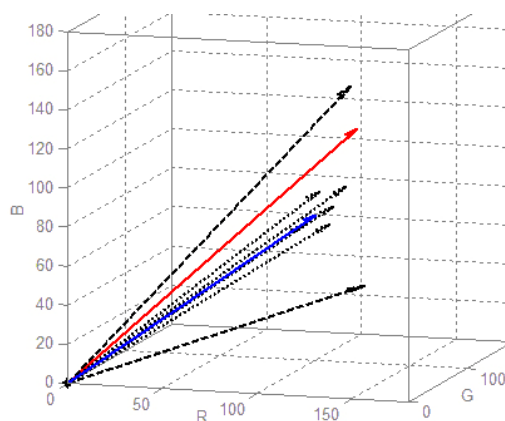


Figure 3-1: Our shadow model.

These vectors stand for the different RGB values of one pixel. The red vector is the background value (value in background model). The black vectors are the observed values in the video sequence. (The dotted vector is the potential shadow point value, and the dashed vector is the foreground value). The blue vector is the expected shadow value.

Under the proposed shadow model, when a surface point is in the shadow region, its value (the dotted vector in Figure 3-1) will fall close to its background value, because the color of the surface in shadow is similar to that in background. Then the three components of the RGB decrease similarly. When the surface point is on foreground objects, its value (the dashed vector in Figure 3-1) will be changed significantly and far from the background value, since the colors of foreground objects are usually different from that of the background.

Our hypothesis is that, at a given surface point, the effect of shadows on the background remains constant over a short period of time, because the illumination condition is expected to remain constant during such a short time. The effects of shadows are also similar, since shadows all result from blocking direct lights. Hence, one pixel's shadow values in recent sequences can be close to its background value vector (the red vector in Figure 3-1). These values cluster together, denoted by a beam of vectors in the RGB color space (the dotted vectors in Figure 3-1). Therefore, we can learn from the temporal information to determine a pixel's behavior under

shadows. However, we do not know which value is the actual shadow value of one pixel in recent frames. In our approach, we use the characteristics of shadow to predict the possible shadow value. For a given pixel, its observed value is a possible shadow value if it satisfies a luminance reduction constraint. We also use the chromaticity consistency to describe the possible shadow value. The chromaticity of foreground objects is usually distinct from that of the background. However, there could be cases where the chromaticity of the foreground objects could be similar to that of the background. The chromaticity alone will not be enough to characterize the shadow. Therefore, we propose to use spatial information to improve the description. Gradients can be applied to capture that information unless gradients are the same for both foreground and background. Unlike obtaining samples in each frame in the background model, shadows may not appear at the same pixel in each frame. So the statistical approach may suffer from insufficient training samples. To overcome this problem, the chromaticity descriptor is for the whole video sequence (global) and gradients descriptor is for a small window centered a pixel (local) in our approach. We call a pixel position with a possible shadow value potential shadow point. We can estimate the expected shadow value of the pixel from those potential shadow values in its temporal history in order to determine the actual shadow points in the current image.

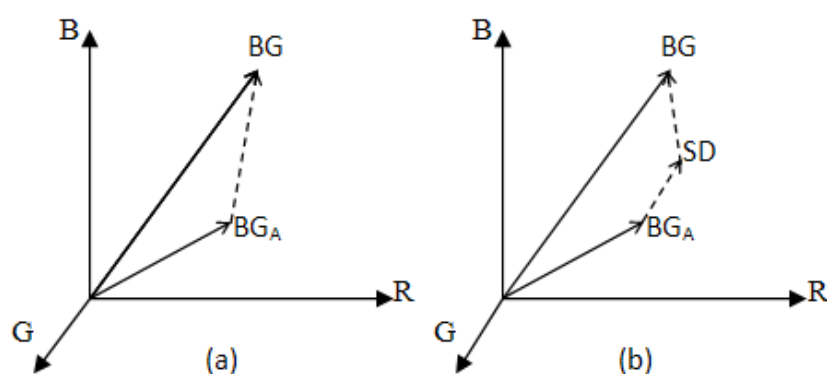


Figure 3-2: Two shadow models in the RGB color space.

The linear model (a) proposed in [15] and the nonlinear model (b) being proposed here.

Figure 3-2 shows two shadow models. Here, BG_A is the value of ambient light in Eq. 3.3 and BG is the background value of pixel. SD is the shadow value when pixel is shaded. Usually, the spectral power distribution (SPD) function of the ambient illumination is different from that of direct light sources. Hence the directions are different (compare BG_A with BG). If we assume that all light sources have the same SPD, we can get the linear model of cast shadow proposed in [15] (shown in Figure 3-2(a)). In this model, the shadow values of a pixel can only fall on the line from the ambient value BG_A to background value BG . However, direct light sources often have different SPDs in complex illumination conditions. When those direct light sources are partially or totally blocked, the shadow values of the same pixel in different images will fall close to the line from BG_A to BG . Therefore, we propose the nonlinear model shown in Figure 3-2(b). Our model does not limit light sources to have the same SPD. The shadow value can fall anywhere close to the vector BG .

3.2 Learning Moving Cast Shadow

3.2.1 Shadow Luminance Classifier

In the first step of our algorithm, we use a shadow luminance classifier to filter out those pixels that are unlikely to be shadow points by checking each foreground pixel detected by the GMM. From the Eq. 3.3, we know that the three components of a pixel value will decrease under cast shadows, which indicates that the pixels in shadows have a lower luminance. We assume that this decrease will be similar for all the three color components. We use the term potential shadow points to refer to those pixels with less luminance than the corresponding background. Then we use the shadow luminance classifier to determine those points.

Our approach is based on a modification of the color model proposed by Horprasert et al. [3]. In this model, the change of a pixel's value is captured by the color model that separates brightness distortion from chromaticity distortion component in the RGB color space. This is shown in Figure 3-3,

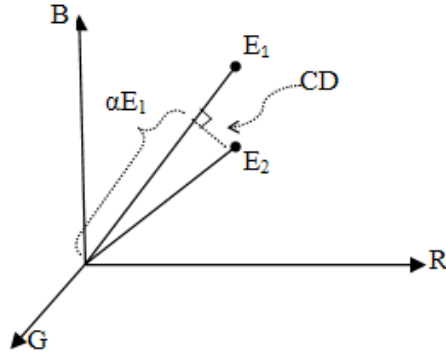


Figure 3-3: The color model proposed in Horprasert et al. [3].

where E_1 represents the latest background value of a pixel obtained from GMM and E_2 represents the current value of the pixel in a current frame. The difference between E_1 and E_2 is decomposed into two components, brightness distortion α and chromaticity distortion CD , where

$$\begin{cases} \alpha = \arg \min(\phi(\alpha_i)) = \arg \min((E_2 - \alpha_i E_1)^2) \\ CD = \|E_2 - \alpha E_1\| \end{cases} \quad (3.4)$$

The above requires searching for the minimum which could be quite time consuming. In our work, we compute brightness distortion and chromaticity distortion as follows:

$$\begin{cases} \alpha = \frac{\langle E_1, E_2 \rangle \|E_2\|}{\|E_1\| \|E_2\|} = \frac{\langle E_1, E_2 \rangle}{\|E_1\|^2} \\ CD = \sqrt{[E_{2,r} - \alpha E_{1,r}]^2 + [E_{2,g} - \alpha E_{1,g}]^2 + [E_{2,b} - \alpha E_{1,b}]^2} \end{cases} \quad (3.5)$$

where $\langle \cdot, \cdot \rangle$ is the inner product operator, and $\|\cdot\|$ is the norm of a vector.

We can easily see that $\alpha < 1$ if E_2 is a shadow pixel in a current frame. For each foreground pixel in M_1 (Eq. 2.6), we use α to represent the luminance reduction and then consider a pixel as potential shadow point if it satisfies the conditions:

$$\alpha_{\min} < \alpha < \alpha_{\max}, \quad 0 < \alpha_{\min} < \alpha_{\max} \leq 1 \quad (3.6)$$

where α is the brightness distortion between E_c and E_b , E_c is the observed RGB value of a pixel in current image and E_b is the background RGB value of the pixel. The parameters α_{\min} and α_{\max} define maximum darkness and maximum brightness of the shadow respectively. The two parameters are small in dark shadow (0.4~0.7) and large in light shadow region (0.7~1.0).

We define:

$$M_2 = \{X \mid X \in M_1 \wedge \alpha_{\min} < \alpha_X < \alpha_{\max}\} \quad (3.7)$$

Although the shadow luminance classifier is not accurate enough, it can remove most of the foreground objects area (or foreground points). After that, most of points in the resulting pixel set will be actual shadow points, except that there are still some noisy isolated pixels and foreground points. Those noisy isolated pixels can also be ignored, which does not affect the detection of the foreground objects. So our main task is to separate shadow points from a few foreground points.

3.2.2 Updating the Shadow Model

For each pixel in the image, we record its temporal information as its shadow track. When the pixel is a potential shadow point in one image, we cannot confirm whether it is an actual shadow point, but we still keep its value in the temporal history for that pixel. Thus, not all of the temporal information is reliable and truly reflect the behavior of the pixel in shadow.

Shadows reduce the luminance but retain the chromaticity and the texture, when compared to the reference background. Here, we evaluate the reliability of the pixel's historical shadow record (spatio-temporal context) using spectral and spatial features. We define the historical shadow record as a 2-D vector (ω, ν) , where ν is the historical potential shadow RGB value, ω is the confidence weight, the reliability of this value.

Based on the chromaticity and the local gradient consistency, we compute the confidence weight $\omega = \omega_c \omega_g$, where ω_c is the chromaticity confidence weight and ω_g is the gradient confidence weight.

In complex illumination environments, the intensity of direct lights is generally larger than that of ambient lights. And the umbra is darker than penumbra. Hence the luminance reduction of shadow pixels will be in a large range (for example 0.3~1.0). If a surface point is shaded in the umbra and also in the penumbra in the several recent frames, the values in those frames will all be kept in the history. Those shadow values are all reliable, but they may not cluster together since these values could be much different. That will lead to an unreliable estimation of the behavior of this surface point in shadow. Therefore, we classify the historical shadow values into three

categories namely high shadow ($0 < \alpha < 0.4$), medium shadow ($0.4 < \alpha < 0.7$), and low shadow ($0.7 < \alpha < 1$).

Global descriptor: Chromaticity confidence weight ω_c

The chromaticity distortion indicates the spectral difference between potential shadow value and background color value of a pixel. In Eq. 3.3, when the chromaticity of three lights (ambient light, diffuse light and specular light) are similar, no matter how much β is, the chromaticity distortion does not change much, for example in indoor scenes. When the chromaticity of the ambient light is very different from that of the diffuse light and specular light, the chromaticity distortion will change significantly, which fail to satisfy the assumption of chromaticity consistency. For example, in outdoor scenes, the chromaticity of the ambient light is more bluish than that of the sunlight (the SPD of the ambient light is different from the sunlight), since the ambient light is the reflected light of blue sky. Even if a chromaticity distortion in shadows is small, it does not mean that the pixel with a smaller chromaticity distortion is a shadow pixel. It could be a foreground pixel with chromaticity similar to background.

The chromaticity characteristic of shadow points has been used in earlier studies, for instance, as a deterministic condition to classify the shadow pixels in [3][4][5], or as a statistic to distinguish if a point is subject to chromaticity distribution of shadow in [12]. Here we use chromaticity distortion to define the chromaticity confidence weight. Using various video sequences with different illumination conditions, we construct the histogram of CD values, where we define the CD value to be chromaticity distortion between a potential shadow point and the corresponding background point. We observe that these different histograms have the same tendency: the values near $CD = 0$ have significantly larger probability. When the CD value increases, the probability decreases exponentially.

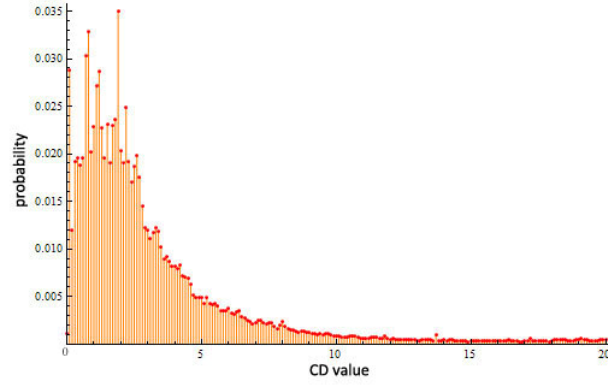


Figure 3-4: The CD-histogram showing the distribution of chromaticity distortion in shadows.

Figure 3-4 shows the distribution of CD for all potential shadow points in one video sequence. To overcome the problem of lack of data samples for statistical methods, in our approach, we establish one CD histogram for all pixels in a video sequence and update the histogram for each frame. That mechanism can accelerate the convergence speed and improve the adaptability. Independent of the chromaticity of each pixel, the chromaticity distortion is similar since the shadows appear in a similar way and in the same environment. The histogram is updated as follows:

$$H_t(CD) = (1 - \gamma) H_{t-1}(CD) + \gamma h_{t-1}(CD) \quad (3.8)$$

where $H(CD)$ is the histogram of CD in a video sequence. $h(CD)$ is the histogram of CD in one frame, $H(CD)$ is a global level condition updated using the local histogram $h(CD)$, t is time index, and γ is the learning rate. In order to transform this distribution into a confidence weight, we define the chromaticity confidence weight as follows:

$$\omega_c = H(CD), \quad CD \geq 0 \quad (3.9)$$

Thus, the confidence weight for chromaticity is the probability of occurrence of

CD .

Local descriptor: Gradient confidence weight ω_g

Some foreground objects have features very similar to the background in terms of spectral characteristics. The global chromacity distrotion may not be able to distinguish such regions. To handle this problem, we consider the spatial characteristic of shadow points. The shadow area is semi-transparent and presents the same textural characteristics between current frame and reference background image. Within a local area under shadow, we can expect that pixels have similar energy change. In Eq. 3.3, for a small local region, those incoming light has vectors with same direction. Therefore, when the lights are blocked somewhat, only the energy of the lights incident on the surface decreases and local luminance reduces at a similar ratio. To capture such local consistency of shadows, we define a local normalized gradient cross correlation (see Eq. 3.10).

We define a local area, 3×3 window centered at a potential shadow point, and use Eq. 3.10 to calculate the correlation. We call this correlation value the gradient confidence weight. We calculate the correlation between p_p , the potential shadow point in current image, and p_b , the corresponding point in background image as follows:

$$\omega_g = Corr(p_p, p_b) = \frac{\sum_{u \in W} (\|\nabla_p(u)\| \|\nabla_b(u)\|) + \varepsilon}{\sqrt{\sum_{u \in W} \|\nabla_p(u)\|^2 \sum_{u \in W} \|\nabla_b(u)\|^2 + \varepsilon}} \quad (3.10)$$

where ε is a small smoothening constant (set to 10 in our experiment) used to alleviate the effect of noise in the case of textureless surface. ∇_p and ∇_b are

respectively the gradient vector of current point and that of its background point in the same position. $\nabla(u) = (\nabla_x(u), \nabla_y(u))$ is the gradient vector of the point. We use the gray scale to calculate the spatial gradient.

The larger w_g is, the more similar a potential shadow point and its background point are.

3.3 Detecting Moving Cast Shadows

3.3.1 Detecting Shadow Regions with Temporal Information

From the analysis of our shadow model, the previous shadow values at the same pixel position should be similar. The main issue is to obtain a good estimation of shadow in current frame. An alternative that usually works quite well is to estimate the value of pixels using a moving average. In our approach, we estimate the shadow value of a pixel as a weighted average of its previous shadow value. Typically, the values in the very distant past are supposed to be weighted to zero, and the weights increase smoothly. But in our approach, we assign different weights to the previous values according to the characteristic of shadows.

For each potential shadow point X in M_2 (Eq. 3.7), we have a sequence of historical records including three shadow class categories (high, middle, low) (ω_i, v_i) , $i = 1, 2, \dots, N$. We select N' ($N' < N$) records from one category of its historical records. The N' records have the larger weight than the rest other records. We consider these N' records as the most reliable ones, and use the weight function to estimate $E(X)$, the expected value of the shadow point:

$$E(X) = \frac{1}{\sum_{j=1}^{N'} \omega_j} \sum_{j=1}^{N'} \omega_j v_j \quad (3.11)$$

The parameters N and N' can be chosen as needed, and reflect the adaptability of our algorithm for different illumination conditions.

Then we compare each potential shadow point with its corresponding expected shadow point by calculating α_X and CD_X value between X and $E(X)$. Here, we propose the following luminance constraint and chromaticity constraint to determine the similarity between the potential shadow point and the expected shadow point.

Luminance constraint:

Define:

$$M_3 = \{X \mid X \in M_2 \wedge 0.8 < \alpha_X < 1.2\} \quad (3.12)$$

Essentially, this means that a potential shadow point should have a similar luminance with the expected shadow point.

Chromaticity constraint:

We use an automatic threshold selection to detect the actual shadow point from M_3 . Construct a histogram of the chromaticity distortion CD of the points in M_3 . Suppose a detection rate r and compute the corresponding threshold T_{cd} . The CD values in the histogram are in an increasing order. Each CD value i has a corresponding probability $P(i)$. We sum the $P(i)$ up by increasing order of CD ,

until we find the CD value T_{cd} , which makes $\sum_{i=0}^{cd} P(i) > r$ for the first time. It satisfies the condition that r is the ratio of the potential shadow points in M_3 , whose CD values are smaller than T_{cd} .

$$T_{cd} = \arg \min_{cd} \left(\sum_{i=0}^{cd} P(i) > r \right) \quad (3.13)$$

where cd is the CD value in M_3 , $P(i)$ is the probability value at i in the histogram.

In our approach, we have used r 90%, which works well for both indoor and outdoor scenes. Thus, we can get the final set of shadow points:

$$M_4 = \{X \mid X \in M_3 \wedge CD_X < T_{cd}\} \quad (3.14)$$

3.3.2 Spatial Information for Shadow Correction

When foreground objects are similar to the background in terms of shadow properties, shadow detection error may occur (see blue pixels Figure 3-5). To improve the accuracy of the shadow detection, we use spatial information to correct the detection error. According to the geometric property of shadow, we know that typically shadows are around foreground objects and shadow pixels cluster together. Hence, most of the boundary of an actual shadow region will be adjacent to the background pixels. We analyze each connected shadow region based on the geometric property of shadows. Using depth first search (DFS) or breadth first search (BFS), we find the connected shadow regions in M_4 and get the boundary of each shadow

region. If the percentage of boundary pixels adjacent to background pixels is greater than 50%, the shadow region is considered as the actual shadow. Otherwise, this region will be corrected to foreground region. Figure 3-5 shows an example, where the blue pixels are corrected to foreground pixels.



Figure 3-5: Shadow detection error corrected by spatial information.

The foregoing discussion is summarized in the shadow detection algorithm shown in Figure 3-6.

```

Shadow Detection Algorithm
for each video sequence do
   $M_1$  = foreground pixels detected by GMM;
   $M_2 = M_3 = M_4 = \Phi$ ;
  for each pixel X in  $M_1$  do
     $\alpha_X = \text{compare}(X, B_X)$ ;
    if  $\alpha_{\min} < \alpha_X < \alpha_{\max}$  then
       $M_2 = M_2 \wedge X$ ;
       $\omega_x = \omega_c \omega_g$ ;
      update the history of X;
    end if
  end for

  for each pixel X in  $M_2$  do
    compute expected shadow value  $E_X$ ;
     $(\alpha_X, CD_X) = \text{compare}(X, E_X)$ ;
    if  $\tau_1 < \alpha_X < \tau_2$  then
       $M_3 = M_3 \wedge X$ ;
    end if
  end for

  compute the threshold  $T_{cd}$ ;
  for each pixel X in  $M_3$  do
    if  $CD_X < T_{cd}$  then
       $M_4 = M_4 \wedge X$ ;
    end if
  end for
  perform geometric correction.
end for

```

Figure 3-6: Shadow detection algorithm.

Chapter 4: Results and Discussion

In this chapter, we present results of the proposed method on a suite of indoor and outdoor video sequences and a discussion on the parameters in our algorithm. ‘Laboratory’, ‘Intelligent Room’ and ‘Hallway’ are indoor scenes, ‘Campus1’, ‘Campus2’ and ‘Highway’ are outdoor scenes. We also compare the quantitative accuracy of the proposed method with other approaches.

In the experiments, we set $\alpha_{\min} = 0.7$, $\alpha_{\min} = 1$ for all the sequences except ‘Highway’. For ‘Highway’, $\alpha_{\min} = 0.3$, since the energy of direct light sources is much larger than that of ambient light in ‘Highway’. We set $N = 30$, $N' = 10$, $r = 90\%$ for the shadow detection stage.

The algorithm has been implemented in C#.net and processes 6 frames a second for a frames size of 320×240 pixels. It has been tested on a Intel Core(TM)2 Duo 2.53GHz processor without any filtering or morphological operations.

4.1 Qualitative Results

In order to show the effectiveness of proposed method, the results presented here are raw data without any post-processing. In all sequences, you can see isolated points called salt and pepper noise, which can be easily removed by other methods, such as morphology-based method.

In the figures, the first column is the original sequence, the second column the foreground and shadow detected by GMM, the third column the shadows detected by our approach and the fourth column the final foreground objects.

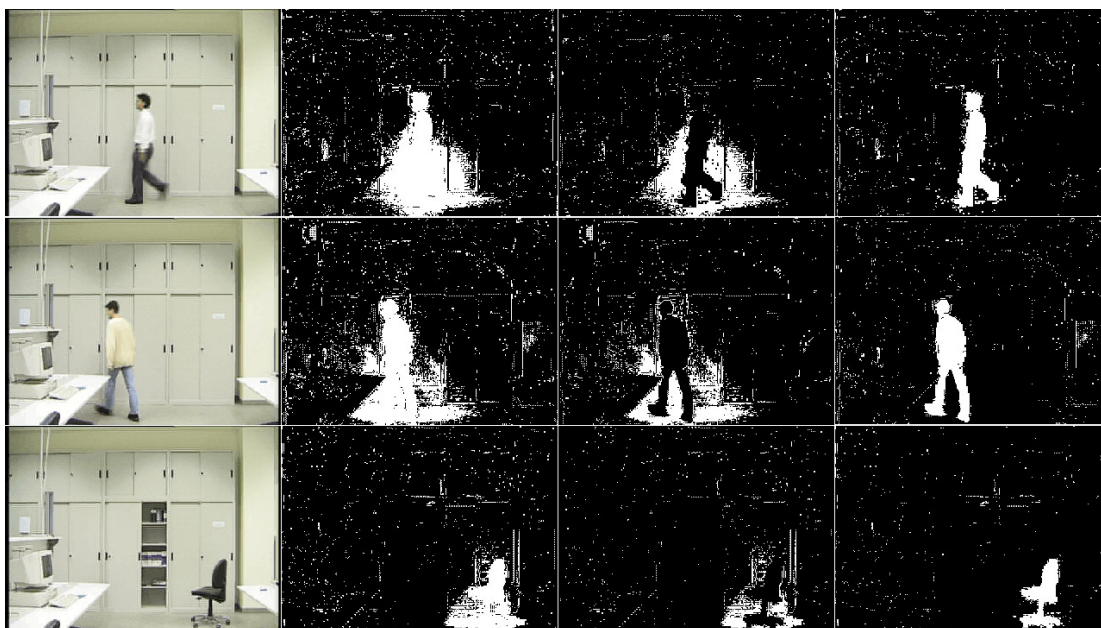


Figure 4-1: Frame 158, frame 299 and frame 885 of 'Laboratory'

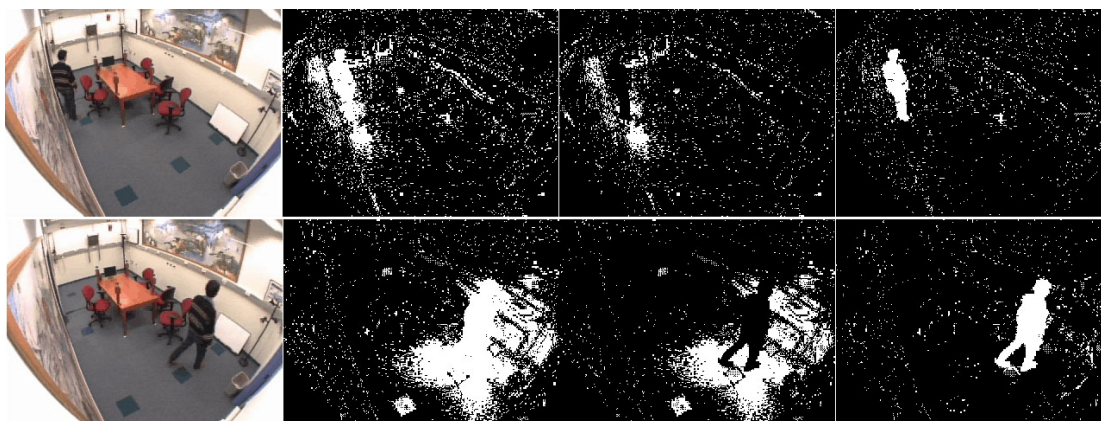


Figure 4-2: Frame 92 and frame 297 of 'Intelligent Room'.



Figure 4-3: Frame 197 and frame 447 of 'Hallway'.

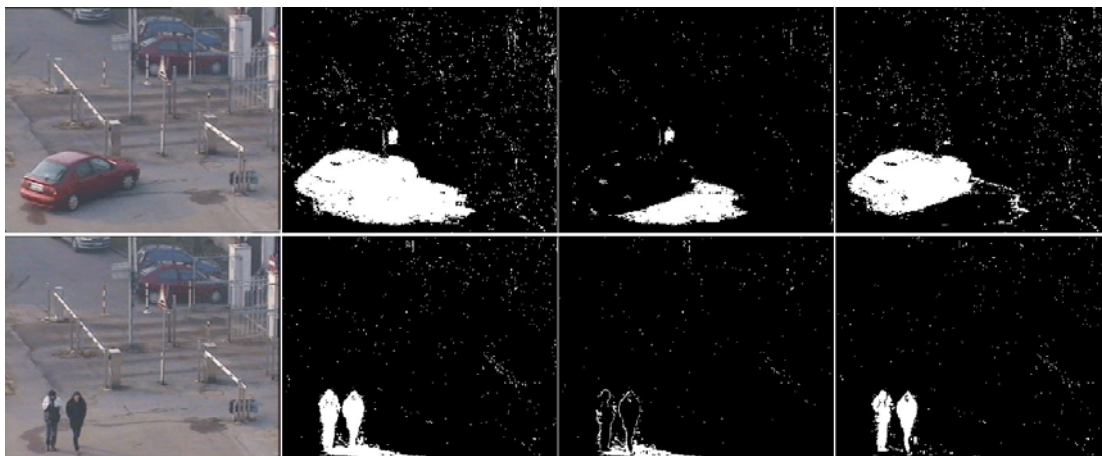


Figure 4-4: Frame 61 and frame 418 of 'Campus1'.

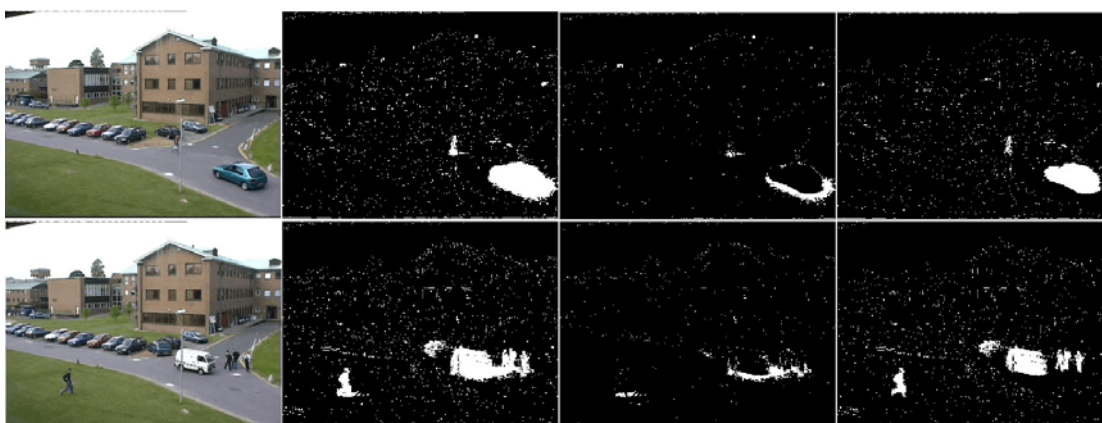


Figure 4-5: Frame 517 and frame 878 of 'Campus2'.

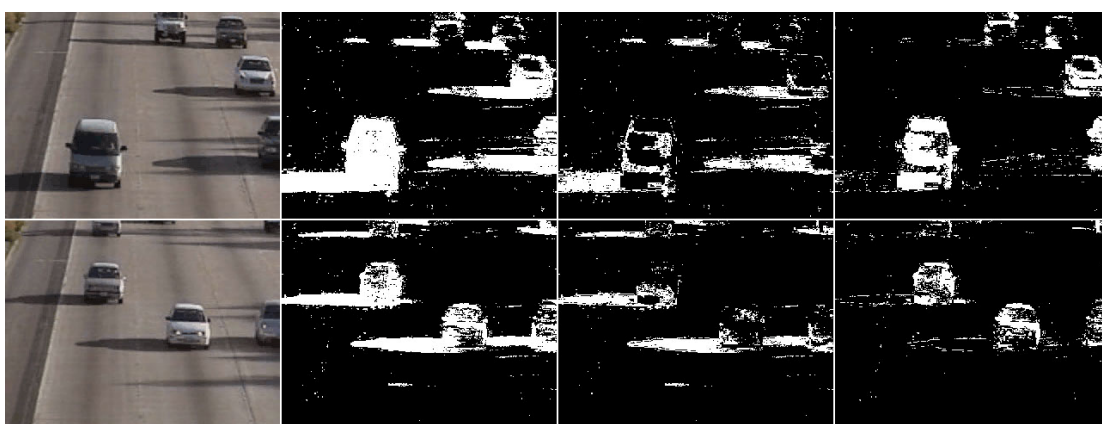


Figure 4-6: Frame 220 and frame 247 of 'Highway'.

In these sequences, shadows range are from small to large, dark to light, next or adjacent to the moving object. The type, size and speed of objects also could vary

considerably. In a special case, if foreground objects move so fast that they will stay fewer frames in the scene, there will be fewer data samples for shadows. The convergence of parameters in the algorithm become slower compared to ordinary motion speed and the learning process will cost more time. This is a common difficulty for all statistics-related approach. In the ‘Laboratory’ sequence, the shadows of persons and chair are detected effectively without foreground information. Besides, different foreground objects blocked the lights in a similar way and the effect of shadows cast by different objects is quite similar. In the ‘Hallway’ sequence, the illumination environment is more complex: multiple light sources, reflections on the floor and large penumbra regions. The ‘Campus1’ and ‘Campus2’ sequences are noisy sequences. In Figure 4-4, we can see that the right part of car’s shadow is not detected accurately, because the shadow appears first on the surface, when there are no historical records to use. From all the sequences, we can see that the shadows cast on different surfaces can be detected effectively.

4.2 Quantitative Results

To evaluate the proposed method quantitatively, we use two metrics: shadow detection rate η , related to the percentage of shadow pixels wrongly detected as foreground, and shadow discrimination rate ξ , related to points both wrongly detected as foreground and shadow pixels, defined in [18]. The formulation of the metrics is as follows [18]:

$$\eta = \frac{TP_S}{TP_S + FN_S} \quad \xi = \frac{\overline{TP_F}}{TP_F + FN_F} \quad (4.1)$$

where TP_S and TP_F are the respective number of shadow pixels and foreground

pixels correctly detected. FN_S is the number of shadow pixels wrongly detected as foreground. FN_F is the number of foreground pixels wrongly classified as shadow or background. $\overline{TP_F}$ is the number of ground truth pixels of the foreground objects minus the number of pixels detected as shadows, but belonging to foreground objects. We also use another two metrics: foreground detection rate ζ , related to the percentage of foreground pixels wrongly detected as shadow, and balancing score FScore, which is a balance between shadow detection rate and shadow discrimination rate:

$$\zeta = \frac{TP_F}{TP_F + FN_F^*} \quad FScore = \frac{2\eta\xi}{\eta + \xi} \quad (4.2)$$

where FN_F^* is the number of foreground pixels wrongly detected as shadow.

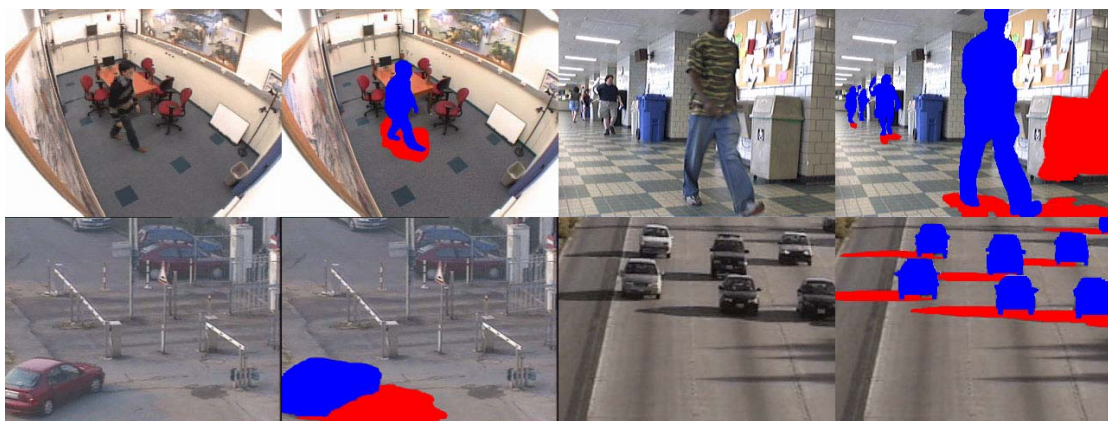


Figure 4-7: Some ground truth samples.

Table 4-1 shows the quantitative results of our proposed method in indoor and outdoor sequences. Our algorithm has an excellent performance under the different illumination conditions.

Table 4-1: Quantitative results of our method in indoor and outdoor sequences

	$\eta\%$	$\xi\%$	$\zeta\%$	FScore%
Intelligent Room	81.09	96.52	96.06	88.13
Hallway	90.17	97.48	97.01	93.68
Campus1	84.34	94.88	94.64	89.30
Highway	78.48	84.67	78.31	81.46

4.3 Comparative Result

4.3.1 Validation with SNP

The following comparison with the SNP (statistical non-parametric named in [18]) algorithm of Horprasert et al. [3] is to validate the performance of the proposed algorithm. As mentioned in section 3.2.1, SNP give a computational color model. For a given pixel, the expected background value $E = [\mu_r, \mu_g, \mu_b]$ is computed from N training static background images, which do not have foreground objects, and $S = [\sigma_r, \sigma_g, \sigma_b]$ is standard deviation. For each pixel in a new frame $X_t = [X_r, X_g, X_b]$, brightness and chromaticity distortions from the background value are given by:

$$\left\{ \begin{array}{l} \alpha_t = \frac{\left(\frac{X_r \mu_r}{\sigma_r^2} + \frac{X_g \mu_g}{\sigma_g^2} + \frac{X_b \mu_b}{\sigma_b^2} \right)}{\left[\frac{\mu_r}{\sigma_r} \right]^2 + \left[\frac{\mu_g}{\sigma_g} \right]^2 + \left[\frac{\mu_b}{\sigma_b} \right]^2} \\ CD_t = \sqrt{\left[\frac{X_r - \alpha_t \mu_r}{\sigma_r} \right]^2 + \left[\frac{X_g - \alpha_t \mu_g}{\sigma_g} \right]^2 + \left[\frac{X_b - \alpha_t \mu_b}{\sigma_b} \right]^2} \end{array} \right. \quad (4.3)$$

It uses the normalized distortion of brightness α' and distortion of the chrominance CD' to classify a pixel in four categories as shown in (24). For i^{th}

pixel in a subsequence,

$$\begin{aligned}
 a_i &= \sqrt{\frac{\sum_{i=1}^N (\alpha_i - 1)^2}{N}} & b_i &= \sqrt{\frac{\sum_{i=1}^N CD_i^2}{N}} \\
 \alpha_i' &= \frac{\alpha - 1}{a_i} & CD_i' &= \frac{CD_i}{b_i}
 \end{aligned} \tag{4.4}$$

Pixel classification is then performed as follows:

$$\begin{aligned}
 \text{Foreground:} & \quad CD_i' > \tau_{CD} \text{ or } \alpha_i' < \tau_{\alpha_0} \text{ else} \\
 \text{Background:} & \quad \alpha_i' < \tau_{\alpha_1} \text{ and } \alpha_i' > \tau_{\alpha_2} \text{ else} \\
 \text{Shadow:} & \quad \alpha_i' < 0 \text{ else} \\
 \text{HighlightedBackground:} & \quad \text{otherwise}
 \end{aligned} \tag{4.5}$$

The thresholds in (4.5) are automatically selected from the histograms of α' and CD' , which are computed from the N training background images, with a detection rate.

The classification or segmentation can be represented as the following Figure 4-8 mapped in the RGB color space.

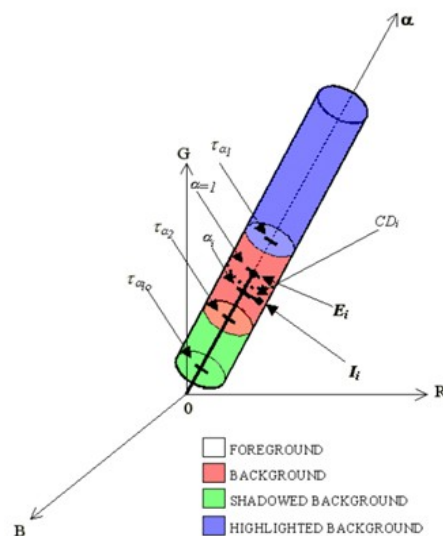


Figure 4-8: The segmentation approach of SNP [3].

E_i represents the expected color of a given pixel and I_i represents the color value of the pixel in the current frame. This figure comes from [18].

The SNP algorithm is time-consuming due to its complex normalization procedure and constructing the histogram for each pixel. Further, its background model and histograms are not updated (if updated, the computation cost becomes too much). Thus, for some complex illumination condition, the thresholds will be inaccurate given the lack of update. Our approach updates the shadow information for each pixel, is easy to implement, and fast to adapt to new illumination conditions.

The SNP algorithm uses the same CD threshold to separate background and shadow from foreground. This may not be suitable for noisy sequences such as ‘Campus1’. If the CD threshold is large, most shadow regions may be detected, but also more foreground regions are incorrectly classified as shadow. On the other hand, if the CD threshold is small, the shadow detection rate will decrease. Figure 4-9 presents results showing the differences between the SNP algorithm and our method on ‘Campus1’.

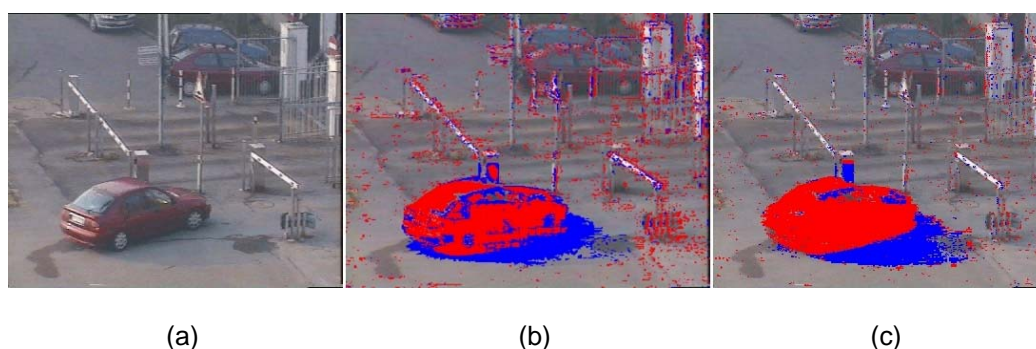


Figure 4-9: Qualitative comparative results with the SNP algorithm.

(a) original, (b) results with SNP, (c) results with our method.

SNP's result of ‘Campus1’ comes from [18].

The SNP algorithm finds the difference between current pixel value and its

expected background value, but our algorithm compares current pixel value to its expected shadow value. Our approach is more accurate. For example, suppose a current shadow pixel's brightness distortion α is 0.8 (compared to its background value). Thus, in Figure 4-8, at the point $\alpha=0.8$ on the α -axis, in the circular section centered point 0.8(cross section of cylinder), the value on this circular section can all be regard as shadow point by the SNP algorithm. This is not accurate, because some foreground pixel also can be on this circular section. But our approach can decide which region on that circular section is shadow point value region, not all the circular section. Our shadow expectation method can more accurate in deciding the possible shadow values in the RGB color space. In estimating the expected shadow value, we not use the spectral feature but the spatial feature. This is a more accurate mechanism.

4.3.2 Comparison with Other Methods

We show the quantitative comparison results of indoor and outdoor sequences in Table 4-2 and Table 4-3. The results of other proposed approaches are taken from the corresponding references. Our method produced the best overall result on the FScore metric and higher result on the shadow detection rate and discrimination rate, for both indoor & outdoor scenes.

Table 4-2: Quantitative comparison on surveillance sequences

	Intelligent Room			Campus1		
	$\eta\%$	$\xi\%$	FScore%	$\eta\%$	$\xi\%$	FScore%
Proposed	81.09	96.52	88.13	84.34	94.88	89.30
ILT[19]	88.23	89.05	88.64	81.40	92.61	86.64
DNM1[18]	78.61	90.29	84.05	82.87	86.65	84.72
SP[18]	76.27	90.74	82.88	72.43	74.08	73.25
SNP[18] (for [3])	72.82	88.90	80.06	80.58	69.37	74.56

DNM2[18]	62.00	93.89	74.68	69.10	62.96	65.89
----------	-------	-------	-------	-------	-------	-------

Table 4-3: Quantitative comparison on surveillance sequences

	Hallway			Highway		
	$\eta\%$	$\xi\%$	FScore%	$\eta\%$	$\xi\%$	FScore%
Proposed	90.17	97.48	93.68	78.48	84.67	81.46
Physical Feature[20]	82.05	90.47	86.05	70.83	82.37	76.17
Physical[16]	71.69	88.25	79.11	72.34	84.98	78.15
Kernel[15]	72.40	86.70	78.91	70.50	84.40	76.83
GMSM[13]	60.50	87.00	71.37	63.30	71.30	67.06

4.4 Discussion on Parameters

To explore the effect of certain parameters on the result of shadow removal, we will change specific parameters and remain all else parameters. By linearly increasing the learning rate (0.01~0.15)(Eq. 3.8), the quantitative result of our algorithm are almost not changed (see Figure 4-9(a)). And when $N = 20,30,40$ or $N' = 5,10,15$, (Section 3.31) the result are also changed slightly. It is suggested that our algorithm is not sensitive to these three parameters. For a changing illumination condition at time t , foreground, background and shadow are all affected in the scene. So they will have a similar RGB change compared to previous background. For example, all of them become reddish at sunset. Even if the learning rate is not chosen exactly appropriately and therefore $H(CD)$ is not updated properly, but when we compute expected value for every element in $M2$ under that learning rate, all the expected value have similar error. That is to say, the poor learning rate changes the value, but does not change the alignment or entire relative relation. Hence, when we compute $M4$, the alignment is not changed and those values least possible to be

shadow are still least possible. Similarly, for N and N' , If they are selected not exactly properly, it will totally affect all the pixels in M_2 . But M_3 and M_4 are affected slightly.

Linearly increasing the detection rate (0.60~1.00), the shadow detection rate drops slowly while the discrimination rate and FScore rise sharply (see Figure 4-9(b)). There is a high point on the curve at 0.90. This value works well for all the video sequences in the benchmark suit.

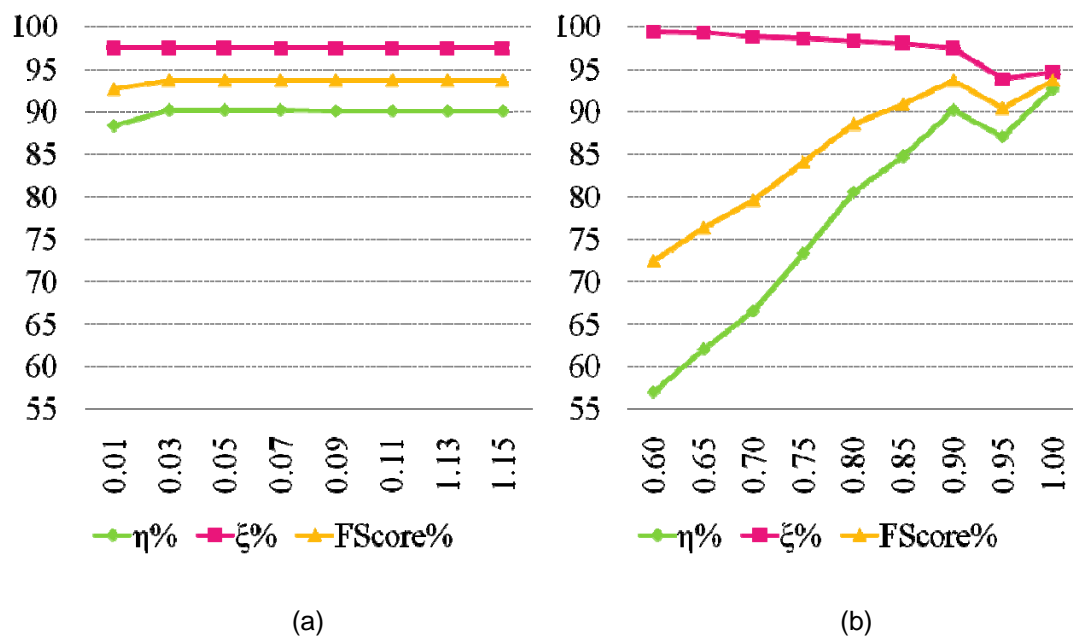


Figure 4-9: Effect of learning rate and detection rate on quantitative result of our algorithm.

(a) effect of learning rate, (b) effect of detection rate.

Chapter 5: Conclusion & Future Work

5.1 Conclusion

In this thesis, we have presented a new shadow model and a pixel-based algorithm capable of detecting cast shadows in various scenes. The method combines statistical information with a spatio-temporal context with geometric information about the image region. It requires only a short time to learn the shadows and is robust to detect the cast shadows. The framework of this algorithm gives the idea that we can give a confidence to the potential shadow point to learn the shadows. The confidence can be composed of the shadows' invariant characteristics. And it chooses the appropriate historical potential points to estimate the shadow value in the next frame. Thus, it can be effective in detecting the cast shadows. Qualitative and quantitative results presented in this thesis validate this approach.

5.2 Future Work

At last step of this approach, a better definition of the similarity between expected shadow values and current values could produce improved results. The method with detection rate for automatic threshold selection should be proved theoretically, although it is effectively in the typical benchmark video sequence. Additionally, there needs to be an effort to improve the implement of the algorithm, for instance the execution time could be significantly reduced with software code optimization.

Object detection is the first stage in many video processing applications. The common method, used to detect the moving objects, is not accurate, which will dramatically influence subsequent operation. Our long term aim is to model and analyse the human actions. Thus, the next step is the tracking of the objects after we

have detected & removed the shadows.

References

- [1] C. Stauffer and W.E.L. Grimson. (2000). Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8), pp. 747-757.
- [2] P.W. Power and J.A. Schoonees. (2002). Understanding background mixture models for foreground segmentation. *Proc. Image and Vision Computing*, pp. 267-271.
- [3] T. Horprasert, D. Harwood, and L. S. Davis. (1999). A statistical approach for real-time robust background subtraction and shadow detection. *IEEE International Conference on Computer Vision*.
- [4] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti. (2001). Improving shadow suppression in moving object detection with HSV color information. *IEEE International Conference on Intelligent Transportation Systems*, pp. 334-339.
- [5] O. Schreer, I. Feldmann, U. Goelz, and P. A. Kauff. (2002). Fast and robust shadow detection in videoconference applications. *Proc. Fourth IEEE Int'l Symp. Video Proces. and Multimedia Comm.*, pp. 371-375.
- [6] E. Salvador, A. Cavallaro, and T. Ebrahimi. (2004). Cast shadow segmentation using invariant color features. *Computer Vision and Image Understanding*, pp. 238-259.
- [7] A. J. Joshi, S. Atev, O. Masoud, N. Papanikolopoulos. (2007). Moving shadow detection with low- and mid-level reasoning. *IEEE International Conference on Robotics and Automation*.
- [8] A. T. Nghiem, F. Bremond, M. Thonnat. (2008). Shadow removal in indoor scenes. *IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance*.
- [9] J. Stander, R. Mech, and J. Ostermann. (2002). Detection of moving cast shadows for object segmentation. *IEEE Transactions on Multimedia*, 1(1), pp. 65-76.

- [10] A. Leone, C. Distanto, F. Buccolieri. (2007). Shadow detection for moving objects based on texture analysis. *Pattern Recognition*, 4(4), pp. 1222-1233.
- [11] Z. Wei, F. Xiang Zhong, Y. Xiao. Kang. (2007). Moving cast shadows detection using ratio edge. *IEEE Trans. Multimedia*, 9(6), pp. 1202–1214.
- [12] Z. Liu, K. Huang, T. Tan, and L. Wang. (2007). Cast shadow removal combining local and global features. *IEEE Conf. on CVPR*, pp. 1–8.
- [13] N. Martel-Brisson and A. Zaccarin. (2007). Learning and removing cast shadows through a multidistribution approach. *IEEE Trans. PAMI*, 29(7), pp. 1133–1146.
- [14] J. B. Huang and C. S. Chen. (2008). Learning moving cast shadows for foreground detection. In *The Eighth International Workshop on Visual Surveillance (VS), in conjunction with ECCV*.
- [15] N. Martel-Brisson and A. Zaccarin. (2008). Kernel-based learning of cast shadows from a physical model of light sources and surfaces for low-level segmentation. In *CVPR*.
- [16] J. B. Huang and C. S. Chen. (2009). A physical approach to moving cast shadow Detection. In *IEEE Conf. on ASSP*, 19(24), pp. 769-772.
- [17] B. T. Phong. (1975). Illumination for computer generated pictures. *Commun. the ACM*, 18(6), pp. 311-317.
- [18] A. Prati, I. Mikic, M. Trivedi and R. Cucchiara. (2003). Detecting moving shadows: Formulation, algorithms and evaluation. *IEEE Trans, PAMI*, 25(7), pp. 918-924.
- [19] Y. Y. Hu and R. D. Wang. (2008). Moving shadow removal based on ILT. *International Conference on Cyberworlds*, pp. 127-130.
- [20] J. B. Huang and C. S. Chen. (2009). Moving cast shadow detection using physics-based features. In *Proc. CVPR*.