# AUTOMATIC REMOVAL OF COMPLEX SHADOWS FROM INDOOR VIDEOS

Deepankar Mohapatra

Thesis Prepared for the Degree of

MASTER OF SCIENCE

UNIVERSITY OF NORTH TEXAS

August 2015

APPROVED:

Xiaohui Yuan, Major Professor
Song Fu, Committee Member
Kathleen Swigger, Committee Member
Barrett Bryant, Chair of the Department of Computer Science and Engineering
Costas Tsatsoulis, Dean of the College of Engineering and Interim Dean of the Toulouse Graduate School

Mohapatra, Deepankar. Automatic Removal of Complex Shadows from Indoor Videos. Master of Science (Computer Science), August 2015, 53 pp., 6 tables, 25 figures, 24 numbered references.

Shadows in indoor scenarios are usually characterized with multiple light sources that produce complex shadow patterns of a single object. Without removing shadow, the foreground object tends to be erroneously segmented. The inconsistent hue and intensity of shadows make automatic removal a challenging task. In this thesis, a dynamic thresholding and transfer learning-based method for removing shadows is proposed. The method suppresses light shadows with a dynamically computed threshold and removes dark shadows using an online learning strategy that is built upon a base classifier trained with manually annotated examples and refined with the automatically identified examples in the new videos.

Experimental results demonstrate that despite variation of lighting conditions in videos our proposed method is able to adapt to the videos and remove shadows effectively. The sensitivity of shadow detection changes slightly with different confidence levels used in example selection for classifier retraining and high confidence level usually yields better performance with less retraining iterations.

ACKNOWLEDGEMENTS

I am very thankful to my advisor, family, friends and colleagues in the CoVIS lab for all the support and encouragement I have received through the course of my Masters. In particular, I am grateful to my major professor, Dr. Xiaohui Yuan for constantly guiding me through the course of thesis work. I would not have completed this work if not for the guidance and motivation I received from him. I would like to thank all of my committee members for the time they spent on my thesis.

TABLE OF CONTENTS

Page

# LIST OF TABLES

Page

LIST OF FIGURES

Page

# CHAPTER 1

# INTRODUCTION

Computer vision has been a popular research area in the last decade, applications today are making their mark in the commercial market, and this makes it increasingly important to focus on making computer vision systems reliable and efficient. Background subtraction is a critical step in many computer vision applications ranging from object tracking to action recognition, which require accurate foreground objects.

These object may vary from human object to other objects which may need to be accurately segmented from the background, some prominent examples are cars, bacteria, blood cells etc. Applications using a video sequence, have to segment the foreground in every frame, this usually involves saving a general model for the background, which may or may not update depending upon the algorithm in question. This model is used to segment the foreground object.

However, the foreground object is usually distorted by non-stationary artifacts and noises. These noises may be caused by a variety of factors such as moving trees, inaccurate segmentation algorithm, shadows etc. This poses complications and may result in incorrect outputs. For instance, if a human segmentation is not accurate it might give incorrect results in a human pose estimation algorithm.

A most challenging distortion is shadows of the moving object. Due to its nature of dynamically emerging with objects, which causes regional color changes with respect to the reference image, shadow is usually misclassified as foreground object or part of it.

Shadows differ when lighting condition varies. As shadows can change dynamically with changes in lighting conditions, it is therefore essential to understand how and when shadow properties change in a video.

Ariel et al. [**3**] defines shadow as a photometric phenomenon which occurs when an

object partially or totally blocks the direct light source. Shadows can also be static or dynamic according to [**3**], static shadows are implicitly handled by many foreground detection algorithms such as [**5**], as they have a constantly changing model for the background. Dynamic shadows are those regions which move between consecutive frames, either because the object moves or because the light source moves.

Shadow can be complex to remove as there are a few issues that need to be handled.

(1) Shadow can be formed from a single or multiple light sources. For single light sources, it is easier to predict the direction of the shadow with respect to the light source and the object detected. Whereas in the case of a complex indoor environment with multiple light sources effecting the shadow pattern predicting the direction of the shadow becomes more complex.

(2) Objects can have shadows casted on themselves by other objects. This situation can cause problems for color based algorithms as many portions of the objects may be mis-classified as shadow.

(3) Ambient light scenario (indoor environment with many lights) can cause shadows of different shade. In indoor situations a single shadow pattern may contain many gradients of shadow color.

(4) Shadows can be spatially joint to the object which makes it more difficult to classify shadows.

There have been many methods developed to handle shadow removal in a variety of outdoor scenarios, e.g., traffic monitoring [**7**] and surveillance [**3**]. However, these methods usually assume a single light source (such as the sun) and are facing difficult in indoor lighting where multiple light sources combine to produce complex shadow intensity. Research has been conducted for indoor scenarios [**19**], in which a manually specified threshold was used.

Shadows in indoor scenarios are usually characterized with multiple light sources that produce complex shadow patterns of a single object. An example is shown in Fig. 1.1(a). As

(a) (b) (c)

Figure 1.1. Complex shadow and the background subtraction results. (a) a frame showing complex shadow of different shades. (b) background subtraction result. (c) background subtraction with shadow removal.

a result of multiple shadows casting on the wall, part of the shadow appears brighter in color than the other. Without removing shadow, the foreground object tends to be erroneously segmented, as shown in Fig. 1.1(b); and with shadow removal the optimal body silhouette contains no shadow component, as shown in Fig. 1.1(c).

The inconsistent shades of shadows make automatic removal a challenging task; simple color-based methods are ineffective and could cause shattered object of interest [**19**]. Another issue in shadow removal from videos is time efficiency. Serving as a preprocessing step for video analysis, shadow removal shall take little computational time to ensure real-time performance for the forthcoming processes.

In this thesis, I present a dynamic thresholding and transfer learning-based method for removing shadows in videos of indoor environments with multiple light sources that generate complex shadows. This method categorizes shadows into light shadows and dark shadows based on the color changes induced to the background model. In light shadows, chroma

of a pixel has little changes but its intensity is mostly impacted. Hence, a threshold is dynamically determined to remove light shadows. For dark shadows an online learning method is proposed to identify the unwanted regions. A model is initially trained with manually annotated examples and refined with the videos on-the-fly.

## 1.1. Thesis Contributions

The contributions of this thesis are as follows:

(1) Framework for analyzing dark and light shadows in a complex indoor lighting condition.

(2) Light shadow removal with dynamic thresholding- A dynamic thresholding mechanism is presented, which takes no user input to decide the threshold for the removal of light shadows.

(3) Transfer learning process to remove dark shadows - A pre-trained classifier is used to detect dark shadow pixels. Transfer learning is used to tune the classifier to the indoor environment in question.

## 1.2. Applications

Many current and future applications can benefit from an accurate shadow removal algorithm. As background subtraction is an intermediate step in many applications, reduced noise due to shadows in this step can improve accuracy and efficiency.

(1) Pose detection algorithms [**24**] rely heavily on a noiseless human silhouette. Various types of segmentation and calculations are performed on this silhouette to get estimates of pose in every frame, a removed shadow from the foreground can improve the accuracy of estimation of these poses.

(2) Automatic surveillance systems [**3**], usually face the problem when two human objects appear to be one because of an extended shadow. A shadow removal process can eliminate this behavior of the system, making it more accurate.

(3) Many applications face incorrect predictions because of a disjoint shadow from the body. A disjoint shadow is spatially not connected to the object, in the frame in question. This causes the system to classify two moving objects instead of one [**3**].

(4) Shadowed regions also increase the number of pixels to be processed after background subtraction. Shadow removal will leave the systems with fewer and accurate foreground to process.

## 1.3. Thesis Organization

The rest of the thesis is organized as follows:

- Chapter 2 presents the motivation, related work to shadow removal in videos and, in particular, methods to handle indoor scenarios. It also reviews various methods and algorithms applied in the system.
- Chapter 3 describes various properties and classifications of shadows. These classifications and properties become the basis of differentiating shadows from the foreground object.
- Chapter 4 describes the system in detail. Light shadow removal is explored by analyzing it's behavior and elaborating on the dynamic thresholding process. Dark shadow removal is performed by a transfer learning process which adapts to the new video by building on a pre-trained classifier.
- Chapter 5 consists of evaluation and experimental results on complex shadow patterns.
- Chapter 6 describes conclusion and future work.

# CHAPTER 2

# BACKGROUND

## 2.1. Motivation

Human Pose tracking and body part segmentation has been an area of extensive research in the past few years. Human Pose tracking refers to tracking the pose of a human object as the object changes position and pose. Body Part segmentation is an intermediate step to pose recognition. It tries to segment the human body in question, into separate parts such as head, torso, legs etc.

Research on body part segmentation has been done on both 2D and 3D camera systems. 3D camera systems generally are accompanied with an additional sensor to calculate depth of the object. This makes it trivial to remove shadows and detect object occlusions. On the other hand with a 2D camera system, it becomes increasingly problematic to handle some issues such as,

(1) Shadows casted when a human object walks around. These shadows are classified as foreground by background subtraction algorithms.

(2) It is hard to detect or predict occlusion in a 2D camera system, due to the absence of depth data.

(3) 2D camera systems might need initial calibration to judge the distance of the object from the camera.

### 2.1.1. Human energy consumption calculation with a 2D camera system

A proposed application of human body segmentation using a 2D camera system is human energy calculation. Following is the overall objective of the system.

(1) A system to track energy cost of human activity using 2D cameras.

(2) The proposed system should be computationally inexpensive, i.e. it can run and provide feedback in real time.

(3) System should track individual body parts such as head, hands, torso, etc.

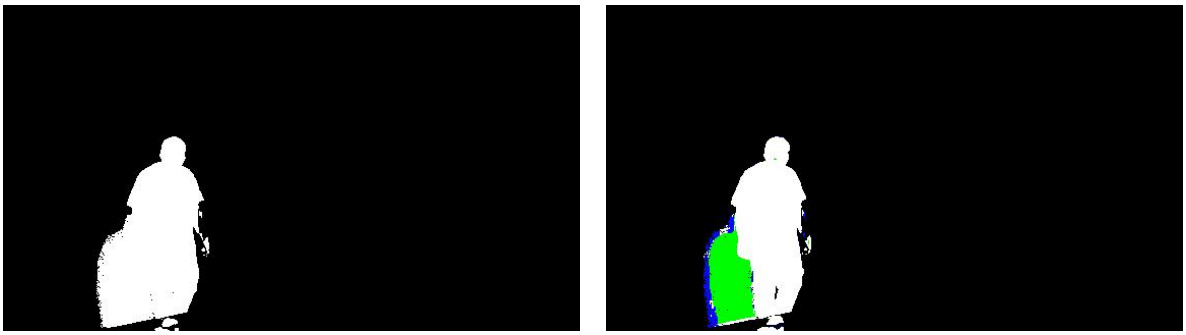
(4) System should handle occlusions of human body parts by predicting their location.

The proposed framework for the above approach is as follows,

(1) The system will use two 2D cameras to track a human object, the purpose of two cameras is to handle total occlusion of some body parts.

(2) Human body parts would be segmented in an initialization phase and features would be captured for the same.

(3) Movement of individual body parts would be tracked by a tracking algorithm in consecutive frames.



(a) (b)

Figure 2.1. Affect of shadow on human object segmentation (a) Frame with shadow classified as foreground object (b) Foreground object separated from shadow.

Background subtraction on each frame is required both for segmentation as well as for tracking in most algorithms, this is done to accurately segment the foreground object before segmenting or tracking. As mentioned in the previous chapter, a very prominent noise element in foreground detection is the presence of shadows, Fig. 2.1 shows foreground object with shadow and shadow removed. Shadows generally accompany the moving object as do not have a fixed gradient, shape or size.

My thesis is focused on the problem of removing shadows from foreground object with focus on complex lighting conditions of indoor environments. Following are the objectives of my thesis,

(1) Remove shadows from real time videos.

(2) The assumed lighting condition would be complex, i.e unpredictable number of light sources, i.e a typical indoor environment.

(3) Presence of static objects with shadows, some of which may combine with a moving cast shadow to create more complex patterns.

(4) Low computational complexity.

## 2.2. Related Work

Shadow removal is a challenging problem in both still images [**2**, **12**] and videos. Although methods that deal with still image can be applied to video frames, their performance degrade and the computational complexity is usually too high for practical applications [**6**].

To remove shadows from videos, various color models have been explored to characterize their dynamic changes. Cucchiara et al. [**9**] proposed an HSV color space model for shadow removal from videos. The idea is that shadow changes the hue and the saturation components in a certain range while reduces the brightness. The thresholds are derived from the average image luminance and gradient. Gallego and Pardas [**10**] implemented a Bayesian method using brightness and color distortion model for shadow removal.

Amato et al.[**1**] developed a method that employs local color constancy. The values of the background image are divided by the values of the current frame in the RGB space. The method assumes that in the luminance ratio space, a low gradient constancy is present in all shadowed regions due to a local color constancy. A chroma difference model in RGB space was also developed in [**5**].

A 3D cone-shaped illumination model was proposed in [**13**] for background subtraction with shadow removal in indoor surveillance. The work explores the challenges of illumination

changes in indoor environments. Nghiem et al. [**19**] employs chromaticity consistency, texture consistency and range of shadow intensity to remove shadows. However, the the sensitivity and efficiency are in question [**21**]. Homogeneity and texture are also employed in shadow detection and removal.

Asaril et. al. [**4**] developed a shadow removal method based on the homogeneity property of the shadow. Thresholding and boundary removal are used for removing shadows followed by a validation step that checks the percentage of area that has been removed.

Bian et al. [**7**] implemented a method that uses texture autocorrelation to extract the shadow of a vehicle. Later statistical discrimination is used to analyze the extracted portions. Error correction is performed using integer wavelength transform.

Lu et al. [**16**] proposed a shadow removal method based on the direction of shadows using patch based comparison on geometrical properties. The algorithm assumes that the shadow will start at the edge of the object. This is true if the whole object is visible from the camera, otherwise the chances of a disjoint shadow arises. Disjoint shadows are shadows which are not connected spatially to the body.

Jung [**15**] proposes a background subtraction technique coupled with geometrical constraints to detect and remove shadows. A statistical model consisting of rations of neighboring pixel values is used to detect and remove shadows. Morphological post processing is used to eliminate pixels which have been wrongly classified.

Learning-based approaches have been developed to model and remove shadows. Wang et al. [**22**] proposed a dynamic conditional random field model for shadow segmentation in indoor video scenes that uses intensity and gradient features. Temporal and spatial dependencies are unified by the conditional random field. An approximate filtering algorithm is derived to recursively estimate the segmentation field from the observed images.

Martel-Brisson and Zaccarin [**17**] proposed a Gaussian mixture model learning algorithm for detecting shadows. Physical properties of light sources and surfaces are employed

in order to identify a direction in RGB space at which background surface values under cast shadows are found. However, the method is affected by the training phase and the computational complexity results in a long learning time.

Joshi and Papanikolopoulos [**14**] proposed a dynamically adapting algorithm that applies co-training to create a classifier with a small number of manually labeled data. Semi-supervised learning helps in adapting to new environments. Intensity, color and edge features are used to train a support vector machine for shadow removal. Qin et al. [**20**] employed a clustering method to remove shadows. However, complex indoor lighting conditions have not been discussed at length.

Patch-based strong shadow removal is performed by first classifying edges as shadow edges and non-shadow edges in [**23**]. This algorithm tries to detect strong shadow edges to classify a shadow edge classifier, followed by spatial patch smoothing.

Chen, Aggarwal, et al. [**8**] propose a method to replace shadow regions with unshaded background pixels. Spatial constraints are used to improve the shadow detection results. Characteristics of shadow are represented by various descriptors, which in turn help to resolve the run time classification. Assumption here is that human and shadow region are connected components.

Barnich et al. [**5**] proposes RGB and chroma difference to fix the moving object shadow problem. [**18**] suggests that the texture of the shadow is unchanged when compared to the background. This is used to estimate shadow effected areas.

### 2.3. ViBe

ViBe is an adaptive technique to segment foreground objects. It used a model for the background which updates as the frames of the video move on. The update process is unique as it does not replace the oldest value but does it randomly. We choose ViBe as our background subtraction algorithm as it is efficient and accurate. This section describes in brief the working of ViBe.

ViBe starts by defining a model which contains samples for every pixel in the background, in experiments the sample size $N$ is taken as 20. As frames are processed in a loop, each pixel of the new frame is compared with the model on the basis of a radius $R$. If the pixel value falls below the defined radius, the pixel is classified as a shadow pixel.

Classified shadow pixels and randomly chosen neighboring pixels are then updated in the defined model. The update probability is 1/16.

Following are some of the key advantages of using ViBe as a background subtraction algorithm,

(1) Fast computation speed, downscaled version can go upto 350 frames/sec on a native implementation [**5**].

(2) Better resilience to camera motions by a sub-sampled update process on neighboring pixels.

(3) Faster ghost suppression caused due to lighting changes or removal or static objects.

(4) Resilience to noise.

### 2.4. k-Nearest Neighbors

k-nearest neighbors algorithm is a machine learning algorithm which is non-parametric, instance based learning. It is generally used for both regression and classification. The input is a set of feature vectors and associated class labels. The output is decided by a majority vote of the $k$ neighbors. For instance if the value of $k$ is 2, then the nearest 2 neighbors are considered. Another user defined input in the algorithm is the distance metric. MATLAB 2015a defines the following distance metrics,

(1) 'euclidean'

(2) 'seuclidean'

(3) 'cityblock'

(4) 'chebychev'

(5) 'minkowski'

(6) 'mahalanobis'

(7) 'cosine'

(8) 'correlation'

(9) 'spearman'

(10) 'hamming'

(11) 'jaccard'

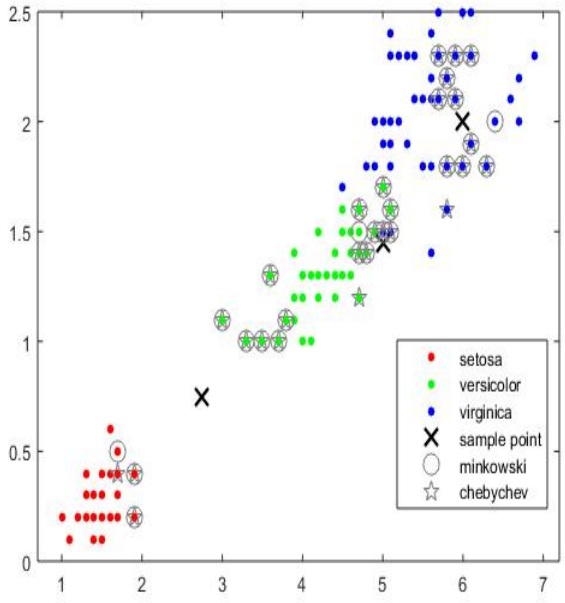



Figure 2.2. kNN plot of fisheriris dataset with minkowski and chebychev distance. k is kept as 11.

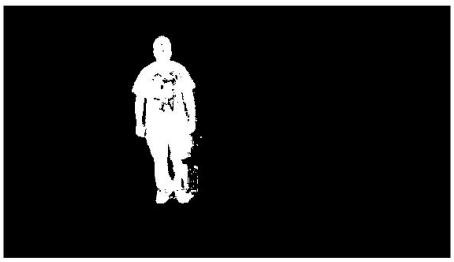


Figure 2.3. Sample frame with k=11 and chebychev distance.

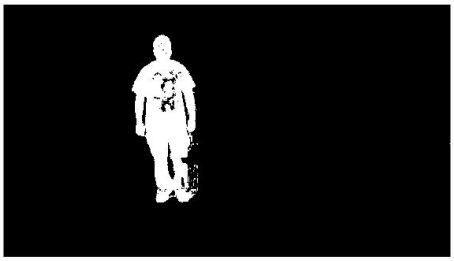


Figure 2.4. Sample frame with k=11 and minkowski distance.

The choice of $k$ and the distance affects the output of a particular application significantly, Fig. 2.2 shows the difference between minkowski and chebychev. This is due to the fact that as the number of neighbors increase the system has an overhead to calculate more distances for each sample, also it is worthy to note that the computation overhead also increases with the increase in samples. The above variation in output is also evident in our experiments later, for instance Fig 2.3 and 2.4 show the difference in the foreground mask with chebychev and minkowski distance respectively. Detailed evaluation will be discussed in the chapter 5.

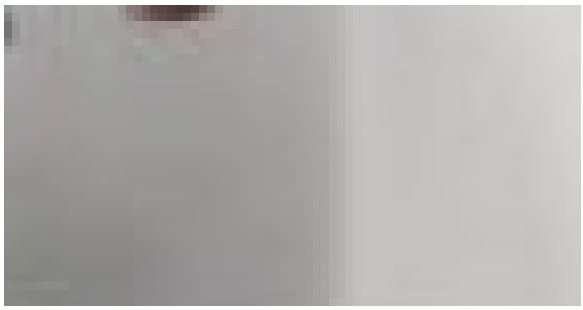
## 2.5. Entropy and Shadows

This section gives a brief overview of entropy calculation and how it relates to the shadow portions. Entropy is a measure of randomness that characterizes the texture of a grayscale image. If an image has a single plane color then the entropy is zero, as the texture of the image increases with more variations the entropy value increases.

We use entropy as one of the features in our transfer learning process. Entropy defines the randomness of a grayscale image. For instance if in a grayscale image all gray values are same, the entropy is zero. In the current implementation entropy has a dual purpose when shadows are concerned.

(1) Entropy of a shadow portion if not zero, lets us to believe that the shadow in question has different gradients to it, and can be assumed to a complex shadow pattern. We

(a)



(b)

Figure 2.5. Clip of complex shadow from a candidate frame. a) is the original frame with the red box for the clipped region and b) is the zoomed in clipped region.

use this to prove and shadow that the shadows in our test videos are indeed complex with varying gradients. Fig. 2.5 shows a clip of a complex shadow region which resulted in an Entropy value of 4.62 proving that the value set represents a changing gradient.

(2) Entropy of a shadowed region can also be used to differentiate between a shadow and a human object.

# CHAPTER 3

## SHADOWS AND ITS PROPERTIES

Before segmenting shadows from video sequences, it is important to understand how and when shadow is casted and what properties can be exploited in order to remove shadows from video sequences. Shadows exhibit properties both with respect to the light source and the object which is casting the shadow. The hue and intensity of a region casted by shadow may vary, largely due to the type/number of light sources and the surface the shadow is casted on.

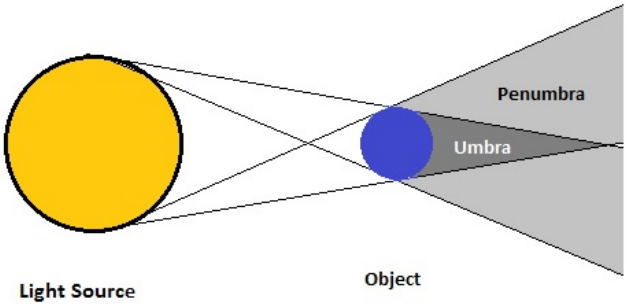



Figure 3.1. Shadow gradients in the form of umbra and penumbra.

Shadows can be broadly classified based on the light source and physical properties it has.

(1) Light source - Light plays a very important role on how a shadow behaves in an environment, Fig. 3.1 shows in general how dark and light regions of shadows are created. If the source is single and fixed we get a predictable pattern of shadow when an object moves into the camera view. Whereas if a shadow is casted by a mixture of different light sources, such as a typical indoor environment, more complex shadows are formed in such cases. Light sources can be,

(a) Static - Light source does not move rapidly, sun can be considered a static source in computer vision applications as the change is not rapid, if weather conditions are stable. Typical static environment is an indoor room with constant light.

(b) Dynamic - Light sources move and cause rapid changes in shadow gradient. If we have a cloudy day and the suns intensity is constantly changing, or an indoor environment where a light source malfunctions and switches on and off.

Light sources can also be,

(a) Single - The source of the light is singular, i.e like the sun or a room with one light source.

(b) Multiple - There are multiple light sources in a complex arrangement. Examples are any common indoor environment like a hallway with multiple light sources.

(2) Physical properties - According to [**1**] shadows can be classified according to their physical properties,

(a) Shadow types -

(i) Umbra - Darkest part directly affected by the object and the light source.

(ii) Penumbra - Mixture of umbra with addition to some light. Has lighter saturation than umbra.

(iii) Overlapping - Overlapping of the above two, commonly caused by ambient light.

(b) Spatial property

(i) Connected - Shadow is spatially connected to the object in question.

(ii) Disconnected - Shadow is not spatially connected the object.

### 3.1. Shadow Properties

Amato et al. classifies the shadow removal algorithms by the following properties that are observed,

(1) If a shadow is casted on pixel $p_{x,y}$ , the intensity of $p_{x,y}$ will decrease by a factor which is proportional to the lighting conditions. On the other hand theoretically the Hue of $p(x, y)$ should not change. This is a very important property which will affect how we come up with a solution to shadow removal.

(2) A region of the background is classified as a shadow region, if the texture of the region is similar to that of the background.

(3) If we have prior knowledge of the background and lighting conditions we can deduce shadows by combining color, shape and size of casted shadows.

## 3.2. HSV Color Space

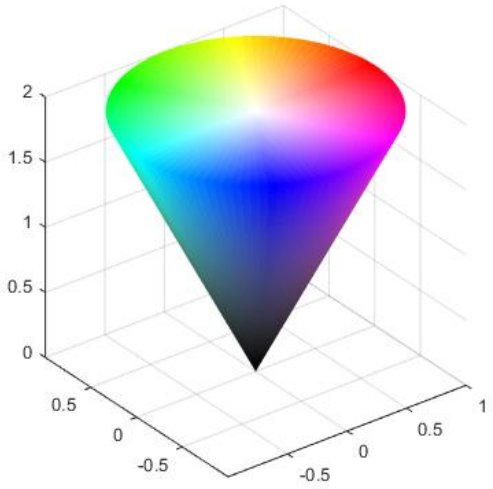



Figure 3.2. HSV representation created in MATLAB.

HSV (hue- saturation-value) is a popular model used in computer graphics. It is a cylindrical representation of the RGB color space, Fig. 3.2 is a generated representation of the model.

(1) Hue is represented by the angle around the central axis.

(2) Saturation is the distance from the axis.

(3) Value is the magnitude of the central axis.

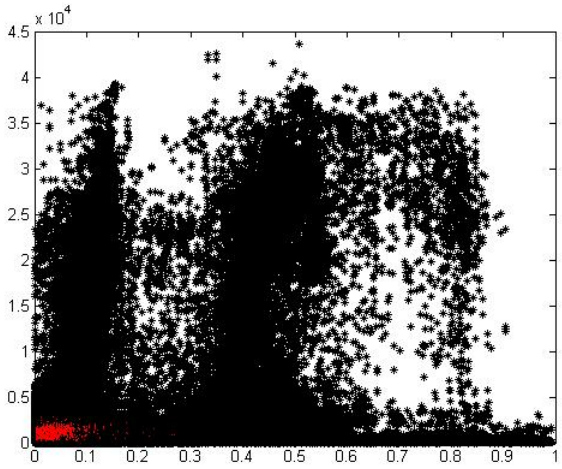



Figure 3.3. Shadow map for light shadows.

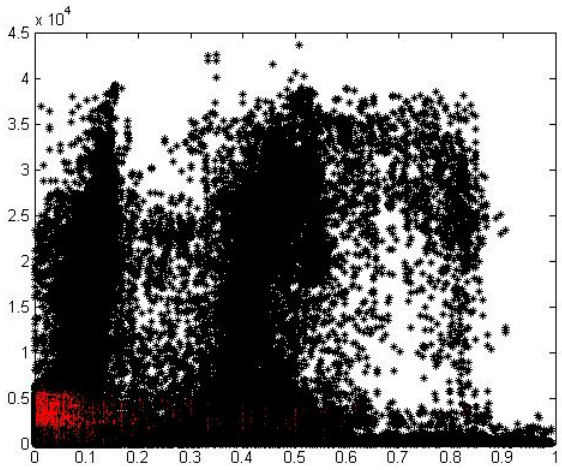



Figure 3.4. Shadow map for dark shadows.

Shadow regions are generally represented by minimal change in hue and value. For shadows the hue should theoretically be zero, but in practice we get a small change for light shadows and a significant change for dark shadows. Value on the other hand has a property of decreasing the intensity of the region, this decrease is also small for light shadows. Saturation on the other hand does not provide a definitive differentiation between the object and shadow.

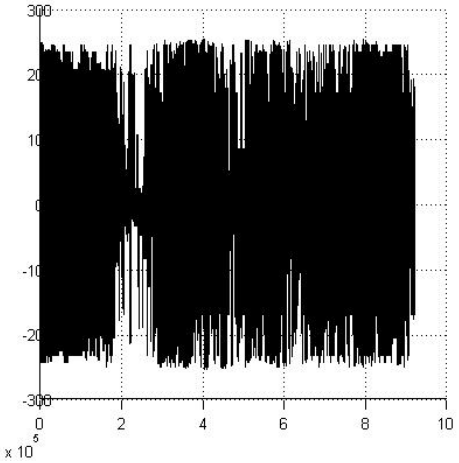



Figure 3.5. Hue difference in consecutive frames.

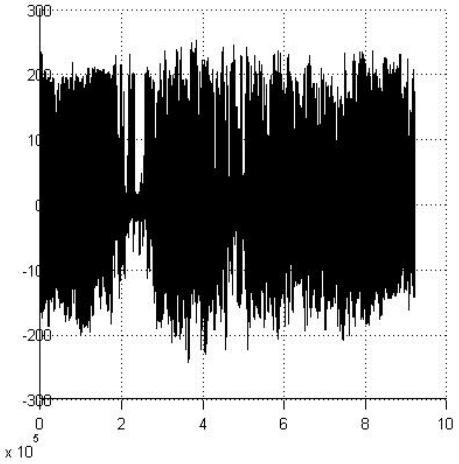



Figure 3.6. Hue difference from averaged model.

## 3.3. Variance in Hue and Intensity Difference

Hue and intensity values are critical in predicting if a particular pixel or region in a specified image/frame is shadow. Theoretically if there is no apparent change in the foreground, the hue and intensity values should not change, however in practice this is not the case.

Fig. 3.5 shows the difference between the hue values among two consecutive frames with no foreground object or light change. Such changes often, are caused by the hard-

ware which captures the frame, i.e. different cameras can result in different changes in hue difference.

One approach to standardize such errors is to make the appropriate comparison with an averaged model instead of a single frame. Fig. 3.6 show the difference between hue values of a frame and an averaged hue model of the background, which in this case contains 20 samples for each pixel in the frame.

# CHAPTER 4

# METHODOLOGY

Shadow is caused by the object that blocks a source of light. Depending on the position of imaging device, light source and the object in question, shadow appears in different shapes, which is complicated when multiple light sources are present.

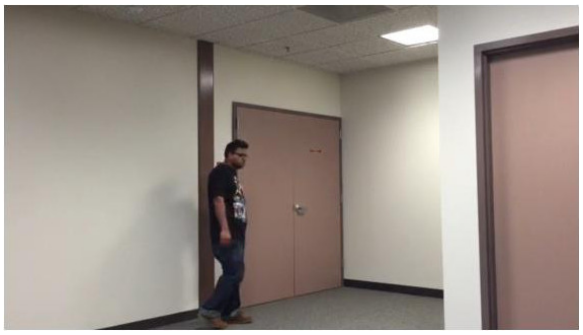
Fig.4.1 illustrates an exemplar frame of an indoor human tracking scenario. The original frame is shown in Fig.4.1(a) and the segmented result is shown in Fig.4.1(b). In this result image, shadows are separated into two kinds: light shadow and dark shadow. In this figure, the human silhouette is depicted in white, the dark shadow is in green, and the light shadow is in blue.

Light shadow usually occurs when the human subject is at a distance to the background wall or there exists other light source to brighten that part of the shadow. Dark shadow, on the other hand, occurs with total (or near total) obstruction of light. The great variation in the shadow intensity makes it difficult to differentiate from background and foreground object.

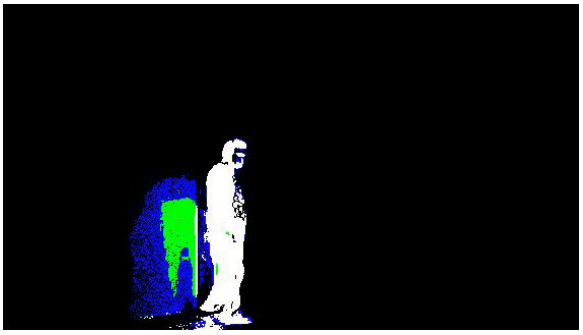
Since light shadows alter the background by slightly dimming its brightness, the hue of the affected pixels has little changes, whereas their intensity decreases proportional to the lighting conditions [**3**]. Hence, given the background model $B$, the light shadow is defined in the HSV color space as follows:

$$S_l = \{I(x,y) \quad | \quad |H(I(x,y)) - H(B(x,y))| < \tau_h \text{ and}$$
$$V(B(x,y)) - V(I(x,y)) < \tau_i\} \tag{1}$$

where $I(x,y)$ denotes an image pixel, $H(\cdot)$ denotes the hue component in the HSV space, and $V(\cdot)$ denotes the intensity component in the HSV space. $\tau_h$ and $\tau_i$are the margins for the hue and intensity differences.



(a)



(b)

Figure 4.1. An example of shadows in an indoor scenario.

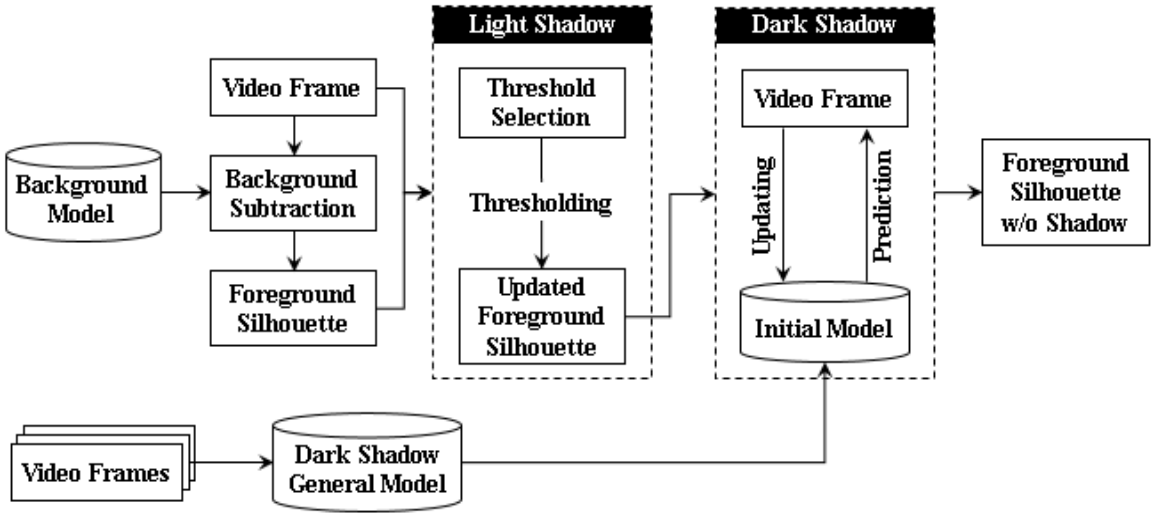



Figure 4.2. Dynamic thresholding and transfer learning-based shadow removal method.

Dark shadows greatly alter the background color, which impact the hue, intensity, and saturation of the affected pixel. We can define the dark shadow following a similar form as

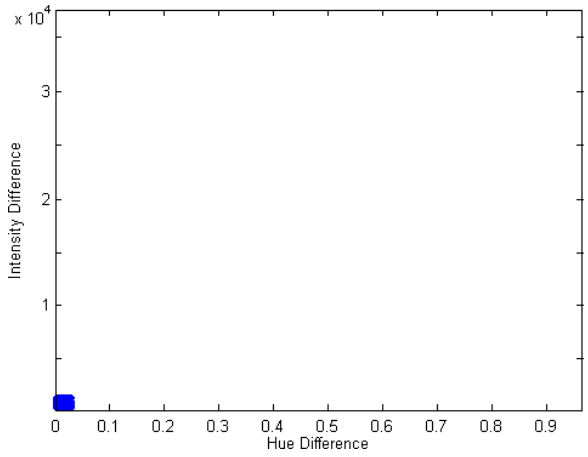



Figure 4.3. Distribution of light shadow pixels

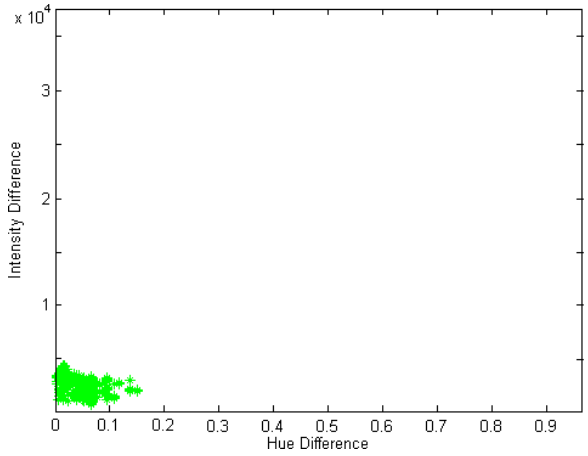



Figure 4.4. Distribution of dark shadow pixels.

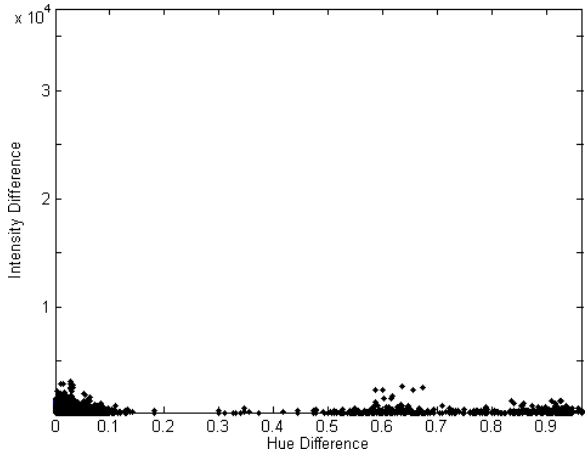



Figure 4.5. Distribution of background pixels

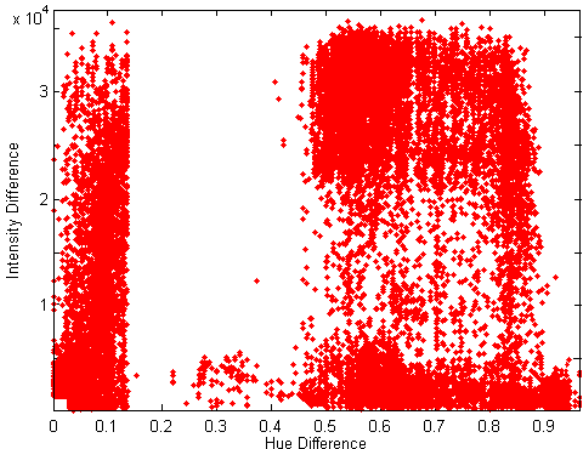



Figure 4.6. Distribution of foreground object pixels.

Eq. (1); yet this definition is also applicable to the true foreground object in dark colors. That is, the color difference is an unreliable feature to differentiate dark shadow from the rest of the video frame. Instead, the color range and texture provide fundamental features but need to be aligned with the video context.

This thesis proposes a dynamic thresholding and transfer learning (DyTaTL) method that deals with light and dark shadows differently based on the aforementioned properties. The framework of the method is shown in Fig. 4.2. A video frame is processed with background subtraction and results in a foreground silhouette that encloses the moving human subject and possibly a variety of shadows in different shade.

Based on the color variation of the foreground object, thresholds are dynamically decided to remove the light shadows. Using annotated video frames as training examples, a classifier is developed as the initial model for the dark shadows in the video under processing. Using the spatial correlation of image pixels, the mostly likely neighboring pixels are recruited as training examples to update the classifier.

## 4.1. Light Shadow Modeling and Dynamic Thresholding

As defined in Eq. (1), light shadow changes only the intensity of a pixel. Yet, due to noise, the hue (i.e., color) of a pixel varies slightly over time. Hence, Eq. (1) is relaxed to

model such variation with a margin $\tau_h$ for the hue difference as follows:

$$S_I = \{I(u,v) \quad | \quad V(B(u,v)) - V(I(u,v)) < \tau_i \quad \text{s.t. } |H(B(u,v)) - H(I(u,v))| < \tau_h\} \tag{2}$$

In this model, the intensity of a shadow-affected pixel decreases, and the difference is within $\tau_i$, subject to the maximum hue change of $\tau_h$ induced by noise. Hence, $\tau_h$ can be estimated by computing the average hue difference of pixels in the background:

$$\tau_h = \frac{1}{QT} \sum_{I(u,v)} \sum_{t} |H(I^t(u,v)) - H(B(u,v))| \tag{3}$$

where $I(u,v) \in I_B$, and $I_B$ denotes the set of background pixels in frames $I^p$ through $I^{p+T}$. $Q$ denotes the total number of background pixels. The superscript $t$ denotes the frame index and is in the range of $\{p, p+1, \ldots, p+T\}$. To compute $\tau_h$ for a video, the candidate pixels are determined with background subtraction. Only those pixels that are in the background in the temporal range $T$ are used.

Alternatively, if there are many initial frames that contain only stationary objects (i.e., background), the entire frame can be used in the estimation of $\tau_h$.

Following the above idea, the intensity difference of the background pixels is estimated as follows:

$$\tau_i = \frac{1}{QT} \sum_{I(u,v)} \sum_{t} (V(B(u,v)) - V(I^t(u,v))) \tag{4}$$

Since shadow always reduces the brightness, it is plausible to assume $V(B(u,v)) - V(I(u,v)) > 0$. This intensity difference accounts for the variations induced by the imaging factors such as noise, quantization error, etc. (as shown in Fig. 4.5) It is also representative

to the changes made by light shadow. Note that Fig. 4.5 depicts a broad range of hue difference for the background pixels.

Given that the background is stationary, it is expected that both the intensity and hue differences are fairly small. The existence of large hue difference is caused by noise and quantization error. For a typical video frame with 169,016 background pixels, the number of pixels with hue difference greater than 0.2 is 2,798, greater than 0.5 is 2,765, and greater than 0.9 is 1,837. It is clear that the percentage of pixels with large hue difference is very low (in the range of 0.01%).

## 4.2. Transfer Learning for Dark Shadow Removal

Dark shadows are portions in a moving cast shadow that show a greater increase in Hue difference than light shadows. Removing these with a second layer of thresholding will impact the foreground object with many misclassifications.

We propose a second filtering of shadows with the help of a classifier. A pre trained classifier is used to determine if a given pixel is a dark shadow or otherwise. This however is not true for light shadows as training different gradients of light shadows might lead to lower precision during classification.

The classifier also adapts to different indoor conditions by having an initial on-line training to learn the new environment, it should be noted here that this training is only required when the indoor scene changes completely. Hue, Intensity, RGB and Entropy are used as features for the training phase.

(1) Hue and Intensity Hue and Intensity values for each pixel $p_{x,y}$ which is a shadow is use. As absolute values of both may cause misclassifications on the object itself (if the object has dark portions) we use the absolute difference in Hue and Intensity. This difference is computed with respect to the background model.

(2) RGB Absolute RGB values are used to increase the accuracy of the classifier, particularly in cases where Hue and Intensity difference on the object are similar to

the shadow pixels.

(3) Entropy Entropy is used to differentiate objects which have similar RGB values to shadows. Entropy of a shadow pixel $p_{x,y}$ is always lower that the object body. Also, this feature is more accurate because we do not have significant change in intensity on the shadow portions as light shadows have been removed through thresholding.

In contrast to light shadow, dark shadow introduces much brightness and hue changes, which makes it difficult to be separated from the foreground object using thresholding method (as shown in Fig. 4.4 and 4.6). By increasing the threshold for intensity and raising the tolerance factor for hue variation, erroneous removals of foreground object is likely to happen. To address this issue, supervised learning methods have been used [**14**, **17**].

Many machine learning methods work well under an assumption that the training and testing data are drawn from the same distribution. When this distribution changes, the existing models need to be rebuilt from scratch, which is expensive and inefficient.

The open challenge is the efficiency and adaptivity, that is, to be able to quickly process videos that are in different lighting conditions from the training examples.

To remove dark shadows, the idea of transfer learning is adopted which is based on k-Nearest Neighbor (kNN) classifier. The learning method adopted is a general model $H$ for dark shadow is first developed using manually segmented video frames. Let $X = \{x_1, x_2, \ldots, x_N\}$ denote a set of training examples, where each $x_i$ is a feature vector of a pixel, and $Y$ denotes the class label $\{0, 1\}$ with 1 denotes dark shadow pixel and 0 denotes non-dark shadow pixel. Hence, we have

$$H : X \to Y.$$

This model $H$ is used as the base classifier for videos. For each instance in $X$, a set of features are extracted from the video frame as follows:

*Intensity and hue difference ($d_i$ and $d_h$)*

Different from background noise and light shadow, dark shadow introduces much greater

changes to the intensity and hue of a pixel. In particular, the brightness of a shadowed area is reduced. These differences are computed with respect to the average intensity and hue of the background model:

$$d_i(u, v) = \bar{H}(B(u, v)) - H(I(u, v)),$$

$$d_h(u, v) = \bar{V}(B(u, v)) - V(I(u, v)).$$

*Pixel color in RGB space (r, g, and b)*

Comparing to the intensity and hue difference, RGB color gives an approximate range of the shadow, which complement the difference feature.

*Local entropy (e)*

Local entropy is used to differentiate the foreground object that might have similar color to the shadows:

$$e(u, v) = -\sum_i p_i log p_i,$$

where $p_i$ is the probability of a color in a M by M window. Due to the greater homogeneity of the shadow region, its entropy is lower than that of the object.

Hence, each instance $x_i \in X$ consists of the above six components: $\{d_i, d_h, r, g, b, e\}$. Note that each feature component is normalized by its respective dynamic ranges to avoid learning bias induced by magnitude. Also, to overcome high storage requirements and low efficiency in kNN algorithm, we adopt the reduced nearest neighbor method [**11**] to keep the model concise.

When a new video is processed, $H$ is applied to identify dark shadow pixels in the video frames, and the neighboring pixels of the most confident shadow are recruited as training examples to update this model, which make $H$ fine-tuned to the variations of the new video such as brightness and tone changes. Our assumption is that the close neighboring pixels of a dark shadow pixel is most likely to be a dark shadow pixel as well. In addition, any new example must satisfy the minimum intensity and hue difference as defined in Eq. (4) and

(3). Hence, the new examples $I(u, v)$ must satisfy the following criteria to be selected for updating the base classifier $H$ :

$$\begin{cases} D(I(u,v), \hat{I}(u,v)) \leq \tau_d \\ V(B(u,v)) - V(I(u,v)) > \tau_i \\ H(B(u,v)) - H(I(u,v)) > \tau_h \end{cases}, \tag{5}$$

where $\hat{I}(u, v)$ is a dark pixel with high confidence and $\tau_d$ is the neighbor distance.

Another issue is when to update classifier $H$ with pixels from the new video. Shadows are not necessarily present as the video starts. In DyTaTL, the algorithm starts re-training process when there are sufficient number of dark shadow pixels identified in a video frame, and the training process continues until there is very few pixels satisfies the criteria in Eq. (5).

### 4.3. Algorithm

Algorithm 1 presents the DyTaTL learning-based dark shadow removal method. In this, $\beta$ is a boolean indicator that controls if re-training of $H$ is performed. It is initialized to 1 and set to 0 when there is no new examples identified in the current video frame classification. $\sum_i y_i$ gives the total number of dark shadow pixels and $\epsilon_s$ is the minimum number of dark shadow pixel to perform classifier update. Set $S$ holds the new training examples and is initialized with an empty set. Given a dark pixel, an instance is added to $S$ when it satisfies Eq. (5). Function $C(\cdot)$ gives the confidence of the prediction of an instance, and the minimum confidence of a dark pixel to serve as a start point of finding new examples is $\epsilon_c$.

A $(h, w, N)$ dimensional model is maintained for the hue values of the background, where $h$ and $w$ are the height and width of each frame in the video sequence, N is the number of samples to be stored in the model, we choose N=20 which is the same for the background model described in [**5**].

First each pixel in the mask that is classified as a foreground, is checked for its Hue difference with the hue model, this value along with the intensity difference of that pixel is

**Algorithm 1** Transfer learning-based dark shadow removal.

**Input**: video $V$ and the base classifier $H$

**Output**: an image map $\hat{I}$ of dark shadow

$\beta \leftarrow 1$

**for** $t = \{1, 2, \ldots, T\}$ **do**

  Extract features from image $I^t \in V$: $I^t \rightarrow X$,

  where $X = \{x_1, x_2, \ldots, x_N\}$ and $x_i = \{d_i, d_h, r, g, b, e\}$

  Apply $H$: $X \rightarrow Y$

  **if** $\sum_i y_i > \epsilon_s$ and $\beta = 1$ **then**

    $S \leftarrow \emptyset$

    **for all** $x_i$: $H(x_i) = 1$ and $C(x_i) > \epsilon_c$ **do**

      Find neighboring pixel $x_j$ within distance $\tau_d$

      **if** $x_j$ satisfies Eq. (5) and $H(x_j) \neq 1$ **then**

        $S \leftarrow S \cup x_j$

      **end if**

    **end for**

    **if** $|S| > 0$ **then**

      Update $H$ with examples in $S$

    **else**

      $\beta \leftarrow 0$

    **end if**

  **end if**

**end for**

filtered through thresholding. $\tau_{Hue}$ and $\tau_{Intensity}$ are the two dynamic thresholds that are used. This step removes the light shadows which had been classified as foreground.

For removing dark shadows, a kNN based classifier is chosen which uses training data

from a varied set of indoor shadow patterns. In the initialization phase the transfer training module collects specific samples to add training data to the existing classifier. These specific samples of pixels are relevant and specific to the indoor environment in question.

The process starts by first identifying dark shadows with the current training data, for each pixel $p_{x,y}$ which is classified as shadow by the existing model all neighboring pixels which are not classified as shadows are considered as a part of specific samples. After the specific samples are collected over the period of $f_n$ frames, the classifier is retrained and transfer learning stops.

The transfer learning in the initialization phase starts when the pre trained classifier gives at least $\tau_s$ pixels as shadow and the object in question has $\tau o$ pixels at a minimum. After such a detection is reached the on-line training will continue for $f_n$ frames.

# CHAPTER 5

# EXPERIMENTAL RESULTS

## 5.1. Experimental Data and Settings

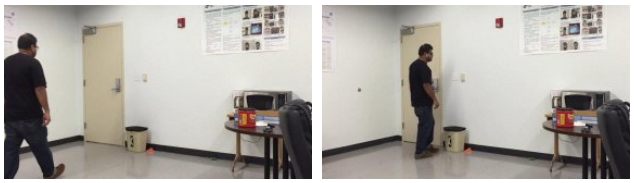
To evaluate the method, 6 indoor videos in rooms and corridors were acquired using two cameras (camera on an iPhone 6 and camera on an ASUS laptop) with different lighting conditions. Table 5.1 lists the properties of videos used in experiments, and exemplar frames are depicted in Fig. 5.1.

Table 5.1. Videos acquired for experimental evaluations.

| Videos | Color Depth | Frame Rate | Resolution | Lighting Condition |
|---|---|---|---|---|
| A | 24 | 30 | 320 × 568 | Bright |
| B | 24 | 30 | 320 × 568 | Moderate |
| C | 24 | 30 | 320 × 568 | Dim |
| D | 24 | 30 | 320 × 568 | Bright |
| E | 24 | 30 | 720 × 1280 | Moderate |
| F | 24 | 30 | 720 × 1280 | Variable |

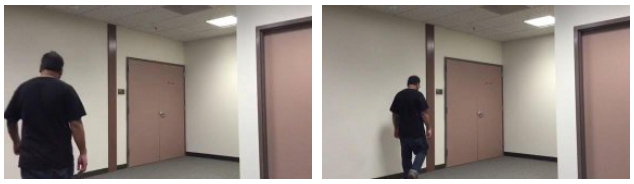
During the implementation, ViBe [**5**] is adopted as the background subtraction method for its simplicity and efficiency. However, DyTaTL method can be combined with any similar method for shadow removal from videos. In ViBe, each pixel in the background model consists of a set of values that describe the possible color range, which is updated randomly in the process of background subtraction. The size of this set is suggested to be 20 based on empirical evaluations of the efficiency and accuracy [**5**], which is adopted in the implementation.

In the experiments, a minimum number of dark shadow pixels in a video frame ($\epsilon_s$) is used to control if and when the classifier retraining starts, and this threshold is set to 300. When selecting pixels as new training examples, dark pixel confidence ($\epsilon_c$) is 90%. For a

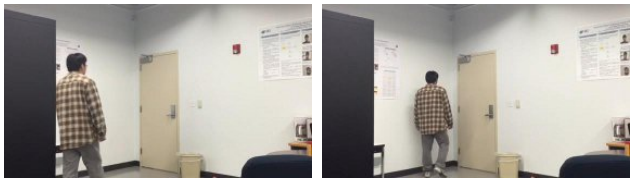


(a)

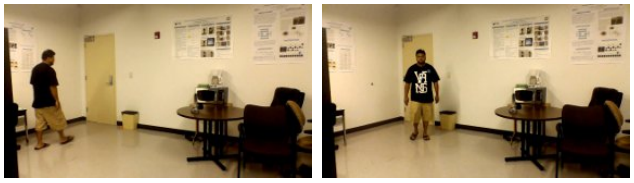
(b)



(c)



(d)



(e)

(f)

Figure 5.1. Exemplar frames from testing videos. (a)-(f) correspond to the videos listed in Table 5.1.

kNN classifier with $k = 11$, this translates to a positive majority vote of $10 : 1$ or $11 : 0$. And pixels in the 4-neighborhood, i.e., $\tau_d = 1$, of the most confident dark pixel are candidate training pixels. The distance metric of kNN classifier is Euclidean distance.

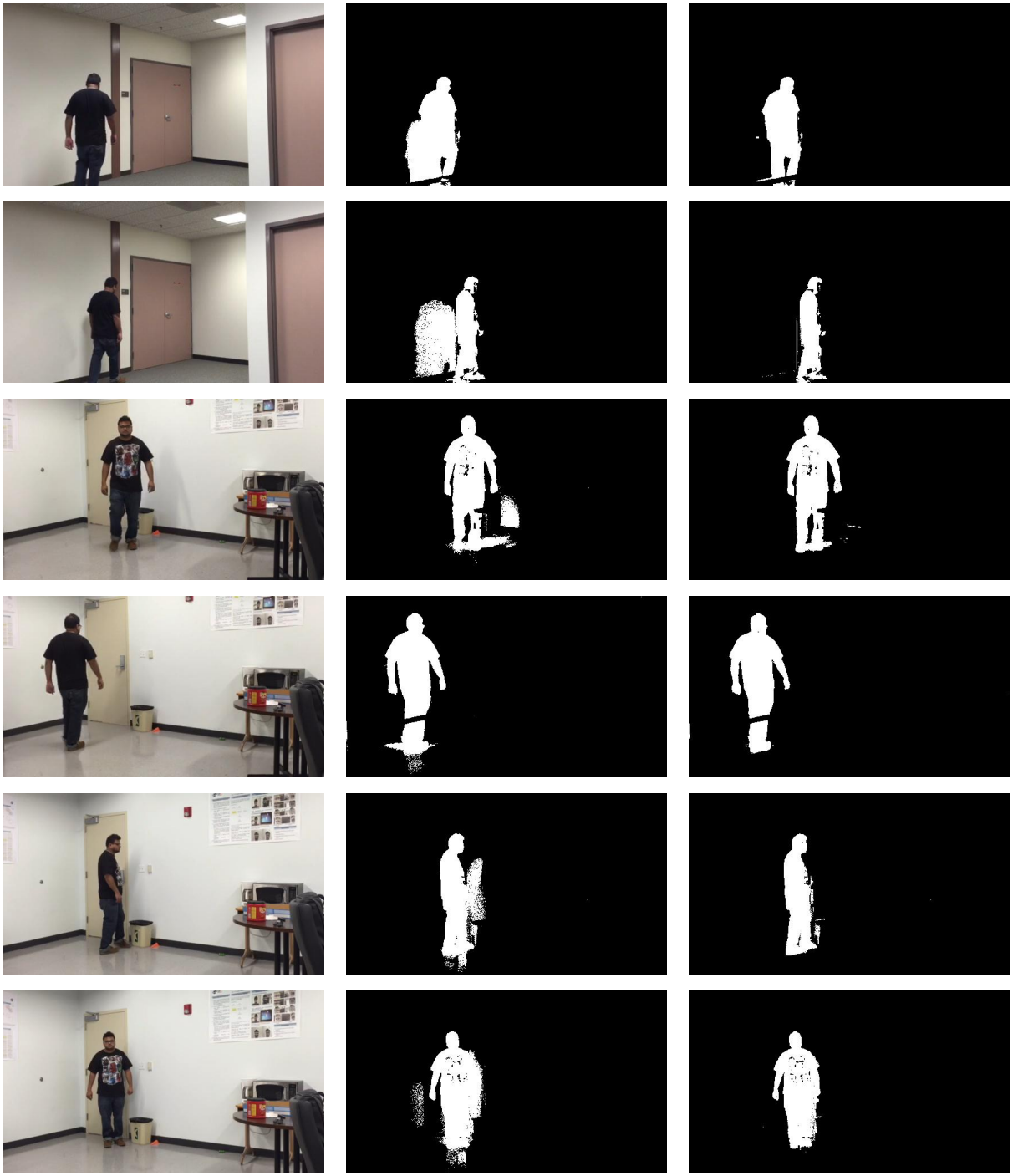
Fig. 5.2 illustrates exemplar frames of shadow removed foreground segmentation results. The left column depicts the original frames from videos; the middle column depicts the background subtraction results using ViBe method; the right column depicts the shadow removed foreground segmentation results using DyTaTL method. It is clear that when shadows are present the foreground object is greatly distorted in the background subtraction results.

The shadow caused erroneous foreground regions could be connected to or disconnected from the human silhouette as well as vary in size and shape. It is demonstrated that DyTaTL method successfully removes the shadows and introduces little distortions to the foreground object. Note that there are voids (dark pixels) inside the human silhouette or imperfect foreground segmentation in the final results, which are, however, inherited from the background subtraction outcomes. Also shown in these examples is that the lighting conditions in these video frames are clearly different and hence the brightness of shadow varies. DyTaTL method is able to adapt to the videos and identify shadows correctly.

## 5.2. Accuracy Analysis

Since evaluation of the performance of shadow removal is in question, it is needed to have reference images of shadows only. However, it is extremely challenging to delineate the shadow region in a video frame even for manual tracing. Alternatively, the ground truth was prepared with images of human silhouette. Another consideration is to exclude errors from the background subtraction process. Due to noise and similar color of human figure to the background, the output of background subtraction usually contains erroneous segmentation.

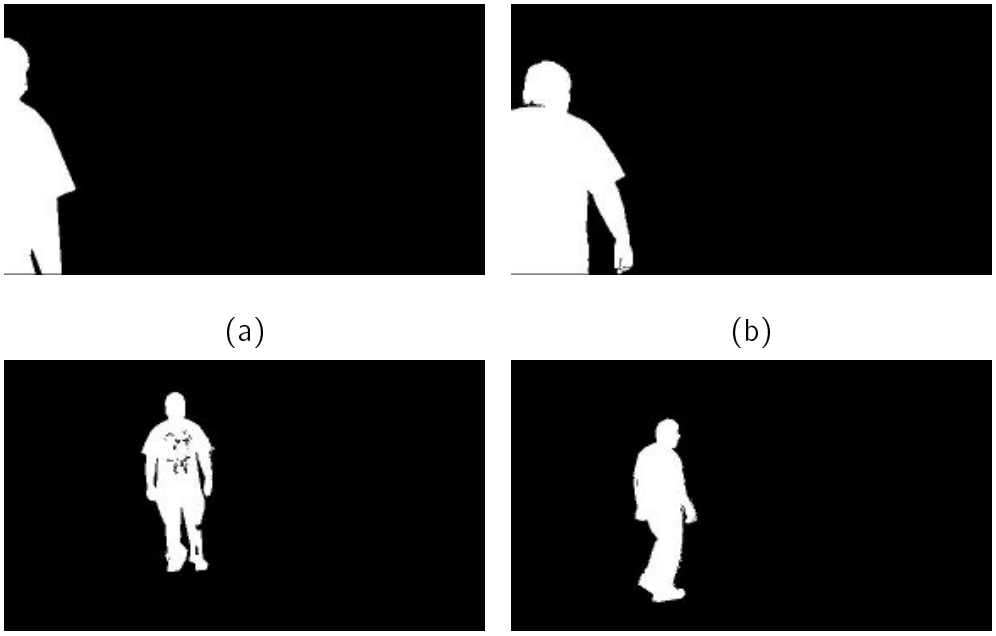
To suppress the impact of such error to the evaluation of shadow removal, ground truth is based on the output of the background subtraction procedure that excludes the shadow areas by manual tracing on the resulted foreground object. Fig. 5.3 depicts a few



(a) (b) (c)

Figure 5.2. Exemplar results. (a) are the original video frames. (b) are the background subtraction results using ViBe. (c) are the shadow removed results using DyTaTL method.

examples of ground truth of human silhouette. Fig. 5.3(c) depicts a ground truth frame that contains errors (black pixels in the upper body) from the ViBe method [**5**]. In the experiments, 60 reference images were created with manually segmented human silhouette, among which 25 contains very little shadows and 35 contains significant amount of light shadows, dark shadows, or mixture of both.



(a) (b)

(c) (d)

Figure 5.3. Ground truth of human silhouette. (a) and (b) are the ground truth images with little shadows. (c) and (d) are the ground truth images with significant amount of shadows.

A key factor for dark shadow removal is the local entropy that differentiates grayish or dark foreground object from the shadow. Table 5.2 lists the average sensitivity and specificity and the corresponding standard deviation of detecting dark shadows using three window sizes: $M = 3, 5,$ and 7. Experimental results show that the detection accuracies using these three window sizes are very close: 96.9%, 96.9%, and 96.8% for $M = 3, 5,$ and 7, respectively. In comparison, the performance of detecting dark shadows is examined in frames with a small amount of shadows and the ones with significant amount of shadows separately. This is be-

cause the much larger number of non-shadow pixels could overwhelm the overall performance when the shadow pixels are limited. The specificity for all cases is above 98% with very little variations (less than 2% standard deviation).

The sensitivity, however, varies greatly, especially for the small shadow cases. For frames with small shadow regions, the sensitivity is at 58.5% for window size of 3 by 3 and is at 48.1% for window size of 7 by 7. However, for frames with large shadow regions, the sensitivity achieves 88.3% and 87.1%, respectively. The much lower sensitivity for the small shadow case is mostly due to the small denominator in computing sensitivity, which also makes the result unstable as shown in the standard deviation. It is evidential that window size of 3 by 3 exhibits the best performance for both small and large shadows. Hence, we use $M = 3$ throughout the rest of the experiments.

Table 5.2. Average sensitivity (Sen.) and specificity (Spe.) of classifying shadows with different window sizes for entropy calculation. The values in parenthesis are the corresponding standard deviation.

| Shadow | 3 by 3 | | 5 by 5 | | 7 by 7 | |
|---|---|---|---|---|---|---|
| Region | Sen. | Spe. | Sen. | Spe. | Sen. | Spe. |
| Small | 58.5% | 98.3% | 55.2% | 98.5% | 48.1% | 98.6% |
| | (36.1) | (0.9) | (34.6) | (0.7) | (33.2) | (0.7) |
| Large | 88.3% | 98.2% | 87.6% | 98.2% | 87.1% | 98.1% |
| | (5.5) | (1.5) | (5.8) | (1.6) | (6.2) | (1.8) |

Table 5.3 lists the average sensitivity and specificity of shadow detection with different number of neighbors in kNN classifier. The results with small shadow regions exhibit much lower sensitivity with great variation compared to the cases with large shadow regions. However, the disparity of sensitivity with respect to the number of neighbors is trivial (within 2%). In addition, the average accuracy regardless of the shadow size is about 97%.

In general video frames usually contain much more non-shadow pixels and our selection of $k$ shall aim to maximize the correct detection of shadow pixels, i.e., largest sensitivity. Based on this criterion, $k = 15$ yielded the greatest overall average sensitivity of 76.5%.

Table 5.3. Average sensitivity (Sen.) and specificity (Spe.) of classifying shadows with different number of neighbors in kNN classifier. The values in parenthesis are the corresponding standard deviation.

| Shadow | k = 7 | | k = 11 | | k = 15 | |
|---|---|---|---|---|---|---|
| Region | Sen. | Spe. | Sen. | Spe. | Sen. | Spe. |
| Small | 59.3% | 98.5% | 58.5% | 98.3% | 58.4% | 98.2% |
| | (37.6) | (0.8) | (35.1) | (0.9) | (35.7) | (1) |
| Large | 87.3% | 98.4% | 88.3% | 98.2% | 88.6% | 98.1% |
| | (5.6) | (1.4) | (5.5) | (1.5) | (5.3) | (1.5) |

## 5.3. Classifier Retraining

In experiments, four frames from two videos were used and selected 4221 dark shadow pixels and the equal number of non-dark shadow pixels as the training examples to create a base kNN classifier, which is then applied to the process of the other videos. The retraining process starts only if there are significant number of dark pixels (i.e., $\epsilon_s$) identified by the base classifier (or the updated classifier). However, the retraining can be conducted in two means: continue for a certain number of frames or retain the classifier only if there are a significant number of dark shadow pixels in a frame. In the first case, the retraining ends at some specific number of iterations.

By adjusting the maximum number of frames, the training process is easily controllable. However, it could run into problems in that the subsequent frames may contain no dark shadow pixels, and hence the classifier is not fully adapted to the video. Fig. 5.4 depicts the average number of new training examples recruited to update the base classifier in the

early stage of processing a new video. The maximum number of frames used is 100. The error bar marks the standard deviation among the trainings in all videos. The average clearly shows the declining number of new examples recruited for retraining, which implies a training convergence.

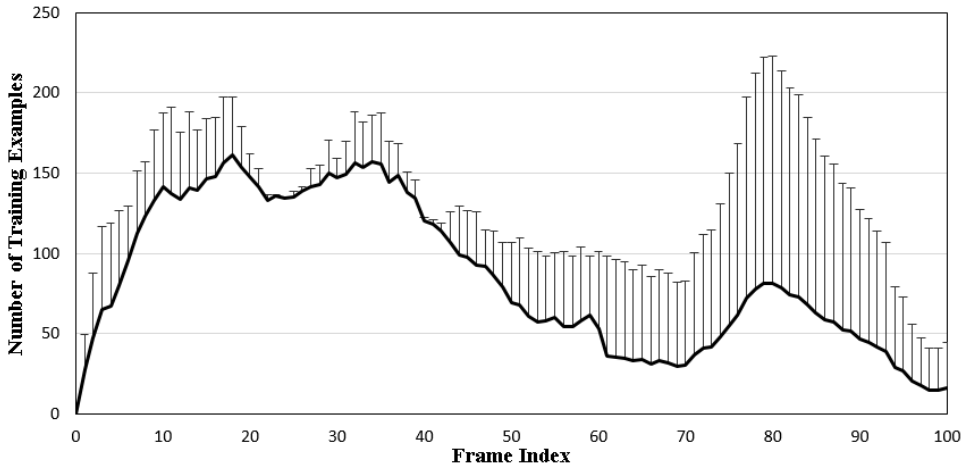
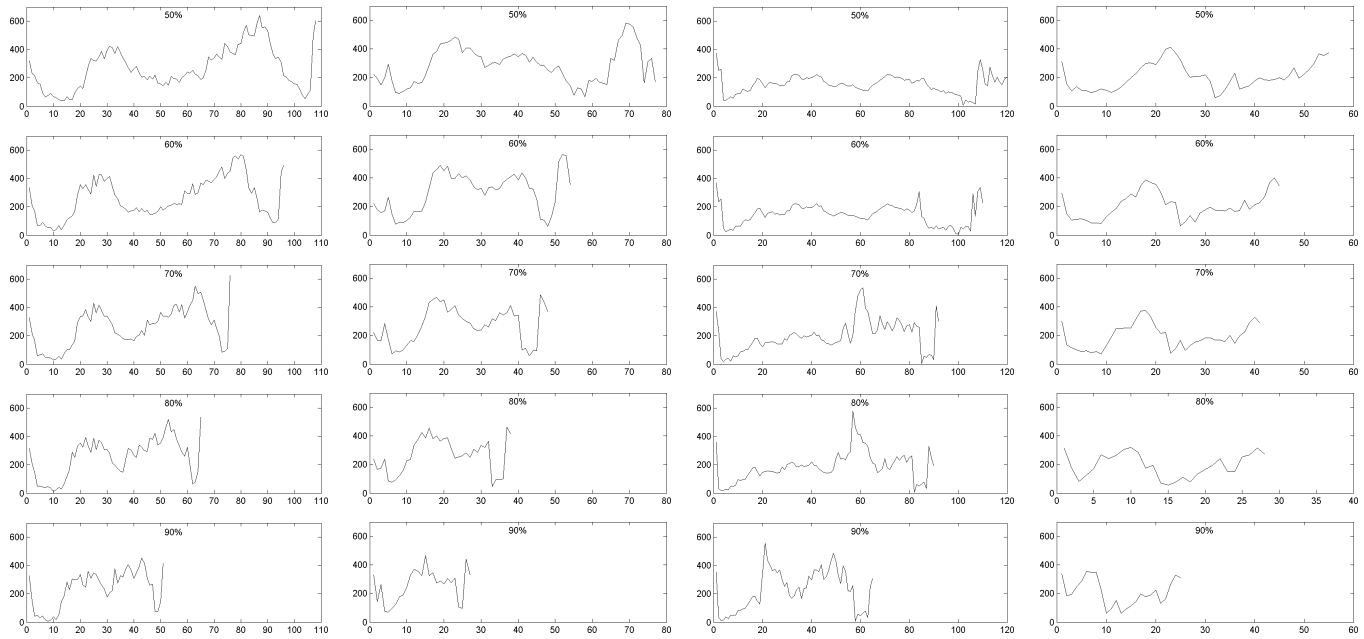



Figure 5.4. The average number of new training examples recruited to update the base classifier in the process of a new video. The error bar shows the standard deviation among testing cases.

Alternatively, the retraining process continues until there are no additional updates. In this case, all frames throughout the entire video that contain the minimum number of dark pixels will be involved and the retraining process could end any time. Fig. 5.5 illustrates the number of new examples recruited in the process of videos using confidence levels greater than 50%, 60%, 70%, 80%, and 90% from top row to bottom row, respectively. Note that the x-axis gives the training iterations instead of the continuous frame index. It is clear that when higher confidence level is used in example selection the training process completes sooner. Different from the previous strategy, the number of new examples shows no declining trend. However, the training process takes much shorter iterations especially when high confidence is used.

Table 5.4 lists the average and peak number of new examples in the retraining process.




(a) Video A (b) Video B (c) Video E (d) Video F

Figure 5.5. The number of new training examples recruited to update the base classifier throughout the entire video using different confidence.

Table 5.4. Average new examples recruited for classifier updates.

| | | Confidence Level | | | | |
|---|---|---|---|---|---|---|
| | Videos | 50% | 60% | 70% | 80% | 90% |
| Average | Video A | 269 | 263 | 264 | 256 | 237 |
| Number | Video B | 288 | 303 | 272 | 267 | 257 |
| of New | Video E | 155 | 146 | 199 | 188 | 230 |
| Examples | Video F | 209 | 206 | 194 | 186 | 210 |
| Peak | Video A | 641 | 569 | 627 | 536 | 452 |
| Number | Video B | 580 | 564 | 485 | 462 | 466 |
| of New | Video E | 374 | 370 | 539 | 579 | 561 |
| Examples | Video F | 411 | 402 | 376 | 353 | 354 |

The average number of new examples recruited changes very little across all confidence levels. Given that the retraining process with higher confidence runs shorter, the total number of new examples recruited from the video is less. The peak number of new examples, on the other hand, decreases when high confidence is used. This is mostly due to the stringent condition applied to the example selection.

The average performances of shadow removal using these confidence levels are listed in Table 5.5. The specificities in all cases are very close and are at 98.2% with small variations. The sensitivities change slightly and when the confidence level is at 100% the retrained classifier exhibits the best sensitivity at 76.6% with a standard deviation of 27.0.

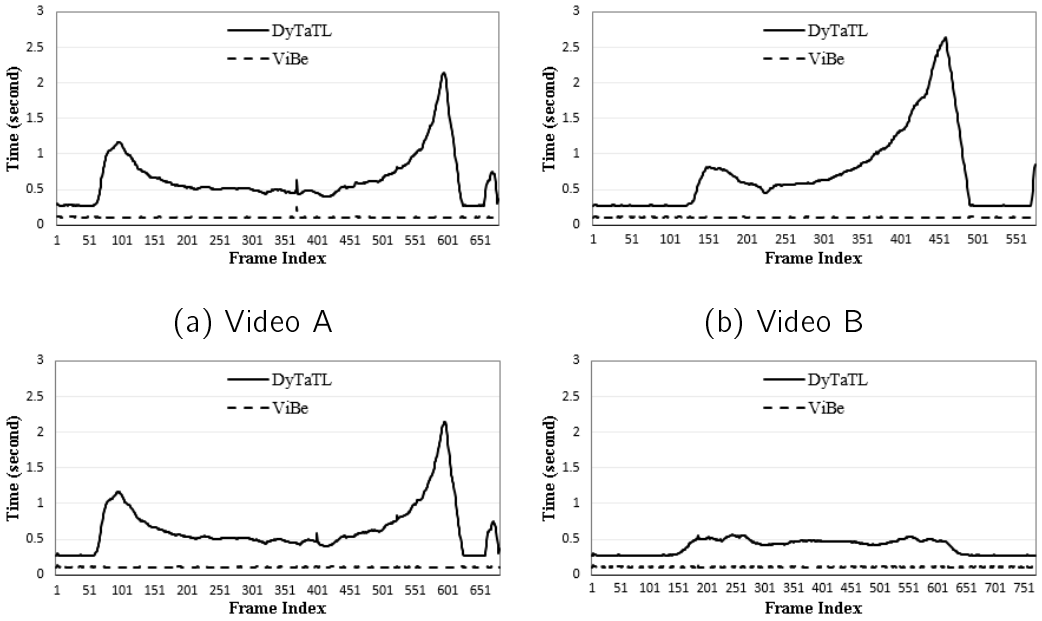
Table 5.5. Average sensitivity and specificity using different confidence levels. The values in parenthesis are the corresponding standard deviation.

| | Confidence Level | | | | | |
|---|---|---|---|---|---|---|
| | 50% | 60% | 70% | 80% | 90% | 100% |
| Sensitivity | 75.6% | 76.0% | 75.7% | 76.3% | 76.3% | 76.6% |
| | (28.3) | (27.5) | (27.5) | (27.1) | (27.4) | (27.0) |
| Specificity | 98.2% | 98.2% | 98.2% | 98.2% | 98.2% | 98.2% |
| | (1.3) | (1.3) | (1.3) | (1.3) | (1.3) | (1.3) |

## 5.4. Efficiency Analysis

DyTaTL algorithm and ViBe method are implemented with MATLAB and tested in a PC system with Intel Core i7-4770 CPU at 3.40GHz and 16GB memory. Table 5.6 lists the average time to process a frame in videos using ViBe for background subtraction and using DyTaTL for shadow removal. The average time of DyTaTL method is in the range of half a second and varies greatly between videos, whereas ViBe takes an average of 0.11 seconds.

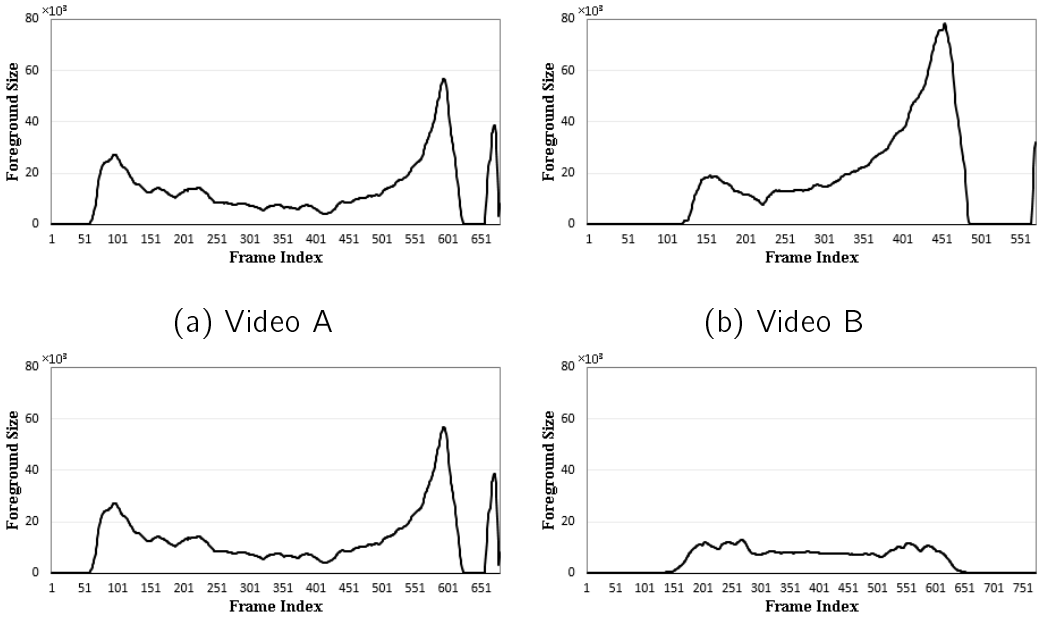
There are two major factors that affect the efficiency of DyTaTL method: foreground object size and computation of distances in kNN classifier. Fig. 5.6 illustrates the "stack




(a) Video A (b) Video B

(c) Video E (d) Video F

Figure 5.6. Time used to process each frame in the videos. The solid curve depicts the time used by DyTaTL method. The dash curve depicts the time used by ViBe method.




(a) Video A (b) Video B

(c) Video E (d) Video F

Figure 5.7. The size of foreground object appears in each frame of the videos.

Table 5.6. The average time (in second per frame) used for background subtraction and shadow removal. The standard deviation is in parenthesis.

| Videos | A | B | E | F |
|---|---|---|---|---|
| ViBe | 0.110 | 0.109 | 0.109 | 0.111 |
| | (0.006) | (0.003) | (0.002) | (0.003) |
| DyTaTL | 0.532 | 0.610 | 0.532 | 0.284 |
| | (0.349) | (0.558) | (0.348) | (0.100) |

plots"[1] of the processing time of video frames. At the start of these videos, there is no moving foreground object and the processing time of DyTaTL is about the same as that of the ViBe, and both are in the order of one tenth of a second.

In videos a, b, and e, the human subject walks into the field of view and then walk toward the camera. Hence the foreground object segmented with ViBe grows in size, which is shown in Fig. 5.7. It is clear that the trend of foreground size coincides with the change of the processing time using DyTaTL. Fig. 5.6(d) shows the average processing time of a video with a human subject walking across the field of view, that is, the size of the foreground object varies slightly, as illustrated in Fig. 5.7(d). In this case, the average time used by DyTaTL is comparable to the time used by ViBe. It is evidential that the processing time of DyTaTL method is proportional to the foreground object size.

In the process of the four videos a, b, e, and f, the re-training starts at frame 68, 132, 68, and 169, respectively. By comparing the foreground object size and the average time used, it can be concluded that updating kNN imposes little impact to the processing time of a frame. However, when making decisions with kNN, the algorithm faces much greater computational expense for calculating the distances between the foreground pixels and the examples in the kNN model and making selections of the nearest neighbors.

[1]The "stack plot" puts the time used by DyTaTL above the time used by ViBe.
So the values marked by DyTaTL represent the total time to complete processing the frames.

# CHAPTER 6

# CONCLUSION

Todays fast paced application development and research has led to the adoption of computer vision applications in the commercial market. Some of the common applications today revolve around the tracking of humans. Thus, it is getting increasingly essential to build robust and accurate systems. Pose detection is an algorithm which is gaining much popularity especially in the health care domain. Pose detection generally involves detecting and tracking human body parts individually. Majority of such applications use background subtraction as an intermediate step before any further processing takes place.

Background subtraction involves getting the human silhouette and separating all other background stationary objects. Noise may be induced in many forms during this process, one such unwanted noise in many applications is a moving cast shadow. Shadows are casted when a moving object blocks a source of light. Indoor lighting conditions complicate the way that shadows are formed with the inclusion of static objects and multiple light sources, which creates complex shadow patterns when casted and are characterized by varied gradients of hue and intensity values within a casted region.

Shadows in indoor scenarios are usually characterized with multiple light sources that produce complex shadow patterns of a single object. Without removing shadow, the foreground object tends to be erroneously segmented. The inconsistent hue and intensity of shadows make automatic removal a challenging task. In this thesis, a dynamic thresholding and transfer learning-based method is presented for removing shadows. DyTaTL method suppresses light shadows with a dynamically computed threshold and removes dark shadows using an online learning strategy that built upon a base classifier trained with manually annotated examples and refined with the automatically identified examples in the new videos.

The experimental results demonstrate that despite variation of lighting conditions in

videos our proposed method is able to adapt to the videos and remove shadows effectively. The sensitivity of shadow detection changes slightly with different confidence levels used in example selection for classifier retraining and high confidence level usually yields better performance with less retraining iterations.

To select window size for entropy calculation, our experimental results demonstrate that the detection accuracies using different window sizes are very close. In the cases of small shadow regions, the sensitivity varies, which is due to the small denominator in calculating the sensitivity. Our results show that window size of 3 by 3 exhibits the best performance for both small and large shadow regions.

In the evaluation of efficiency, updating kNN imposes little impact to the processing time of a frame. When making decisions with kNN, however, the algorithm faces much greater computational expense for calculating the distances between the foreground pixels and the examples in the kNN model and making selections of the nearest neighbors.

## 6.1. Future Work

Shadow removal in video sequences for indoor situations are new to the field of research. Shadows act as a noise element in many applications. In this section I would like to discuss possible future research opportunities that can be extended from my thesis.

(1) The current implementation is written in MATLAB 2014a. A native code implementation such as in C/C++ can be tested for efficiency. Efficiency of the system would scale up with such kind of an implementation.

(2) Chromatic shadows occur in indoor situations when the light source is colored, such as bright red or blue. Currently the implementation only handles achromatic shadows. More research and experiments on such situations may make the system to also handle chromatic shadows.

# APPENDIX

## MATLAB IMPLEMENTATION OF ViBe

```
% VIBE method for video background extraction
% USAGE: vibe('myVideoFile');
%
% Deepankar Mohapatra, Xiaohui Yuan
% COVIS LAB - UNIVERSITY OF NORTH TEXAS



function vibenew(vidFile)

N = 20;   % Number of examples per pixel that used as the model
R = 400;   % Max distance to be considered closely matched color value
nnCt = 2; % Number of closely matched examples in the model

% amount of random subsampling
fi = 16;

% Load file as grayscale video
vidSrc = vision.VideoFileReader(vidFile, 'ImageColorSpace', 'Intensity');

vidInfo=info(vidSrc);
w=vidInfo.VideoSize(1);
h=vidInfo.VideoSize(2);

model=zeros(h,w,N);     % background model. N values for each pixel
dist=zeros(h,w,N);      % distance matrix
segMap=zeros(h,w);      % background removed image
vidPlayer = vision.VideoPlayer();
vidPlayer2=vision.VideoPlayer();
```

```
% Initialize model with the first N images
for ii=1:N
  img= step(vidSrc)*256;
  model(:,:,ii)=img;
end

testCounter=0;
startN=0;
totalTime =0;
while ~isDone(vidSrc)
    tic;
  img = step(vidSrc)*256; % Acquire a frame

  % Process each frame
  for ii=1:N
    dist(:,:,ii)=(img-model(:,:,ii)).*(img-model(:,:,ii));
  end

  distR=zeros(size(dist));
  distR(dist<R)=1;
  distRSum=sum(distR,3);
  segMap=single(double(distRSum<nnCt)*255);

  %Step 2.1update model
  %Create Random Map of which pixels to update with a probability of 1/16

  randProbMap=randi(fi,h,w);
  randTemp=zeros(h,w);
  randTemp(randProbMap==1)=1;
  randRangeMap = randi(20,h,w);
```

```
updateMap=randTemp .* randRangeMap;

%find relevent index
updateInd=find(updateMap>0);
mapIndex=find(segMap<250);
Ind=intersect(updateInd,mapIndex);

%update model
modelIndex = Ind +h*w*(updateMap(Ind)-1);
model(modelIndex)=img(Ind);

%Step 2.2 Update neighbours
randProbMap=randi(fi,h,w);
randTemp=zeros(h,w);
randTemp(randProbMap==1)=1;
randRangeMap = randi(20,h,w);
updateMap=randTemp .* randRangeMap;

%find relevent index
updateInd=find(updateMap>0);
Ind=intersect(updateInd,mapIndex);

%update model
modelIndex = Ind +h*w*(updateMap(Ind)-1);

%Create neighbour vector
probableNeigh = randi([-1 2],size(modelIndex));
probableNeigh(probableNeigh==0)= -(h);
probableNeigh(probableNeigh==2)= h;
NeighIndex = modelIndex + probableNeigh ;
```

```
  removeInd=find(NeighIndex<=0 | NeighIndex>=h*w);
  NeighIndex(removeInd)=[];
  Ind(removeInd)=[];
  model(NeighIndex)=img(Ind);
  step(vidPlayer, segMap);
  step(vidPlayer2,img/256);
end

release(vidPlayer);
release(vidSrc);
end
```

# REFERENCES


[1] A. Amato, M. G. Mozerov, A. D. Bagdanov, and J. Gonzalez, *Accurate moving cast shadow suppression based on local color constancy detection*, IEEE Transactions on Image Processing 20 (2011), no. 10, 2954–2966.

[2] E. Arbel and H. Hel-Or, *Shadow removal using intensity surfaces and texture anchor points*, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (2010), no. 6, 1202–1216.

[3] A. Ariel, I. Huerta, M. G. Mozerov, F. X. Roca, and J. Gonzez, *Moving cast shadows detection methods for video surveillance applications*, Augmented Vision and Reality 6 (2012), 23–47.

[4] M. A. Asari, U. U. Sheikh, and S.A.R. Abu-Bakar, *Object's shadow removal with removal validation*, IEEE International Symposium on Signal Processing and Information Technology (Giza), Dec 2007, pp. 841–845.

[5] O. Barnich and M. Van Droogenbroeck, *Vibe: A universal background subtraction algorithm for video sequences*, IEEE Transactions on Image Processing 20 (2011), no. 6, 1709 – 1724.

[6] C. Benedek and T. Sziranyi, *Bayesian foreground and shadow detection in uncertain frame rate surveillance videos*, IEEE Transactions on Image Processing 17 (2008), no. 4, 608–621.

[7] J. Bian, R. Yang, and Y. Yang, *A novel vehicle's shadow detection and removal algorithm*, International Conference on Consumer Electronics, Communications and Networks (Yichang), April 2012, pp. 822 –826.

[8] F. Chen, B. Zhu, W. Jing, and L. Yuan, *Removal shadow with background subtraction model vibe algorithm*, International Symposium on Instrumentation and Measurement, Sensor Network and Automation, Dec 2013, pp. 264–269.

[9] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, *Detecting moving objects, ghosts, and shadows in video streams*, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2003), no. 10, 1337–1342.

[10] J. Gallego and M. Pardas, *Enhanced bayesian foreground segmentation using brightness and color distortion region-based model for shadowremoval*, IEEE International Conference on Image Processing (Hong Kong), Sept. 2010.

[11] G. W. Gates, *The reduced nearest neighbor rule*, IEEE Transactions on Information Theory (1972), 431–433.

[12] G. Han, D. Cosker, C. Li, and M. Brown, *User-aided single image shadow removal*, IEEE International Conference on Multimedia and Expo (San Jose, CA), July 2013, pp. 1–6.

[13] J.S. Hu, T.M. Su, and S.C.i Jeng, *Robust background subtraction with shadow and highlight removal for indoor surveillance*, IEEE/RSJ International Conference on Intelligent Robots and Systems (Beijing), Oct 2006, pp. 4545–4550.

[14] A.J. Joshi and N.P. Papanikolopoulos, *Learning to detect moving shadows in dynamic environments*, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008), no. 11, 2055–2063.

[15] C.R. Jung, *Efficient background subtraction and shadow removal for monochromatic video sequences*, IEEE Transactions on Multimedia (2009), 571 – 577.

[16] Y. Lu, H. Xin, J. Kong, B. Li, and Y. Wang, *Shadow removal based on shadow direction and shadow attributes*, International Conference on Intelligent Agents, Computational Intelligence for Modelling, Control and Automation and International Conference on Web Technologies and Internet Commerce (Sydney, NSW), Dec 2006, p. 37.

[17] N. Martel-Brisson and A. Zaccarin, *Learning and removing cast shadows through a multidistribution approach*, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007), no. 7, 1133–1146.

[18] K. Nakagami and T. Nishitani, *The study on shadow removal on transform domain gmm*

*foreground segmentation*, International Symposium on Communications and Information Technologies (Tokyo), Oct 2010, pp. 867–872.

[19] A. T. Nghiem, F. Bremond, and M. Thonnat, *Shadow removal in indoor scenes*, IEEE International Conference on Advanced Video and Signal Based Surveillance (Santa Fe, NM), Sept 2008, pp. 291–298.

[20] Y. Qin, S. Sun, X. Ma, S. Hu, and B. Lei, *A background extraction and shadow removal algorithm based on clustering for ViBe*, International Conference on Machine Learning and Cybernetics (Lanzhou), July 2014, pp. 52–57.

[21] M. Sofka, *Commentary paper on "shadow removal in indoor scenes"*, IEEE International Conference on Advanced Video and Signal Based Surveillance (2008), 299–300.

[22] Y. Wang, K.-F. Loe, and J.K. Wu, *A dynamic conditional random field model for foreground and shadow segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006), no. 2, 279–289.

[23] Q. Wu, W. Zhang, and B.V.K.V Kumar, *Strong shadow removal via patch-based shadow edge detection*, IEEE International Conference on Robotics and Automation (Saint Paul, MN), May 2012, pp. 2177 – 2182.

[24] D. Ramanan Yi Yang, *Articulated human detection with flexible mixtures of parts*, , IEEE Transactions on Pattern Analysis and Machine Intelligence (2012), 2878 – 2890.