ILLUMINATION COMPENSATION IN VIDEO SURVEILLANCE ANALYSIS

A Dissertation
Presented to
The Academic Faculty

by

Michael Ryan Bales

In Partial Fulfillment
Of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

May 2011

ILLUMINATION COMPENSATION IN VIDEO SURVEILLANCE ANALYSIS

Approved by:

Dr. D. Scott Wills, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Linda M. Wills, Co-Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Jongman Kim
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Ayanna Howard
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Justin Romberg
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. David Bader
College of Computing
Georgia Institute of Technology

Date Approved: March 21, 2011

ACKNOWLEDGEMENTS

Many acknowledgement sections begin with the words "This work is the culmination…" To me, more precisely, this work is a record of a significant—but not singular—aspect of my education. While it comprehensively discusses the motivation, experiments, and results of my research, a great deal more was learned than will probably ever be written down. Here I give thanks to the many people with whom I had the privilege of working, who influenced not only this work, but also the experiences that were earned during its conduction.

First, I express my deep appreciation to my advisors, Dr. D. Scott Wills and Dr. Linda M. Wills, for their mentorship, encouragement, fervor, technical insights, and innumerable paper revisions. My experience was much more fruitful and enjoyable as a result of working under their guidance.

I also appreciate Dr. Jongmyon Kim, Dr. Ayanna Howard, and Dr. Justin Romberg for serving on my committee, and for providing valuable feedback and insight that strengthened my research.

I am especially grateful for the opportunity to work with such a diligent and innovative research group. Special thanks to Dana Forsthoefel for her tireless efforts on numerous experiments and publication endeavors, and to Dr. Brian Valentine and Dr. Senyo Apewokin for their collaboration, stimulating discussions, and guidance in navigating the vagaries of Georgia Tech.

I particularly appreciate the opportunity to work as a GRA with GTRI, during which time I worked on a wide variety of interesting projects, and with a very dedicated and enthusiastic group. Many thanks to the folks at GTRI/SEAL, including (but not limited to, and in no particular order) Roger Dickerson, Bill Wilson, Sean Begley, Mike Brinkmann, Brian Hudson, John Trostel, Stan Sutphin, Dan Campbell, and John Porcello. It has indeed been a pleasure.

As my 23-year career as a full time student comes to a close, I am inspired to express thanks to the faculty—past and present—of De Soto School District 73 for their dedication and enthusiasm. These include Connie Tindall, Joan Drake, David Hagerman, Chris Woelich, Rick Ruble, Rod Cable, and Roger Eberhardt, to name a few. In addition, many thanks to my undergraduate and graduate advisors at UMR, Dr. Daryl Beetner and Dr. Steve Watkins, for all of their contributions toward shaping my academic career.

Most of all, my deepest gratitude goes to Julie, my wife; Dennis Bales and Amy Henson, my parents; Kristin, my sister; and my extended family for their lifelong support in shaping the person I am today.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BB	BigBackground
HSI	Hue, Saturation, Intensity
MAD	Mean Absolute Difference
MCD	Maximum Component Difference
MM	Multimodal Mean
SAD	Sum of Absolute Differences

SUMMARY

Problems in automated video surveillance analysis caused by illumination changes are explored, and solutions are presented. Controlled experiments are first conducted to measure the responses of color targets to changes in lighting intensity and spectrum. Surfaces of dissimilar color are found to respond significantly differently. Illumination compensation model error is reduced by 70% to 80% by individually optimizing model parameters for each distinct color region, and applying a model tuned for one region to a chromatically different region increases error by a factor of 15. A background model—called BigBackground—is presented to extract large, stable, chromatically self-similar background features by identifying the dominant colors in a scene. The stability and chromatic diversity of these features make them useful reference points for quantifying illumination changes. The model is observed to cover as much as 90% of a scene, and pixels belonging to the model are 20% more stable on average than non-member pixels. Several illumination compensation techniques are developed to exploit BigBackground, and are compared with several compensation techniques from the literature. Techniques are compared in terms of foreground / background classification, and are applied to an object tracking pipeline with kinematic and appearance-based correspondence mechanisms. Compared with other techniques, BigBackground-based techniques improve foreground classification by 25% to 43%, improve tracking accuracy by an average of 20%, and better preserve object appearance for appearance-based trackers. All algorithms are implemented in C or C++ to support the consideration of runtime performance. In terms of execution speed, the BigBackground-based illumination

compensation technique is measured to run on par with the simplest compensation technique used for comparison, and consistently achieves twice the frame rate of the two next-fastest techniques.

CHAPTER 1

INTRODUCTION

1.1. Motivation

Video surveillance cameras are commonplace in modern society. They provide security in public transportation terminals, along national borders, and in commercial venues from gas stations and retail markets to banks and government buildings. They monitor vehicular traffic flow for infrastructure management and are used to prosecute violators of traffic lights and speed limits. They augment many homeowners' personal security systems, and have been distributed throughout cities—both large and small—to observe and prosecute criminal offenses in public spaces.

Instances of these applications are rapidly increasing. According to market studies, as of 2009 there were an estimated 30 million surveillance cameras deployed in the United States alone, with the global market for surveillance cameras estimated to increase from $4.9 billion in 2006 to $9 billion in 2011 [1]. Many U.S. cities have deployed video surveillance systems to help combat crime, provide early warning for acts of terrorism, and improve response for public safety.

- Chicago, IL deployed an estimated 6000 cameras in schools and public transit stations [2]
- San Francisco, CA deployed 71 cameras for low frame rate recording for post-event image retrieval [2]
- Denver, CO has a total of 259 cameras in high activity locations [2]
- New Orleans, LA deployed 240 wireless digital cameras at a cost of $4.5 million for anti-crime and homeland security purposes [3]
- Baltimore, MD has a network of 480 cameras [4]

- St. Paul, MN deployed 60 cameras in its downtown district with a $1.2 million grant [5]
- Madison, WI installed 32 cameras with a $388,000 grant [5]
- Pittsburg, PA added 83 cameras to its network with a $2.58 million grant [5]
- Boston and 8 surrounding communities are one of 64 urban regions receiving funding under the Urban Area Security Initiative [6]

However, the benefits and appeal of such systems are not limited to major metropolitan areas, but are spreading into smaller municipalities as well.

- Pittsburg, CA (pop. 60,000) [7]
- Newnan, GA (pop. 33,000) [7]
- Salisbury, MD (pop. 24,000) [7]
- Scottsbluff, NE (pop. 14,000) [6]
- Liberty, KS (pop. 95) [6]

These cameras generate a phenomenal amount of data. Most surveillance cameras simply feed into monitors watched by human operators, or record video or snapshots to non-volatile storage and rely upon human analysts to later identify objects and interpret events of interest. Human observers offer the potential of being able to respond to events in real time. However, humans can easily be bored, distracted, or overwhelmed by the amount of data they have to process, and thereby miss important events. As a rough estimate, 30 million cameras operating at 30 frames per second at color VGA resolution (640 x 480 pixels x 3 bytes per pixel) produces 6.6 petabits of uncompressed video *per second*. Even with modern compression technologies, storage space for video is finite, and a great deal of video is eventually overwritten without being seen if some external alert or report does not draw attention to it.

Automated video surveillance has become progressively more attractive for monitoring environments that are tedious, difficult or dangerous for human operators to monitor. With the proper algorithms and system architecture, video processors can devote unwavering attention to a scene and extract important information about objects and events that the human visual system is ill-equipped to detect. As costs for imagers and computing hardware have decreased, and as computing capability has advanced to handle data-rich video streams, embedded video surveillance systems are poised to revolutionize automated surveillance applications in traffic management, point-of-access monitoring, and threat detection. While many useful computer vision processes can take advantage of large-scale computing platforms, it is infeasible to dedicate a top-of-the-line machine to every surveillance camera. The cell phone, gaming and digital media industries have played significant roles in driving the development of such processors, which are capable and low-cost, yet resource limited in terms of power consumption and memory. There is a compelling need to focus on developing video surveillance algorithms for these small-scale, low-power embedded platforms. Combining these dedicated vision processors with cameras to form distributed smart sensors will give rise to integrated virtual surveillance environments, in which objects are constantly accounted for, actions are recognized, and the state of the monitored environment is comprehensively updated.

Some steps toward automated video surveillance systems have already been taken. Chicago's city-wide surveillance system has tested video analytics for automatically detecting loiterers, abandoned objects, and suspicious behavior [1], [2]. Facial recognition and tracking algorithms are now sufficiently advanced to provide useful results [1]. However, significant work in the field remains to be done. For example, in 2005 the Secure Border Initiative proposed a "virtual fence"—dubbed SBInet—to secure the border between the U.S. and Mexico. The prototype system suffered from outdoor environmental factors such as camera towers swaying in wind, low data bandwidth, high latency, high false positive detection rates, and difficulty correlating targets with geographical position in large open spaces [8]. Other problems that automated surveillance systems must address include sudden illumination variation, automatic calibration of a system to an environment, partitioning of processing effort between camera nodes and central stations, and integration by central stations of the information being distilled by camera nodes. As practical automated video surveillance systems are deployed, we will learn a great deal about how to synthesize vast amounts of diverse data—recognized and tracked objects, observations from multiple viewpoints, and analyses of behavior—to make decisions and take action.

1.2. Overview of Video Processing Pipeline

A typical video processing pipeline is shown in Figure 1, depicting some of the most common steps used to perform useful, modern video surveillance tasks. The first stages of most surveillance algorithms involve the separation of foreground (changing regions of interest) from background (stationary regions or uninteresting motion) by modeling the background, and noting variations between the background model and the current scene. A variety of change detection algorithms have been proposed for this purpose [9], [10], [11]. Implementations of background models vary, but all are generally statistical representations of the persistence of image features that an application defines as uninteresting. Pixels classified as foreground are further distilled into blobs or object representations, which are then analyzed to recognize or track objects or identify events. Each stage of the pipeline produces a higher level of abstraction with a more compact data representation.

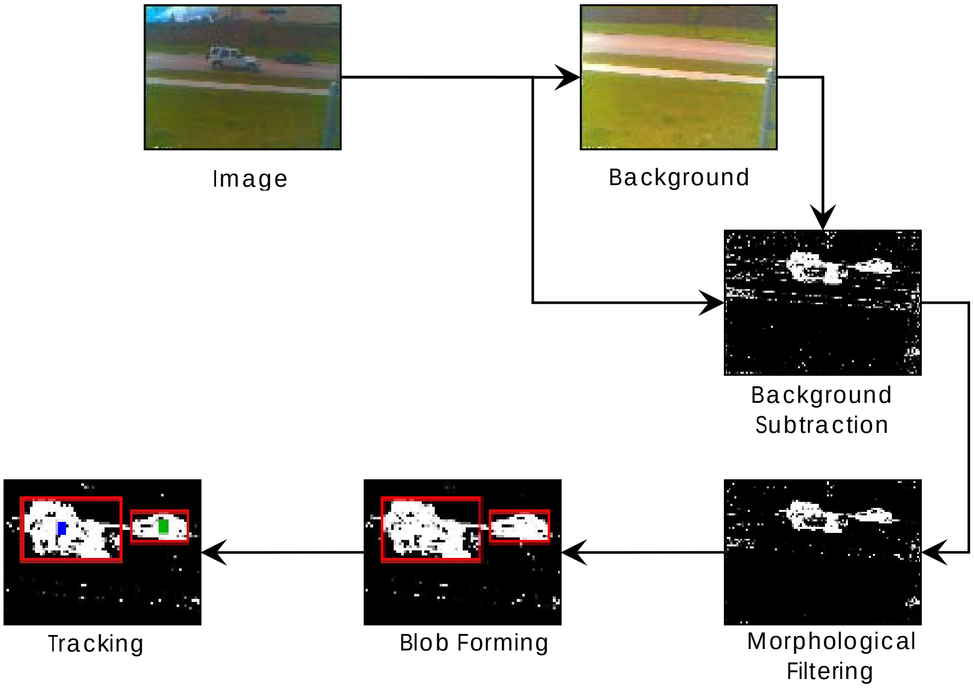


Figure 1. Overview of typical video processing architecture.

Illumination variation in a scene is a challenge to most background models. As temporary cloud cover and artificial lights change a scene's illumination, background pixels fail to match their background model counterparts and are falsely interpreted as foreground. Such a surge in the number of foreground pixels often taxes downstream processes because object tracking or recognition routines must sift through additional data. Salient foreground features can be masked by surrounding background under new illumination conditions, and objects of interest may be difficult to localize. Thus, real-time performance becomes harder to maintain, and analysis of the objects of interest becomes less accurate (or impossible in extreme cases) because algorithms are fooled into thinking that changing background regions are as important as moving objects. The conceptual motivation behind this work is easily seen in Figure 2, which shows some scenes before and after significant lighting changes. Common video surveillance applications, such as traffic management and municipal surveillance, operate in uncontrolled environments and are frequently subjected to unpredictable illumination variations. It is desirable for surveillance algorithms to monitor such scenes, reliably observe foreground objects, and filter out persistent background despite such lighting changes. This dissertation presents a method of compensating for sudden illumination changes by using reliable, dominant background features as reference points, and using chromatically sensitive transformations to adjust images to better resemble the scene under the original illumination condition. By performing this correction early in the process to improve change detection, the quality of subsequent processes is improved.

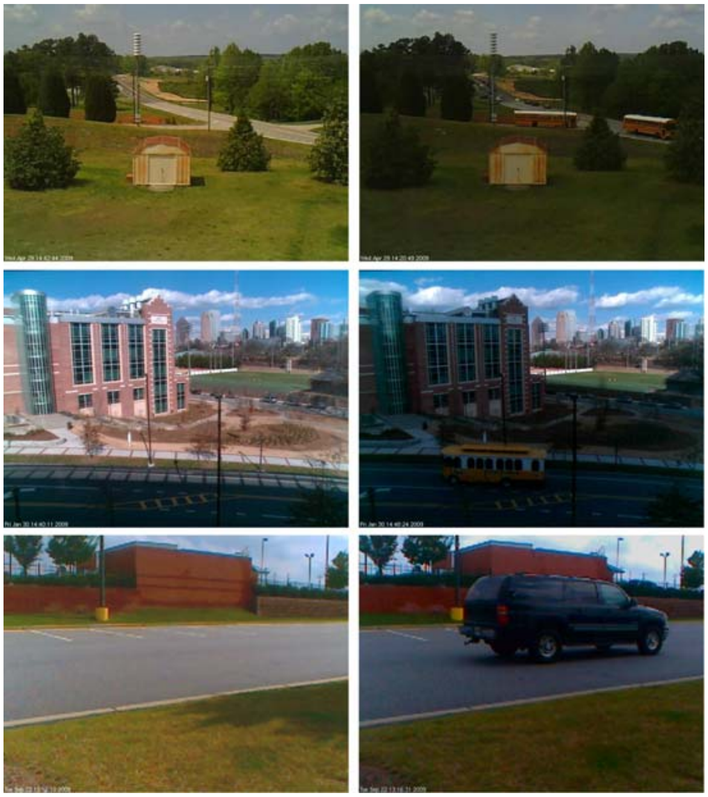


Figure 2. Examples of illumination changes causing perceptual problems in surveillance environments.

1.3. Problem Statement and Research Contributions

The objective of this research is to develop effective, real-time techniques to make surveillance video more robust to illumination changes for automated analysis algorithms. Sudden illumination changes are unpredictable events to video surveillance algorithms, and often cause those algorithms to falter in their "understanding" of a scene. These algorithms often cannot distinguish between illumination change and occlusion. This dissertation discusses the problem of illumination change in the context of current video surveillance theory, and presents and evaluates approaches for solving problems related to this topic. Three contributions to the field are presented. The first contribution presents a pixel clustering technique called BigBackground, which is based on chromatic region consistency. This technique results in a model of large stable regions of a scene, making it possible to distinguish light changes from occlusions and separating surfaces that are likely to respond differently to a given illumination change. The second contribution presents a short-term illumination compensation method based on BigBackground. This approach to illumination compensation is motivated by observed chromatic dependence of surface responses to illumination change. Several models for compensation are explored, and results are compared with other compensation approaches from the literature. The third contribution expands BigBackground as a general approach that can be used alone or integrated into independently-developed techniques. Several compensation techniques are comprehensively evaluated in an object tracking pipeline, where interactions with kinematic and appearance-based trackers are explored. Classification accuracy, tracking accuracy, runtime, and quality of object appearance are used as evaluation metrics. The inclusion of these contributions into the standard video processing pipeline is depicted in Figure 3.

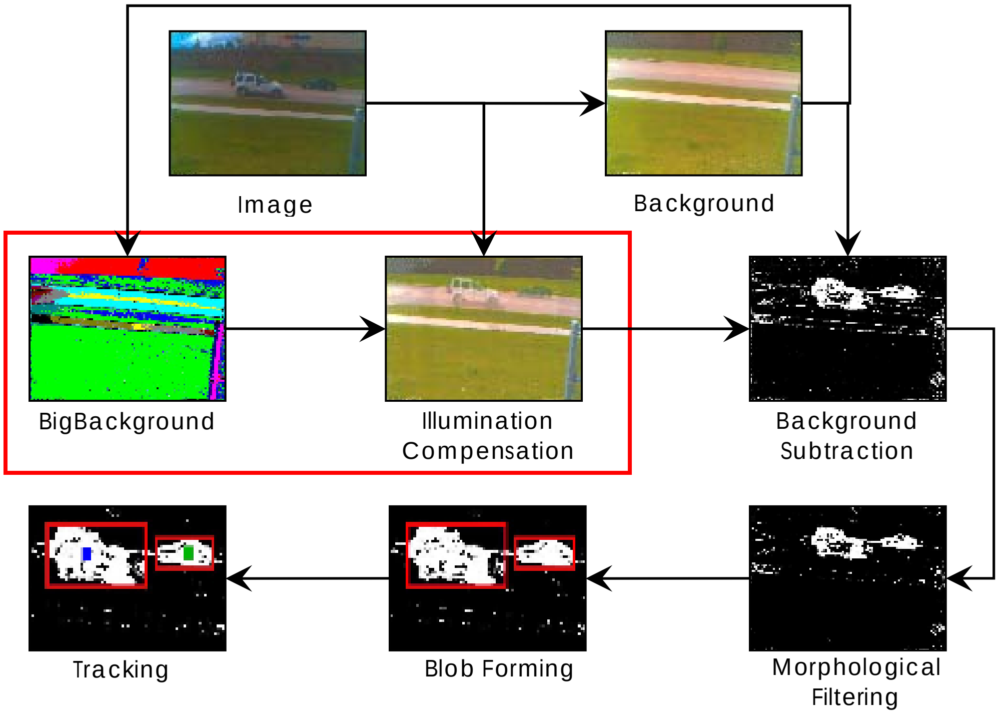


Figure 3. Video processing pipeline modified for illumination compensation.

1.3.1 Contribution 1

First, data are presented from a series of experiments in which color targets are subjected to controlled changing light conditions. The data indicate that chromatically similar surfaces respond similarly to a given illumination change, while chromatically dissimilar surfaces respond less similarly. This observation encourages the development of a color-centric illumination compensation technique to improve compensation accuracy. A background model is presented (called BigBackground) that identifies large, stable, salient background features based on chromatic popularity within a scene. The characteristics of the BigBackground (BB) model and its sensitivity to changes in input parameters are evaluated in terms of stability and scene coverage.

1.3.2 Contribution 2

Many vision processing tasks begin by distinguishing pixels belonging to salient features from those belonging to static regions or uninteresting motion. A background model is often used to learn the general appearance of a scene. A pixel is then classified as salient if its current appearance falls outside its expected value based on past observations. Sudden illumination changes occur faster than background models can adapt, and cause large numbers of pixels to fall outside their expected appearances. The masking effect of a surge in saliently classified pixels makes it difficult (or impossible) to extract true objects of interest. Illumination compensation addresses this problem by separating the effects of illumination change from interesting changes in the scene, and maintaining consistent perception of the scene. Several approaches to illumination compensation have been explored. This contribution exploits the chromaticity dependence of illumination change response to develop a BigBackground-based illumination compensation technique. Hand-marked ground truth images are used to evaluate the accuracy of foreground/background classification using this technique, and the technique's responses to changes in input parameters are examined. The technique is compared with several approaches from the literature in terms of classification accuracy and runtime.

1.3.3 Contribution 3

The characteristics of several illumination compensation techniques are comprehensively examined by applying these methods to an object tracking problem. Two new variations of BigBackground-based illumination compensation are presented and tested. The first takes into account scene-wide changes in BigBackground regions before performing local compensation. The second applies the BigBackground concept to an independently developed compensation technique, demonstrating that BigBackground is useful as a general approach. The impact of illumination compensation on several state-of-the-art tracking algorithms is evaluated. Each tracker exploits a different set of features with which to establish object correspondence across frames. Trackers using pure kinematics and combinations of spatial and spectral distributions are considered. Tracking accuracy is considered across many frames and video environments; thus this evaluation method is more thorough than the (commonly used) hand-picked frame approach of Contribution 2.

1.4. Summary of Results

The key results of this dissertation can be summarized as follows.

- The effects of illumination changes on surfaces are quantitatively shown to depend on the chromaticities of those surfaces [13]. By customizing compensation models for distinct colors, model effectiveness is improved. Applying a model customized for one surface to a surface of different hue multiplies the error rate by an average factor of 15. Mathematical model choice for illumination compensation has reduced impact on compensation accuracy when the model is customized for each chromatically distinct region.

- A background model is presented that identifies large, stable scene components by extracting dominant chromatically self-similar regions [14]. These regions often cover more than 50% of a scene—and can cover over 90%—and are 20% more stable on average than other regions in the scene.

- An illumination compensation method based on the BigBackground concept is presented [14]. Compensation model parameters are customized for each region, allowing chromatically distinct regions to respond independently to the same illumination change. When used to aid foreground detection, this approach decreases false positives by an average of 83% compared with no corrective action, and decreases false positives by 25%-43% compared with other compensation methods from the literature.

- Variations of BigBackground-based illumination compensation are explored that demonstrate that the BigBackground concept can be integrated into independently developed compensation techniques, establishing BigBackground as a useful general methodology [15].

- The discussed illumination compensation methods are comprehensively evaluated in an object tracking application [15], [16], [17]. Kinematic and appearance-based trackers are tested, and techniques are evaluated in terms of tracking accuracy and object appearance quality. The BigBackground-based techniques improve object tracking by an average of 20%, and produce less distortion in object appearance.

1.5. Dissertation Overview

The remainder of this dissertation is organized as follows. Chapter 2 presents the results of controlled experiments that test the chromatic dependency of illumination change effects on captured images. These results form the foundation and motivation for Chapters 3 and 4. Chapter 3 introduces and characterizes a model for extracting large, stable, chromatically self-consistent background regions. Chapter 4 discusses previous work on handling illumination change, presents an illumination compensation technique based on the region model of Chapter 3, and compares with other popular illumination compensation methods. Chapter 5 evaluates the impact of illumination compensation techniques on several state-of-the-art object tracking applications, each of which uses a different feature set for establishing correspondence. Conclusions and future work are discussed in Chapter 6.

CHAPTER 2

CHROMATICITY DEPENDENCE OF ILLUMINATION CHANGE RESPONSE

2.1. Introduction

The wide proliferation and relatively low cost of USB web-cameras make them attractive sensors for inexpensive computer vision platforms. Such platforms are useful for many applications including video surveillance, tracking, and recognition. Algorithms in these applications often rely on a degree of perceptual constancy to function properly. They observe trends in color pixel values to learn the appearance of background, and to identify features of interest. Illumination change is a common problem that such vision algorithms must face. Changes in lighting intensity, spectrum, or physical position alter the appearance of otherwise unchanged pixels, and can affect how a scene is perceived. It is desirable to compensate for illumination changes to improve the robustness of vision algorithms. Before effective illumination compensation can be performed, however, it is necessary to quantify the effects of lighting changes on images.

The purpose of this work is to determine the sensitivity of illumination change models to chromaticity, with the aim of improving illumination compensation techniques. Prior work in the field has discussed the problems of modeling illumination change globally—using one set of model parameters across the entire image—and has described the benefits of dividing images into arbitrary tiles. By considering each tile individually, the effects of spatially varying illumination and surface reflectance can be accommodated. This chapter considers the relationship between illumination changes and surface color, and demonstrates that the effectiveness of illumination models can be improved by segmenting an image into chromatically dissimilar regions and separately computing compensations for each region. Spatial regionalization combined with chromatic regionalization will likely lead to additional benefits, but is not a requirement and such compound effects are not examined here.

A set of color targets is illuminated by a controllable light source, and sets of images are taken with a web-camera under varying intensity levels and spectra. Choice of illumination model, chromatic regionalization of model parameters, and the web-camera's driver settings are examined for their effects on the effectiveness of illumination compensation. These controlled illumination experiments show that the choice of illumination model becomes less important when such chromatic regionalization is used. Computing illumination compensation models for each chromatic region reduces error by 70% to 80% on average as compared to applying a global compensation model across the entire image. Applying a model customized for one color to a color of different hue results in 15 times the error of that color's custom model. These trends can guide the development of computationally efficient illumination compensation techniques for webcam-based vision platforms.

2.2. Related Work

Illumination changes are generally categorized into two types: internal changes involve changes to the intensity or spectrum of the light source, while external changes result from the physical movement of the source with respect to the scene. Several studies have provided insight on the nature of scene response to internal illumination change. In particular, the choice of illumination model has received considerable attention as a tradeoff between computational complexity and accuracy.

Mindru et al. [18] compare diagonal and affine transformations using linear regression to determine the optimal matrix and offset values. These methods are tested on image sequences that featured changes in both illumination and viewing angle, with the conclusion that the affine transformation is generally worth taking the time to compute. The authors note that the offset values of the affine transformation are always significant, and that the affine transformation is especially more successful than the diagonal transformation in cases of extreme lighting change and is therefore worth the additional complexity. Gros [19] tests eight illumination models on images of a static scene in RGB space. A least median squares algorithm is used to find the optimum global parameter values for each model, and the error remaining between image pairs after compensation is calculated. For the case of intensity change, multiplication of the pixel triple by a single coefficient are found to be sufficient to account for most of the change, with more complicated models reducing the error marginally further. Spectral changes in the illumination source require models that adjust each color channel independently of the others. Finlayson et al. [20] show that under certain conditions, several color constancy theories can be achieved by a diagonal matrix transform if an appropriate change of basis is applied to the sensor response function.

The experiments discussed above consider several mathematical models for describing illumination change. However, each model is evaluated by optimizing its parameters to minimize global mean square error. Such minima can surely be found, but this approach of pure optimization assumes that all points in the scene should respond similarly to an illumination change, and neglects natural factors such as surface color, composition, or orientation and position with respect to the light source. The work presented in this chapter explores the role of chromaticity in the illumination change response of surfaces, and provides a foundation for chromatically-oriented illumination compensation in color video processing.

2.3. Experimental Setup

This section describes the physical setup of the experiments, followed by analyses of the resulting data. The experiments use a fixed-focus Logitech USB webcam on a stationary mount, as would be used in a surveillance application. A target was constructed consisting of twenty color chips (five hues with four saturations each) placed on a sheet of dark foamboard (Figure 4a). The color target was oriented perpendicularly to the camera's optical axis, with the camera positioned 1.5 meters away. A light source was constructed using nine standard, independently controllable light fixtures (Figure 4b). A plastic diffuser was used to diminish shadows, and the experiment was conducted in a dark room without external light sources. Lights were turned on three at a time to provide three consistent, discrete intensity levels. In addition, three bulb types were used (two incandescent, one fluorescent) to produce changes in spectrum. Bulb information is given in Table 1. The light source was located coaxially with and 3 meters behind the camera, and elevated 0.5 meters to reduce glare. A light meter was used to ensure that the light distribution across the target was uniform. The Scene Brightness data was collected for the mid-intensity condition (6 lights on), and was measured at the camera facing the target 1.5 meters away.

During the first stage of data collection, the webcam's driver is set to automatic until the gain, exposure, contrast, and white balance settings stabilize. This step is performed on a mid-intensity scene. The driver is then switched to static operation, so the same settings are used for all subsequent image captures. Ten images are captured for each intensity level and bulb type, after allowing 10 minutes between each transition to allow the light source to reach steady state. The second stage of data collection repeats this process with the driver left on automatic, allowing the webcam to adjust for each scene. The 10-image sequences are captured after the driver stabilizes.

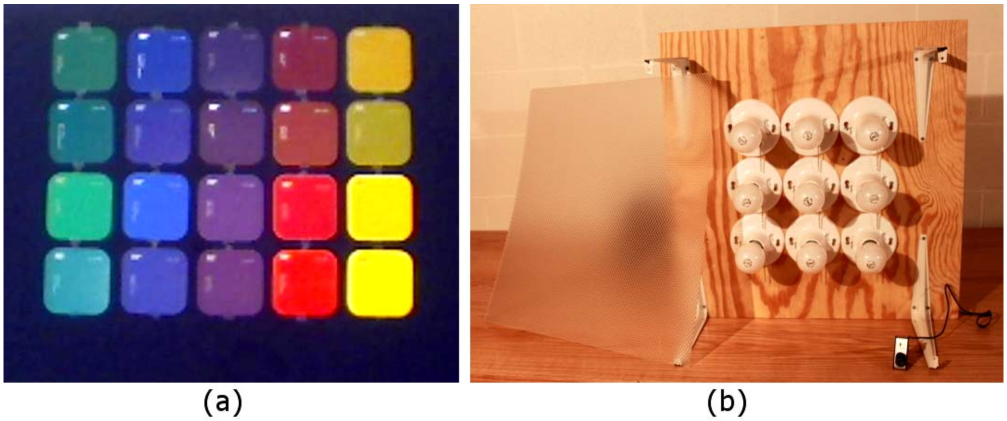


Figure 4. Equipment used for controlled illumination experiment. (a) Color chips arranged on target. (b) Light board featuring a 3x3 arrangement of light fixtures, and detached diffuser. The Logitech camera is visible in the bottom right.

Table 1. Information about the light bulbs used in controlled illumination experiments

				Average Scene	
	Type	Power (W)	Output (lum)	Brightness (lux)	Note
A	Incandescent	60	630	36	Full Spectrum
B	Incandescent	52	710	34	Soft White
C	Fluorescent	13	825	37	Soft White

2.3.1 Target Color Consistency

Sensors based on CMOS and CCD technologies are subject to many noise sources: temperature fluctuations, support electronics and the digital conversion process can all introduce noise into the final image. These effects can cause minor differences between images taken of an otherwise static scene. Thus, temporal and spatial averaging steps are used to minimize noise effects in our analysis of illumination changes.

The image sets are first tested for color consistency in the absence of illumination changes. Each set of 10 images is averaged together, and the average standard deviation if the images in each set is found to be less than 1.6% in RGB space. One average image is generated for each illumination condition, and is used in subsequent experiments. The color chips shown in Figure 4a are chosen to provide manageable regions of reasonably consistent colors. Computations are performed on 40 x 40 pixel windows within each chip, chosen to exclude the text and labels visible on the chips.

2.3.2 Model Effectiveness and Chromatic Locality

Three mathematical models are tested for their effectiveness at illumination change compensation. The set of evaluated models is limited to those lowest in computational cost. These models are shown in Equations 1-3, where P is the 3x1 RGB pixel being transformed, D is a 3x3 diagonal matrix, and T is a 3x1 translation vector. Models are computed in RGB space. For each lighting transition, a least mean square algorithm is used to compute the optimum parameters (α, β, γ, x, y, and z) for each model. First the parameters of each model are tuned for global application across the image. The image is compensated by transforming each pixel by the model being tested. Then the mean absolute difference (MAD) is computed between the compensated image and the original. The calculation for MAD is shown in Equation 4, where N is the number of pixels in the regions being compared; R, G, and B represent the pixel components, and subscripts 1 and 2 denote the regions being compared. Mean absolute difference is acceptable as a suitable metric for evaluating model performance, since the primary interest lies in compensation to assist downstream object detection and tracking algorithms. Next, the models are optimized for and applied to each of the 20 color chips in the target, and the MAD is computed between each pair of compensated and original chips.

$$D * P = \left[\alpha R \quad \beta G \quad \gamma B\right] \tag{1}$$

$$P + T = \left[R + x \quad G + y \quad B + z\right] \tag{2}$$

$$D * P + T = \left[\alpha R + x \quad \beta G + y \quad \gamma B + z\right] \tag{3}$$

$$\text{MAD} = \frac{1}{N} \sum_{i=1}^{N} \left| R_{i,1} - R_{i,2} \right| + \left| G_{i,1} - G_{i,2} \right| + \left| B_{i,1} - B_{i,2} \right| \tag{4}$$

Table 2 shows the average MADs for the cases of no model applied, models applied globally, and models customized for individual color chips. Three lighting transitions are tested in which three, six, and nine identical lights are turned on—for example, transitioning from three lights to six lights (3-6) of the same type. Data is presented for the webcamera driver set to static operation, and for the driver set to automatic adjustment. Table 3 is organized in the same fashion and shows data for changes in light spectrum where A (full spectrum incandescent), B (soft white incandescent), and C (soft white fluorescent) denote light bulb types with different spectra.

Table 2 shows that in the case of globally calculated models, model selection has a significant impact on goodness-of-fit. The D*P+T model consistently results in the lowest error, and all three global models noticeably degrade as the magnitude of the intensity change increases. However, computing model parameters separately for each color not only achieves 70% to 80% lower error than the globally applied models, but also achieves a more consistent error rate regardless of the model used or the magnitude of the intensity change. Enabling the automatic driver measurably improves globally applied models, and does not significantly affect the performance of color-wise models, which still achieve 40% to 50% lower error than the globally computed models. This indicates that it is no more difficult to compensate for simultaneous changes in intensity and driver settings than it is to compensate for intensity change alone. Thus, from a steady-state point of view, it is reasonable to leave the automatic driver enabled to improve the camera's dynamic range.

To obtain the data in Table 3, images are compared that were taken under illumination from different light bulb types but similar intensities (e.g., 6 type A bulbs versus 6 type B bulbs). The Uncompensated MAD row shows that the raw image differences caused by spectrum changes are lower in magnitude than those caused by the intensity changes. However, the automatic webcamera driver does little to mitigate the effects of the spectral changes. For changes in spectrum, optimizing the models based on chromatic region achieves 70% to 80% lower MAD error than the globally optimized models. Also, the global models gain little benefit from the automatic driver.

Table 2. Effectiveness of illumination models and color regionalization on reducing MAD error caused by intensity changes. Number pairs 3-6, 6-9, and 3-9 denote the magnitude of the intensity transition (ex: transitioning from 3 bulbs to 6 bulbs of the same type).

		Driver Static			Driver Automatic		
		3-6	6-9	3-9	3-6	6-9	3-9
Uncompensated MAD		145.2	101.7	246.9	24.4	18.4	10.4
Global	D*P	28.8	25.6	41.9	10.1	7.3	8.1
	P+T	21.2	19.9	32.2	9.3	6.5	8.1
	D*P+T	12.4	18.8	22.8	7.8	5.9	7.5
Chromatic Regions	D*P	4.3	4.3	4.5	4.7	3.9	4.6
	P+T	4.8	4.9	5.3	4.5	4.0	4.6
	D*P+T	4.2	4.3	4.6	3.7	3.5	3.7
Avg Error Reduction		76%	79%	84%	52%	42%	45%

Table 3. Effectiveness of illumination models and color regionalization on reducing MAD error caused by spectrum changes. Letter pairs A-B, B-C, and A-C denote the bulb type transition (ex: transitioning from type A bulbs to the same number of type B bulbs).

		Driver Static			Driver Automatic		
		A-B	B-C	A-C	A-B	B-C	A-C
Uncompensated MAD		34.7	39.6	42.6	38.5	42.0	41.7
Global	D*P	24.5	30.7	33.9	26.9	24.3	27.5
	P+T	18.6	26.3	27.5	19.8	19.3	19.1
	D*P+T	17.4	23.8	25.4	18.3	17.2	17.7
Chromatic Regions	D*P	9.8	8.9	6.7	10.0	8.2	8.6
	P+T	5.0	5.0	5.1	5.1	4.6	4.4
	D*P+T	4.1	4.3	4.5	4.4	3.9	3.8
Avg Error Reduction		70%	78%	81%	71%	73%	75%

2.3.3 Specificity of Model Parameters to Color

It is observed that color-specific models achieve much better results than applying a model tuned to an entire image. Presumably, this effect is because a globally computed model is a compromise between the many colors and surfaces present in a scene. Next, the degree to which the illumination model parameters are color-specific is evaluated. This experiment is conducted by again applying a least mean square algorithm to compute the optimum parameters for each of three models for each color chip in the target. This time, the optimum models for each chip are applied to each of the other chips, and the MAD error is calculated between each compensated chip and its instance in the original image. This demonstrates how well the model parameters for each color work for each of the other colors. The chip-wise results for the each transformation model are shown in Tables 4-7, followed by the average results over all chips for each model in Table 8.

The first column in Tables 4-7 show the chip color for which each model is computed. The labels (G, B, P, R, and Y) indicate green, blue, purple, red, and yellow respectively, while the number denotes different saturations. The Self-Correct column shows the MAD error for the model applied to the chip for which it was optimized. The Similar Hue column shows the average MAD error resulting from the model being applied to other chips of similar hue (i.e., the model for G1 applied to G2, G3 and G4). The Dissimilar Hue column shows the average MAD error resulting from the model being applied to the remaining chips. Tables 4 and 5 show the application of each color chip's transformation to the other chips during intensity changes, averaging the results for each bulb type. Tables 6 and 7 show the application of each color chip's transformation to the other chips during spectrum changes, averaging the results for each bulb transition. The chip-wise data is averaged to produce the summary of Table 8.

Table 4. Effectiveness of color-specific illumination models on reducing the MAD error of various surface colors for the 3-6 intensity transition.

	D*P			P+T			D*P+T		
	Self-	Similar	Dissimilar	Self-	Similar	Dissimilar	Self-	Similar	Dissimilar
Chip	Correct	Hue	Hue	Correct	Hue	Hue	Correct	Hue	Hue
G1	4.4	19.0	53.7	4.8	11.9	22.6	4.2	30.7	59.2
B1	4.2	12.8	53.1	4.6	9.3	25.1	4.5	10.8	31.3
P1	3.6	17.5	46.9	4.0	6.9	23.5	3.7	10.1	45.7
R1	4.2	18.3	54.3	5.6	25.3	22.6	5.4	39.2	36.1
Y1	4.0	22.7	68.9	4.5	24.8	33.8	4.1	43.7	68.7
G2	4.2	17.1	54.8	4.3	13.0	24.5	3.7	31.5	68.7
B2	3.9	10.7	44.4	4.2	9.7	24.5	3.8	20.3	71.4
P2	4.4	11.2	40.2	4.6	6.3	23.8	3.5	16.6	86.6
R2	3.5	15.7	53.8	5.0	20.6	25.7	4.5	40.1	66.7
Y2	4.5	24.9	62.2	5.1	32.4	30.4	4.4	72.1	98.3
G3	4.7	18.4	62.3	4.8	13.8	29.2	4.0	36.9	104.5
B3	3.6	13.1	47.7	4.1	16.6	35.7	3.6	43.2	124.0
P3	3.6	13.4	36.5	4.1	6.5	23.1	3.3	29.5	103.9
R3	4.1	14.3	65.2	5.3	20.1	40.4	5.2	32.5	84.7
Y3	6.4	21.1	77.9	5.7	27.3	49.7	5.3	42.4	96.8
G4	4.7	18.9	44.0	4.9	12.4	25.8	4.4	25.1	46.3
B4	3.6	10.9	41.9	4.1	9.0	27.0	3.2	25.5	112.6
P4	3.9	10.4	38.5	4.3	6.3	24.4	3.5	15.0	82.5
R4	5.3	23.8	83.0	5.7	25.7	47.3	4.9	50.9	123.7
Y4	5.2	20.0	74.2	5.5	33.3	54.1	3.9	68.7	157.6

Table 5. Effectiveness of color-specific illumination models on reducing the MAD error of various surface colors for the 3-9 intensity transition.

	D*P			P+T			D*P+T		
	Self-	Similar	Dissimilar	Self-	Similar	Dissimilar	Self-	Similar	Dissimilar
Chip	Correct	Hue	Hue	Correct	Hue	Hue	Correct	Hue	Hue
G1	4.3	20.3	70.5	5.0	23.4	35.9	4.6	35.0	51.6
B1	4.5	17.9	70.8	5.1	9.1	41.0	5.1	8.6	39.4
P1	3.5	19.8	63.6	4.4	11.0	37.8	3.9	14.2	49.3
R1	4.5	26.6	70.2	6.2	32.3	40.5	6.0	42.5	49.7
Y1	4.1	59.6	78.3	5.2	76.7	69.5	5.0	93.1	102.3
G2	4.6	18.3	74.2	4.8	23.4	36.0	3.9	46.7	91.8
B2	4.0	16.5	61.8	3.9	8.4	39.5	3.2	22.1	88.4
P2	4.4	12.5	57.1	4.4	9.1	37.2	3.5	19.9	87.4
R2	3.6	23.8	69.5	4.9	27.2	47.2	4.3	46.7	77.3
Y2	4.5	62.4	73.1	4.8	68.8	54.0	4.4	66.1	119.2
G3	5.1	19.5	79.3	5.3	25.5	50.0	4.1	57.4	138.5
B3	3.6	24.9	71.5	4.1	11.5	43.9	3.2	44.2	146.4
P3	3.4	14.2	54.1	4.3	14.1	37.0	3.9	34.1	92.3
R3	4.6	20.8	85.8	6.2	29.8	73.3	5.6	48.8	142.8
Y3	7.5	54.5	120.9	8.3	75.3	65.0	6.3	62.5	162.6
G4	5.2	21.8	61.9	5.0	26.6	44.3	4.7	34.8	52.2
B4	3.7	16.1	59.8	4.5	8.3	43.0	3.8	23.7	118.0
P4	3.9	11.6	55.9	4.0	9.0	37.0	3.2	19.0	87.0
R4	5.8	36.2	106.1	6.8	27.4	70.3	5.7	61.6	155.8
Y4	5.8	60.5	127.2	9.6	91.8	72.1	7.2	88.7	232.9

Table 6. Effectiveness of color-specific illumination models on reducing the MAD error of various surface colors averaged over all spectrum transitions for low intensity.

	D*P			P+T			D*P+T		
	Self-	Similar	Dissimilar	Self-	Similar	Dissimilar	Self-	Similar	Dissimilar
Chip	Correct	Hue	Hue	Correct	Hue	Hue	Correct	Hue	Hue
G1	11.1	41.7	67.0	5.6	13.6	30.1	5.1	24.3	61.2
B1	5.4	8.1	33.8	5.1	6.5	28.4	4.8	8.3	40.5
P1	4.4	10.8	31.7	3.8	8.4	24.0	3.5	12.2	39.0
R1	22.6	31.1	166.8	4.7	21.9	27.9	4.6	26.2	40.7
Y1	17.6	111.0	329.8	5.7	31.9	33.3	3.0	56.0	105.1
G2	5.8	27.4	42.6	4.4	14.8	24.6	4.1	25.1	56.8
B2	4.1	8.3	32.6	3.9	6.2	26.6	3.6	13.8	51.7
P2	5.2	10.8	33.0	4.7	7.0	24.4	4.4	10.3	45.9
R2	19.1	35.9	635.6	4.2	21.6	27.5	3.4	15.1	63.8
Y2	16.6	120.0	347.1	6.3	30.7	28.1	3.9	53.4	89.9
G3	13.3	61.3	97.1	5.8	23.1	43.5	5.4	38.2	93.0
B3	4.4	10.8	36.0	4.3	8.1	29.7	4.0	32.2	105.6
P3	4.6	12.0	29.0	4.3	8.7	25.2	4.0	21.8	65.6
R3	21.5	33.4	486.5	5.5	23.4	34.2	3.9	17.7	89.2
Y3	5.8	20.5	59.4	5.1	28.3	51.7	4.6	30.1	51.7
G4	5.8	28.2	29.6	5.2	14.0	24.7	4.8	25.1	35.1
B4	4.5	7.8	32.0	4.4	6.1	26.3	3.9	19.8	86.6
P4	5.2	9.4	31.8	4.6	7.4	25.4	4.3	12.4	58.2
R4	15.3	37.2	440.7	7.3	29.3	28.0	3.3	34.3	119.8
Y4	4.3	20.2	65.2	4.9	31.3	56.3	4.4	43.1	83.5

Table 7. Effectiveness of color-specific illumination models on reducing the MAD error of various surface colors averaged over all spectrum transitions for high intensity.

	D*P			P+T			D*P+T		
	Self-	Similar	Dissimilar	Self-	Similar	Dissimilar	Self-	Similar	Dissimilar
Chip	Correct	Hue	Hue	Correct	Hue	Hue	Correct	Hue	Hue
G1	6.7	23.5	59.1	5.5	15.7	39.4	5.2	26.5	57.3
B1	5.9	11.3	45.4	5.7	9.4	39.0	5.4	12.7	59.2
P1	4.0	9.5	41.8	3.9	7.8	35.3	3.4	12.8	54.4
R1	12.3	51.0	88.9	7.0	25.2	41.9	6.3	38.4	52.7
Y1	7.3	95.7	77.9	6.5	66.1	47.0	4.8	71.4	121.8
G2	4.4	21.5	46.4	4.1	18.1	34.3	3.7	36.7	70.7
B2	3.3	11.6	43.1	3.3	9.6	37.4	2.9	20.2	75.8
P2	3.8	10.6	42.2	3.7	8.5	34.9	3.4	13.5	58.1
R2	9.5	49.6	147.4	5.0	24.9	50.8	4.8	35.0	51.1
Y2	7.2	121.3	105.4	5.8	70.7	44.2	4.7	73.1	112.6
G3	6.3	33.0	67.8	5.1	27.4	49.5	4.8	44.9	94.6
B3	3.5	16.3	48.3	3.5	16.2	47.4	2.9	43.2	132.2
P3	4.7	10.9	42.9	4.6	9.1	38.6	3.8	30.2	89.8
R3	19.4	123.9	506.4	6.3	35.3	73.9	5.7	34.0	110.2
Y3	6.4	55.8	69.2	6.9	63.4	61.0	5.9	46.7	103.6
G4	5.0	23.0	42.6	4.8	15.5	36.2	4.6	28.3	44.4
B4	4.4	10.6	43.1	4.3	9.3	38.1	3.6	22.9	103.9
P4	3.9	9.4	44.7	3.8	8.2	38.5	3.1	15.2	81.8
R4	11.7	87.0	363.7	6.9	24.2	48.3	5.4	45.3	142.8
Y4	5.2	56.4	70.4	5.3	70.0	72.1	4.8	70.8	179.3

Table 8. The average MAD error of chromatically-optimized illumination models applied to identical, similar and dissimilar colors. Data is shown for small and large intensity changes, and for changes in spectrum with low intensity (3 bulbs) and with high intensity (9 bulbs).

		Intensity (3-6)		Intensity (3-9)		Spectrum (Dim)		Spectrum (Bright)	
		AVG MAD	STD MAD	AVG MAD	STD MAD	AVG MAD	STD MAD	AVG MAD	STD MAD
D*P	Self Correct	4.4	0.9	4.5	1.0	8.5	6.2	6.7	3.9
	Similar Hue	22.5	14.7	27.9	17.0	41.4	41.4	41.6	37.9
	Dissimilar Hue	58.1	17.7	75.6	20.4	140.4	198.5	99.8	120.0
P+T	Self Correct	5.0	1.2	5.3	1.5	5.0	0.9	5.1	1.2
	Similar Hue	24.3	20.5	30.4	26.1	22.1	16.1	26.7	22.3
	Dissimilar Hue	37.8	12.6	48.7	13.6	39.7	10.3	45.4	11.6
D*P+T	Self Correct	4.3	0.9	4.6	1.1	4.3	0.7	4.5	1.0
	Similar Hue	38.0	20.7	43.5	23.2	32.4	17.5	36.1	18.9
	Dissimilar Hue	90.5	41.5	104.3	48.7	82.1	32.2	89.8	36.1

The data supports the initial hypothesis—that optimum illumination models depend heavily on surface color, and that even with spatially uniform lighting, globally-tuned models are insufficient to compensate all of the colors and surfaces in a scene. Models achieve low error rates when applied to the color for which they are tuned (similar to error in Table 3). The models work better for surfaces of similar hue than for surfaces of different hues. The data in Tables 4-8 demonstrate that model error decreases because the appearances of chromatically dissimilar surfaces are indeed changing distinctly, and not just because the optimization problem is less constrained with fewer pixels to fit.

Table 8 shows the data for 3 illumination models, and for 4 separate lighting transitions: dim to medium intensity (3-6), dim to bright intensity (3-9), dim spectrum changes (transitions between 3 bulbs of each type), and bright spectrum changes (transitions between 9 bulbs of each type). Regardless of the illumination change type, corrections do well when applied to the chromatic regions for which they are calculated. As intensity changes increase in severity, compensation effectiveness decreases for

Similar Hue and Dissimilar Hue regions. Error increases by an average factor of 6 when chromatically optimized models are applied to colors of similar hue, and by an average factor of 15 when models are applied to colors of dissimilar hue. The diagonal transformation D*P proves most effective for Similar Hue regions during intensity changes, while the translation P+T is most effective for Similar Hue regions during spectrum changes.

2.4. Application to Realistic Surfaces

To test these observations on a more realistic scene with a wider diversity of surfaces, new sets of images are captured of a scene populated with various objects positioned at various angles (Figure 5). A square region was selected from each of 12 objects in the scene (3 each of blue, red, green, and yellow). The objects differ in saturation and surface reflectance. Table 9 shows the results of applying each chromatic region's illumination change model to similar and dissimilar chromatic regions, formatted similarly to Table 8. Observations drawn from previous data hold for this scene as well. Models applied to surfaces with Similar Hue in the scene of Figure 5 are slightly less effective than for the controlled surfaces of Figure 4b due to differences between chromatic regions in surface reflectance and orientation.

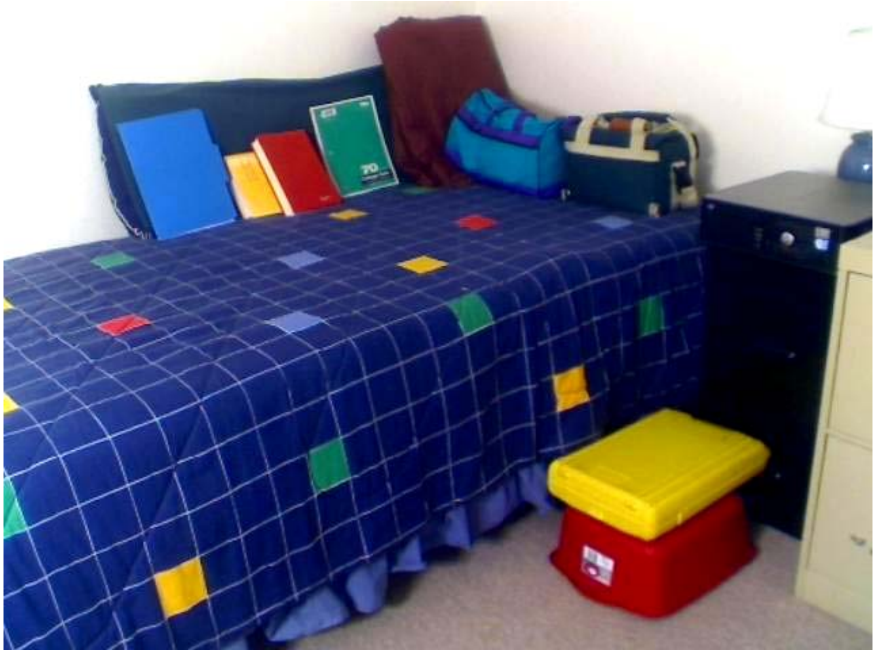


Figure 5. A scene of realistic surfaces featuring a diversity of colors, surface reflectances, and orientations.

Table 9. Average MAD error from applying three illumination models to the chromatic regions for which they were computed, chromatic regions with similar hue, and chromatic regions with dissimilar hue. Here, the models were applied to various surfaces in a realistic scene (Figure 5)

		Intensity (3-6)		Intensity (3-9)	
		AVG MAD	STD MAD	AVG MAD	STD MAD
D*P	Self Correct	4.0	1.8	4.1	1.8
	Similar Hue	17.8	8.2	23.8	9.5
	Dissimilar Hue	68.4	14.1	83.7	12.1
P+T	Self Correct	6.0	2.4	7.3	3.0
	Similar Hue	40.1	11.8	53.8	16.9
	Dissimilar Hue	68.0	12.2	93.8	16.7
D*P+T	Self Correct	5.5	2.3	6.7	2.9
	Similar Hue	48.9	21.9	57.9	24.7
	Dissimilar Hue	101.8	25.9	121.5	30.5

2.5. Conclusions

This chapter shows the significance of color to illumination changes in images captured by low-cost webcameras. Three illumination models are evaluated for their effectiveness in accounting for changes in lighting intensity and spectrum. The more complicated D*P+T model results in the smallest error out of the three models tested. However, by computing model parameters independently for each chromatically distinct region (without necessarily dividing the image into arbitrary spatial tiles), MAD error is reduced by an average of 70% to 80% compared with that achieved by globally calculated models. Furthermore, chromatic regionalization drastically reduces the variation in error due to model choice. This suggests that the least computationally expensive model (P+T) could be chosen in some applications to improve runtime performance in exchange for an acceptable penalty to accuracy. It is demonstrated that surfaces with different hues have significantly different illumination change responses, and that applying a chromatically optimized model to a surface of dissimilar hue increases MAD error by an average factor of 15. Finally, data is presented that suggests a webcamera's automatic driver does not generally increase the complexity of illumination corrections. The driver does not reduce illumination compensation effectiveness for spectrum changes, and does help stabilize images after intensity changes.

This work explicitly tests the color dependency of illumination change in a way not seen in prior literature, and provides a compelling argument for using color regionalization in illumination modeling. The next two chapters demonstrate and evaluate techniques to exploit these relationships in a comprehensive compensation algorithm using automatic color regionalization.

CHAPTER 3

BIGBACKGROUND: CHROMATIC REGIONALIZATION FOR LARGE BACKGROUND FEATURES

3.1. Introduction

Images represent large amounts of raw data. A typical uncompressed color image that is 640 pixels wide and 480 pixels tall represents nearly 1 megabyte of data. In recent years digital cameras have dropped in price, increased in resolution, and proliferated into phones and portable computing platforms. Computers tasked with video surveillance algorithms must sort through this data, and are generally trying to extract a relatively small amount of information. Is there an intruder? Are people or vehicles present? Where are they going, and how fast are they moving? The human visual system is adept at examining a scene in varying resolutions, able to pick out buildings, vehicles, and people. In contrast, computers operate by carrying out mathematical and logical operations on very small sets of numbers, and must first approach digital images by examining them at the pixel level.

Background models are important early vision algorithms in computer video surveillance. They provide a means for distilling large amounts of raw image data into useful preliminary classifications: which pixels are important, and which can be ignored. Background subtraction is a form of change detection: an image of a scene's background is subtracted from a new image, and non-zero pixels (or pixels for which the difference is sufficiently greater than zero) probably represent new, potentially interesting objects and are classified as foreground. A variety of background models have been proposed, which are most easily separated into two broad categories: recursive and non-recursive. Recursive background models use each new image to directly update the current state of the model. When computing the new model, a learning rate controls how much emphasis is placed on the new image versus the existing background image. Fast learning rates allow the model to adapt quickly, but risk improperly adapting to slowly moving foreground objects. Slow learning rates resist the influence of slowly moving foreground, but take much longer to respond to legitimate background changes. Because every frame contributes to the new model, the effects of scene features can persist indefinitely, though the learning rate can be such that a feature's contribution will become negligible. Non-recursive models maintain a history of the previous N frames, and compute a new background image based on those frames. While different frames in the history may be weighted differently, rates of adaptation are determined by the size of the history. Once the frames containing a particular feature have been rotated out of the history, that feature has no impact on the background model.

Rapid illumination changes make background model-based change detection difficult because illumination changes and object occlusions are difficult to distinguish at the pixel level. Illumination changes can alter the appearance of the entire scene, causing most truly background pixels to fail to match their background models.

While most image and video processing algorithms begin at the pixel level, the ultimate goal of most of these algorithms is to recognize, track, or make decisions based on objects or activities in the monitored environment. Thus, an important step in automated video surveillance is the aggregation of pixels into larger, higher-order abstractions. Abstraction allows the machine to begin to “see” a scene in terms of its content, rather than in terms of individual pixels. Content-oriented knowledge is less sensitive to pixel-level variations, and can be used to maintain consistent perception of a scene despite environmental changes.

In this chapter, an approach is presented for extracting regional statistics that identify large, permanent background objects such as roads and buildings. Applications of such a model were suggested in the previous chapter, as it was demonstrated that chromatically dissimilar regions respond differently to illumination change. This approach to regional statistic extraction is motivated by the observation that many surveillance scenes contain dominant, relatively homogeneous background structures. The presented approach identifies such structures by determining the dominant colors in a scene, and then mapping where those colors occur. The BigBackground procedure is described, and its sensitivities to the parameters that dictate its function are explored. The BigBackground model is found to cover between 50% and 90% of most scenes, while BigBackground pixels are found, on average, to be 18% more stable than non-BigBackground pixels.

3.2. Related Work

Image segmentation has been explored in various forms for three decades, and is still an active field as new applications, computational abilities, and underlying mathematical frameworks are developed. Several surveys of color image segmentation techniques are available [21], [22], [23]. Cheng et al. [21] reviews color image segmentation as a combination of grayscale segmentation techniques (histogram thresholding, feature space clustering, edge detection and region-based techniques) with different color spaces. Fundamentally, segmentation algorithms identify region boundaries such that regions are homogeneous within themselves relative to the heterogeneity across regions. The active contour approach to segmentation expresses this concept explicitly in the form of differential geometry [22]. Parameters such as boundary length, curvature, and appearance are viewed as energies, and partial differential equations are evolved until region boundaries settle to minimize those energies.

Freixenet et al. [23] discuss region-based segmentation and methods for using edge and region information to refine initial segmentation results. In split-and-merge techniques, an image is subdivided into smaller subregions. Each subregion that does not meet the required homogeneity criteria is recursively subdivided until every subregion is internally consistent (in terms of edges, gradients, color, etc). Then adjacent subregions that share desired traits are remerged. Region growing techniques begin with seed pixels distributed about the image. Neighboring pixels that share certain similarities with the seed are merged with the seed. This approach is sensitive to initial seed placement, and a great deal of work on region growing techniques has focused on choosing seed points.

Christoudias et al. [24] present the Edge Detection and Image Segmentation (EDISON) system, which combines segmentation and edge detection with the philosophy that the two operations are complements of each other. Specifically, a properly segmented image should feature proper edges at segment boundaries. The EDISON method relies on the popular mean shift algorithm to recursively localize the modes of the image's color space, and incorporates a weight for each pixel that is proportional to the pixel's distance to an edge. The result allows segmentation of distinct regions with weak boundaries separating them from their surroundings.

An alternative segmentation procedure is presented by Felzenszwalb et al. [25]. The Efficient Graph-Based Image Segmentation (EGBIS) identifies distinct region boundaries when appearance differences across a potential boundary are greater than the appearance differences within at least one of the resulting regions. This adaptive, non-local measure of saliency aims to segment features relative to their surroundings and their internal consistency.

Segmentation traditionally enforces an adjacency requirement. A region is only allowed to extend to its most distant direct neighbor, and associations are not made between non-adjacent objects of similar appearance. It is common for otherwise similar surfaces to be bisected by occlusions or interspersed background features and considered as separate regions. An image may have hundreds of distinct segments depending on algorithm selection, parameter adjustment, and scene complexity. Conversely, the adjacency requirement can also result in an oversimplification of scene features. A space that contains internal variation or islands of dissimilar appearance may be classified as a single region if the transition to a neighboring region is strong relative to the interior. Such variation could be due to the presence of distinctly different colors or reflectances. As discussed in the previous chapter, different colors tend to have different responses to illumination change, making their aggregation a liability to illumination change.

Examples of the EDISON and EGBIS segmentation methods (using parameters suggested by the authors) are shown in Figure 6 alongside the results of the BigBackground identification approach that will be presented in this chapter. The number below each false color image indicates the number of distinct scene regions identified by the technique. Scenes (b)–(e) show several instances of interlaced colors that are aggregated into single regions by the segmentation techniques, but which appear in the BigBackground map as pixels belonging to interlaced (but independent) regions. Also visible are spatially bisected regions of similar appearance that are treated as separate segments, but which BigBackground unifies.

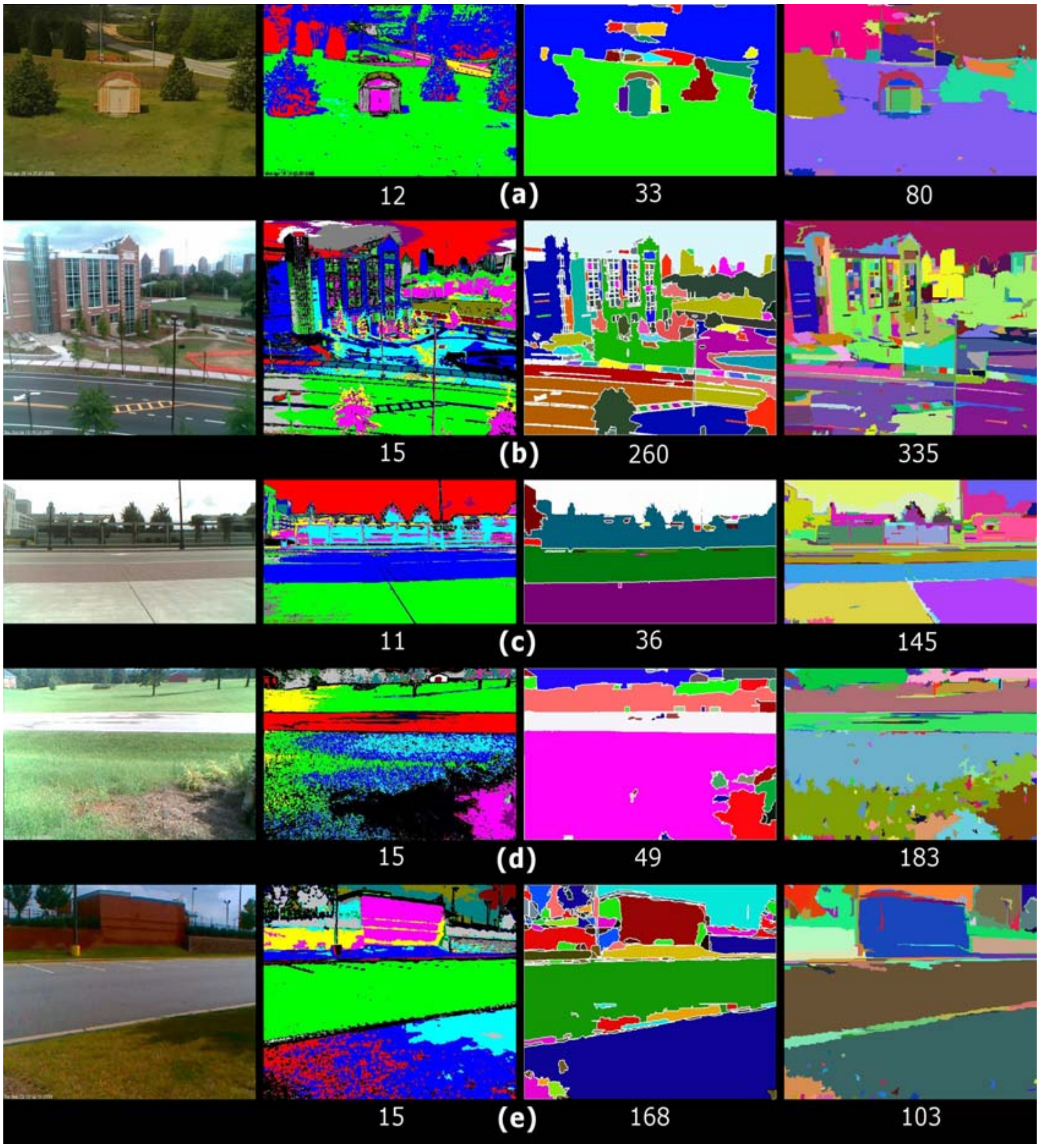


Figure 6. Comparison of BigBackground maps with segmentation procedure results. Left to right: original image, BigBackground region map, EDISON segmentation, EGBIS segmentation. Numbers below images indicate the number of distinct regions found in the scene. Note that for the BigBackground maps, all pixels of the same color belong to the same region. For the segmentation techniques, pixels that are the same color but are spatially disjointed represent independent regions.

In this work, it is desired to obtain a model of the dominant background features in a scene, and to recognize similar objects as unified surfaces regardless of spatial orientation. Small regions with infrequently occurring colors are more likely to be occluded or to vary with time, making them less appealing as reference points. While segmentation could be used in conjunction with a minimum size constraint, a regional feature that inherently accounts for size, does not rely so heavily on initial conditions, and allows nonadjacent similar surfaces is more attractive.

The concept of BigBackground shares some philosophical similarities with the frameworks of stels and locales. Stels are introduced by Jojic et al. [26] as ‘structuring elements’, and represent distinct regions of self-similarity within an image class. A class is considered to be a set of images that share similar features, such as images of faces, or images of roads and power lines. A variety of metrics can be used to distinguish between stels, such as pixel intensity, texture, or color. Stels are identified by an unsupervised Bayesian hidden model. For each pixel, the probability that the pixel belongs to each of the stels is calculated. The primary benefit is that the same palette of stels should be identified for all images in the same class, regardless of differences in environmental factors or imaging conditions.

The locales framework described by Drew et al. [27], [28] serves to localize features of interest within an image. Statistics are calculated on a tile-by-tile basis. The tile statistics may include average intensity, texture, mass, centroid, or histogram. These geometric features are measured in terms of the pixels within a tile. A tile is said to possess a particular feature if a minimum density of the pixels within the tile have that feature, and a locale is defined by the envelope of tiles that share that feature. It is not required that every pixel belong to a locale, nor is it necessary that locales are mutually exclusive. This contrasts with traditional segmentation, in which segmented regions do not overlap and every pixel belongs to a segment.

3.3. BigBackground

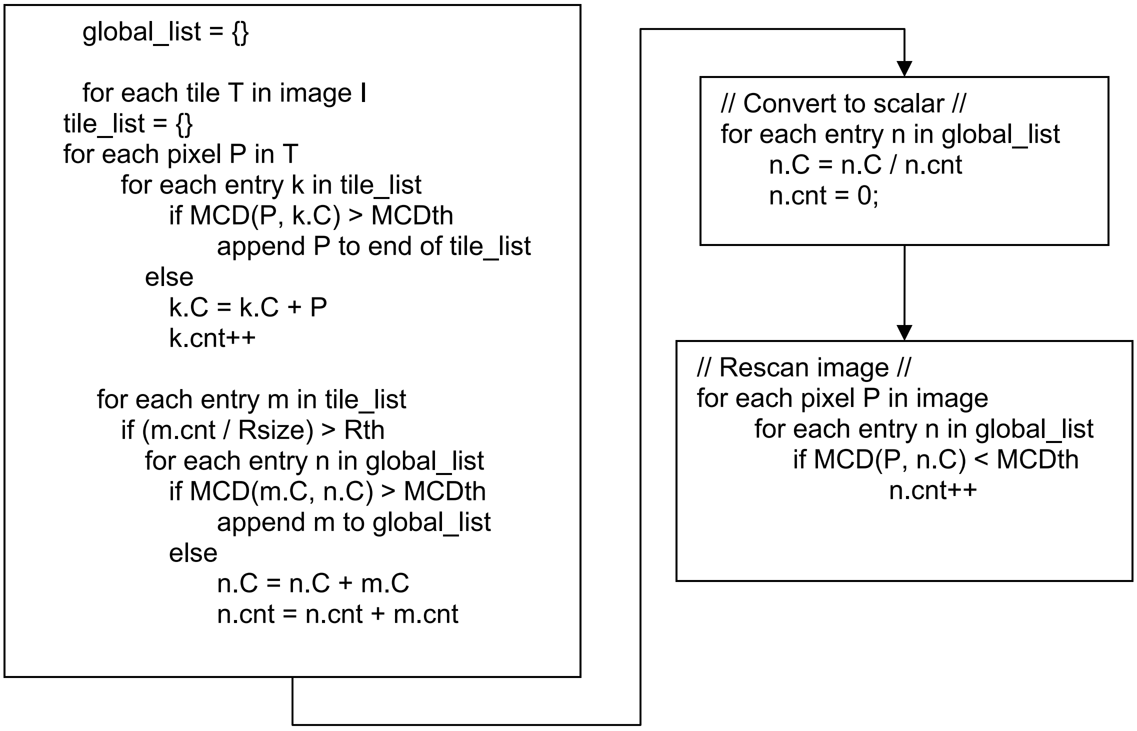
Within a scene, there are often large, stationary objects of relatively homogeneous color. Buildings, roads, and tree lines are examples. The size of these objects lends confidence that their appearances are unlikely to change, and that at least some portions of them will remain visible, even as occluding objects traverse the scene. BigBackground (BB) is based on the premise that these large background objects will be comprised of the most common colors in a scene, and therefore a relatively small color palette can be found which represents many of the pixels in the image. Once the most common colors are identified, a map can be generated that points each pixel that matches a BB color to that color in the palette. Each pixel that does match a color from that palette is said to belong to that color's region. Since each object is likely to respond to illumination changes uniformly over their local surfaces, it is possible to compare the colors of BB pixels before and after a lighting change to measure the effect on each region. BigBackground regions *could* be extracted from every frame of a video stream, but this would risk erroneous region extractions if large transient objects pass through the scene, and would be computationally wasteful because the background regions of interest are unlikely to change often. To keep from incorporating transient objects in the BB model, and to better distinguish between changing background and occlusion, the BB algorithm is applied to the output of a background model such as Approximated Median, Mixture of Gaussians, or Multimodal Mean [29]. Multimodal Mean is chosen for this study, and is described in Section 3.1. By extracting BB regions from the background model, it is possible to only recompute BB after significant changes occur in the background.

The maximum component difference (MCD), rather than sum of absolute differences (SAD), is chosen for most of the described pixel-comparison routines. The MCD is the largest difference between two color components, as shown in Equation 5. The SAD is simply the sum of the individual component differences between two pixels. Conceptually, the MCD is a better measure of how similar two pixels are, while the SAD better conveys how different two pixels are.

$$\text{MCD} = \text{MAX}(\text{abs}(R1-R2), \text{abs}(G1-G2), \text{abs}(B1-B2)) \qquad (5)$$

The BigBackground model is comprised of two data structures. The first is a reduced color palette C_0—a list of RGB colors, each of which represents a BB region. The palette is made of the most frequently occurring colors that satisfy a spatial density constraint. The second structure is a map with an entry for each pixel. Map elements are indices into the reduced color palette list. If a pixel in an image matches one of the reduced color palette colors, that pixel's entry in the map points to the corresponding color palette entry, and the pixel is said to belong to the BB region. Pixels that do not match any palette colors are assigned null entries in the map. The process of assigning pixels to palette entries in the map is called "branding".

The process for identifying BigBackground colors is as follows. The image is separated into square tiles (Rsize x Rsize), and a list of the colors present in each tile is generated according to the following rule: if a pixel matches any of the colors already on the list within a threshold, then that pixel is averaged with that color; otherwise, it is appended to the end of the list (Figure 7). Once an entire tile has been evaluated in this way, colors that occur frequently enough (more than Rth percent of the tile area) are added to a global color list by the same mechanism. If a tile color matches a global color within a threshold, it is ratiometrically added to the global entry; otherwise it is appended to the end. Once the entire image has been processed, the global color list is sorted in descending order by frequency of occurrence. The colors in the list are then converted from a ratiometric representation to a scalar representation: the running sum for each color channel is divided by the number of times it was observed. The image is rescanned to determine how many pixels match each scalar color list entry, as the average colors may have drifted since early tiles were examined. The list is once again sorted in order of descending pixel matches, and the Cnum most frequently observed colors are saved to form the reduced color palette C_0. The BB map is constructed by comparing each pixel from the background model with each entry in the color palette. Each pixel's entry in the map is the C_0 index of the color that best matches that pixel.



```
  global_list = {}

  for each tile T in image I
tile_list = {}
for each pixel P in T
     for each entry k in tile_list
         if MCD(P, k.C) > MCDth
             append P to end of tile_list
       else
         k.C = k.C + P
         k.cnt++

   for each entry m in tile_list
     if (m.cnt / Rsize) > Rth
       for each entry n in global_list
         if MCD(m.C, n.C) > MCDth
             append m to global_list
         else
             n.C = n.C + m.C
             n.cnt = n.cnt + m.cnt
```

```
// Convert to scalar //
for each entry n in global_list
    n.C = n.C / n.cnt
    n.cnt = 0;
```

```
// Rescan image //
for each pixel P in image
     for each entry n in global_list
        if MCD(P, n.C) < MCDth
                n.cnt++
```

Figure 7. Process for identifying BigBackground. Color list entries such as m and n consist of an RGB triple (denoted C) and a count of the number of times a color is observed (cnt).

An important feature of the BB model is that while a color must have a minimum density within a tile before it can be considered for inclusion into the global list, connectedness (direct adjacency) is not a requirement. It is therefore possible for surfaces composed of interleaved colors, such as grass or brick, to be modeled by BB as well as well as solid homogeneous surfaces. Additionally, it is not necessary for a pixel to belong to any BB region.

3.4. Stability Evaluation of BigBackground

A study is presented that evaluates the stability of BB regions compared to the stability of the overall image. BB's behavior under different threshold values is also characterized. This study uses results from BB and another background model known as Multimodal Mean, which is introduced fully in [29], but is briefly described here.

Multimodal Mean models each pixel as a finite set of possible average pixel values. If a pixel from the current image matches within a threshold of one of its possible averages, that average is updated with the value of the current pixel. The model tracks how frequently each mode has been observed, and how long it has been since each mode was observed. The match count and average of each mode are periodically decimated to prevent outdated information from persisting too long after the scene has changed, and to avoid integer overflow. Pixels from the current image that do not match any of their Multimodal Mean cells are declared foreground, and a new mode cell is created to track what may be the start of a new background value. Typically, 3 or 4 modes are allowed per pixel.

In this section, the relative temporal stabilities of BB pixels and non-BB pixels are evaluated. Illumination change effects are considered in the next section. Six test videos with no appreciable illumination changes are chosen to test BB stability. Sample images from these videos are shown in Figure 8. As part of the evaluation of BB's stability, Multimodal Mean is applied to a preamble period of each test sequence. By the end of that period, Multimodal Mean produces a stable background model, and a predominance image is created in which each pixel assumes its own most frequently observed mean color. The BB algorithm is applied to the predominance image to find the most dominant, sufficiently clustered colors throughout the scene. This results in a palette of common colors, and any pixel that matches one of these colors within a threshold is mapped to that entry in the color palette. This process is called “branding”. Pixels which do not correspond to a BB color do not belong to the BB model, and are assigned an index of zero in the map. BB pixels receive an index from 1 to Cnum. For each sequence, after computing the predominant image and the BB model, the next 100 frames in the sequence are analyzed. Each pixel in each new frame is compared with its corresponding pixel in the predominant image, and the number of BB and non-BB pixels are counted that continue to match their original predominant pixel values within an MCD threshold.

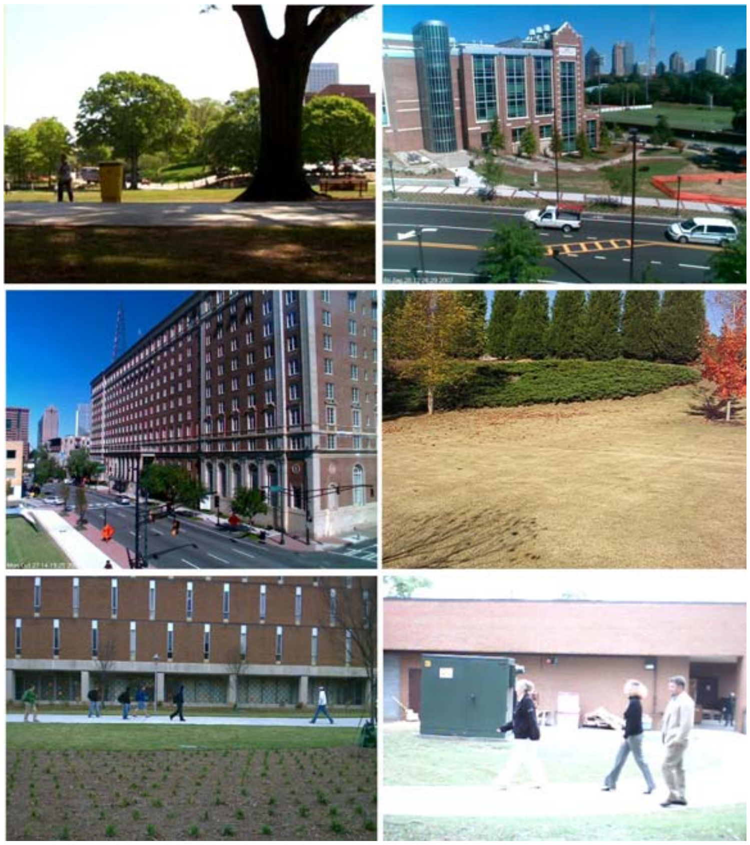


Figure 8. Samples from the videos used in the BigBackground stability experiments. (a) Shady, (b) City, (c) Biltmore, (d) Yard, (e) Courtyard, and (f) Sidewalk.

The evaluation of BB stability is done in two steps. First, the number of BB pixels that match their predominance values is compared with the number of non-BB pixels that match their predominance values. Because both sets of pixels are compared with their individual values in the predominant image, and BB and non-BB serves only as a spatial classification, this serves as an apples-to-apples comparison that reveals if the subset of pixel positions identified as BB is more likely to stay consistent than the remaining pixels. In the next step, the BB pixels are compared with their reduced color palette colors. The non-BB pixels are still compared with their own predominance values. Comparing these percentages reveals how well BB pixels match when described by a small color palette.

Table 10 summarizes the average BB stability statistics as observed in six video sequences. These averages are computed over 45 trials for each sequence, in which all combinations of the parameter choices for Cnum {5, 10, 15, 20, 25}, Rsize {8, 16, 32}, and Rth {10, 20, 30} are tested. The precise effects of these parameters are described in the next section.

Table 10. BigBackground Coverage and Stability in 6 scenes. These are the average results obtained from all combinations of the parameters Cnum={5, 10, 15, 20, 25}, Rsize={8, 16, 32}, and Rth={10, 20, 30}.

Seq	% Branded	NonBB % Match	BB % Match	BB % Match (small palette)
Shady	49.4	30.2	72.2	59.2
City	23.7	87.9	94.7	76.6
Biltmore	31.3	82.8	92.2	72.3
Yard	46.1	35.0	62.7	48.1
Courtyard	48.6	88.7	95.3	71.0
Sidewalk	58.7	50.9	66.8	60.7

First examining the 'NonBB % Match' and 'BB % Match' columns, it is observed that the pixels branded as BB match their predominant image values significantly more often than the unbranded pixels match their predominant values. The 'BB % Match (small palette)' column shows that when BB pixels are compared with their entry in a reduced color palette, the number of BB pixels that match drops by 20% or less. The '% Branded' column shows what percentage of pixels is identified as BB in each scene. In summary, Table 10 shows that the subset of pixels identified as BB is more stable on average than the remaining pixels. Using a small color palette to represent BB pixels still captures a very large set of the pixels that would have matched using their own custom models.

3.5. Parameter Characterization of BigBackground

This section examines BB's responses to variations in its thresholds and tuning parameters. The process of generating the BB model depends on three major parameters: the number of colors allowed in the color palette (Cnum), the size of the tiles used to create regional color lists (Rsize), and the pixel density required for a color to be preserved in the global color list (Rth). Two metrics are considered when evaluating the importance of these parameters: BB coverage (what percentage of a frame's pixels are classified as BB), and stability (what percentage of the BB pixels continue to match their predominant pixel values). Figure 9 shows the effect of different color palette sizes on BB coverage. Figure 10 shows the effect of different color palette sizes on the number of branded pixel matches. The reduced color palette is formed by sorting observed average colors by number of matches, and creating the palette from the Cnum most popular colors. By increasing Cnum, additional colors from the sorted list are included, allowing more pixels to match a palette color. Also, the total coverage of BB increases. The law of diminishing returns applies: since the most frequent colors are chosen first, any new colors added to the palette will not contribute as many pixels as those colors that have come before. Figures 9 and 10 show that as the size of the palette increases, the number of pixels identified as BB also increases, but the percentage of those pixels that match their model color over the course of the sequence tends to decrease slightly. This indicates that the BB colors with the fewest member pixels (and therefore the last to be added to the palette) are somewhat less stable than the most popular colors.

The parameters Rsize and Rth are observed to have very small, erratic effects on BB coverage and stability. Rsize is iterated through 8, 16, and 32, while Rth is iterated through 10, 20, and 30. These parameters rarely influence coverage or stability by more than two percent. The direction of the change depends heavily on the scene; increasing Rsize increases coverage in some sequences, while decreasing coverage in others. This small, erratic response suggests that Rsize and Rth can be chosen to maximize computing performance. For instance, increasing Rsize from 8 to 32 reduces the overhead of processing each tile, and increasing Rth from 10 to 30 places more stringent requirements on color density within each tile, thereby reducing the number of colors to be sorted and searched through in the global list. The main tradeoff to consider when tuning BB parameters is BB coverage versus BB stability. Additional pixels identified as BB tend to be less stable than those pixels previously identified. Color palette sizes (Cnum) of 15 to 20 are generally observed to capture the most significant background structures without capturing unnecessarily small features. Examples of predominance images and their BB-produced region maps are shown in Figure 11. False colors are used to highlight the separation of BB regions. Black pixels do not map to a BB color.

After analyzing the distribution of colors in the BB palette, it is observed that several palette entries are occupied by colors that a human observer would call the same. As a result, several significant regions in the scene are left out of the model because subtle color variations of a few large surfaces occupy most of the color palette entries. It is desirable to obtain a chromatically diverse color palette for two reasons. First, a diverse palette allows the effects of a lighting change to be observed on a wider range of surface types. Second, a diverse palette is able to cover more of the scene with the BB model. An experiment is conducted to justify this assumption using the image sequences with illumination change. The sum of absolute difference is computed between every BB color pair. The sum of absolute difference is also computed between every pair of RGB Translation illumination compensation factors. Measurements are made of the separation of the colors in the BB palette and the separation between the illumination compensations of those colors, and the data is organized in an XY scatter plot to observe any correlation (Figure 12).

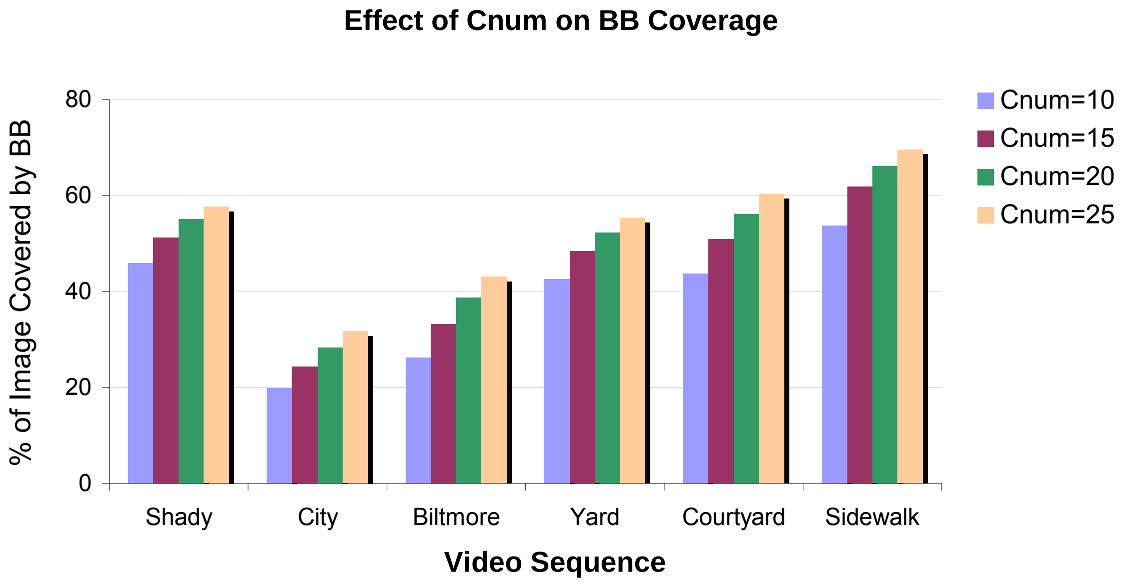


Figure 9. Increasing the size of the color palette (Cnum) increases the number of pixels belonging to BigBackground.

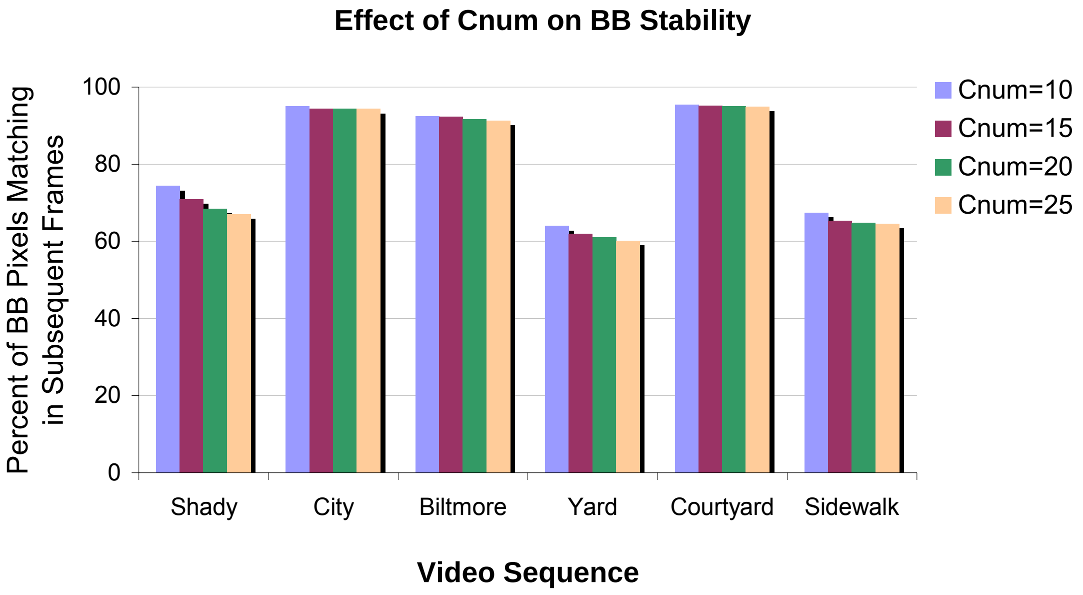


Figure 10. The additional pixels incorporated into the BigBackground model by increasing Cnum match slightly less often.

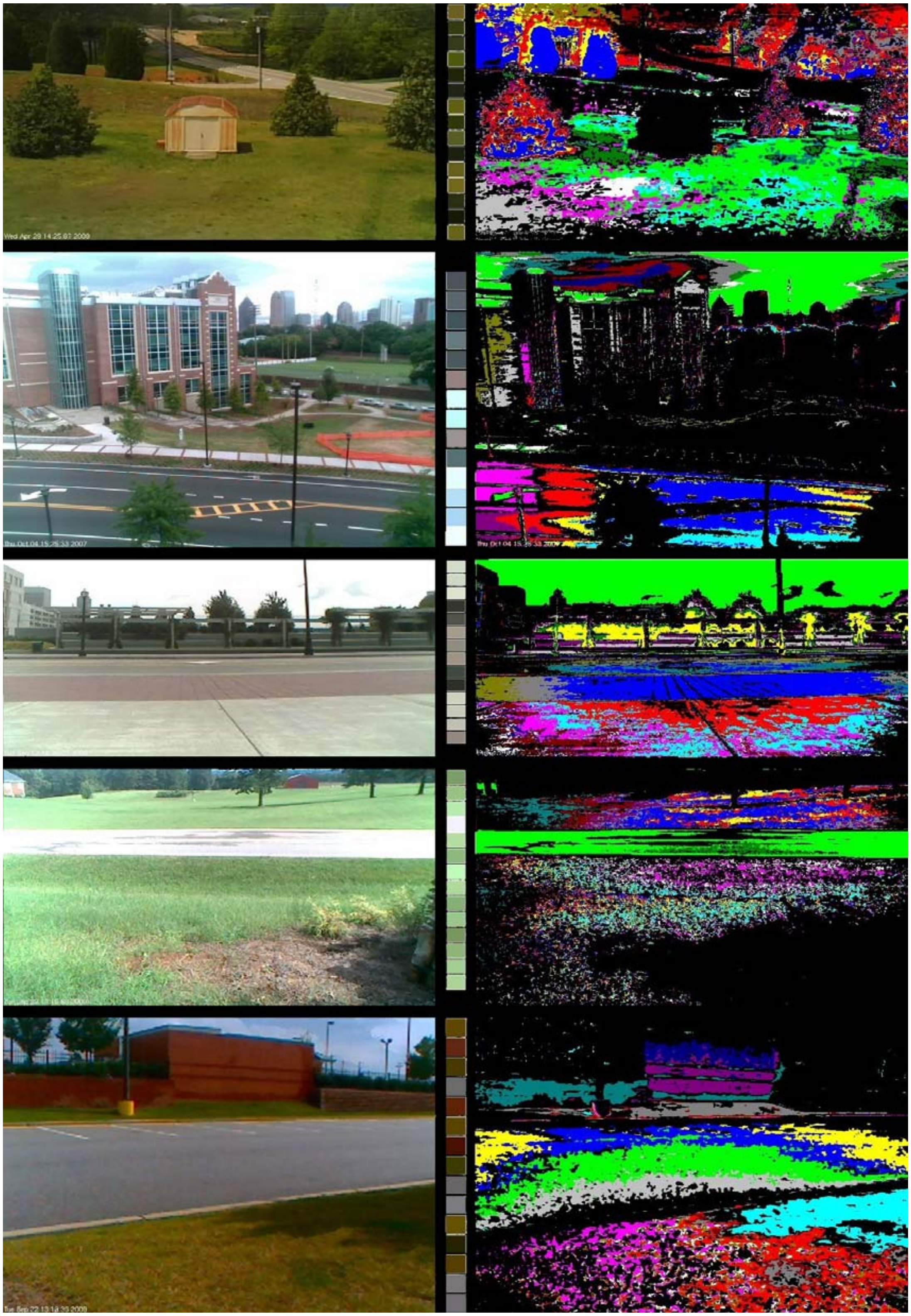


Figure 11. Identification of BigBackground regions. Left: Samples of Multimodal Mean predominant images from evaluation sequences. Center: Color palettes. Right: False-color BigBackground maps. Each color represents a different BB region. For these examples, Cnum=15, Rth=20, and Rsize=16.

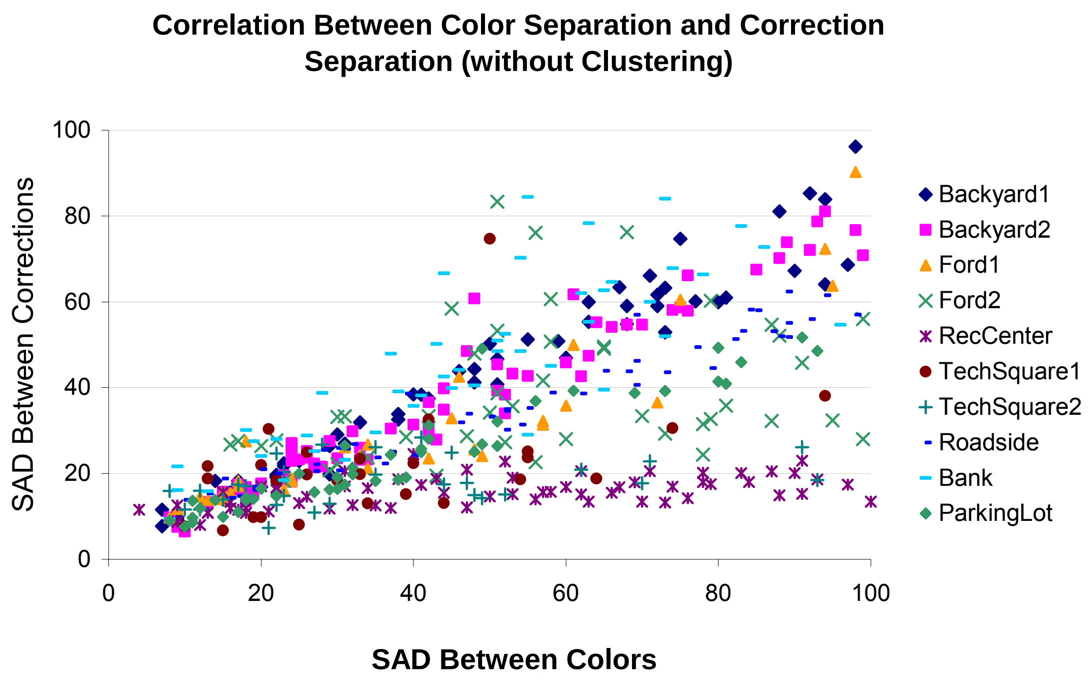


Figure 12. Correlation between color separation and correction separation. Pairs of colors that have a small sum of absolute difference tend to have a small SAD between their illumination corrections as well, indicating that the pair could possibly be treated as a single color for compensation purposes.

It is apparent from Figure 12 that palette colors that are very close together have very similar compensation factors. A strong linear relationship can be seen between color separation and correction separation for each sequence. In order to improve color palette diversity, the following color clustering step is added to the BB color-finding algorithm: after producing the scalar color list, all of the colors in the list are examined, and colors that match within a clustering maximum component difference (ClMCD) of each other are organized into a single linked list. The color palette then consists of an array of linked lists, rather than an array of individual colors. The parameter ClMCD represents the maximum color distance allowed between similar colors. If a pixel matches any one of the colors in a list, it is branded with the index of that list instead of the index of a specific color. The weighted average of each linked list of colors is used to represent the list for calculating illumination compensation.

The previous experiments are repeated to observe the effect of clustering on palette diversity and BB coverage. Figure 13 shows the correlation between average color separation and compensation separation. Some linear trends are still present, but the entire mass of data points has shifted significantly up and to the right. This signifies that there is now a greater difference between palette entries and their respective compensations, and that the palette has been chromatically diversified.

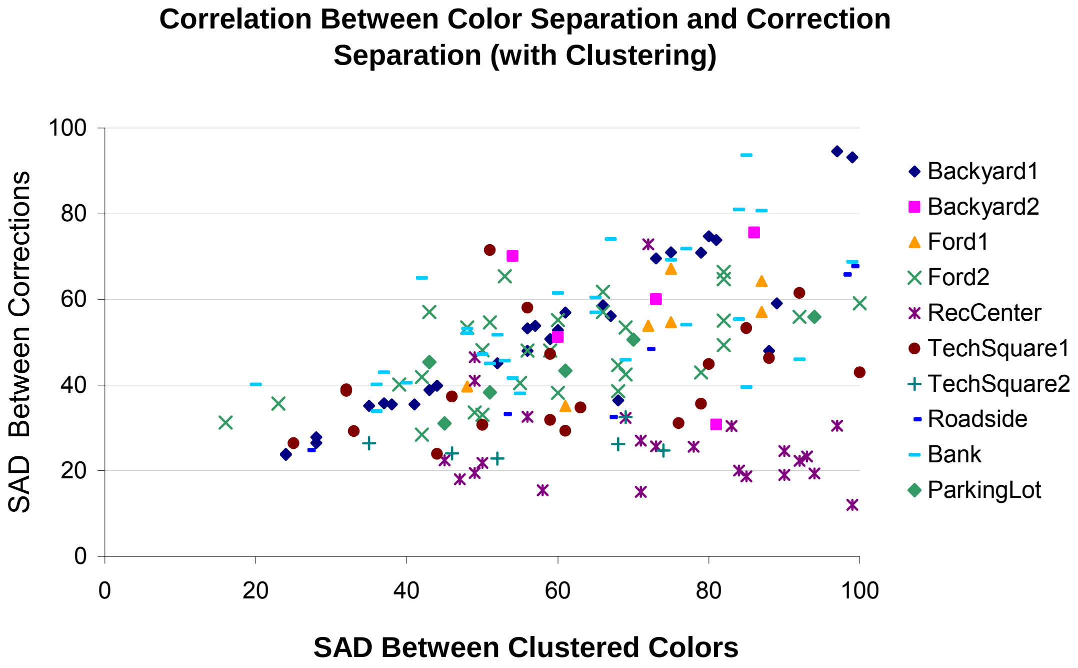


Figure 13. Correlation between color group separation and correction separation after additional clustering. Far fewer color pairs have SADs less than 100.

The stability experiment is repeated to observe the effects of clustering on BB coverage and relative stability. The effects of parameters Rsize and Rth are also measured for the algorithm with clustering. Table 11 shows that again, the pixels branded as BB match significantly more often than the remaining pixels. Compared with the values in Table 10, the clustering process slightly decreases stability. However, the BB model covers significantly greater image area—generally increasing by 20% or more. This increase in coverage more than makes up for the stability decline, and indicates that a greater number of pixels are being matched with nearly the same reliability. A larger percentage of BB pixels also match within the reduced color palette.

Table 11. Stability for BB with color clustering.

Seq	% Branded	Non-BB % Match	BB % Match
Shady	68.9	26.6	62.6
City	52.8	86.1	92.3
Biltmore	64.5	78.7	89.1
Yard	65.4	28.2	57.9
Courtyard	76.4	84.6	93.9
Sidewalk	81.8	42.9	62.8

The next experiment examines the effects of the BB parameters Cnum, Rsize, Rth, and the new clustering maximum component difference threshold (ClMCD) on BB coverage and stability. Figures 14 and 15 show the results for varying Cnum. The results shown are the averages for each sequence over all combinations of Rth, Rsize, and ClMCD. Again, a law of diminishing returns in coverage is observed, as well as slight decreases in the proportion of BB pixels that consistently match. Because colors are added in order of popularity, subsequent palette entries contribute fewer pixels and are less stable than preceding entries.

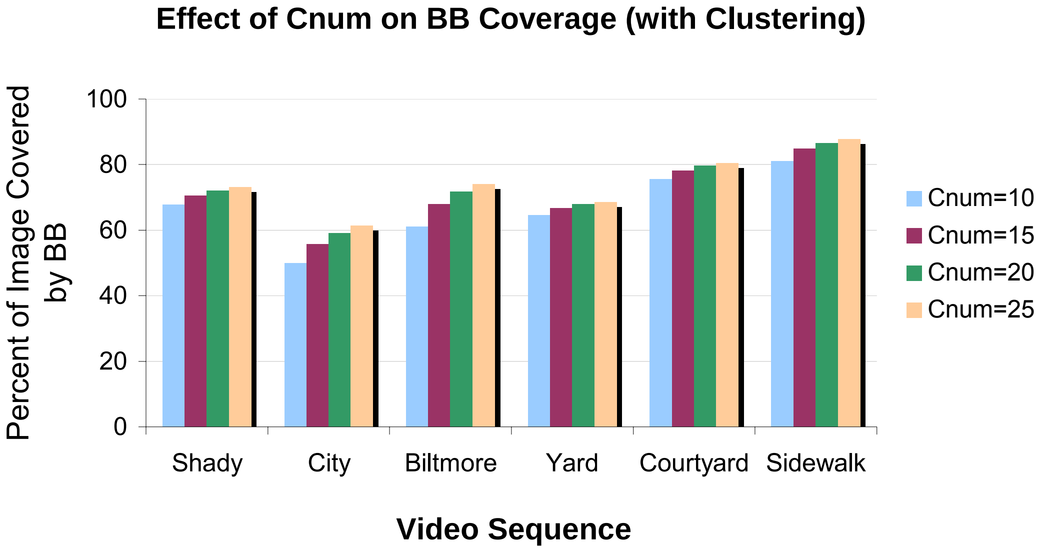


Figure 14. The effect of increasing the size of linked list color palette size on BB coverage. The clustering step succeeds in increasing coverage between 20% and 40% over the non-clustering algorithm for the test sequences.

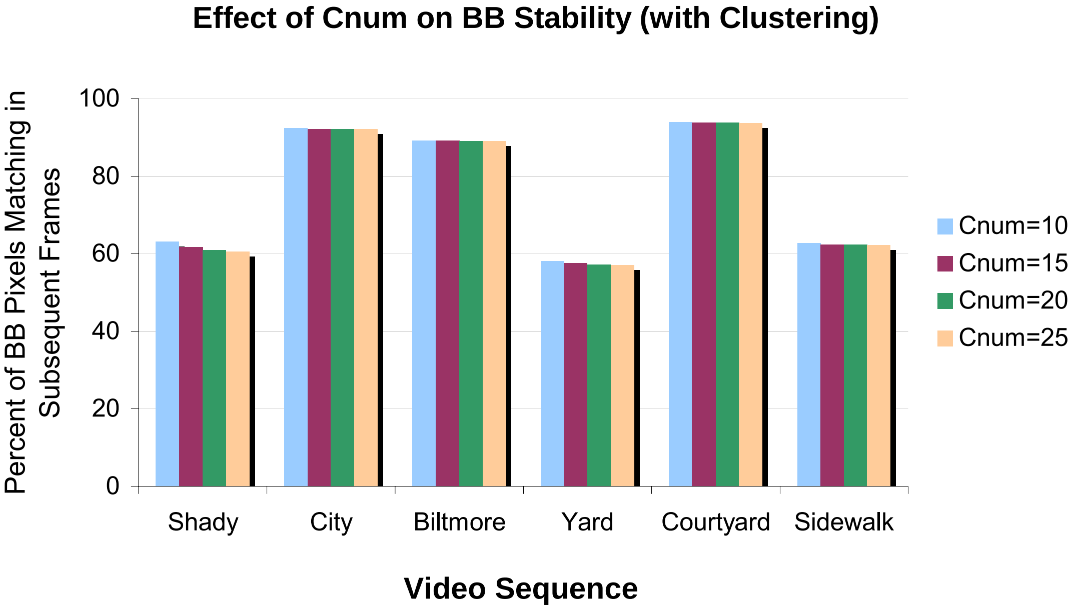


Figure 15. The effect of increasing the size of linked list color palette on BB pixel stability. A general decrease in stability is observed when compared to the pre-clustering stability measurements (in Figure 10), but the sharp increase in overall coverage more than compensates for the stability decline.

While Rsize and Rth have a negligible effect when no clustering is used, their effects increase in magnitude as the ClMCD threshold is increased. The plots in Figures 16 – 21 show the performance of different Rth and Rsize combinations for different ClMCD values and a constant Cnum of 20 for three representative sequences (Biltmore, Shady, and Courtyard). The Rth parameter takes on values of 10, 20, and 30, while Rsize takes on values of 8, 16, and 32. Increasing the ClMCD shifts the overall coverage percentage upward, and the match percentage downward. However, the curves for different ClMCD values are not parallel, and demonstrate that as ClMCD increases, the effects of changing Rth and Rsize become more dramatic. Changes in Rsize have the most dramatic impact. Small Rsize values—which correspond to small tile sizes during color segmentation—lead to the greatest BB coverage. The Rth parameter follows a similar relationship but is less pronounced. Small Rth and Rsize values result in a larger number of colors in the global list. A small Rsize means that the average colors found during the initial color search stage are more localized, and aren't competing with other slightly different colors for inclusion into the global list as they would if they were in the same tile. This process allows similar colors to be clustered into color groups after they have been identified and added to the global list. Examples of the effects of the clustering step on BB region coverage and association are shown in Figure 22. The sample images show the false-color BB maps from Figure 11 on the left (in which no clustering was used), and new false-color maps produced by BB with clustering on the right. The average color palettes produced without and with clustering are shown in the center left and right, respectively. These images reveal the increase in BB coverage (as additional pixels are mapped to BB), as well as the improvement in perception (as similar surfaces that were previously regarded as separate regions are now associated with the same region). Increased diversity in color palette hue and shading is also observed.

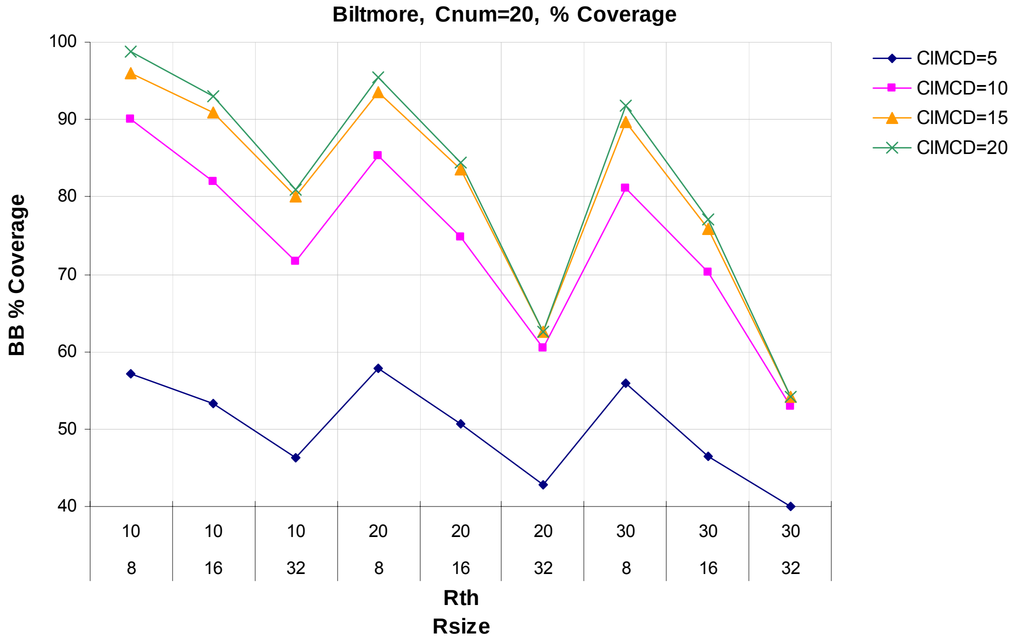


Figure 16. BigBackground coverage of the Biltmore sequence as a function of CIMCD, Rth and Rsize.

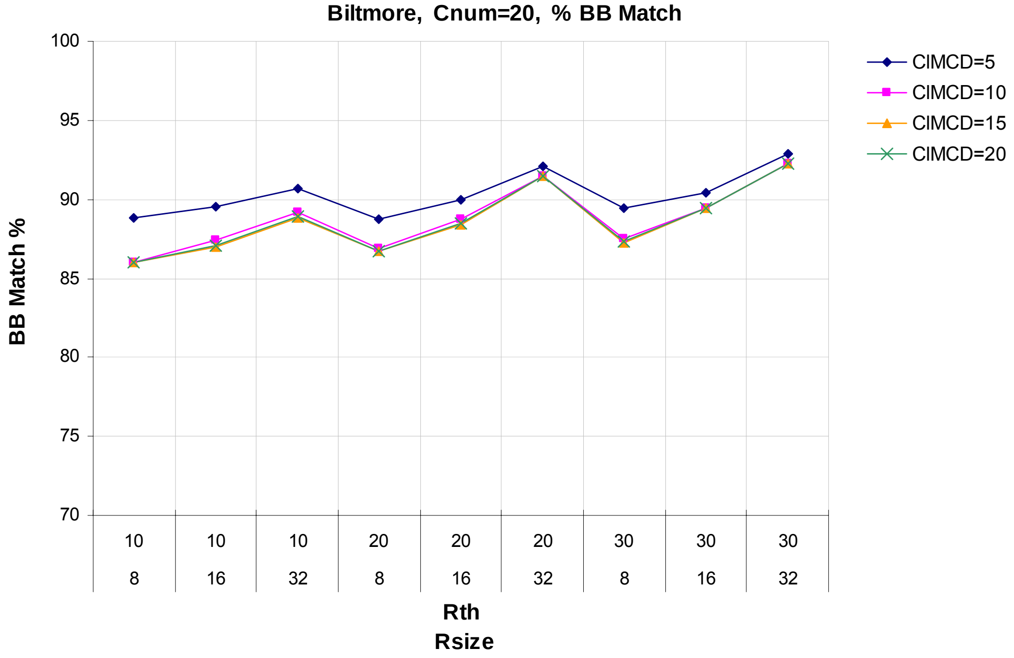


Figure 17. BigBackground stability in the Biltmore sequence as a function of CIMCD, Rth and Rsize.

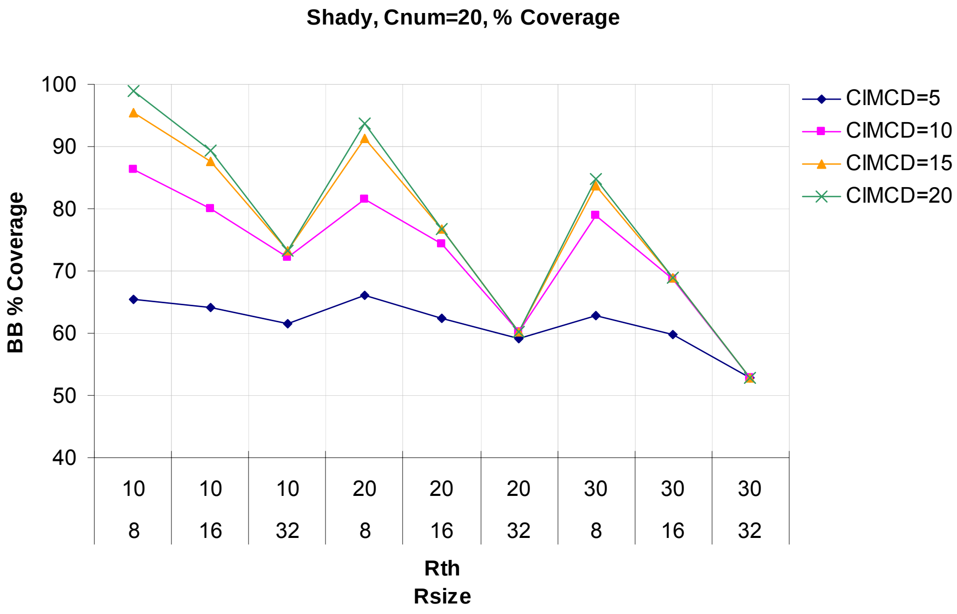


Figure 18. BigBackground coverage of the Shady sequence as a function of CIMCD, Rth and Rsize.

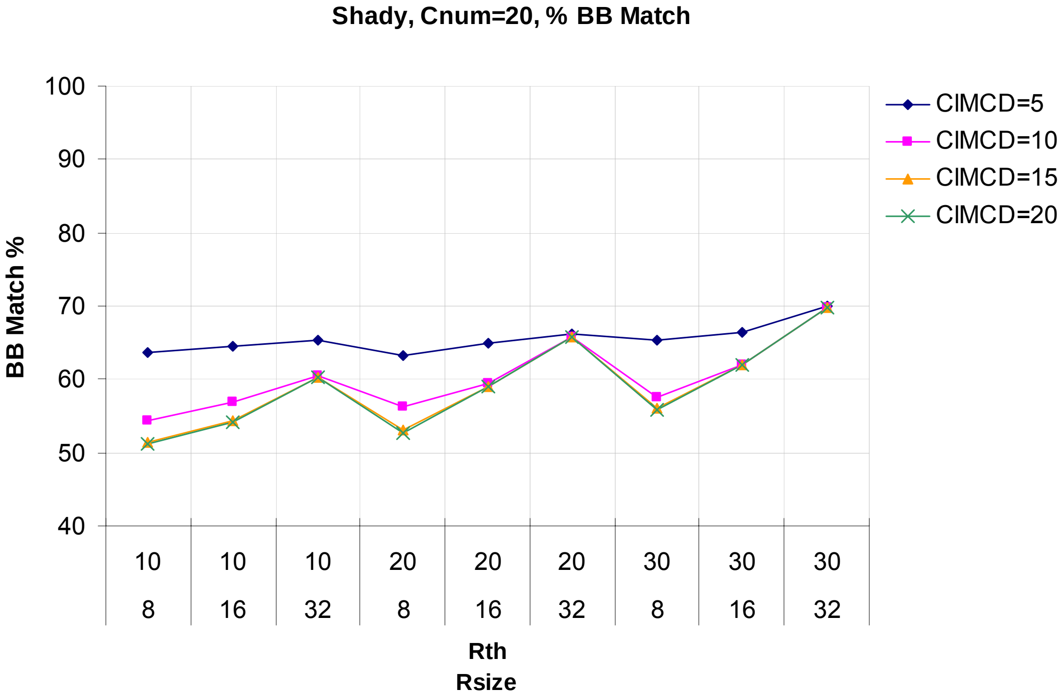


Figure 19. BigBackground stability in the Shady sequence as a function of CIMCD, Rth and Rsize.

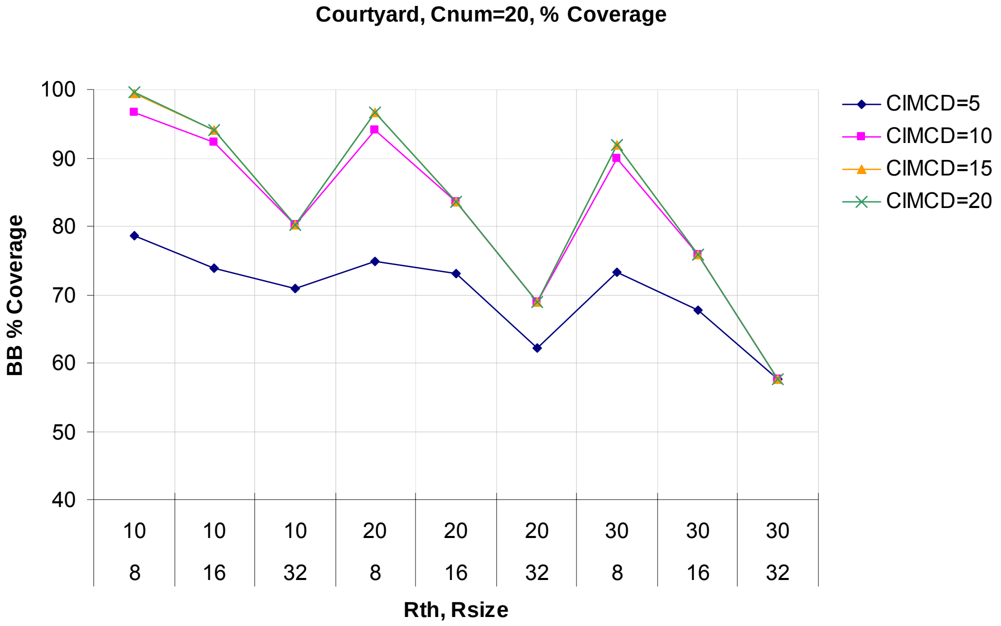


Figure 20. BigBackground coverage of the Courtyard sequence as a function of CIMCD, Rth and Rsize.

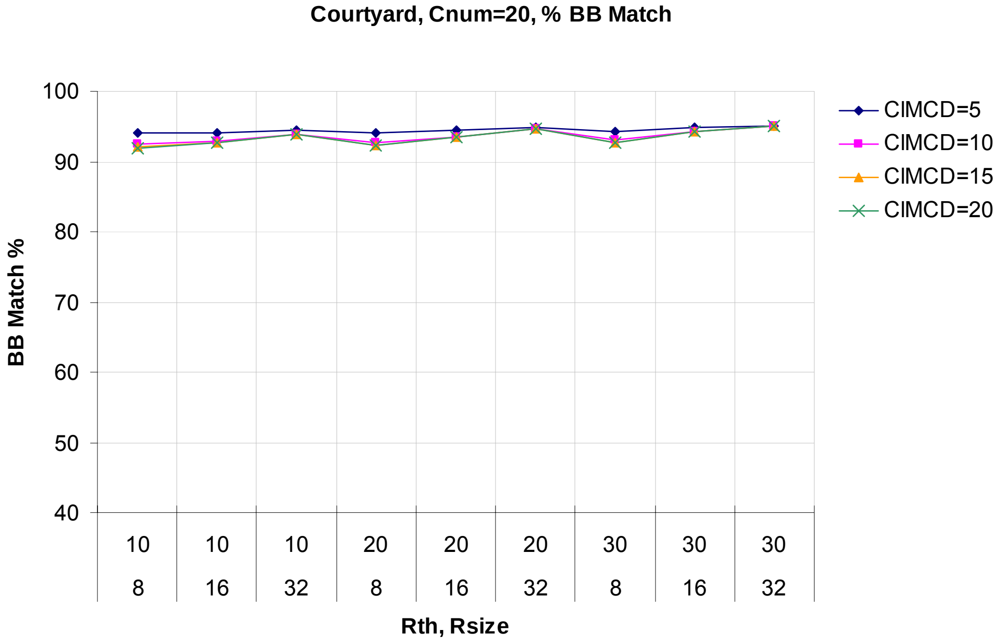


Figure 21. BigBackground stability in the Courtyard sequence as a function of CIMCD, Rth and Rsize.

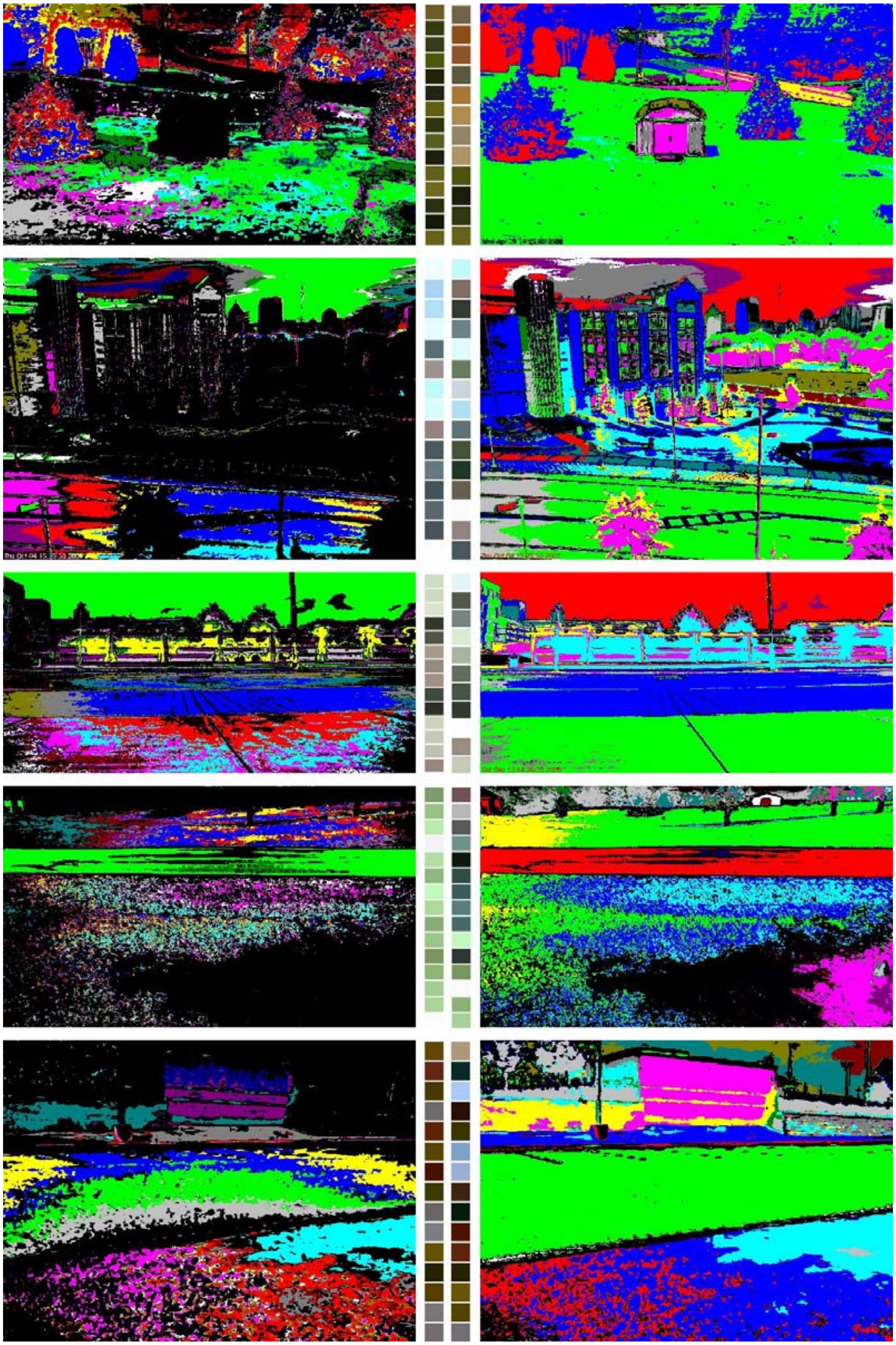


Figure 22. Identification of BigBackground regions using linked-list color palette. Left: Samples of Multimodal Mean predominant images from evaluation sequences. Left Center: Color palettes without clustering. Right Center: Average color palettes with clustering. Right: False-color BigBackground maps. Each color represents a different BB region. For these samples, Cnum=15, ClMCD=15, Rth=20, and Rsize=16.

3.6. Conclusions

While computers must begin to examine an image pixel by pixel, it is necessary to view a scene in terms of its content. An important step in computer scene understanding is the aggregation of individual pixels into high-level spatial abstractions. This chapter has presented a model for representing large, stable, self-consistent background features, and a method for identifying these features based on chromatic dominance. Parameters such as tile size, minimum color density, and palette length were tested for their effects on BigBackground coverage and stability. It was shown that after the initial color search, an additional clustering step that combines similar colors into a linked list structure significantly increases color palette diversity and coverage. The linked list color palette also provides a more logical way of increasing acceptable color matches than simply increasing the classification threshold. BigBackground was found to comprise 50% to 90% of several representative surveillance scenes. Pixels belonging to the BigBackground model were found to be 20% more stable on average than non-BigBackground pixels. In the next two chapters, the size and stability characteristics of BigBackground features are exploited to form color-centric approaches to illumination compensation.

CHAPTER 4

BIGBACKGROUND-BASED ILLUMINATION COMPENSATION

4.1. Introduction

Illumination variation in a scene is a challenge to most background models. As temporary cloud cover and artificial lights change a scene's illumination, background object pixels fail to match their background model counterparts, and are falsely interpreted as foreground. Such a surge in the number of foreground pixels often taxes downstream processes, as object tracking or recognition routines must sift through additional data. Salient features can be masked by surrounding background under new illumination conditions. Thus, real-time performance becomes harder to maintain, and analysis of the objects of interest becomes less accurate. It is desired for surveillance algorithms to monitor such scenes, reliably observe foreground objects, and filter out persistent background regardless of changes in illumination.

This chapter presents a computationally efficient technique that quantifies and compensates for lighting variations. This technique uses the concept of BigBackground, which identifies large, permanent background objects such as roads and buildings. The resulting BigBackground model is used as a calibration anchor to quickly, quantitatively estimate the effects of lighting changes on stable regions in the scene. These estimates are used to produce lighting compensation factors that can be applied to estimate the scene's appearance under the original lighting condition, and to extend the useful life of the background model without requiring complete reinitialization. Applying an illumination compensation technique based on BigBackground decreases average false positives by 83% compared to no corrective action, and decreases average false positives by 25% to 43% compared to competing compensation techniques from the literature. During run-time tests, the BigBackground-based algorithm performs among the fastest techniques tested at 15 to 20 frames per second.

4.2. Related Work

Several techniques have been explored for dealing with illumination changes in video analysis and image processing. Some involve direct compensation to improve image quality, while others simply recognize if two images are of the same scene. Online and offline learning systems have been employed. A wide range of models for representing illumination change have been described with several degrees of reliance on physical properties of light. The general goal is to transform an image of a new lighting condition (I_2) to match the illumination condition observed in an earlier image (I_1) while preserving the features of I_2.

The majority of techniques used to resolve illumination change problems rely on color information. Fundamental work is presented by Gros [19] as several linear and nonlinear transformation models are explored to account for illumination change. Static scenes are observed as illumination is varied in a controlled way, and a least median square algorithm chooses the best coefficient values for minimizing the error between the original image and the image of the scene under new illumination. For changes in illumination intensity (and not in spectral distribution), the multiplication of the RGB pixel vector by a single constant is shown to be sufficient to reduce most of the error, although adding a translation vector to the RGB vector (i.e., adding a scalar offset to each color component) is also fairly effective. Spectral changes in the light source require more complicated transformations to significantly reduce error, such as multiplication of the RGB vector by a full 3x3 matrix. Also, for the spectral change case, the addition of a translation vector decreases error better than multiplication by a constant. Since a spectral shift in the light source would cause each color channel to respond differently, the translation vector better accounts for such a change.

Experiments presented by Bales et al. [13] demonstrate that the effects of illumination change have a significant dependence on chromaticity. Color targets are subjected to controlled illumination changes, and several mathematical models are tested for how effectively they account for illumination changes. All models improve in effectiveness when tuned individually for each color. Furthermore, compensation parameters that are optimized for one color are found to remain effective when applied to other colors with similar hue, and rapidly lose effectiveness when applied to colors of dissimilar hue. These observations on the chromatic dependency of illumination change response motivate the illumination compensation approach presented here.

There are four general approaches to handling illumination change: illumination invariance, physics-based and photometric stereo modeling, local area statistics, and spatial correction. Illumination invariance methods attempt to formulate algorithms such that the data they process are not affected by illumination change. These algorithms typically use edges and gradients [30], [31] or chromaticity [28], [32], [33], [34] instead of raw RGB pixel values. Edges are derived from local features of a discrete approximation to the gradient field, which measures intensity differences between adjacent pixels. While edges are often present despite illumination levels, illumination changes can still affect the apparent strength of an edge, and a threshold mechanism must be used to determine which edges are significant [30]. Several color spaces are available that separate color from intensity information, such as YCbCr and HSV. Chromaticity values are calculated as the ratio of the intensity of a color channel to the total intensity of the pixel. Chromaticity is the simplest intensity-separating color space derived from RGB values. Special color spaces alone are typically insufficient for accounting for illumination change, because changes in light source spectrum alter the colors in the scene. They are instead used as components of more complex algorithms to provide cues about color stability.

Other approaches to illumination compensation estimate a scene's surface properties to estimate response to changing light conditions. Horn [35] describes in detail the concepts of photometric stereo, in which a scene is decomposed into irradiance and reflectance components. The irradiance map is generally assumed to be a smooth function, and high-frequency features are assumed to be caused by changes in reflectance between objects. The primary drawback to these techniques is the requirement of a controlled calibration mechanism with well-defined relationships between different illumination conditions. Calibration often must be computed ahead of time offline, and the photometric stereo process is computationally expensive. Weiss et al. [36] extract intrinsic (reflectance and illumination) images by assuming a constant scene reflectance and time-varying illumination, and by approaching the problem as one of maximum likelihood estimation. In [37], Hager and Belhumeur assume a Lambertian scene and use at least three images of a scene under linearly-independent light source directions, from which albedo and surface normal can be computed. This set of basis images can be linearly combined to depict the scene under a new illumination condition. In [38], shadows are removed from images by computing background images for the same scene over different periods of time, and then decomposing these backgrounds into a reflectance image and a set of illumination images. The technique presented by Wu et al. [39] first estimate the camera's response function by observing the scene with several different camera exposure times. Then online, two or more images of the scene are captured using different known exposure settings. Assuming the scene's illumination stays constant during capture, the images with different exposures can be distilled into radiance maps using the estimated camera response function, and then fused to form an image that is less sensitive to lighting fluctuation.

A data-driven approach by Miller et al. [40] uses a large set of images of a control color palette under varying lighting conditions to learn color space response. Principal component analysis is used to derive the most statistically significant color transform pairs. This approach does not model lighting transformation directly; instead, the resulting color eigenflows are used to test if two images are of the same scene under different illumination. This seems to be a useful aid in scene recognition, but results are not given for response to occlusion. True illumination compensation is also avoided by Finlayson et al. [41], where instead the goal is to identify which image regions are illuminated by the same light sources. An algorithm is presented to use images from a chromagenic camera, which takes two images of a scene—one normally and one through a colored filter. A physics-based reflection model is used by Makihara et al. [42] to estimate color transformations starting with a single reference color. Because it is difficult to automatically obtain reference colors from unknown lighting conditions, the proposed method uses human interaction to learn color transformations. Upon finding a new color pair, the algorithm updates a color transformation matrix and the algorithm repeats. If the transformation does not successfully match an object in the scene with a reference texture image of the object, human intervention is required to facilitate the match, and the transformation model is updated with the new color pair.

Statistics computed locally about individual pixels provide a computationally inexpensive approach to illumination compensation. In [43], Young et al. propose two compensation models. In the first (called the first-order model), the average intensity is calculated for a window centered about each pixel in I_1 and I_2. The size of the window influences how large and small features are compensated differently, and generally ranges from 3 x 3 to 31 x 31 pixel squares. In the second model (called the second-order model), both the local averages and standard deviations are used. The first and second-order models are given in Equations 6 and 7, respectively. Here, $\overline{I_1}$ and $\overline{I_2}$ represent the mean pixel value within the window centered about (x, y) in the original image and in the image being compensated. The standard deviations for the same windows are given by σ₁ and σ₂. This second-order model is also proposed by Lu et al. [44] for block-based illumination compensation in multi-view video coding. Instead of computing statistics for windows centered about each pixel, the statistics are computed for each fixed-size macroblock, and then applied to all of the pixels within that macroblock to reduce computational cost.

$$I_2(x,y)' = \frac{\overline{I_1}}{\overline{I_2}} I_2(x,y) \tag{6}$$

$$I_2(x,y)' = \left(I_2(x,y) - \overline{I_2}\right)\left(\frac{\sigma_2}{\sigma_1}\right) + \overline{I_1} \tag{7}$$

A third method based on local statistics is presented by Kamikura et al. [45] with the intended application of illumination compensation for video coding. The approach can also be applied to compensating surveillance video, and the motion estimation factors can be omitted. A pixel-wise affine transformation is used of the type shown in Equation 8. The gain and offset parameters (c and d, respectively) are chosen to minimize mean square error, with the optimal solution given by Equation 9. The statistics used to calculate c and d are given in Equation 10. Again, I_1 and I_2 are the originally illuminated image and the currently illuminated image, respectively, while R is the region over which the statistics are calculated and N is the number of pixels in region R. A single (c, d) pair for the entire image is chosen by calculating pairs for all of the tiles in the image, and choosing the pair that occurs most frequently.

An illumination change is modeled by observing changes in the pixels of an image, but such changes can be caused by either illumination or occluding objects. The drawback to local area statistics techniques is the implicit assumption that each pixel in I_2 should match its corresponding pixel in I_1. These methods make no distinction between persistent background pixels or pixels belonging to occluding objects, so the resulting compensations tend to drive all pixels towards their appearance in image I_1. Also, tiles or regions that contain significantly different surfaces can result in averages that fail to properly compensate either surface.

$$I_2^{'}(x,y) = cI_2(x,y) + d \tag{8}$$

$$\begin{aligned} c &= \frac{N \cdot S - P \cdot Q}{N \cdot T - P^2} \\ d &= \frac{T \cdot Q - P \cdot S}{N \cdot T - P^2} \end{aligned} \tag{9}$$

$$\begin{aligned} P &= \sum_{x,y \in R} I_1(x,y) \\ Q &= \sum_{x,y \in R} I_2(x,y) \\ S &= \sum_{x,y \in R} I_1(x,y) \cdot I_2(x,y) \\ T &= \sum_{x,y \in R} I_2^{\,2}(x,y) \end{aligned} \tag{10}$$

The technique described by Suau et al. [46] also uses first and second order statistics, but computes these statistics over multiple tile resolutions and fuses the results. For each tile resolution, the image is divided into equal numbers of horizontal and vertical tiles, and the mean and variance of each tile's luminance channel are computed. Bilinear interpolation expands these statistics into matrices with the original image's dimensions. The original image is then mean-variance normalized toward a target illumination average and standard deviation level, as given in Equation 11, where Y is the original luminance channel, Y' is the compensated luminance channel, L is the number of resolutions, M_k is the bilinearly interpolated mean image for resolution k, V_k is the bilinearly interpolated variance image for resolution k, and μ_0 and σ_0 are the target mean and standard deviation levels, respectively. This approach is presented as a preprocessing step and is used in conjunction with a Mixture of Gaussians background model. Rather than compensate a new image to more closely resemble an image depicting the original illumination condition, all images are normalized toward a preset ideal illumination condition defined by μ_0 and σ_0. The multiresolution aspect of the method reduces the technique's sensitivity to tile size selection, and it is stated that all of the resolutions used must be larger than the objects of interest in the observed scene. However, the extra passes required for each resolution and the bilinear interpolation steps significantly increase the computational complexity of the process. Because only the intensity channel is compensated, there is not a mechanism for handling changes in light source spectra.

$$Y' = \frac{1}{L}\sum_{k=1}^{L}\left[\left(Y - M_k\right)\cdot\frac{\sigma_o}{1+\sqrt{V_k}} + \mu_0\right] \tag{11}$$

Another illumination compensation method that exploits local area statistics is given by Vijverberg et al. [47]. Rather than compensate the image directly, this technique uses histogram analysis to tailor the thresholds used for foreground/background classification. Laplacian, Gaussian, and two-component Gaussian models are considered, and the model is chosen that best describes the distribution of background difference pixels (the difference between the background image and the current image). The mean μ_k and standard deviation σ_k from the best-fitting model are used to derive the classification threshold of Equation 12, where k denotes the component of a multimodal background model, $\Delta(x,y)$ is the background difference, and $T_k=\mathrm{MAX}(T_{min}, 1.5\sigma_k)$. This technique is intended for global illumination changes, and, as described, does not include a mechanism for handling partial changes.

$$F(x,y) = \begin{cases} foreground & \text{if } \Delta(\mathrm{x},\mathrm{y}) - \mu_k > T_k \forall k \\ background & \text{else} \end{cases} \tag{12}$$

Spatial correction methods attempt to adjust localized lighting effects to achieve a smoother, more balanced reflectance function. Skin tone is a commonly exploited reference color for balancing illumination across faces to improve facial recognition [48], [49], [50]. Block-based histogram equalization is used in [48] to improve contrast, followed by the categorization of the type of illumination present in the scene. Then an illumination compensation model is applied that corresponds to that lighting condition. Skin color distributions are studied in [49] under several lighting conditions in the YCrCb color space, and proposes a correction to red component saturation to improve skin color segmentation in strong light. In [50], skin color is identified in faces in the first frame. This color is used to track humans in the remaining video. The appearance of skin under new illumination conditions is compensated for by the application of a skin reflectance model, which consists of the reflectance coefficient of skin as a function of incident light wavelength.

The work in [51], [52], and [38] focuses on correcting particular types of illumination variation. Static glare removal is considered in [51]. Grayscale background images are computed as the median of a window of 10 frames. Background differencing is used between the current and previous background models. The algorithm identifies regions which have increased in brightness and are brighter than the average grayscale value of the image. Pixels that meet these requirements are classified as static glare. The technique presented in [52] uses separate daytime and nighttime background models of a scene. These images are segmented based on illumination and motion, and the results are fused to produce an illumination-enhanced night image in which the effects of artificial lights are reduced. Principal component analysis is used in [38] to estimate the illumination image and time-varying reflectance images of a scene. The estimated components are then used to cancel out illumination changes. The illumination image is computed from the input image directly rather than from a background model to avoid detecting transient shadows as movement. These techniques address spatial intra-frame illumination variation rather than time-varying inter-frame variation. The goal for intra-frame correction is to modify regions within a scene to make it appear more uniformly illuminated by a distant, diffuse source, removing artifacts that result from the locations and orientations of objects with respect to the light source. Inter-frame compensations maintain constancy during temporal lighting changes.

4.3. Approach

As discussed by Horn [35], an observed scene radiance F is proportional to the product of the reflectance R of the scene surfaces and the irradiance E striking the scene (13).

$$F(x, y) = E(x, y) \cdot R(x, y) \qquad \textbf{(13)}$$

These functions generally depend on the point (x,y) being considered in the scene and on the relative angles between the light source, the surface, and the observer as in (14). Here, the general change in the appearance of a scene F due to a change in illumination is described as functions of changes in intensity (I), spectrum (ω), or incident and observed azimuth (θ_i, θ_o) and elevation (φ_i, φ_o) with respect to the scene.

$$\Delta F(x, y, \theta_o, \varphi_o) = \Delta E(I, \omega, \theta_i, \varphi_i) \cdot R(x, y) \qquad \textbf{(14)}$$

The observer (the camera) is stationary for most surveillance applications, and because we are concerned only with internal illumination changes we assume that the light source is also stationary (or is moving sufficiently slowly that it can be approximated as stationary). Therefore, the dependence of the scene change on changes in the relative positions of the source and observer can be removed (15). Changes in the observed scene S are then due to changes in illumination intensity and spectrum applied to the reflectance of the scene, which accounts for the surface properties of each point (including color).

$$\Delta F(x,y) = \Delta E(I,\omega) \cdot R(x,y) \tag{15}$$

Compensation methods that rely on local pixel statistics tend to base a pixel's compensation on the average change of nearby pixels within a window. This approach approximates all surfaces as responding equally to a given illumination change and takes the form of (16), or (17) if the model has separate parameters for each color channel to accommodate spectrum changes. Here the average illumination change is taken over the window or region containing the pixel (x,y) being compensated.

$$\Delta F(x,y) = \overline{\Delta E(I)} \tag{16}$$

$$\Delta F(x,y) = \overline{\Delta E(I,\omega)} \tag{17}$$

Experiments presented in [13] demonstrate the dependence of illumination change response on surface color. Surfaces of significantly different color are shown to respond differently to the same illumination change, and compensation models are shown to be much more effective when optimized for individual colors. This data suggests that compensation models of the form of (16) and (17) may work acceptably for regions comprised of the same surface, but will begin to fail for windows containing dissimilar surfaces (particularly dissimilar hues). We propose an illumination compensation model in which scene changes depend on intensity and spectrum changes as well as surface color C, as defined in (18). The C term represents only surface color, which is a significant but easier to compute component than a full description of a surface's reflectance properties.

$$\Delta F(x, y) = \overline{\Delta E(I, \omega)} \cdot \overline{C(x, y)} \tag{18}$$

4.4. BigBackground as an Illumination Anchor

The results in the preceding chapter show that BB can be relied upon as a relatively stable set of pixels. Here it is demonstrated that this characteristic can be used as a point of reference for calculating illumination changes. As illumination changes in a scene, the new values of BB pixels can be compared with their values from the BB color palette and the average effect on each of those color regions can be quantified.

The general approach is to detect prevalent changes in the BB regions of an image, identify which changes are due to changes in illumination, and from that data, formulate a mathematical operation that will transform pixels from the current image into something closer to what was observed in images before the lighting change.

The most computationally efficient means of observing changes in a BB region is to find the region's new average pixel value. A foreground object that obscures part of a background region would tend to pull the region's average color away from the true background color. Therefore, one of the conditions in Equation 19 must be met before a BB pixel is allowed to contribute to its region's new average color, where H_x and S_x denote the pixel's Hue and Saturation, respectively. The subscript x can take on values of 1 to denote a pixel in the original lighting condition, or 2 to denote a pixel in the new lighting condition being compensated.

$$\begin{gathered} S_1 > 12\% \ \text{AND}\, S_2 > 12\% \ \text{AND} \left|H_1 - H_2\right| < 8\% \\ S_1 > 12\% \ \text{AND}\, S_2 > 12\% \ \text{AND} \left|S_1 - S_2\right| < 8\% \\ S_1 < 12\% \ \text{AND}\, S_2 < 12\% \end{gathered} \tag{19}$$

The thresholds used in these rules were determined empirically, and are constant for all sequences. If only hue comparisons are made without the saturation condition, many false mismatches were encountered in dark and gray-colored regions, such as asphalt and concrete. If one of these conditions is met, there is a reasonable probability that the pixel indeed represents the same BB region, but is being observed under different illumination. In that case, the pixel's RGB values contribute to the region's new average color. If the hues or saturations are too different, the pixel is considered likely to be a temporary occluding object, and does not contribute to the region's average color. To summarize, all of the pixels that belong to the same BB region and that satisfy one of the rules in Equation 19 are averaged together, thereby computing the BB region's new average color under a potentially new lighting condition.

Knowing each BigBackground region's original appearance (under the original lighting condition) and new color (under a new lighting condition) forms the basis for our approach to illumination compensation. For each BB region, the parameters for a compensation model are independently computed. The sequences used for evaluation, address the issue of local lighting effects are presented next, and four possible compensation models are evaluated.

4.5. Video Test Set

Ten image sequences featuring significant occlusion and illumination change were captured to evaluate the algorithms discussed. Sequences were captured using off-the-shelf USB webcameras at 30 frames per second with 640 x 480 pixel resolution. Table 12 describes the important features of the test sequences used. Samples of the video sequences are shown in Figures 23 and 24. The first column shows the scene before a lighting change, the second column shows the scene after a lighting change, and the third column shows the desired ground truth images of ideal foreground/background segmentation. The ground truth images are generated by hand, where white pixels represent ideal foreground and black pixels represent ideal background.

Table 12. Properties of Test Sequences with Illumination Change

Sequence	Lighting Change	Foreground Objects	Background Behavior
Backyard1	Large	Distant vehicles	Rural, yard, treeline
Backyard2	Large	Distant vehicles	Rural, yard, treeline
Ford1	Partial, large	Mid-range vehicles	Urban, buildings
Ford2	Partial, small	Mid-range vehicles	Urban, buildings
RecCenter	Very small	Mid-range pedestrian	Indoors, desaturated
TechSquare1	Small	Mid-range vehicles	Urban, buildings
TechSquare2	Small	Mid-range vehicles	Urban, buildings
Roadside	Moderate	Close vehicles	Rural, field, treeline
Bank	Large	Close vehicles	Building, parking lot
ParkingLot	Moderate	Distant vehicles	Parking lot, desaturated

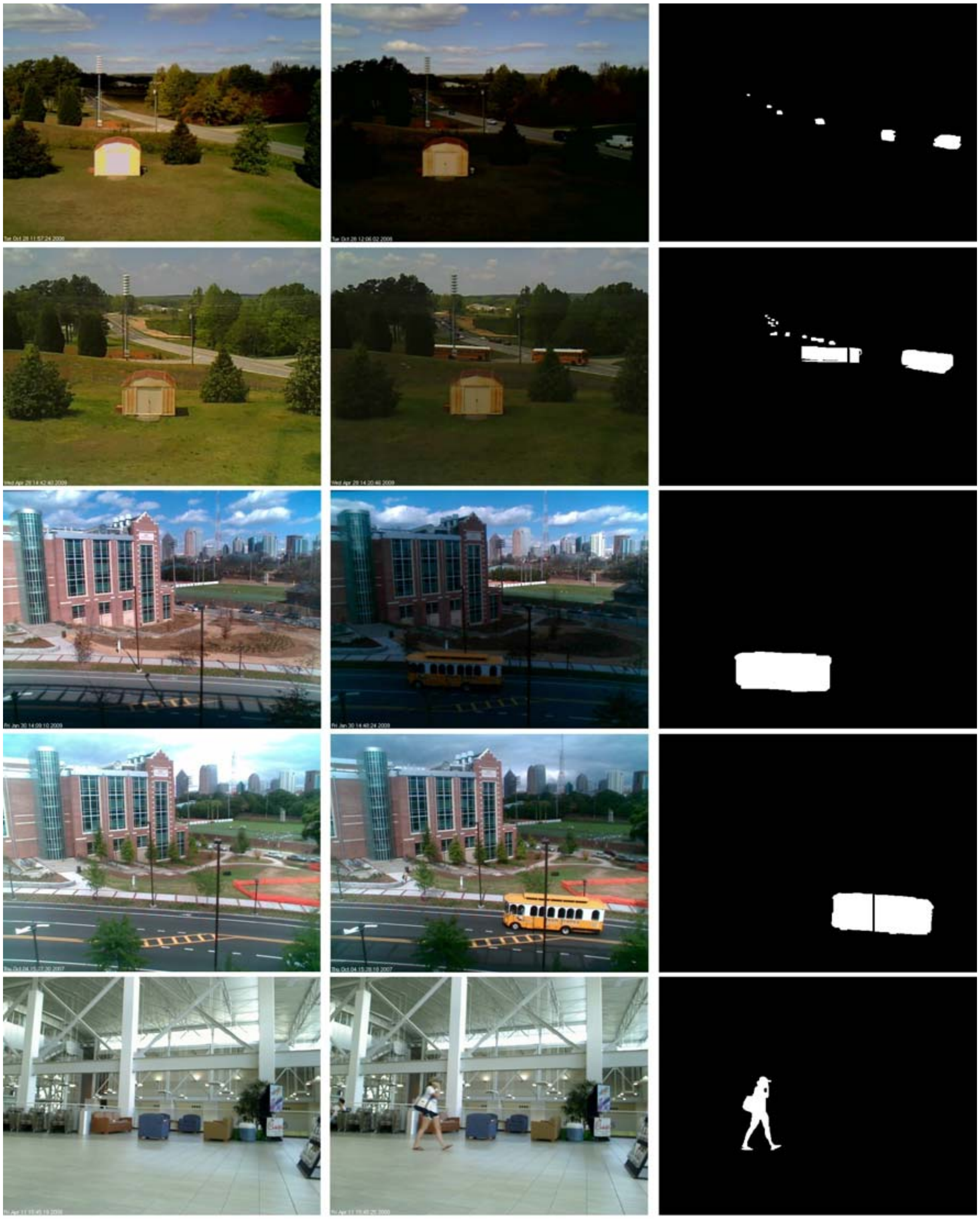


Figure 23. Samples 1-5 of illumination change sequences used for ground truth evaluation. Left to right: Initial image, post-lighting change image, and hand-marked ground-truth image from the sequences used to evaluate illumination compensation methods. Top to bottom: Backyard1, Backyard2, Ford1, Ford2, and RecCenter.

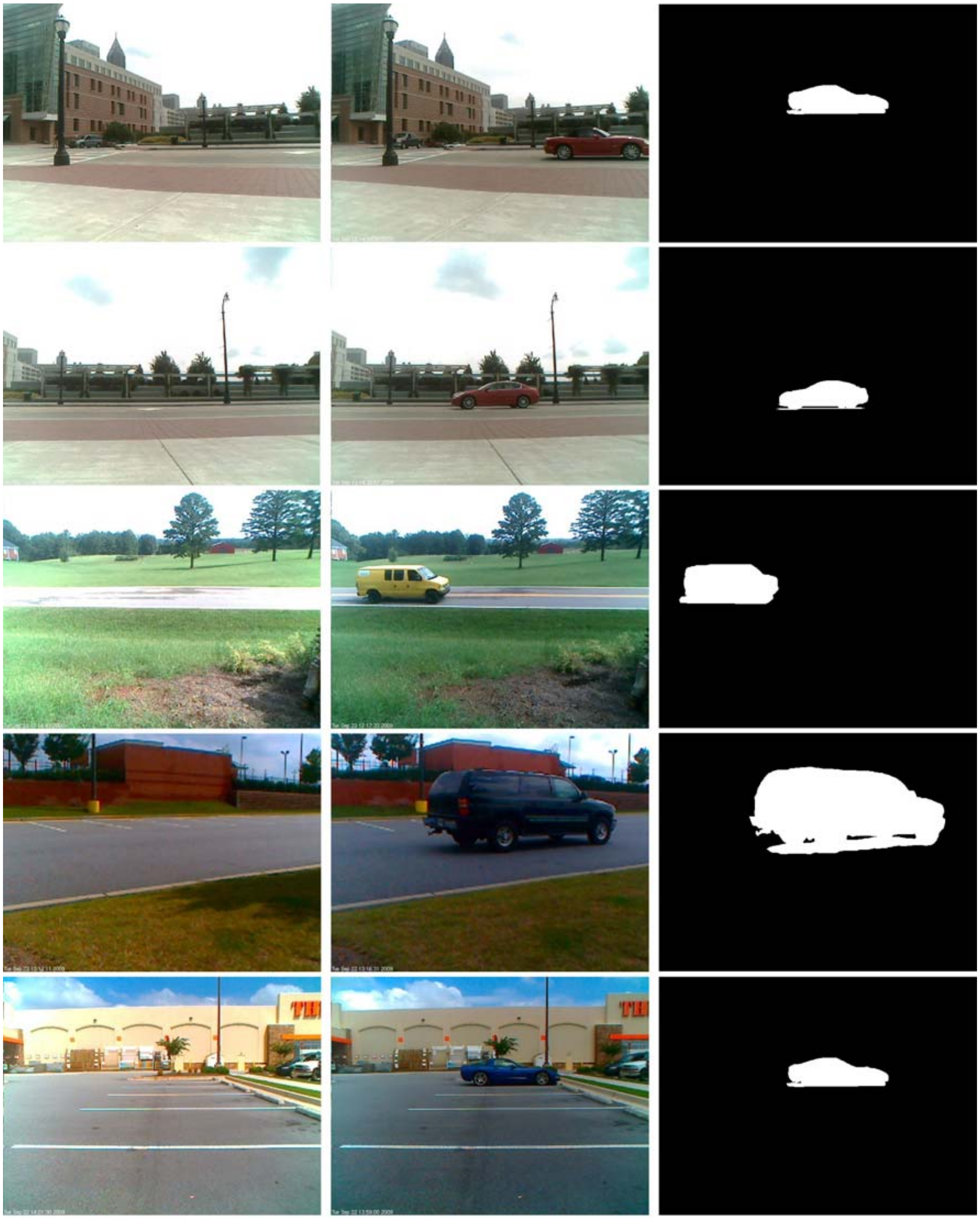


Figure 24. Samples 6-10 of illumination change sequences used for ground truth evaluation. Left to right: Initial image, post-lighting change image, and hand-marked ground-truth image from the sequences used to evaluate illumination compensation methods. Top to bottom: TechSquare1, TechSquare2, Roadside, Bank, and ParkingLot.

4.6. Local Illumination Changes

Some video scenes feature global lighting changes, in which the entire visible scene responds to new illumination fairly evenly. However, in many cases, rolling cloud cover or small lamps being turned off and on result in local lighting changes. Sources of illumination might vary with intensity across a scene, resulting in different lighting transformations from one side of the scene to the other. Different objects with similar surfaces may be oriented differently with respect to the light source, thereby responding differently to the same lighting change. To better accommodate these cases, the image is segmented into square tiles. New region averages are then computed for each BB region that appears within each tile. Each tile is treated independently, so the transformations within each tile are the best fit for the local lighting conditions. As observed in the Relative Operating Characteristic (ROC) plot in Figure 25, larger tile sizes (up to the limit of treating the entire image as one large tile) tend to lead to higher false positive rates, incidentally increasing the true positive rates as more pixels are classified as foreground. Data points corresponding to smaller tile sizes tend toward the left axis of the plot. This effect is more visible in the sample images of Figure 26. The Ford2 sequence features a sharp partial illumination change in the back half of the image. Global compensation without tiling results in large portions of false foreground. The Techsquare1 sequence features an illumination change that is more spatially uniform, so while tiling does improve quality in some areas, the effect is less pronounced. Using a very small tile size can result in aperture artifacts: such a small portion of the image is examined at one time that even occluding objects are driven into the background. For the scene types tested here, a tile size of 32 pixels per side responds well to local effects without masking occlusions irrecoverably. The optimal value will depend somewhat on the relative sizes of the objects of interest.

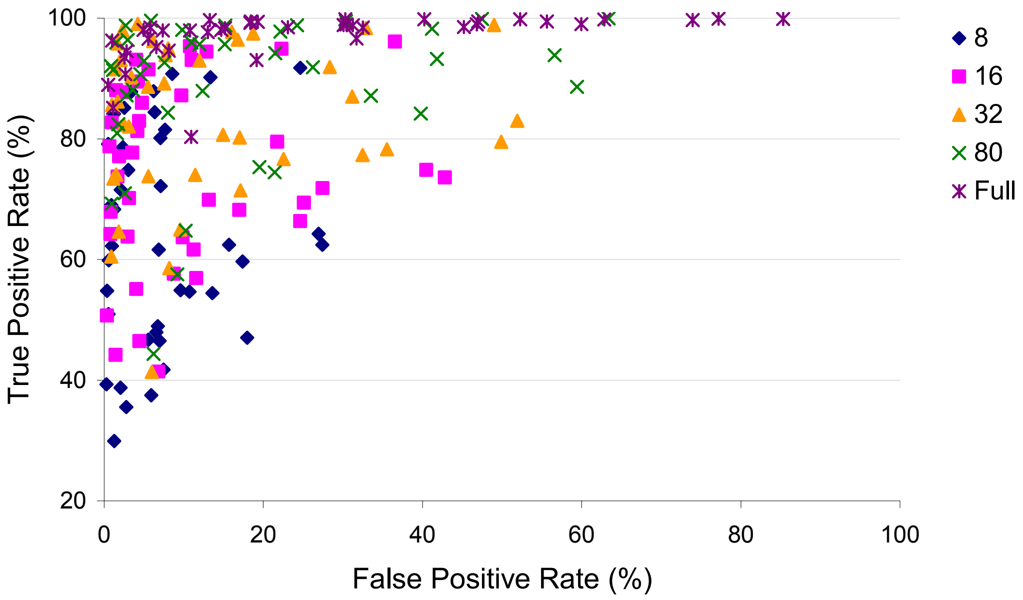


Figure 25. An ROC plot showing the effects of different tile sizes (8, 16, 32, and 80 pixels per side) on foreground/background classification accuracy. The data points shown were computed from ten video sequences using classification thresholds of 5, 7, 10, and 15.

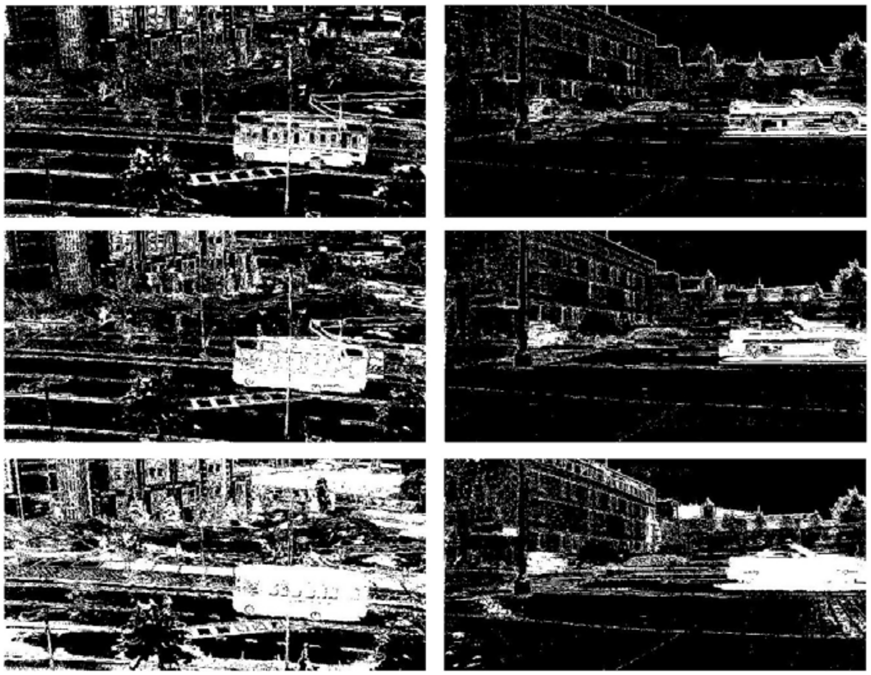


Figure 26. Sample images from the Ford2 (left) and TechSquare1 (right) test sequences, illustrating classification results when the illumination compensation tile size is 8 pixels per side (top), 32 pixels per side (middle), and full image (bottom). For these sequences, Cnum=15, Rsize=16, Rth=10, and MCDth=7.

4.7. Illumination Model Selection

As described in [19], several transformation models are available for dealing with illumination change. This section experimentally determines which model is most effective for exploiting BB features to perform illumination compensation. In this investigation, four transformation models are explored for accuracy and consistency. These models are chosen as the best balance between accuracy and computational cost. More complex models tested in [18] and [19] are able to account for the illumination changes of more pixels, but the increase in accuracy is small compared to the extra number of operations required. Of the four methods examined here, the first method treats illumination changes purely as translation operations, and computes the difference between the current BB region average and the original BB region color. The second method calculates the ratio of the original region color to the new average color. This treats illumination change as a gain operation, and applies a multiplier to member pixels of a given BB region. The third method, like the first, is translation-based, but the original colors and the averages are first converted to HSI space where the differences are calculated. Correcting pixels in a new image thus requires converting that pixel to HSI space, adding the corresponding region's average HSI differences, and converting back to RGB. The fourth method is similar in form to the second, but again operates in HSI space. Pixels and region averages are converted to HSI, the ratios between the new region averages and the original region averages are calculated and applied to each pixel, and the result is converted back to RGB space. The derivations of the models' parameters from BB are summarized in Table 13, where the subscript i denotes the BigBackground region under consideration; R, G, and B denote the color channels of the BB average color regions; and α, β, and γ denote the compensation parameters for each color channel. The subsequent application of these models to pixels belonging to BB is shown in Table 14, where the subscript i again denotes the BB region to which the pixel and compensation parameters belong. Non-BB pixels are compensated with the parameters corresponding to the BB region with the hue that best matches the pixel.

Table 13. Computing illumination compensation models parameters.

RGB Translation	$\alpha_i = R_{i,orig} - R_{i,new}$ $\beta_i = G_{i,orig} - G_{i,new}$ $\gamma_i = B_{i,orig} - B_{i,new}$
RGB Gain	$\alpha_i = R_{i,orig} / R_{i,new}$ $\beta_i = G_{i,orig} / G_{i,new}$ $\gamma_i = B_{i,orig} / B_{i,new}$
HSI Translation	$\alpha_i = H_{i,orig} - H_{i,new}$ $\beta_i = S_{i,orig} - S_{i,new}$ $\gamma_i = I_{i,orig} - I_{i,new}$
HSI Gain	$\alpha_i = H_{i,orig} / H_{i,new}$ $\beta_i = S_{i,orig} / S_{i,new}$ $\gamma_i = I_{i,orig} / I_{i,new}$

Table 14. Application of illumination compensation models to pixels belonging to BB.

RGB Translation	$P_i' = (\alpha_i+R, \beta_i+G, \gamma_i+B) = P_{RGB} + T_{RGB}$
RGB Gain	$P_i' = (\alpha_i R, \beta_i G, \gamma_i B) = D^* P_{RGB}$
HSI Translation	$P_i' = (\alpha_i+H, \beta_i+S, \gamma_i+I) = P_{HSI} + T_{HSI}$
HSI Gain	$P_i' = (\alpha_i H, \beta_i S, \gamma_i I) = D^* P_{HSI}$

The four compensation models are evaluated in terms of how well image I_2 is segmented into foreground and background after the compensation is applied. A representative image that contains a significant amount of foreground occlusion is chosen from each sequence, and a corresponding ground truth image is generated by hand-labeling the proper classification of each pixel. The automatically segmented compensated image is compared with the ground truth. Pixels that are labeled as background in the ground truth image and as foreground in the segmented image are counted as false positives. Pixels that are labeled as foreground in the ground truth image and as background in the segmented image are counted as false negatives. These are converted to percentages by dividing the false positives by the true number of background pixels, and dividing the false negatives by the true number of foreground pixels. An ROC curve is presented in Figure 27 which shows the false positive rate on the x-axis, and the true positive rate on the y-axis. To produce the data points for this curve, each of the compensation models is tested using all combinations of tile sizes {8, 16, 32, 80} and maximum component difference thresholds {5, 7, 10, 15}. Of the four models, the RGB Translation model most consistently yields the lowest false positive rates, which are the artifacts that must be minimized during illumination changes. The behavior of each model can be observed in some sample scenes in Figure 28. False positive and false negative rates are shown in Figures 29 and 30, respectively; these plots show the relative performance of the models for parameters set to Cnum=15, Rsize=32, Rth=20, MCDth=7, and tile size=16. In the Ford2 and Roadside sequences in particular, one can observe how the gain and HSI models leave considerably more foreground noise than the RGB Translation model. Due to the averaged nature of how compensations are computed from the BB color regions, the multiplicative models tend to overcompensate many pixels during significant lighting changes. Small variations in the pixels being transformed are amplified outside the range of the classification threshold. Also, compensations performed in the HSI color space occasionally suffer from hue and saturation artifacts, and produce exaggerated colorizations of some tiles that could cause problems in downstream processes. Therefore RGB Translation is used in the remaining experiments.

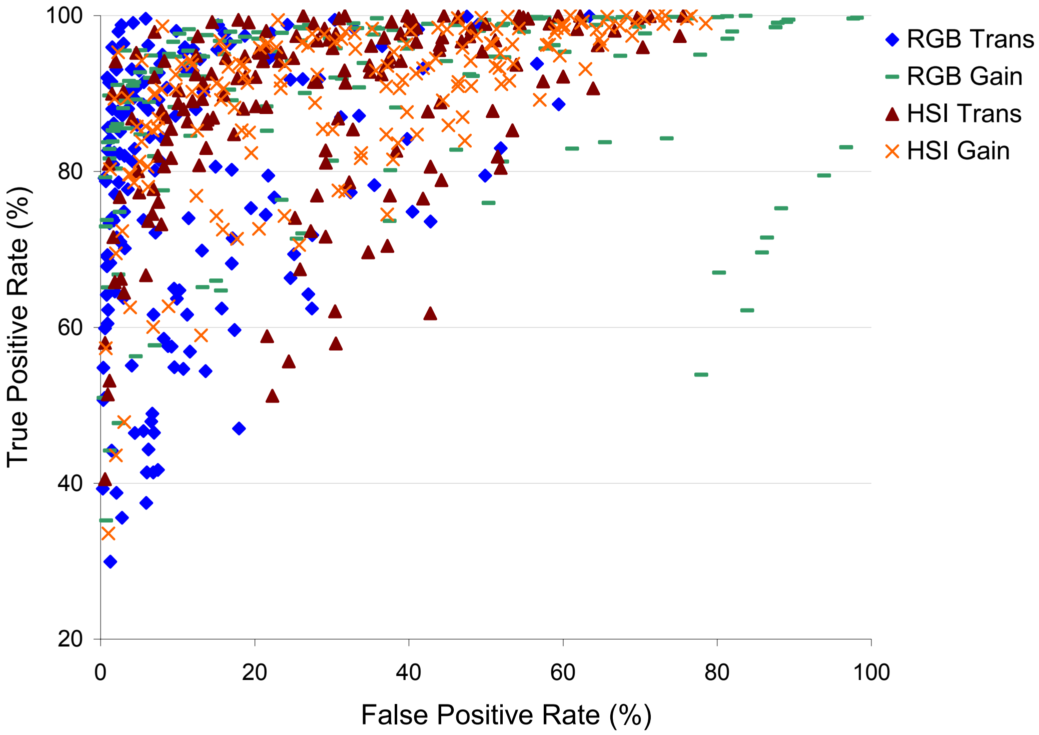


Figure 27. ROC plot for four mathematical models for illumination compensation. The data points shown were computed from ten test sequences, sweeping MCDth over {5, 7, 10, 15} and tile size over {8, 16, 32, 80}. The RGB Translation model stays more consistently concentrated in the low false positive range, which are the errors that must be minimized during illumination changes.

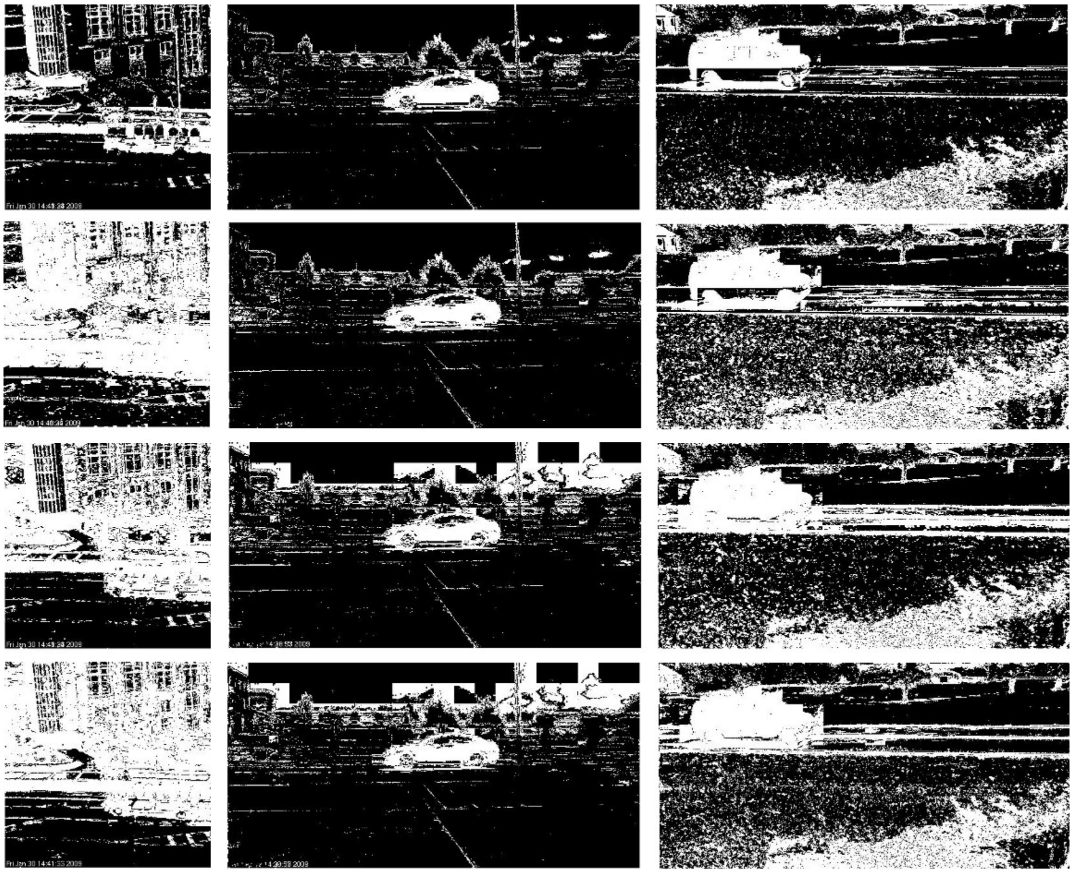


Figure 28. Sample images of foreground/background classification for four illumination compensation models. Left to right: Ford2, Techsquare2, and Roadside sequences. Top to bottom: RGB Translation, RGB Gain, HSI Translation, and HSI Gain. These samples were processed with Cnum=15, Rsize=16, Rth=10, MCDth=7.

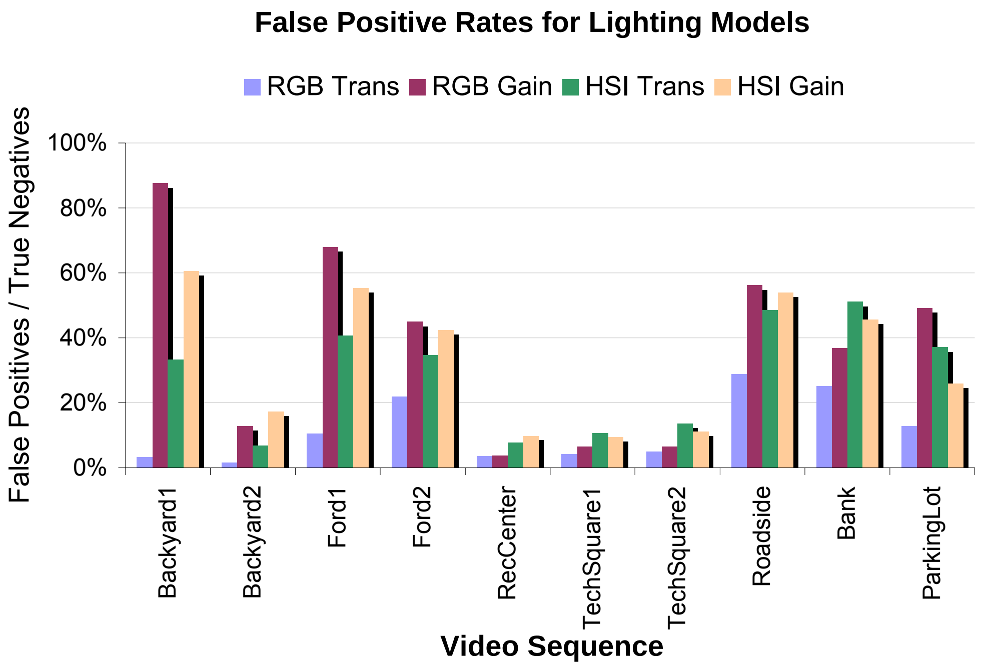


Figure 29. Comparison of four compensation models in terms of false positives / true negatives. The RGB Translation technique consistently achieves the lowest false positive rate.

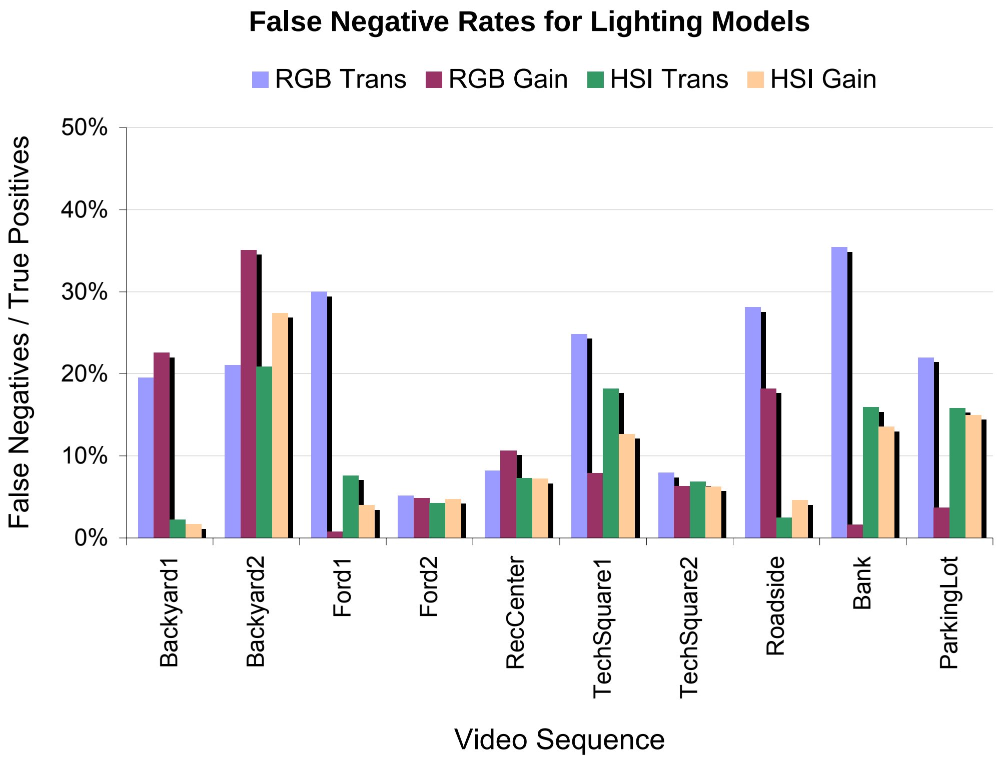


Figure 30. Comparison of four compensation models in terms of false negatives / true positives.

4.8. Comparison to Other Methods

In this section, the BB-based illumination compensation method is compared to the illumination compensation techniques described in [43], [44], [45], and [46] in terms of accuracy and execution time. These techniques are chosen for comparison because they are of similar structure and complexity to the novel approaches presented. They do not require extensive calibration, do not rely on assumptions about the scene environment or light sources, and are not designed around models for compensating specific targets (such as faces). Data used for compensation is extracted directly from pixels near the regions of interest.

4.8.1 Accuracy Comparison

In this experiment, the settings shown in Table 15 are used for the BB process. To make results more comparable, the competing methods are coded to calculate correction statistics for each fixed tile, and to apply those corrections to all pixels in the tile rather than recalculating statistics for a new window centered about each pixel. True and false positive rates are used for method evaluation, and are shown succinctly in the ROC plots of Figure 31. Data points on the ROC plot are generated by applying the five described compensation methods to 10 video sequences, and sweeping maximum component difference thresholds and tile sizes through several combinations (MCD = 5, 7, 10, 15; tile size = 8, 16, 32, 80). The top plot of Figure 31 compares the proposed method with the 1st-order and 2nd-order techniques. The bottom plot compares the proposed method with the MinMSE and multiresolution techniques.

Table 15. Settings used when comparing BB compensation to other methods. These parameters do not affect the other compensation methods.

Cnum	15
CIMCD	15
Rth	20
Rsize	32

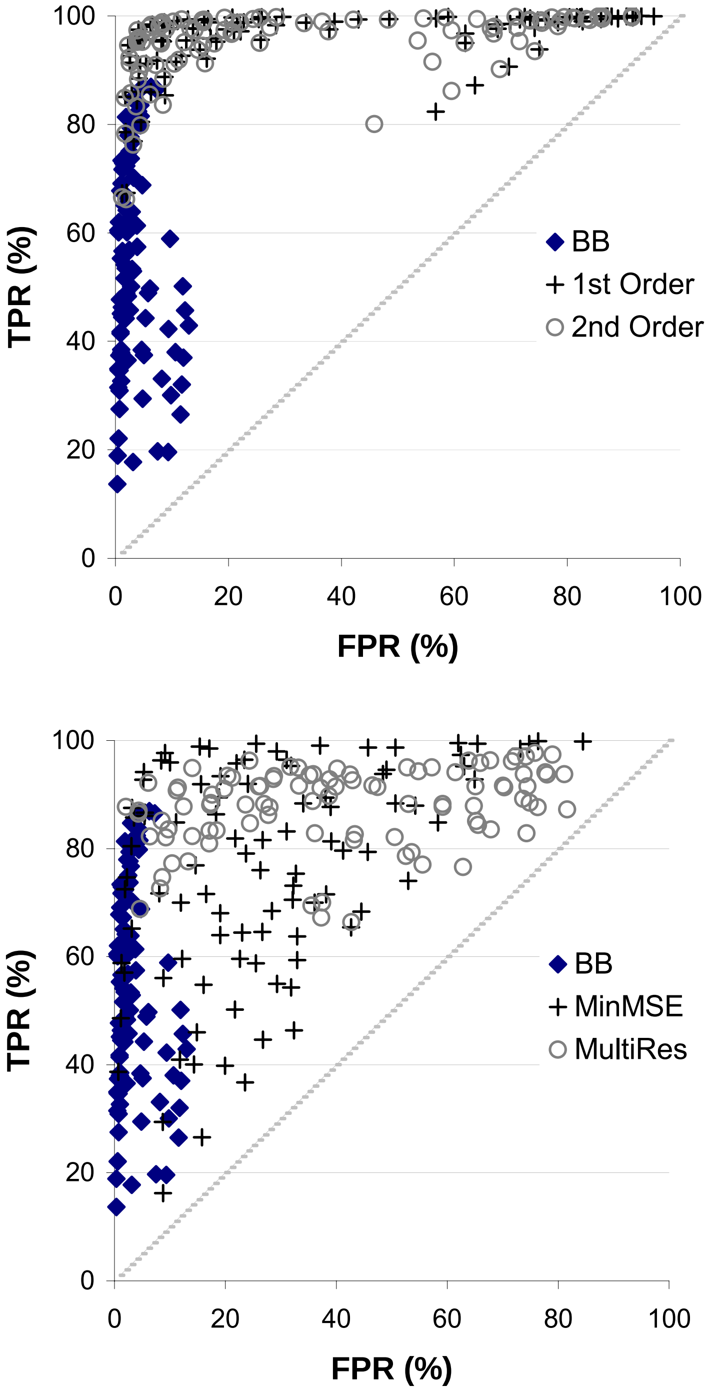


Figure 31. ROC plots for five illumination compensation techniques. Data was generated by processing ten test sequences and sweeping MCDth over {5, 7, 10, 15} and tile size {8, 16, 32, 80}. For the Multiresolution technique, the following four resolution sets were tested: {2, 4, 8, 16}, {4, 8, 16, 32}, {8, 16, 32, 80}, and {2, 4, 8, 16, 32, 80}.

For the BB-based method and the 1st Order, 2nd Order, and MinMSE methods, foreground/background classification is performed using the maximum component difference of the three color channels for each pixel. The multiresolution compensation method [46] is implemented here using the YCbCr color space. Because it does not compensate for changes in light source spectrum, full-color pixel comparisons lead to very high false positive rates. Therefore, foreground/background classification is performed using only the intensity of each pixel (i.e., (R+G+B)/3). The values for μ_0 and σ_0 were set to 128 and 40, respectively, as suggested by the authors.

When no compensation is applied, the false positive rate is often greater than 90%. The BB-based compensation technique results in less than 20% false positives, and performs especially well compared to the other methods during extreme lighting changes. By decreasing Rsize to 8 and Rth to 10 to improve BB coverage, the false positive rate is kept below 10%. However, this generally comes at the expense of a 10% to 20% increase in false negatives. Sequences RecCenter, TechSquare1, and TechSquare2 feature slight changes in intensity, resulting in fewer false positives for all five methods. Figure 32 shows examples of the foreground/background segmentation achieved using each compensation method. The most useful conclusions are drawn from the combination of Figures 31 and 32. While the local area statistic approaches presented in [43], and [44] typically have high true positive rates, the unpredictable number of remaining false positives presents a challenge to subsequent processes. The techniques from [45] and [46] have wide variability in both false positives and true positives. For the purposes of illumination compensation, where widespread false positives cause the most difficulty, the consistently low rates of false positives produced by the BB-based technique is a more useful operating range on the ROC curve. While fewer true positives are produced, enough remain that in the absence of distracting false positives, objects of interest can be found more easily.

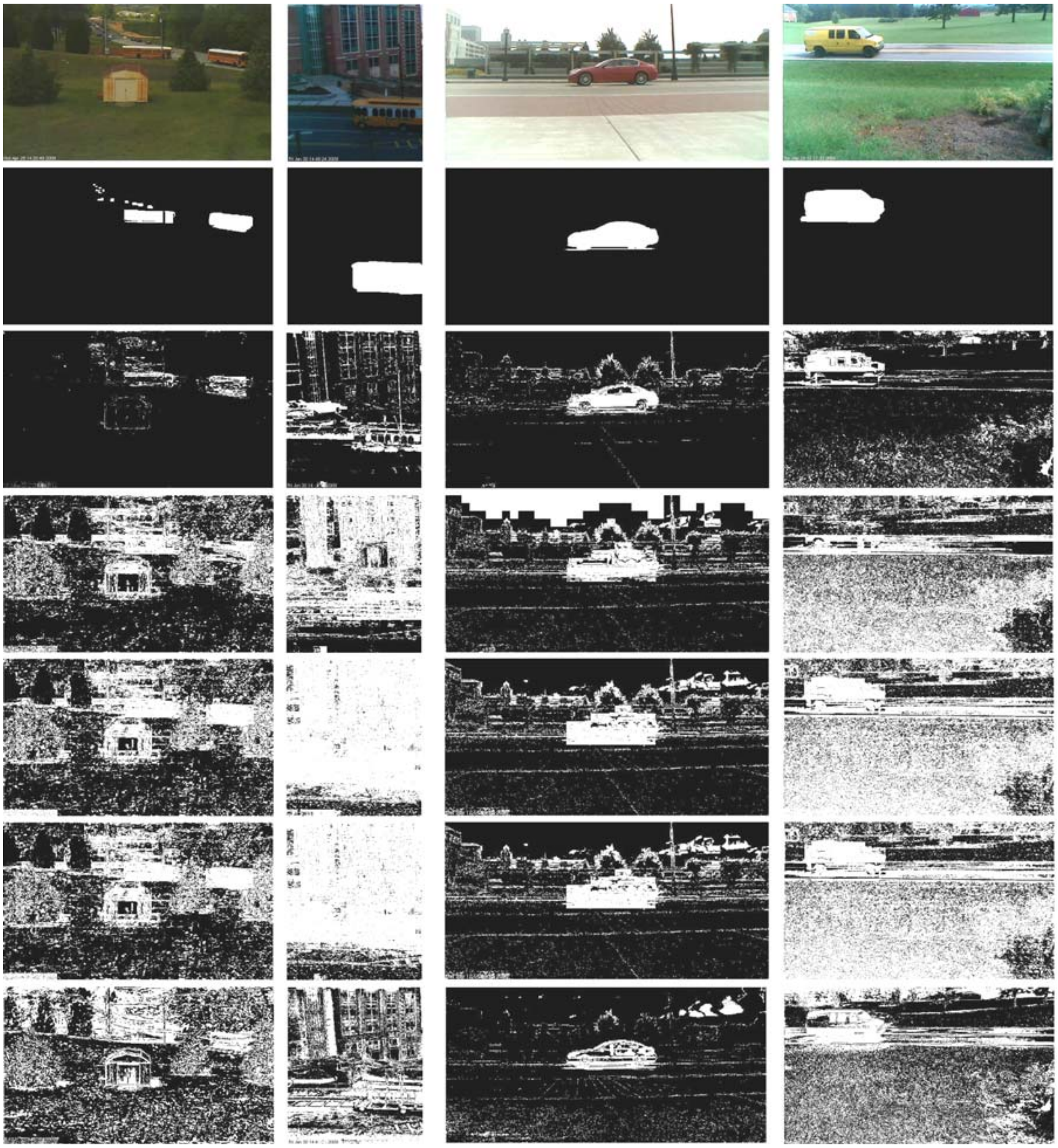


Figure 32. Segmentation results for illumination compensated scenes. Left to right: Backyard2, Ford1, TechSquare2, and Roadside sequences. Top to bottom: Original scene, Ground Truth, BB-based, MinMSE, First-order, Second-order, and Multiresolution compensation methods. For these examples, the parameters used were MCD=7, illumination tile size = 32, Rth=10, Rsize=8.

4.8.2 Execution time comparison

Each compensation method was coded in the C programming language, and executed on a PC running Ubuntu 10.04 and equipped with a 3.4 GHz Pentium D and 1GB of RAM. The same coding style was used for each algorithm, so while additional optimizations may be possible to improve absolute frame rate, this serves as a useful comparison for relative performance. Each trial—consisting of a combination of test sequence, compensation method, and tile size—is run 3 times; the standard deviation for each trial set is measured to be less than 2 ms. Data collection and file I/O processes are not included in these measurements. The average runtimes (in frames per second) are shown in Figure 33 for each sequence and method. Each runtime represents the average of nine trials: three trials for each of three tile sizes (8, 16, 32). The remaining parameters during this experiment were set to Rth=20, Rsize=16, and Cnum=15.

The execution time of the BB-based compensation method is on par with—and occasionally about 10% faster than—that of the first-order method. The proposed method consistently runs at more than twice the frame rate achieved by the second-order and Min MSE methods. The multiresolution method performs two passes through an image and bilinearly interpolates two statistics matrices per resolution, generally requiring 2-3 seconds per frame. All of the compared methods require considerable use of floating point calculations, while the BB-based method primarily uses integer arithmetic. Because the same number of pixels is processed regardless of tile size, larger tile sizes (and therefore fewer tiles) reduce the overhead incurred by processing each new tile. Finally, the BB-based compensation method requires slightly more time to execute for scenes with lower BB coverage, since non-BB pixels require a search of the color palette to find the closest color match.

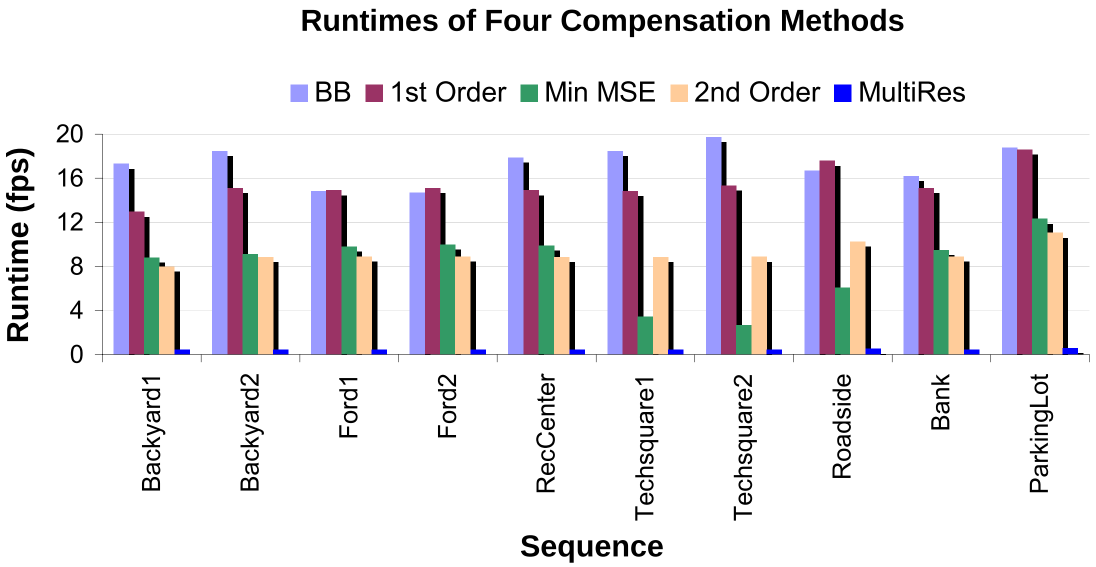


Figure 33. Runtime performance (in frames per second) of five illumination compensation techniques.

4.9. Conclusions

This chapter has used BigBackground, a stable feature identifier based on chromatic similarity, as a set of reference points with which to compute illumination compensation corrections. BigBackground employs the hypothesis that large, stable regions can be identified by the most popular colors in the scene. Experiments show that pixels identified as BigBackground are more stable than non-BigBackground pixels, and that the process of clustering similar colors across image tiles improves the efficiency of the color palette and allows the model to account for large percentages of scenes. Experiments show that the BigBackground model is effective at quantifying illumination changes by using simple RGB translation to account for those changes. Multiple cameras, multiple points of view, complex physical models, and special training sets are not used. False positives—the primary complications to change detection caused by illumination changes—are greatly reduced in foreground/background classification compared to competing algorithms. Applying an illumination compensation technique based on BigBackground decreases average false positives by 83% compared to no corrective action, and decreases average false positives by 25% to 43% compared to other compensation techniques from the literature. Resulting foreground/background images possess less clutter and feature better isolated and well-defined objects of interest. In addition, the execution time of the proposed technique is measured to be similar to a simple first-order, tile-oriented compensation approach, and is less than half of the time spent by second-order and multiresolution techniques.

When the local statistics-based compensation methods from the literature work well, they are effectively (but inadvertently) taking advantage of the BigBackground characteristic. When the processing tile is small enough that it contains a uniform homochromatic surface, its pixels are likely to respond similarly to illumination changes as observed in the experiments of Chapter 2. Thus a single compensation model tuned for all of the pixels in the tile works well. It is when the processing tile contains surfaces of multiple colors—either adjacent or interlaced—that such methods are most likely to fail.

Due to the tile-based approach for local compensation, the BigBackground-based technique can sometimes dissolve the interior of an object if a tile is completely enclosed by the object and the object satisfies one of the hue and saturation conditions. In the next chapter, a variation of BigBackground-Based illumination compensation is introduced to mitigate this effect. The previously discussed compensation techniques are applied to an object tracking problem for more thorough evaluation, and to determine the effects of illumination compensation on appearance-based tracking mechanisms.

CHAPTER 5

EFFECTS OF ILLUMINATION COMPENSATION ON OBJECT TRACKING

5.1. Introduction

Object tracking is a popular application for video processing, and is often called for in traffic management and pedestrian surveillance systems. An important first step in many object tracking pipelines is change detection: separating foreground (changing regions of interest) from background (stationary regions or uninteresting motion). Background models are used to statistically represent uninteresting aspects of the scene, and are subtracted from current images. Foreground pixels are aggregated together to form ‘blobs’ or objects, which are analyzed for similarities with previously observed objects for tracking. Accurate change detection is extremely challenging in scenes that experience illumination changes. As temporary cloud cover obscures the sun or artificial lights are turned on and off, the appearance of the entire scene changes and often causes otherwise unchanged background features to be misinterpreted as foreground. Salient moving objects then become lost in the clutter of false foreground.

This chapter demonstrates the effectiveness of various illumination compensation techniques on high-level tracking algorithms. Previously investigated methods are described for recognizing large, chromatically consistent regions (called BigBackground) in the scene, and for exploiting these regions as reference points in a compensation technique for illumination changes. Building upon previous work presented in [14], the concept of BigBackground is expanded into a general class of approaches by considering two variations of BigBackground-based illumination compensation. The first variation integrates a global measurement of a scene’s response to illumination change into the local compensation process. The second variation integrates the BigBackground concept into the second-order compensation technique used by Young et al. [43] and Lu et al. [44]. It is demonstrated that in addition to the compensation techniques designed specifically for BigBackground, the BigBackground concept can be usefully combined with independent techniques.

In video tracking pipelines that rely on change detection for object localization, there are two points at which illumination compensation can affect algorithm performance. First, the compensated image can improve change detection accuracy. Second, if the compensated image is passed to an appearance-based tracking algorithm, that algorithm will extract a compensated version of an object's appearance descriptor. Both influences of illumination compensation on the tracking process are examined here, and BigBackground-based approaches are found to achieve a favorable balance of false foreground elimination and object appearance preservation. The compensation technique that integrates a global view of illumination change with locally-performed adjustment is found to increase the accuracy rate of foreground detection by an average of 16% over the purely local approach, with negligible effect on the false positive rate.

Compensation method performance is evaluated in an object tracking problem with eight test sequences. Three tracking mechanisms are demonstrated: one based purely on kinematics, and two that rely on different appearance models of objects of interest. These trackers are characterized on scenes with constant illumination. During illumination change events, the BB-based compensation techniques improve tracking accuracy by an average of 20% over other compensation methods for each tracking approach. Because real-time embedded systems are of interest here, the algorithms described are designed with low computational cost in mind. Evaluation sequences are captured with stationary, inexpensive USB webcameras that are generally oriented orthogonally to the plane of motion.

5.2. Related Work

A general survey of object tracking techniques is provided by Yilmaz et al. [53]. The described approaches are motivated by a variety of applications, and are suited for tracking many different object types depending on the problem domain. Trackers have been presented that focus on specific types of traffic, such as vehicles or pedestrians.

Several point detection mechanisms have been proposed for producing trackable feature sets, such as SIFT points [54], [55], SURF points [56], and multiresolution critical points [57], [58]. These points are often fed into particle or other statistical filters.

The mean-shift algorithm has received considerable attention in the last decade as a segmentation-oriented tracking mechanism. Originally proposed by Comaniciu et al. [59], mean-shift tracking uses some feature space (such as color, texture, or edges) in which to describe the object model. An iterative gradient ascent algorithm is employed to minimize the Bhattacharyya distance between the target model and the candidate region observed in the scene. Classical mean-shift tracking requires that an object's boundaries spatially overlap between consecutive frames. It is also necessary to provide the algorithm ahead of time with models of the objects to be tracked, or to include a means for detecting objects of interest so the model can be formed from the first observation. These limitations have been addressed by recent work such as [60], which tracks fast-moving objects or objects in low frame rate video with multiple kernels, and [61], which tracks objects by considering all of a blob's fragments (if any) and using a voting strategy to determine the fragment that best describes the object. In an environment that may contain multiple objects with unknown appearances, a foreground object detection mechanism is best used to detect and learn the appearance of a new object upon its first instance. This approach has the additional advantage of reducing the search space to a short list of objects.

Computationally-sensitive trackers have been described that first identify blobs by background modeling and change detection, and then distinguishing each blob with a small, simple set of figures such as kinematic principles [17], object-strip color [62], or spectral distribution [63]. These approaches are attractive because their features are efficient to compute, and they have inherently manageable search spaces.

The work in this chapter focuses on the common surveillance applications of pedestrian and vehicular traffic tracking. Because gesture recognition and object orientation are not needed in this case, complex object representations such as skeletons, articulated and geometric shapes are unnecessary. For computational efficiency, a center of mass representation is preferred in conjunction with a kinematic tracking mechanism. The algorithm described by Apewokin et al. [17] is used as a representative kinematic tracker. This tracker is inspired by principles of motion correspondence that are well-described by Rangarajan and Shah [64]. Maximum velocity and small velocity change constraints are exploited to establish correspondence between blobs. Trackers based on object-strip color [62] and a spatio-spectral model [63] are also implemented to observe the effects of illumination compensation on appearance-based tracking descriptors. The general processing framework used here consists of background subtraction, morphological erosion, blob formation, and blob tracking. As mentioned by Masoud and Papanikolopoulos [65], this framework approach is preferable for its efficiency and versatility.

Object tracking algorithms commonly exploit assumptions such as constant illumination to make the tracking problem tractable. In real applications, such assumptions are often violated [53]. This chapter examines the application of novel illumination compensation techniques to several video sequences featuring significant illumination change, and evaluates the effectiveness of the techniques on three modern tracking algorithms.

5.3. Approach

The concept of BigBackground (BB) is based on the observation that most surveillance scenes contain large, stationary, chromatically self-similar regions such as buildings, sidewalks, roads, and treelines. The BB algorithm recognizes these regions by identifying the most common colors in the scene. It is then possible to construct a small color palette that represents a relatively large proportion of the scene, and map which pixels are likely to belong to large, stationary objects.

This dissertation proposes an approach to handling illumination change based on using BB as a calibration anchor. BigBackground identifies large stable regions that are likely to change only due to illumination, and allows occlusions to be differentiated from illumination changes. The effects of illumination change are then quantified by comparing BB regions, while the effects of transient occluding objects are prevented from corrupting the compensation. The different responses of differently colored surfaces to a given illumination change are accommodated in the model by customizing the compensation parameters for each BB region.

Several adaptive background models have been proposed such as Mixture of Gaussians [12], Multimodal Mean [29], and sliding window-based techniques [11]. Adaptation is necessary to avoid having to restart the system due to subtle appearance changes or uninteresting motion. The rate at which these models adapt is critical. A model must adapt slowly to avoid prematurely absorbing transient salient objects that would then cease to be detected. It is assumed that slowly-varying illumination—such as that caused by the sun's natural progression across the sky—would be absorbed into the background model. The proposed methodology improves algorithm response to sudden illumination changes, and assists during the transition period either until the background model has adequate time to adapt, or the illumination condition reverts to its previous state.

The video processing pipeline used here is shown in Figure 34. Illumination compensation is applied to incoming images to improve change detection accuracy. Pixels are then classified as foreground or background, and blobs of foreground pixels are located for tracking. Three tracking algorithms from the literature are considered—each of which relies on a different set of features for establishing correspondence—and the effects of illumination compensation techniques on those features are evaluated. The background and BB models adapt during a preamble period of N frames, and remain constant for the duration of the sequence.

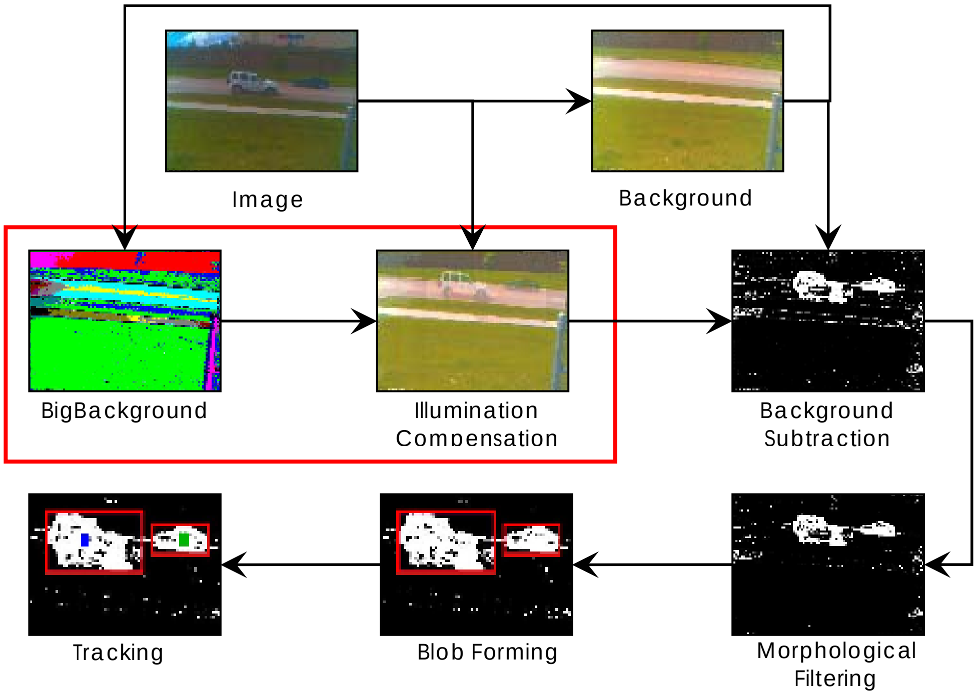


Figure 34: Overview of object tracking processing pipeline with illumination compensation.

5.4. BB-based Illumination Compensation

Changes in illumination can drastically change the appearance of a scene and can cause surveillance vision algorithms to falter in their recognition of the features being observed. Many adaptive background models have been developed that can eventually acclimate to environmental changes. However, the adaptation rates of such models must generally be small to prevent premature adaptation to transient occlusions. It is desirable to efficiently compensate for illumination variation for robust scene understanding. Moreover, it is desirable to base such compensation on reliable visual cues extracted from salient features in the scene itself, rather than to simply use average changes from neighboring pixels.

The next section demonstrates that it is useful to exploit the stability of BB for an illumination compensation model. Three variations of BB-based illumination compensation are discussed here: the first is a simple offset technique first described in [13]. The second is a variation that considers the global appearance of BB before computing local compensations. The third is a hybrid in which the second-order model used by Young et al. [43] and Lu et al. [44] is applied separately to each BB region.

5.4.1 Offset Compensation

As discussed in Chapter 4, after reviewing literature in which illumination models are explored [18], [19], and after evaluating our own models [14], a computationally efficient RGB translation compensation model is chosen. A translation (or offset) value is computed for each color component, and the offset is added to the pixel being compensated. The goal is for the transformed pixel to match the pixel's appearance under the original lighting condition.

The offset value is calculated as follows. The process is performed on a tile-by-tile basis to accommodate localized lighting changes such as partial cloud cover. Within each tile, a new color palette C_1 is calculated by averaging all of the qualifying pixels belonging to the same BB region. A pixel qualifies to be included in the average if it meets one of the criteria in (20), where H_x and S_x denote the pixel's hue and saturation, respectively. The subscript x takes on a value of 1 to denote a pixel in the original lighting condition, or a value of 2 to denote a pixel in the new lighting condition.

$$\begin{array}{l} S_1 > T_S \text{ AND } S_2 > T_S \text{ AND } \left|H_1 - H_2\right| < T_H \\ S_1 > T_S \text{ AND } S_2 > T_S \text{ AND } \left|S_1 - S_2\right| < T_H \\ S_1 < T_S \text{ AND } S_2 < T_S \end{array} \tag{20}$$

The saturation and hue thresholds, T_S and T_H, were chosen empirically to be 12% and 8%, respectively, and have been found to be effective on a wide range of scenes. If hue alone is checked, a large number of false mismatches are identified in unsaturated regions, such as pavement and concrete. If a pixel does not meet one of these criteria, there is a high likelihood that the pixel belongs to an occluding object and should not contribute to the compensation process.

$$\begin{array}{l} C_1[i] = \dfrac{1}{N_i} \displaystyle\sum_{\mathrm{x}}^{tile} \sum_{y}^{tile} I_2(x, y) \\ \quad \forall I_2(x, y) \in C_0[i] \; and \; \text{ satisfy Hue/Sat rules} \end{array} \tag{21}$$

Once the new average color palette C_1 is calculated within a tile (21), the compensation offset for each region is calculated by subtracting the new average from the region's original BB color, as given by the reduced color palette C_0. The compensation parameter is thus the difference between the original and new color palettes. The offset is added to all pixels within the tile that belong to that BB region as shown in (22), where C_0 denotes the original BB color palette, C_1 denotes the color palette for the BB regions in the image being compensated, and $i_{x,y}$ denotes the color palette index of pixel (x,y).

The process is repeated for each tile in the image. The compensated image is then ready for additional processing, such as foreground/background classification.

$$I'(x,y) = I(x,y) + [C_0(i_{x,y}) - C_1(i_{x,y})] \tag{22}$$

5.4.2 Globally-Aware Compensation

In the previously discussed technique, tiles that are completely covered by homogeneous desaturated foreground objects can sometimes be dissolved into the background. The tile-by-tile approach is effective at handling spatial variations in illumination response, but is so localized that foreground objects can be over-transformed. In this globally aware variation, we exploit the largeness of BB to help discern foreground objects from background. Rather than compare pixels from a new illumination condition with the BB colors of the original condition in a single step, in which changes in appearance could be due to illumination or occlusion, we divide the decision process into two stages. First we learn how the BB colors across the scene have changed with illumination by computing the average color of each region, yielding the new global color palette C_G. Considering such a large area reduces the influence of occluding objects on the result. Then within each tile we compare individual pixels from the new image with the colors of C_G to determine which tile pixels should contribute to the local color palette C_1 (23). Corrections are then calculated and applied as defined in (22).

$$\begin{gathered} C_G[i] = \frac{1}{N_i} \sum_{\mathrm{x}}^{width} \sum_{y}^{height} I_2(x,y) \qquad \forall I_2(x,y) \in C_0[i] \\ C_1[i] = \frac{1}{N_i} \sum_{\mathrm{x}}^{tile} \sum_{y}^{tile} I_2(x,y) \\ \forall I_2(x,y) \in C_0[i] \ \mathit{and} \ \mathrm{MCD}(I_2, C_G[i]) < 10\% \end{gathered} \tag{23}$$

5.4.3 Hybrid Compensation

The hybrid compensation method combines the chromatic discrimination of BB with the second-order compensation formula. Rather than computing RGB offsets for each BB region, the mean and standard deviation of each BB region are computed tile by tile and (24) is applied. This demonstrates the effects of chromatic separation on a traditional compensation approach, and addresses the problem of applying local area statistics models to regions containing multiple disparate surfaces.

$$I_2'(x,y) = \left(I_2(x,y) - \overline{I_2}\right)\left(\frac{\sigma_2}{\sigma_1}\right) + \overline{I_1} \qquad (24)$$

5.4.4 Analysis

The ROC plot of Figure 35 compares the two new BB-based variations with the original offset compensation in terms of false positive rates and true positive rates. The Global variation maintains an even tighter distribution around zero false positives than the original offset technique, and produces a larger lobe near the top of the true positive axis. Table 16 shows the changes in true positive and false positive rates for each method (BB-Global and BB-Hybrid) compared with its original counterpart (BB Offset and 2nd order, respectively). True positives are correctly identified foreground pixels. False positives are background pixels that are misclassified as foreground. Note that these numbers represent differences between percentages, not percentages of percentages. A ΔTPR of 12% represents a 12% increase in the number of correctly identified foreground pixels. The BB-Global algorithm increases true positives by an average of 16% with negligible impact on false positives, thus providing more complete object silhouettes. The BB-Hybrid algorithm reduces true positives by 16%, but reduces false positives by 25%. The decrease in false positives makes reduces the likelihood that important features will be lost among surrounding background noise. Sufficient foreground remains to identify objects despite the reduction in true positives; because a significant problem for the 2nd Order technique in these sequences is the masking of objects by surrounding false foreground, the BB-Hybrid method presents an improvement over 2nd Order compensation in most cases.

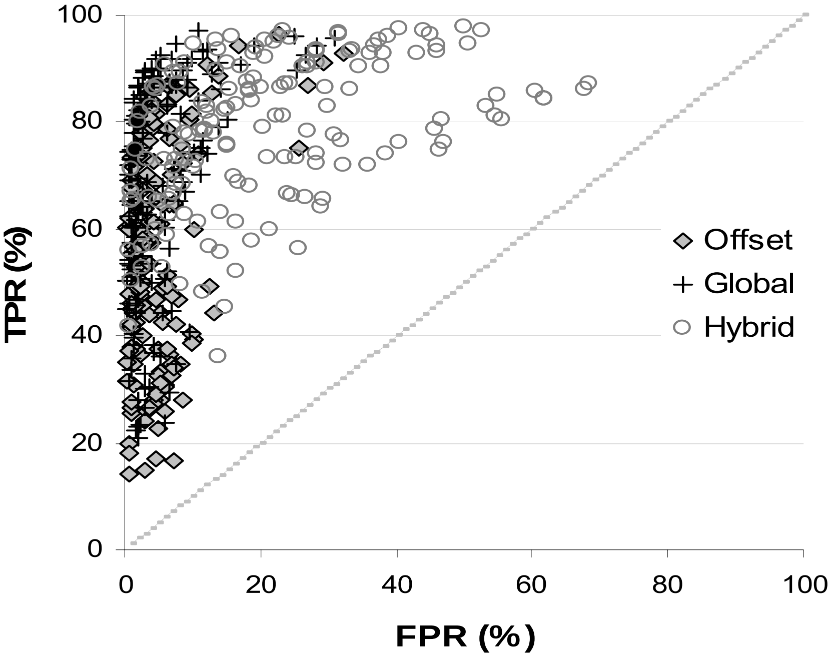


Figure 35. ROC plot comparing new BB-based compensation methods to the original Offset method. To produce these points, compensation tile sizes were swept across {8, 16, 32, 80}, and the MCD threshold was swept across {5, 7, 10, 15}. The same sequences were used as in [14].

Table 16. Relative Improvements of New Illumination Compensation Techniques

			BB-Global vs. BB-Offset		BB-Hybrid vs. 2nd Order	
Sequence	FG Pixels	BG Pixels	ΔTPR	ΔFPR	ΔTPR	ΔFPR
Backyard1	2234	303154	12%	-1%	-10%	-36%
Backyard2	7639	247904	11%	0%	-22%	-23%
Ford1	12086	111284	-4%	-3%	-18%	-24%
Ford2	13900	265766	7%	4%	-17%	-29%
TechSquare1	13083	292587	27%	0%	-17%	11%
Roadside	13003	223092	26%	1%	-11%	-38%
Bank	52562	241432	34%	2%	-16%	-35%
AVG			**16%**	**0%**	**-16%**	**-25%**

The sample images shown in Figures. 36, 37, and 38 give a more detailed look at how the BB-Hybrid and 2nd order techniques compare. It can be observed in all three figures that BB-Hybrid compensation reduces false foreground pixels, (particularly in Figures 36 and 37), and simultaneously preserves the solidity and original appearance of the objects in the color images. The images of Figures 39, 40, and 41 compare the BB-Global variation with the original offset compensation. The BB-Global method produces stronger silhouettes and noticeably preserves object appearance, reducing the translucency that results from overcompensation.

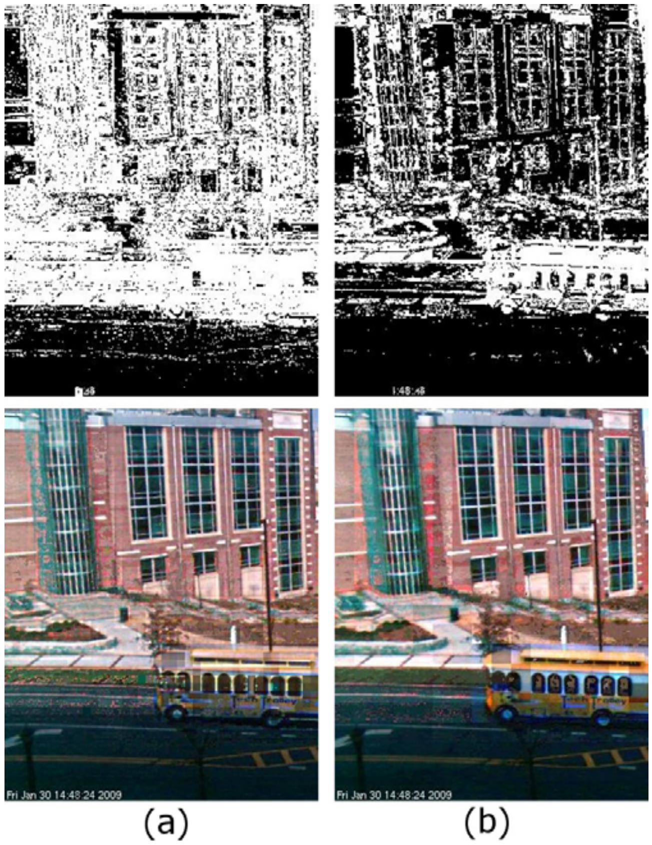


Figure 36. Ford1 sequence examples of foreground masks and compensated images produced by (a) 2nd order compensation and (b) BB-Hybrid compensation.

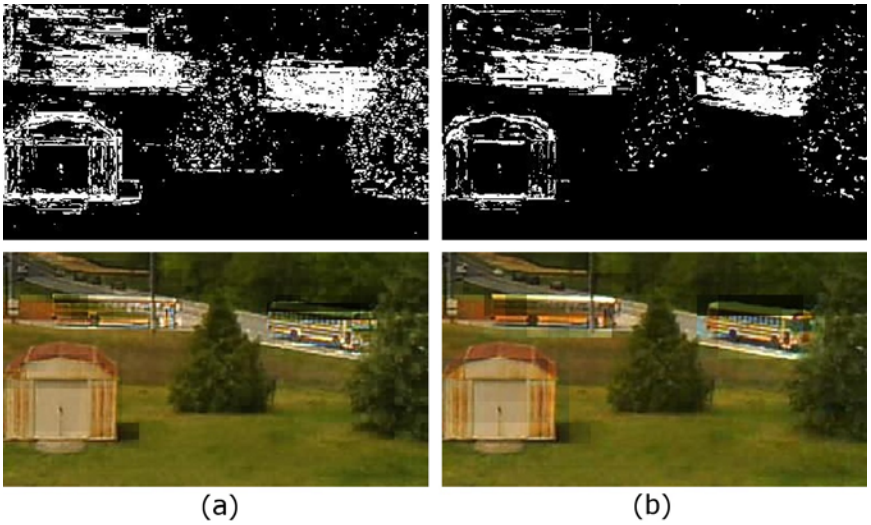


Figure 37. Backyard sequence examples of foreground masks and compensated images produced by (a) 2nd order compensation and (b) BB-Hybrid compensation.

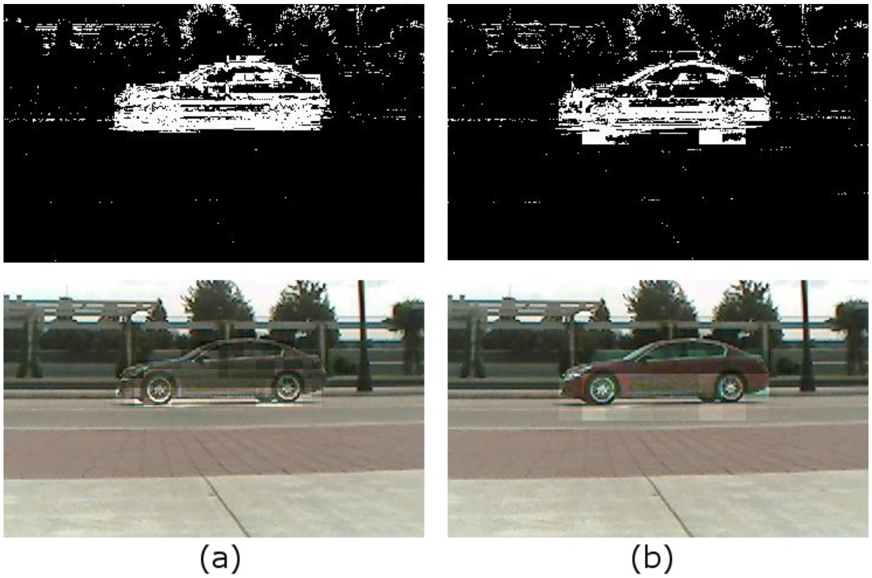


Figure 38. Techsquare1 sequence examples of foreground masks and compensated images produced by (a) 2nd order compensation and (b) BB-Hybrid compensation.

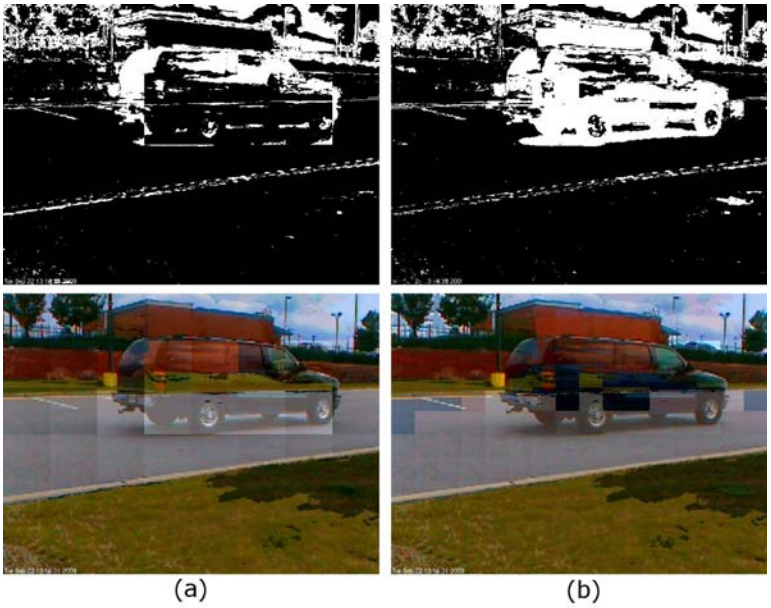


Figure 39. Bank sequence examples of foreground masks and compensated images produced by (a) Offset compensation and (b) BB-Global compensation.

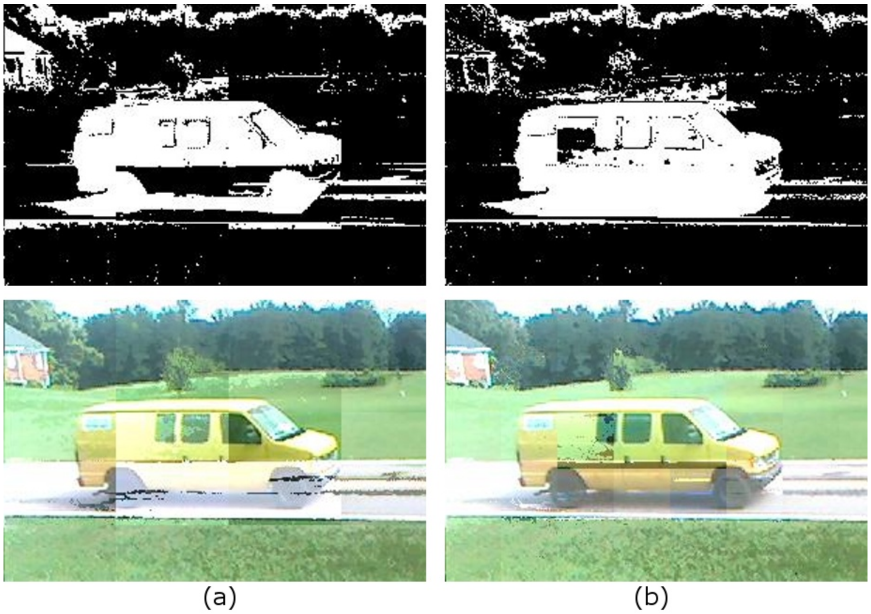


Figure 40. Roadside sequence examples of foreground masks and compensated images produced by (a) Offset compensation and (b) BB-Global compensation.

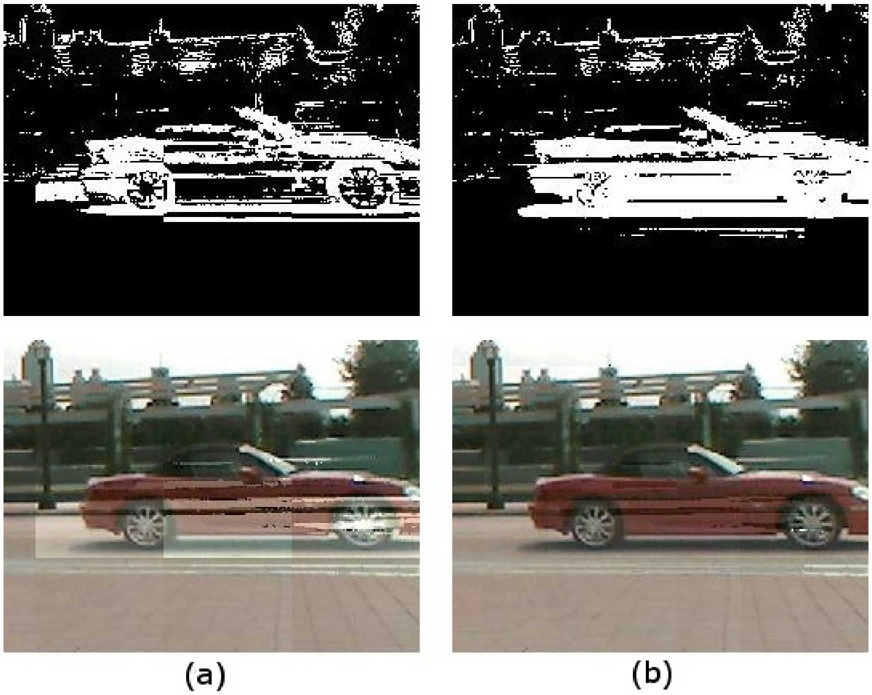


Figure 41. Techsquare2 sequence examples of foreground masks and compensated images produced by (a) Offset compensation and (b) BB-Global compensation.

5.5. Blob Formation

After illumination compensation, change detection is performed to classify pixels as foreground (salient) or background. Regions of high spatial foreground density are classified as objects of interest by scanning for peaks in horizontal and vertical histograms. The number of foreground pixels is counted in each row and column as defined in (25), where F(x,y) is the foreground mask.

$$\begin{aligned} Col[x] &= \sum_{y=0}^{height-1} F(x,y), \quad x \in [0, width) \\ Row[y] &= \sum_{x=0}^{width-1} F(x,y), \quad y \in [0, height) \end{aligned} \tag{25}$$

After obtaining the histograms, the row counts are scanned for contiguous runs of MinYSize or more rows of at least RowTotalTH foreground pixels. Column counts are scanned for contiguous runs of MinXSize or more rows of at least ColTotalTH foreground pixels. An example of this process is shown in Figure 42. The third row has only three foreground pixels. Because the RowTotalTH = 4, this row does not contribute to the run of Y1. Every X,Y combination corresponds to the location of a potential blob. Each subregion (in this case, (X1,Y1) (X1,Y2), (X2,Y1), and (X2,Y2)) is then rescanned to discover if any blobs are masking each other. The (X2,Y1) blob will be ignored when it is scanned individually because its horizontal size is less than MinXSize. The (X1,Y1) blob will be ignored because its row totals will be less than RowTotalTH. In this example, only the (X2,Y2) blob will survive for tracking. The thresholds RowTotalTH, ColTotalTH, MinXSize, and MinYSize can be adjusted based on the size and geometry of the objects of interest.

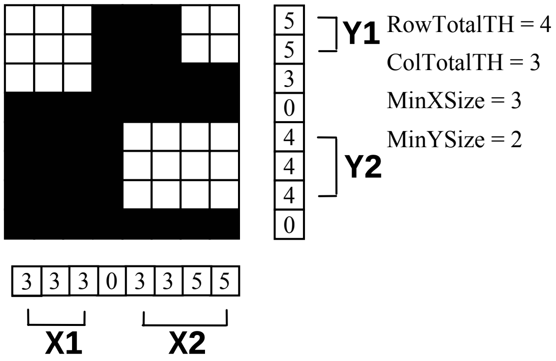


Figure 42. Example of blob-forming process. The number of foreground pixels is counted for each row and column. These counts are examined for long runs of high counts. This process is repeated for each identified region to determine if one blob is masking another, or if high counts are due to many individual small blobs.

5.6. Tracking Techniques

The effects of illumination change compensation techniques are observed on three tracking mechanisms chosen from recent literature. The considered trackers do not rely on prior information about the targets of interest, and use change detection and blob formation to locate and track new targets. In this framework, correspondence is established by searching lists of new and previously observed objects instead of spatially searching around the positions of previously observed objects.

Three tracking methods are implemented for comparison. The kinematic tracking algorithm presented by Apewokin et al. [17] relies solely on object position and velocity. Two appearance-based tracking algorithms that use different appearance descriptors: the object-strip color descriptor presented by Zhang et al. [62], and the spatio-spectral probability presented by Tavakkoli et al. [63]. The kinematic tracker does not rely on appearance information, and therefore benefits from illumination compensation only in object detection. This approach provides a contrast with the appearance-based trackers. Each tracker operates by comparing a list of objects extracted from the current image with a historical list of objects previously observed. Upon establishing a correspondence between an object in each list, the historical list is updated with current information. Objects in the historical list that have not been observed in 10 frames are pruned.

5.6.1 Kinematic Tracking

A low-cost implementation of a kinematic tracking algorithm [17] is chosen as a representative appearance-apathetic tracking approach with which to demonstrate the impact of illumination compensation. This tracker models objects as centers-of-mass. Maximum velocity and small velocity change constraints are employed to achieve correspondence between blobs.

Object tracking information is maintained in two lists: one list for short-term tracking between three consecutive frames (t, t-1, and t-2), and one list for long-term tracking over tens of frames. Short-term tracking is performed first. Objects observed in frame t are matched to objects from frames t-1 and t-2 that minimize change in speed. If an object is unobserved in a previous frame such that speed is unknown, then distance traveled is minimized. The long-term tracker is then used to match objects that were previously occluded. Inertia (persistence of an object to travel in one direction) is used to break ties between objects that are close together. No predictions are made of future object positions; correspondences are established based solely on observed positions.

5.6.2 Object-Strip Color Tracking

Zhang et al. [62] introduce object-strip color (OSC) features as a basis for establishing object correspondence. Blobs of interest are located by background subtraction, and each blob is vertically divided into strips of predetermined size. Each blob is represented by a one-dimensional vector of the average hues of its constituent strips. This formulation is shown in (26), where w is the object width, d is the strip size, h is the object height, (xmin, ymin) is the top left corner of the object bounding box, and i is the index into the strips that comprise the object. Correspondence is established between blobs by maximizing the correlation coefficient between object-strip color vectors (27). OSC vectors of different lengths are shifted against each other to find the optimum match point for the pair. The OSC feature incorporates color and spatial distribution information, while requiring less computational effort than full color histograms. In this implementation, each new object is compared with each previously observed object. The pair with the highest correlation coefficient is associated together and removed from consideration, and the process is repeated with the next-highest match.

$$OSC[i] = \frac{1}{wd} \sum_{x=x\min}^{x\min+w} \sum_{y=y\min+id}^{y\min+2id} Hue(I(x,y))\,;\ \forall i \in \left[0, \left\lfloor \tfrac{h}{d} \right\rfloor\right] \qquad (26)$$

$$R = \frac{\frac{1}{d}\sum_{i=0}^{d-1}\left(OSC_1[i]-\overline{OSC_1}\right)\left(OSC_2[i]-\overline{OSC_2}\right)}{\sqrt{\frac{1}{d}\sum_{j=0}^{d}\left(OSC_1[i]-\overline{OSC_1}\right)^2}\sqrt{\frac{1}{d}\sum_{j=0}^{d}\left(OSC_2[i]-\overline{OSC_2}\right)^2}} \qquad (27)$$

5.6.3 Spatio-Spectral Tracking

Appearance-based correspondence is also used in the spatio-spectral tracking (SST) proposed in Tavakkoli et al. [63]. This technique uses a first-order statistical estimation of each object's photometric appearance as a descriptor for establishing correspondence. Each object is modeled as separate upper and lower halves to accommodate bimodal object appearance (particularly in the case of pedestrians). The YCbCr color space is used to separate color from intensity. A Gaussian probability distribution is assumed, and the mean and covariance matrix of each color component is computed for the upper and lower halves of each object. Thus each object is modeled as a set of four parameters: <**M$_U$**, **S$_U$**> and <**M$_L$**, **S$_L$**> as in (27) where ρ is the covariance between the two components.

$$\mathrm{M} = \begin{bmatrix} \mu_{\mathrm{Cb}} \\ \mu_{Cr} \end{bmatrix}; \quad \mathrm{S} = \begin{bmatrix} \sigma_{Cb}^2 & \rho\sigma_{Cb}\sigma_{Cr} \\ \rho\sigma_{Cb}\sigma_{Cr} & \sigma_{Cr}^2 \end{bmatrix} \tag{28}$$

When an object is detected in a new image, it is tested for the probability that it is associated with each previously observed object. The probability that each pixel of the new object belongs to a previously observed object's bivariate pdf is calculated (20). The average of all of these pixel probabilities is taken, and correspondence is established between each new object and the previous object that yielded the highest probability.

$$f(x,y) = \alpha \cdot \exp\left[\beta\left(\frac{(Cb-\mu_{\mathrm{Cb}})^2}{\sigma_{\mathrm{Cb}}^2} + \frac{(Cr-\mu_{\mathrm{Cr}})^2}{\sigma_{\mathrm{Cr}}^2} - \frac{2\rho((Cb-\mu_{\mathrm{Cb}}))((Cr-\mu_{\mathrm{Cr}}))}{\sigma_{\mathrm{Cb}}\sigma_{\mathrm{Cr}}}\right)\right] \tag{29}$$

$$\alpha = 1/2\pi\sigma_{\mathrm{Cb}}\sigma_{\mathrm{Cr}}\sqrt{1-\rho^2}$$
$$\beta = 1/2(1-\rho^2)$$

The full tracking algorithm includes steps for collision detection and resolution, and Kalman filtering of future position estimates. As the primary interests of this chapter are the effects of illumination change and compensation on tracking appearance descriptors, only the appearance-based correspondence component is replicated here.

5.7. Experiments and Results

For this study, the baseline performance of the trackers on video sequences with constant illumination is first established. The effects of illumination change on tracking success rate are demonstrated, followed by the evaluation of the proposed methods for illumination change compensation as well as four methods from the literature. Each method is evaluated using both automotive traffic and pedestrian traffic environments.

5.7.1 Evaluation Method and Baseline

The video sequences used to test the algorithms in these experiments were captured in a variety of environments. In each environment, video was taken of pedestrian or vehicular traffic during stable lighting conditions and during illumination changes. By establishing the tracking algorithm's baseline performance on these stable lighting sequences, we demonstrate that the chosen background model and change detection method properly detect objects of interest, and that this tracker is suitable for these environments under nominal conditions. When the sequences featuring illumination change are tested with the compensation techniques, the variable is illumination change, and not the environments or traffic characteristics.

Tracking results are checked for accuracy by a human observer. A ground truth of how each object should be tracked is established ahead of time. The following rules are used to evaluate tracking accuracy.

(1) The center of mass must fall somewhere within the object's silhouette.

(2) The object's bounding box must not exceed 2X the object's size.

(3) During occlusions, one of the occluding objects' ID labels must be used.

(4) After an occlusion, the objects' original ID labels must be restored.

(5) If an object's blob becomes fragmented, at least one ID label must carry over from the prior frame.

The baseline performance of the tracking algorithms are given in Table 17, Table 18, and Table 19. The trackers are also tested with illumination compensation turned off. In the absence of illumination change, clear object silhouettes are obtained from background subtraction and the tracker is able to successfully establish correspondence for each object across the scene. This data also shows that in the absence of illumination change, the compensation techniques do not significantly harm tracking quality.

Table 17. Baseline Accuracy for Kinematic Tracker

	None	BB-Offset	BB-Global	BB-Hybrid	MinMSE	1st	2nd	MultiRes
CarsA	100%	96%	98%	99%	99%	99%	100%	100%
CarsB	100%	100%	100%	100%	90%	100%	100%	99%
CarsC	100%	96%	93%	100%	100%	100%	100%	93%
PedA	100%	100%	100%	100%	97%	94%	92%	97%
PedB	97%	100%	97%	97%	81%	95%	89%	100%
AVG	**99%**	**98%**	**98%**	**99%**	**93%**	**98%**	**96%**	**98%**

Table 18. Baseline Accuracy for OSC Tracker

	None	BB-Offset	BB-Global	BB-Hybrid	MinMSE	1st	2nd	MultiRes
CarsA	92%	89%	90%	93%	92%	96%	98%	90%
CarsB	100%	97%	100%	100%	78%	100%	100%	87%
CarsC	100%	93%	93%	100%	88%	85%	85%	95%
PedA	100%	100%	100%	100%	94%	86%	89%	94%
PedB	95%	89%	89%	86%	92%	92%	97%	97%
AVG	**97%**	**94%**	**94%**	**96%**	**89%**	**92%**	**94%**	**93%**

Table 19. Baseline Accuracy for SST Tracker

	None	BB-Offset	BB-Global	BB-Hybrid	MinMSE	1st	2nd	MultiRes
CarsA	92%	87%	92%	92%	88%	91%	92%	92%
CarsB	100%	99%	100%	100%	90%	100%	100%	90%
CarsC	100%	92%	93%	100%	99%	88%	91%	95%
PedA	100%	94%	100%	100%	97%	81%	81%	97%
PedB	97%	100%	100%	92%	95%	89%	89%	97%
AVG	**98%**	**94%**	**97%**	**97%**	**94%**	**90%**	**91%**	**94%**

5.7.2 Test Sequences

Samples of the eight test sequences used in these experiments are shown in Figure 43, depicting six distinct environments that vary in terms of traffic type, direction of travel, and apparent size. In total, over 500 frames containing objects are processed to provide a more thorough evaluation than hand-picking ground truth images. The vehicular sequences Cars1 and Cars2 feature less extreme intensity changes, and each technique allows successful tracking of at least a few objects over several frames. However, the remaining sequences exhibit relatively strong intensity changes. The sequences PETs1 and PETs2 are taken from the publicly available PETs 2001 Dataset 3 [66], which features pedestrian traffic and illumination change caused by cloud cover. These are challenging due to the relatively small size and greater number of the pedestrians being tracked.

The objective is to compensate illumination changes that are strong enough to preclude the tracking of objects in uncompensated images. For the sequences used, object tracking is not possible without compensation because false foreground completely masks the objects of interest. It is demonstrated in Figure 44 that simply adjusting the segmentation threshold is insufficient to accommodate illumination changes. Eight video sequences featuring illumination change are processed, and the maximum component difference (MCD) threshold used to classify foreground and background is swept among {25, 50, 67, 75, 83, 100}. No illumination compensation is used. Two types of errors occur in blob recognition. If the threshold is too low for the environment, background pixels are erroneously classified as foreground and will mask the true objects. If the threshold is too high, foreground pixels are erroneously classified as background and the object’s silhouette is dissolved. Dotted lines represent the percentage of blobs that are not recognized due to dissolving. Solid lines represent the percentage of blobs that are not recognized due to masking. Because there are no ranges of thresholds for which most objects are reliably recognizable, classification threshold adjustment alone cannot solve the illumination change problem.

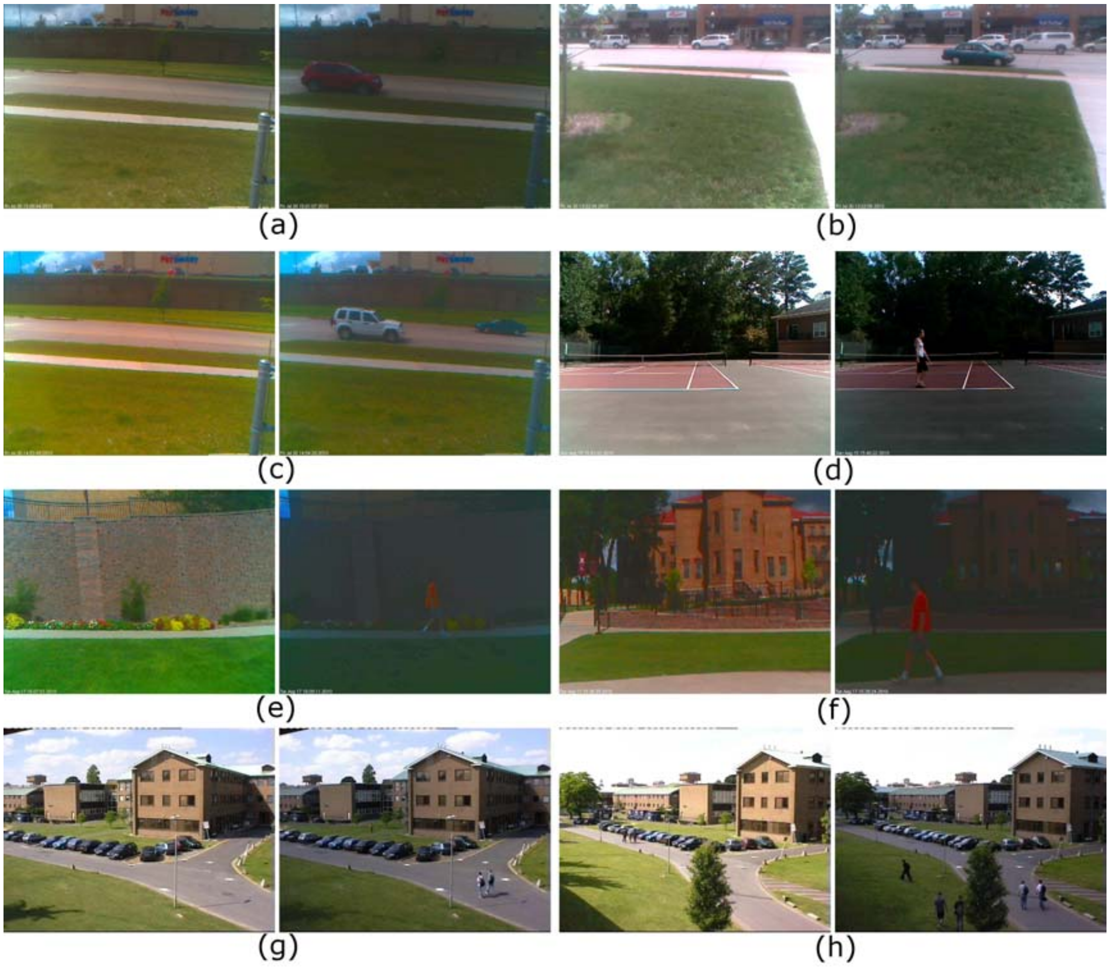


Figure 43. Samples of image sequences used in tracking evaluation before and after illumination changes: (a) Cars1, (b) Cars2, (c) Cars3, (d) Ped1, (e) Ped2, (f) Ped3, (g) PETs1, and (h) PETs2.

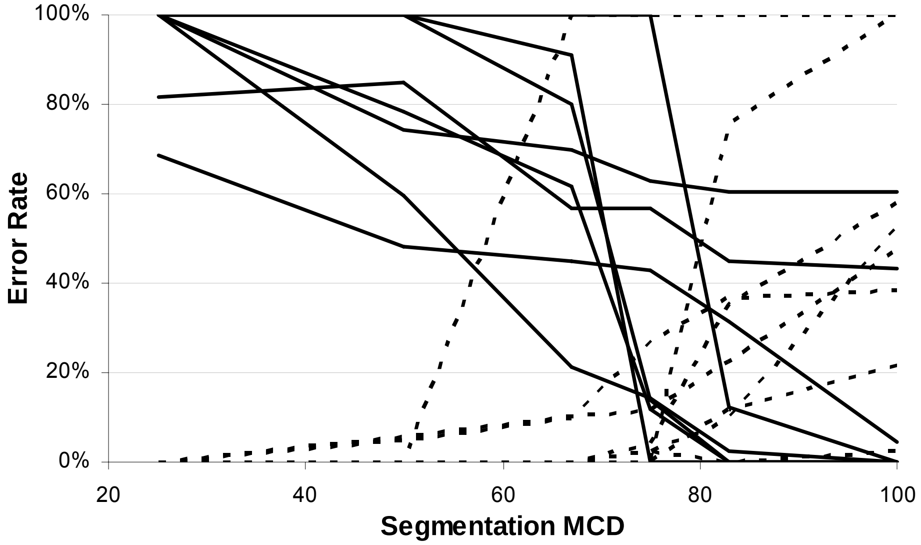


Figure 44. Variation in object identification errors during illumination changes as a function of MCD threshold. Dotted lines denote the percentage of blobs dissolved into the background due to a high MCD threshold. Solid lines denote the percentage of blobs obscured by surrounding false foreground.

Table 20. Parameters Used for Methods and Sequences

	Cars	Peds	PETs
BB-based MCDth	10	10	10
BB-based Tilesize	40	40	20
BB-based blob size	{20, 20}	{10, 35}	{10, 20}
MinMSE MCDth	7	15	10
MinMSE Tilesize	20	20	20
MinMSE blob size	{20, 20}	{10, 35}	{10, 20}
1-, 2-Order MCDth	20	15	15
1-, 2-Order Tilesize	8	8	8
1-, 2-Order blob size	{20, 20}	{10, 35}	{10, 20}
MultiRes MCDth	20	20	20
MultiRes Tilesize	{320, 160, 80 40}	{320, 160, 80 40}	{320, 160, 80 40}
MultiRes blob size	{20, 20}	{10, 35}	{10, 20}

5.7.3 Tracking Accuracy

The BB-based illumination change compensation techniques are compared with four other approaches from the literature. The two techniques discussed by Young et al. [43] use the mean (referred to as 1st order) and mean and standard deviation (referred to as 2nd order) of windows of pixel values. These statistics are calculated for the image being compensated and the image depicting the desired illumination condition. The technique described by Kamikura et al. [45] likewise computes parameters over windows of pixels to minimize the mean-squared error for the pixels in each window. The technique described by Suau et al. [46] adjusts interpolated mean and variance values toward target mean and variance values across multiple resolutions and fuses the results. Because this technique only adjusts the intensity channel and does not manipulate color components, grayscale foreground detection is used for this case. These techniques are chosen for comparison because they are of similar structure and complexity to the proposed methodology. Extensive calibration is not required, no assumptions about the environment or light source are used, and data used for compensation is extracted directly from pixels near the region of interest.

The BB-based techniques and the techniques from the literature do not all have the same optimum operating points. Several combinations of illumination compensation and blob-forming parameters were tested, and only the best results from each method are shown. Wide ranges of classification thresholds (values of 5-20) and compensation tile sizes (8, 10, 16, 20, 32, 40, 80) were tested. The BB-based approaches function best with large tile sizes to better observe BB. In early experiments [16], 1st and 2nd order techniques were tuned to maintain solid object silhouettes, and suffered from excessive false foreground. However, by making the tile size very small (a value of 8) and the classification threshold large (values of 15-20), the 1st order and 2nd order techniques are able to suppress false foreground noise at the expense of object solidity. Small tile sizes are required for these techniques to prevent compensating many disparate surfaces at once. Blob-forming size constraints are adjusted to discriminate blobs from background clutter for each traffic type. Compensation and blob-forming parameters for each technique and sequences are shown in Table 20. Each compensation technique is run identically for all tracking algorithms to produce consistent foreground masks. Tracking algorithm parameters are constant for all compensation techniques.

Table 21. Kinematic Tracking Accuracy During Illumination Changes

Sequence	BB-Offset	BB-Global	BB-Hybrid	MinMSE	1st	2nd	MultiRes
Cars1	94%	95%	98%	86%	60%	63%	95%
Cars2	96%	98%	98%	93%	92%	88%	97%
Cars3	99%	99%	100%	96%	92%	85%	76%
Ped1	98%	100%	91%	5%	30%	37%	63%
Ped2	88%	94%	94%	76%	88%	80%	0%
Ped3	100%	100%	100%	0%	88%	83%	0%
PETs1	87%	83%	96%	20%	92%	93%	83%
PETs2	74%	76%	92%	19%	92%	89%	71%
AVG	**93%**	**94%**	**97%**	**62%**	**82%**	**79%**	**73%**

Table 22. OSC Tracking Accuracy During Illumination Changes

Sequence	BB-Offset	BB-Global	BB-Hybrid	MinMSE	1st	2nd	MultiRes
Cars1	84%	93%	88%	73%	63%	56%	63%
Cars2	70%	74%	71%	60%	78%	64%	54%
Cars3	91%	86%	84%	72%	75%	75%	62%
Ped1	81%	100%	49%	5%	40%	51%	35%
Ped2	72%	74%	78%	48%	58%	46%	0%
Ped3	100%	100%	83%	0%	92%	92%	0%
PETs1	47%	30%	43%	16%	37%	45%	33%
PETs2	53%	50%	45%	21%	39%	50%	34%
AVG	**74%**	**74%**	**69%**	**46%**	**62%**	**59%**	**43%**

Table 23. SST Tracking Accuracy During Illumination Changes

Sequence	BB-Offset	BB-Global	BB-Hybrid	MinMSE	1st	2nd	MultiRes
Cars1	90%	94%	95%	88%	63%	57%	70%
Cars2	87%	91%	89%	88%	82%	80%	83%
Cars3	98%	95%	93%	90%	83%	80%	73%
Ped1	93%	100%	81%	9%	19%	28%	53%
Ped2	88%	94%	88%	50%	78%	78%	0%
Ped3	100%	100%	88%	0%	50%	63%	0%
PETs1	62%	78%	75%	18%	79%	82%	74%
PETs2	68%	74%	87%	21%	89%	87%	66%
AVG	**86%**	**90%**	**88%**	**58%**	**73%**	**72%**	**63%**

5.8. Analysis

The results of the tracking experiments are shown in Tables 21, 22, and 23, in which all combinations of compensation techniques and tracking algorithms are tested. The BB-based compensation techniques improve the accuracy of the kinematic tracker by 10% to 30% over other compensation techniques. The Min MSE method tends to overcompensate foreground objects into the background, resulting in fragmented silhouettes and inconsistent blob positions. The 1st and 2nd order methods are able to resolve most objects, but are still unable to sufficiently reduce false foreground in the Ped1 sequence. The multiresolution method handles mild illumination changes, but also produces excessive false foreground during extreme changes. Because the BB-Global technique does not appreciably affect false positive classification, and because the BB-Offset technique is often capable of resolving objects, the kinematic tracker shows little difference between them. The benefit of the BB-Global technique's improved object appearance is indicated in the SST appearance-based tracker. The BB-based compensation techniques tend to decrease in accuracy in the PETs sequences as pedestrians move farther from the camera, occupy fewer pixels, and blend in with the desaturated road surface.

Variations in performance are more pronounced in the appearance-based trackers. The overall accuracy of the OSC tracker is lower than that of the other two. The OSC appearance descriptor is relatively sensitive to the exact foreground mask used to calculate the average hue vector. Even without occlusions, the IDs associated with object pairs tend to exchange frequently and interrupt tracking continuity. This effect is shown in Figure 45, where the colored tags used to indicate identity have been manually annotated with numbers for clarity in grayscale. The BB-Global compensation technique is used and produces well-defined blobs and low background noise, yet identification labels are traded among objects several times. The BB-Hybrid method suffers from this effect more than the BB-Global method. Objects of interest are sufficiently localized, but the presence of false objects in the background provides the means for the OSC feature to lose reliability. The other tracking techniques frequently prove capable of maintaining an accurate track on objects of interest despite the presence of false objects. The SST approach—though more computationally intensive due to frequent evaluation of the bivariate distribution function—is considerably more stable even without the implementation of its collision detection component, and performs similarly to the kinematic tracker.

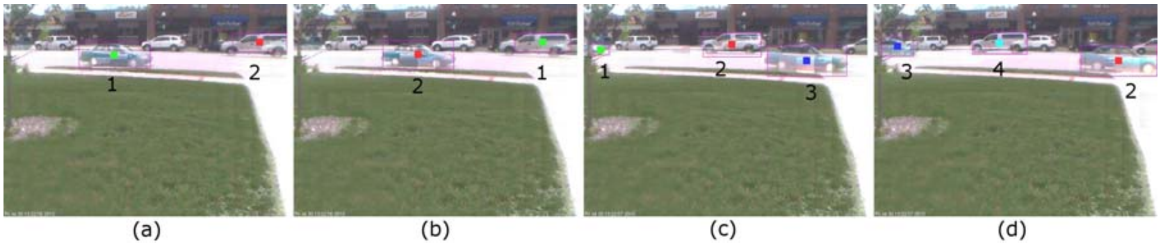


Figure 45. Example of OSC tracker mistakenly trading identities between well-defined objects during the Cars2 sequence. Colored tags and numbers indicate identities as chosen by tracker. (a) Frame 519, (b) Frame 520, (c) Frame 528, (d) Frame 530. Images were captured at 1 fps.

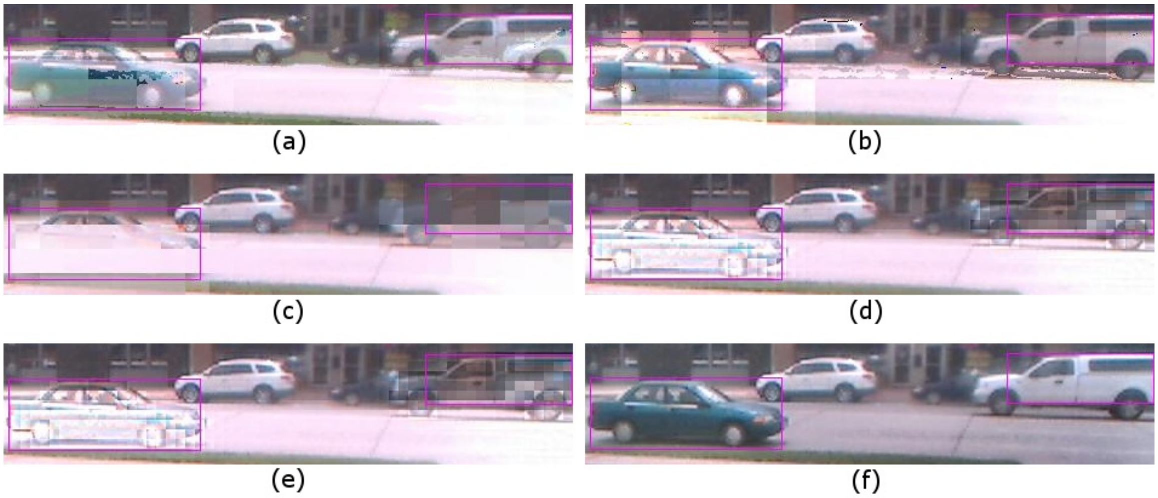


Figure 46. Examples of object appearance in Cars2 sequence after each compensation technique. (a) BB-Global, (b) BB-Hybrid, (c) Min MSE, (d) 1st Order, (e) 2nd Order, and (f) Mutiresolution.

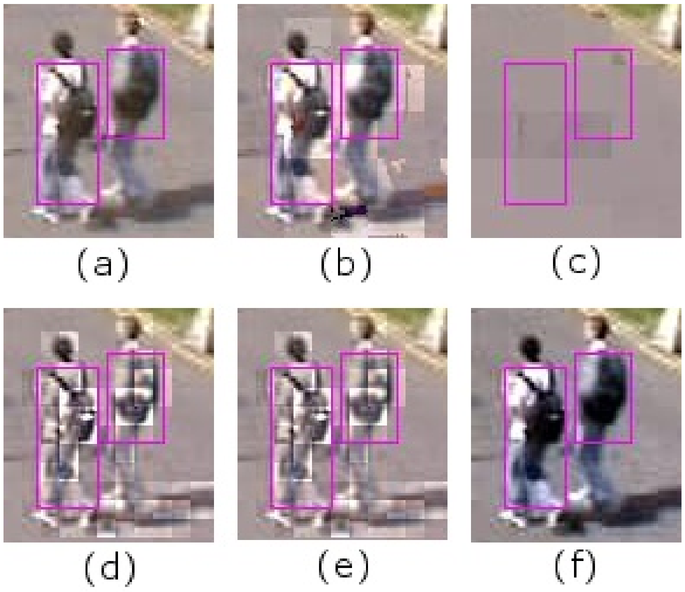


Figure 47. Examples of object appearance in PETs1 sequence after each compensation technique. (a) BB-Global, (b) BB-Hybrid, (c) Min MSE, (d) 1st Order, (e) 2nd Order, and (f) Mutiresolution.

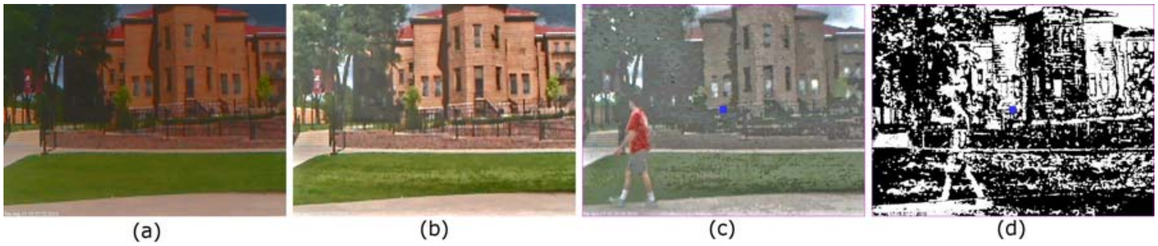


Figure 48. Multiresolution compensation tracking failure. In some instances, the multiresolution compensation technique results in tracking failure because false foreground prevents adequate object location, as shown here. (a) Original background image, (b) compensated background image, (c) compensated new image, and (d) foreground mask produced by comparing intensities of (c) with those of (b).

The samples in Figures 46 and 47 demonstrate the difficulties some appearance-based trackers may have with different compensation approaches. The BB-Global method was used to compensate all of these images for blob extraction, but the appearance in each image is due to the regular corresponding compensation method. The outputs of BB-based methods are shown in (a) and (b), and reveal objects with generally smooth, solid appearance. The MinMSE output is shown in (c), and clearly indicates a tendency to over-optimize the compensated image. This type of transformation can actually increase the magnitude of correspondence metrics used for tracking, depending on the appearance of the background behind the objects being tracked. To a lesser degree, the 1st order and 2nd order compensation methods also blend objects into the background and desaturate their features. Object appearance is fragmented due to differences in compensation factors across tiles. The multiresolution compensation method causes the least appearance distortion, and actually improves contrast in these examples. However, its tracking accuracy is consistently among the lowest of the compensation techniques because it often obscures objects with false foreground. Recall that this technique does not transform one image to look more like another, but rather transforms images toward a common mean and variance distribution. Therefore when using it for illumination compensation, we first compensate the background model with the same parameters for use in change detection. As shown in Figure 48, the processed background model (b) and the processed post-illumination change image (c) differ significantly in appearance, resulting in widely distributed false foreground (d).

As previously mentioned, the 1st Order and 2nd Order techniques leave excessive false positives in the background, making it difficult to resolve true objects. When classification thresholds are increased and tile sizes are decreased to reduce false positives, object solidity is compromised. Compared at this operating point, the BB-Hybrid technique actually improves foreground object silhouettes and prevents them from being drastically dissolved into the background. The BB-Hybrid technique offers a more consistent balance between true positive and false positive classification.

It is logical to compensate an illumination-shifted image and use the result throughout the remainder of the processing pipeline. Next the adverse effects that such compensation processes may have on appearance-oriented tracking are considered. The BB-Global compensation technique is used to preprocess the input to change detection and blob formation. The resulting blob locations and foreground masks are passed to the OSC and SST trackers. However, the trackers receive raw, uncompensated image from which to extract object appearance. Tracking accuracy from this configuration is seen in Table 24. Comparing with the BB-Global columns of Tables 22 and 23, accuracy is improved; since blob and foreground pixel locations are constant, it can be inferred that the improvement is due to the trackers extracting more consistent descriptors from untransformed images.

Table 24. Tracking Accuracy for Raw Images

	OSC		SST	
Sequence	BB-Global	Raw	BB-Global	Raw
Cars1	93%	95%	94%	99%
Cars2	74%	88%	91%	95%
Cars3	84%	92%	95%	99%
Ped1	100%	100%	100%	100%
Ped2	68%	76%	94%	96%
Ped3	100%	100%	100%	100%
PETs1	30%	64%	78%	78%
PETs2	48%	58%	70%	80%
AVG	**73%**	**85%**	**90%**	**93%**

5.9. Execution Time Comparison

This section considers the relative execution speeds for the illumination compensation techniques and the tracking algorithms. Each of the discussed algorithms was implemented in C or C++. Testing was done on a PC running Ubuntu 10.04 and equipped with a 3.4 GHz Pentium D and 1GB of RAM. The same coding style was used for each algorithm, so while additional optimizations may be possible to improve absolute frame rate, this serves as a useful comparison for relative performance. Data collection and file I/O processes were not included in these measurements. Because the tracking sequences consist of several hundred frames instead of a few hand-picked frames as in Chapter 4, trials are not repeated and averaged. Each compensation technique used the same parameters that produced its optimum tracking results in Sections 5.7 and 5.8. The average runtimes (in milliseconds) are shown in Figure 49 and Table 25 for each method. In addition to pure runtime, an additional effort metric is also considered which takes the ratio of runtime (in milliseconds) to tracking accuracy rate (in percent). This gives a sense of how much effort is expended for each compensation technique per correct tracking result.

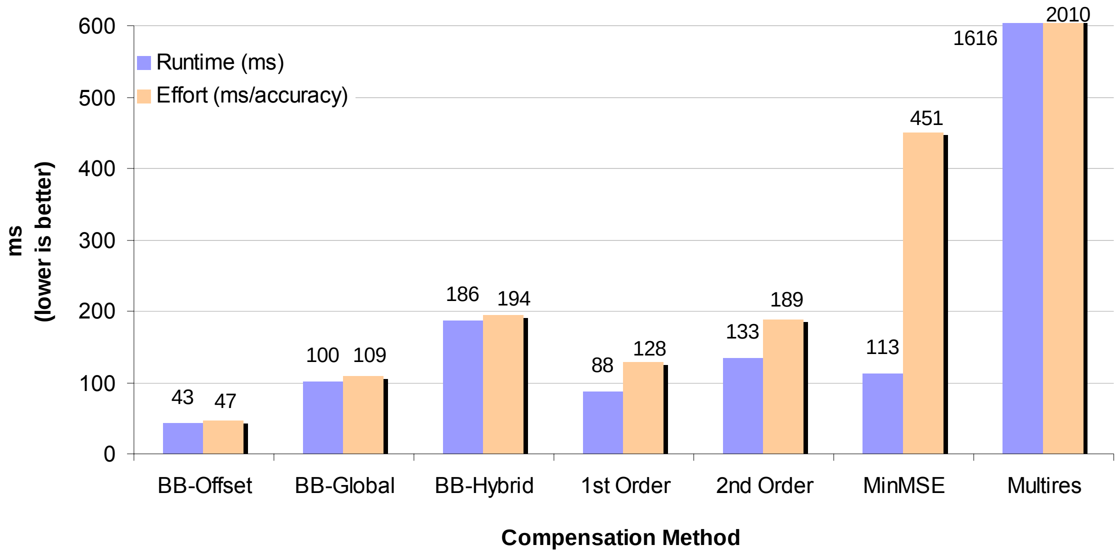


Figure 49. Comparison of runtimes and effort figures of compensation techniques.

Table 25. Average runtimes and effort figures for compensation techniques.

	Runtime (ms)	Effort (ms/accuracy)
BB-Offset	42.7	47.0
BB-Global	100.1	108.6
BB-Hybrid	185.8	193.6
1st Order	87.6	128.1
2nd Order	133.4	189.2
MinMSE	112.8	451.1
Multires	1616.0	2010.2

Because the Multiresolution technique has such a large runtime, the y-axis of the bar graph in Figure 49 is truncated at 600 ms for readability. The trends are similar to those observed in Section 4.8.2. For the tracking sequences, the BB-Offset technique runs in half the time of the 1st Order technique (the next-fastest performer). The BB-Global variation requires an extra pass through the image to determine the mean scene-wide appearance of each BB region, incurring a 100% penalty over BB-Offset as coded. This could be optimized at the expense of memory usage by storing global and local results during a single image pass. The BB-Hybrid technique, which resulted in a significant accuracy improvement over the 2nd Order method, also requires about 40% more time to execute than the 2nd Order method since the 2nd Order computations are essentially being repeated for each BB region. In terms of effort, the BB-based methods achieve low error rates and thus cost less per correct tracking result. Considering this perspective makes up for the increase in BB-Hybrid runtime, placing it on fairly even footing with the 2nd Order technique. Due to their lower accuracy rates, the MinMSE and Multiresolution techniques have substantially higher effort costs than their pure runtimes suggest.

Table 26 shows the runtimes in microseconds for the tracking methods applied to each tracking sequence. The runtimes of single tracking function calls were often smaller than the operating system could resolve, so each tracker was run 1000 times and the resulting accumulated time was divided by 1000. The architecture of the evaluation processing pipeline distills video information into short, compact lists of objects for trackers to analyze. Tracking performance is therefore sensitive to object list formation, and relative runtime increases sharply as the object lists get longer (e.g., if illumination variations or background changes cause false object detections). For this experiment, all images were compensated by the BB-Global technique. Because the same compensation method and blob forming settings were used for each tracker, each tracker received identical lists of blobs to analyze.

The kinematic tracker requires the least overhead per object, relying solely on position information that is stored in the object list data structure. The object-strip color approach must compute an OSC descriptor for each object list entry by scanning the pixels that make up the object. The conversion to the HSI is relatively cheap. Correspondence is established by a one-dimensional convolution-like function between OSC vectors. The OSC tracker takes an order of magnitude longer to run than the kinematic tracker. The spatio-spectral tracker also scans the pixels that make up each object, but uses a costly floating-point bivariate Gaussian probability function to evaluate correspondence likelihoods. While it achieves 15% greater tracking accuracy on average than the OSC tracker, the SST tracker requires an average of 50 times longer to run than the kinematic method and 3 times longer to run than the OSC method.

Table 26. Average runtimes (in microseconds) of tracking algorithms.

	kin	osc	sst
Cars1	1.1	43.3	196.4
Cars2	2.4	64.9	294.4
Cars3	1.1	25.0	112.8
Ped1	3.1	46.9	153.6
Ped2	2.3	25.2	86.2
Ped3	1.0	7.9	34.5
PETs1	6.8	40.5	138.8
PETs2	6.6	29.4	93.0
AVG	**3.0**	**35.4**	**138.7**

5.10. Conclusions

Illumination changes can drastically alter the appearance of a scene, making it difficult to distinguish objects from background noise during change detection. Illumination compensation algorithms improve the quality of change detection, and transform the visual appearance of interesting objects in the process. It is desirable to transform the image so objects of interest can be reliably located while preserving the visual features that distinguish those objects from each other. Four illumination compensation techniques and three object tracking algorithms from the literature were implemented for the purpose of exploring illumination compensation effects on object tracking problems.

In addition to examining the effects of the RGB translation compensation technique (BB-Offset) on tracking methods, this chapter has presented two new variations of illumination compensation based on the BigBackground model. The first variation (BB-

Global) improves the offset technique introduced in Chapter 4 by incorporating scene-wide illumination change measurements into the local compensation model. The second variation (BB-Hybrid) demonstrates that the BigBackground approach can be applied to the benefit of other independently developed techniques.

By obtaining a global sense of BigBackground regions' responses to illumination change and incorporating this information into the compensation model, true positive detection is improved by an average of 16% over an earlier BigBackground-based approach with negligible impact on false positive rates. In addition to producing more coherent object silhouettes, the global approach better preserves the visual appearance of those objects. In contrast, several compensation techniques from the literature tend to dissolve objects, making their appearance a function of the background that they occlude. Appearance-based trackers then have less reliable object descriptors with which to perform correspondence. When the illumination compensation algorithms are applied to three tracking methods, the BigBackground-based techniques result in a 20% average increase in tracking accuracy for each tracker. The hybridization of BigBackground and the 2nd-order compensation method reduces distracting false positives by an average of 25% and improves tracking accuracy by an average of 15% over 2nd-order compensation alone. Finally, it is demonstrated that while BigBackground-based techniques cause less appearance distortion than other compensation approaches, improved appearance-based tracking accuracy can be achieved by using illumination compensated images for object location and extracting appearance descriptors from the raw, uncompensated images. This work has demonstrated the benefits of using stable, chromatically diverse background features as reference points for illumination change compensation in object tracking processes.

CHAPTER 6

CONCLUSION

This dissertation has investigated problems in surveillance video analysis caused by illumination changes, and has presented solutions to improve analysis resilience to these problems. Controlled experiments yielded insight into the underlying responses of scenes to temporal lighting variations. Models and methods were developed to quantify and compensate for lighting variations in a manner consistent with those observations. The novel methods were compared with prior art across several dimensions of evaluation, from pixel-level classification accuracy to high-level object tracking accuracy. Experiments used video sequences captured by low-cost webcameras in a wide range of realistic surveillance environments. Algorithms were developed and implemented in C.

First, experiments with a controllable light source and color targets demonstrated that surfaces of significantly different hue have different responses to a given change in illumination. Also, several illumination models from the literature were shown to achieve different degrees of effectiveness when optimized globally for a scene, but all achieved similar effectiveness rates when optimized for each chromatically distinct region. These effects were also observed in a staged scene containing realistic surfaces at non-uniform orientations, and formed the basis for a novel approach to temporal illumination compensation.

Second, the BigBackground model was developed for representing large, stable, chromatically distinct background features. An algorithm was presented that extracts these background features by finding the dominant colors in the scene, clustering similar colors into a reduced color palette, and mapping relevant pixels to palette entries. The

BigBackground model was evaluated in terms of scene coverage and stability, and its responses to input variation was characterized.

Third, the stability and chromatic diversity traits of the BigBackground model were used as reference points for measuring the effects of illumination change. Differences in BigBackground region appearance before and after a lighting change were applied to transform the post-illumination change image to appear as it would under the original lighting condition. Effectiveness was measured in a change detection application since change detection often produces vast amounts of false foreground during lighting variations. Ground truth images were chosen for several sequences, and were hand-marked for correct foreground/background classification. BigBackground-based compensation was compared with several techniques from the literature.

Fourth, two variations of BigBackground-based illumination compensation were presented. The first variation improved the original technique by incorporating global measurements, while the second variation demonstrated the integration of the BigBackground philosophy into an independently-developed compensation method. These new techniques, along with the previously discussed methods, were evaluated in an object tracking pipeline in which objects were tracked during and after illumination changes. Counting foreground and background pixel classification rates told an incomplete story, as it did not convey what types and sizes of errors could be tolerated by downstream processes. Object tracking provided a framework for evaluating compensation effects on practical, high-level analyses over many hundreds of frames, and thus was a more comprehensive evaluation method than comparing ground truth frames (though studying this low-level behavior was a necessary first step).

6.1. Summary of Results

The main results of this dissertation are summarized here.

6.1.1 Chromatic sensitivity of scene response to illumination change

- The effects of illumination changes on surfaces are quantitatively shown to depend on the chromaticities of those surfaces [13].

- Customizing compensation models for distinct colors improves model effectiveness by reducing SAD error by 70% to 80%. Applying a model customized for one surface to a surface of different hue multiplies the error rate by an average factor of 15, while applying such a model to other surfaces of similar hue only increases the error rate by an average factor of 4 to 6.

- Choice of the mathematical model used for illumination compensation has reduced impact on compensation accuracy when the model is customized for each chromatically distinct region.

6.1.2 BigBackground: dominant chromatic feature extraction

- A background model is presented that identifies large, stable scene components by extracting dominant chromatically self-similar regions [14]. These regions often cover more than 50% of a scene—and can cover over 90%—and are 20% more stable on average than other regions in the scene.

6.1.3 BigBackground-based illumination compensation

- Compensation model parameters are customized for each BigBackground region, allowing chromatically distinct regions to respond independently to the same illumination change. A simple RGB Translation model is shown to be more effective than models based on gain or the HSI color space [14].

- When used to aid foreground detection, this approach decreases false positives by an average of 83% compared with no corrective action, and decreases false positives by 25%-43% compared with other compensation methods from the literature.

6.1.4 Expansion of BigBackground concept to versatile methodology

- A variation of BigBackground-based compensation that incorporates global measurements of scene change increases true positive foreground detection by an average of 16% with negligible impact on false positive classifications (compared with the original, solely-local technique). Object silhouettes are more solid and object appearance is better preserved [15].

- The BigBackground concept is integrated into an independent second-order compensation method from the literature. The resulting hybrid exhibits less sensitivity to the foreground/background classification threshold than the original second-order technique. False positive rates are decreased an average of 25%, greater than the 16% decrease in true positive rates. The hybrid technique also preserves object appearance better than the second-order technique [15].

6.1.5 Effects of illumination compensation on object tracking mechanisms

- Kinematic and appearance-based trackers are tested, and the discussed compensation techniques are evaluated in terms of tracking accuracy and object appearance quality over several hundred frames [15], [16], [17].

- The BigBackground-based techniques improve object tracking by an average of 20% for each tracking approach, and produce less distortion in object appearance.

- Appearance-based tracking accuracy is increased by 3% to 12% by using BigBackground-based illumination compensation to localize objects, and then extracting appearance descriptors from the uncompensated image.

6.2. Future Work

Future work will consider the BigBackground model and related illumination compensation techniques over long time scales by considering the usefulness of the BB model for handling long-term illumination changes, and identifying when illumination changes occur and how that information might be used to update or extend the life of the BB model. It is desirable to avoid repeating the computation of illumination compensations, and instead rely on some easily computed metric to indicate when a model may need to be reinitialized or adapted. This work will examine how long a set of compensations can be reused before they no longer accurately reflect the lighting condition. It will also present techniques for using the illumination models to determine when significant background changes have occurred, signaling the need to update an old background model. The extraction of patterns in illumination changes over long time scales would allow recurring changes to be anticipated.

A problem related to long term adaptation is the interaction between camera and surveillance algorithm. Surveillance cameras have controls for gain, exposure, contrast, and white balance that can often be set statically, or dynamically to accommodate changing conditions. When set dynamically, these parameters allow the camera to provide the most information per pixel. However, on-the-fly adjustment may also compound the effects of lighting changes, or introduce new artifacts. It will be necessary for video analysis algorithms and camera controllers to interact. Timing of camera recalibration events will need to be coordinated to minimize disruptions to algorithms, while setting parameters to maximize the types of data sought by those algorithms.

Another area of future work is to apply architectural considerations to the presented material. The methods presented here can be optimized and evaluated for low-cost, resource-constrained platforms. In such systems, tradeoffs between memory usage and scene coverage by the reduced color palette would be more pronounced. Because the BigBackground model and the compensation techniques consider image tiles independently, there are opportunities for the exploitation of parallelism on multi-core and many-core platforms.

Graphics processing units (GPUs) have long been used to hardware-accelerate lighting effects when rendering artificial scenes, and thus may also be useful for quickly performing detailed transformations for illumination compensation in scene analysis. As shown in Chapter 2, considering chromatically distinct regions independently during an illumination change greatly improves the accuracy of the illumination model. Future work may produce techniques for extracting a model of the illumination source itself by observing distinct colors for their individual responses to illumination changes and measuring the differences in those responses. These observations might also be used to determine the nature of the illumination source (incandescent, fluorescent, or halogen or exterior sunlight) and of the illumination change (binary on/off such as interior lighting, or filtered in the case of outdoor cloud cover). The illumination source model could be applied across an image to reverse illumination changes as a more general, geometrical transformation that GPUs are well-suited to perform.

This dissertation presented one approach for identifying reliable features based on color clustering. Other approaches based on traditional color segmentation techniques, analysis of optical flow, or some combination may also provide useful features with which to partition illumination changes and are worth investigating. An approach that efficiently incorporates natural feature boundaries might eliminate the need for tiling and reduce tile-induced artifacts.

Finally, the applicability of BigBackground can be extended by integrating it with other independent compensation methods to improve resilience to illumination change, and by viewing it as a property akin to center of mass or texture and including it in a stel [26] or locale [27] type framework. There are also opportunities for exploring other applications of BB as a general salient feature set beyond the realm of change detection and surveillance, such as image registration or understanding scenes observed by moving cameras.

REFERENCES

[1] J. Vlahos. "Surveillance Society: New High-Tech Cameras Are Watching You," Popular Mechanics, January 2008.

[2] C. Harris. "Getting An Eyeful," The American City and County. Pittsfield: Nov. 2009. vol. 124, no. 11, pp 38.

[3] M. Moore. "Cities Opening More Surveillance Eyes," USA Today, July 18, 2005.

[4] J. Redmon. "Atlanta Seeks to Add 500 Surveillance Cameras," The Atlanta-Journal Constitution. Oct. 24, 2009.

[5] C. Savage. "U.S. Doles Out Millions for Street Cameras," The Boston Globe. Aug. 12, 2007.

[6] M. Cramer. "Watching From A Distance," The Boston Globe. May 3, 2010.

[7] C. Duvall. "Cameras Give Small Towns Additional Crime-Fighting Tool," Nation's Cities Weekly. March 13, 2006. vol. 29, no. 11, pp. 6.

[8] D. Koeppel. "The Holes Found in U.S. Border Fence Technology," Popular Mechanics. Aug. 2010.

[9] M. Piccardi. "Background Subtraction Techniques: A Review," *Proc. IEEE International Conference on Systems, Man and Cybernetics,* vol. 4, pp. 3099–3104, 2004.

[10] R.J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. "Image Change Detection Algorithms: A Systematic Survey," *IEEE Transactions on Image Processing,* vol. 14, no. 3, pp. 294–307, 2005.

[11] S. Cheung and C. Kamath. "Robust techniques for background subtraction in urban traffic video," *IS&T/SPIE Electronic Imaging—Visual Communications and Image Processing,* vol. 5308, pp. 881–892, 2004.

[12] C. Stauffer and W.E.L. Grimson. "Adaptive background mixture models for real-time tracking," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1999),* pp. 246–252, 1999.

[13] M.R. Bales, D. Forsthoefel, D.S. Wills, and L.M. Wills, "Chromatic Sensitivity of Illumination Change Compensation Techniques," *Int'l Symposium on Visual Computing.* G. Bebis et al. (Eds.): ISVC 2010, Part I, LNCS 6453, pp. 211–220. Springer, Heidelberg (2010).

[14] M.R. Bales, D. Forsthoefel, B. Valentine, D.S. Wills, and L.M. Wills, “BigBackground-Based Illumination Compensation for Surveillance Video,” *EURASIP Journal on Image and Video Processing,* vol. 2011, Article ID 171363, 22 pages, 2011.

[15] M.R. Bales, D. Forsthoefel, B. Valentine, D.S. Wills, and L.M. Wills, “Application of Illumination Compensation Techniques to Tracking Problems,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI),* under review, 14 pages, submitted February 2011.

[16] M.R. Bales, D. Forsthoefel, D.S. Wills, and L.M. Wills, “Illumination Change Compensation Techniques to Improve Kinematic Tracking,” *Proc. IEEE Workshop on Applications of Computer Vision (WACV 2011),* pp. 434–439, 2011.

[17] S. Apewokin, B. Valentine, R. Bales, L. Wills, and S. Wills, “Tracking multiple pedestrians in real-time using kinematics,” *Proc. IEEE Conference on CVPR Workshops,* pp 1-6, 2008.

[18] F. Mindru, L. Van Gool, and T. Moons. “Model estimation for photometric changes of outdoor planar color surfaces caused by changes in illumination and viewpoint,” *Proc. Int’l Conference on Pattern Recognition,* vol. 1, pp. 620–623, 2002.

[19] P. Gros. Color illumination models for image matching and indexing. *Proc. Int’l Conference on Pattern Recognition,* vol. 3, pp. 576–579, 2000.

[20] G.D. Finlayson, M.S. Drew, and B.V. Funt, “Diagonal Transforms Suffice for Color Constancy,” *Proc. of 4th International Conference on Computer Vision,* pp. 164–171, 1993.

[21] H.D. Cheng, X.H. Jiang, Y. Sun, and W. Jingli. “Color image segmentation: advances and prospects,” *Pattern Recognition*, vol. 34, no. 12, pp. 2259-2281, 2001.

[22] D. Cremers, M. Rousson, and R. Deriche. “A review of statistical approaches to level set segmentation integrating color, texture, motion and shape,” *International Journal of Computer Vision,* vol. 72, no. 2, pp. 195-215, 2007.

[23] J. Freixenet, X. Munoz, D. Raba, J. Marti, and X. Cufi. “Yet another survey on image segmentation: region and boundary information integration,” *Proc. Seventh European Conference on Computer Vision: Part III, Lecture Notes in Computer Science* vol. 2352, pp. 408-422, 2002.

[24] C.M. Christoudias, B. Georgescu, and P. Meer, “Synergism in Low Level Vision,” *Proc. 16th Int’l Conf. Pattern Recognition,* vol. 4, pp. 150–155, 2002.

[25] P. Felzenszwalb and D. Huttenlocher, “Efficient Graph-Based Image Segmentation,” *Int’l Journal of Computer Vision,* vol. 59, no. 2, pp. 167–181, 2004.

[26] N. Jojic, A. Perina, M. Cristani, V. Murino, and B. Frey, “Stel Component Analysis: Modeling Spatial Correlations in Image Class Structure,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR 2009),* pp. 2044–2051 2009.

[27] M.S. Drew, W. Jie, and L. Ze-Nian, “Illumination-invariant color object recognition via compressed chromaticity histograms of color-channel-normalized images,” *Sixth Int’l Conference on Computer Vision (ICCV 1998),* pp. 533–540, 1998.

[28] M.S. Drew, Z. Li, and Z. Tauber, “Illumination color covariant locale-based visual object retrieval,” *Pattern Recognition,* vol. 35, no. 8, pp. 1687–1704, 2002.

[29] S. Apewokin, B. Valentine, D. Forsthoefel, D.S. Wills, L.M. Wills, and A. Gentile. “Embedded real-time surveillance using multimodal mean background modeling,” *Embedded Computer Vision* (B. Kisacanin, S. Bhattacharyya, and S. Chai, eds.), Springer, pp. 163–175, 2009.

[30] M.J. Hossain, J. Lee, and O. Chae, “An Adaptive Video Surveillance Approach for Dynamic Environment,” *Proc. Int’l Symposium on Intelligent Signal Processing and Communication Systems,* pp. 84–89, 2004.

[31] Y. Wang; T. Tan, and K. Loe. “A probabilistic method for foreground and shadow segmentation,” *Proc. International Conference on Image Processing,* vol. 3, pp. 937–940, 2003.

[32] M.S. Drew, W. Jie, and L. Ze-Nian. “Illumination-invariant color object recognition via compressed chromaticity histograms of color-channel-normalized images,” *Proc. Sixth Int’l Conference on Computer Vision*, pp. 533–540, 1998.

[33] Z. Ming, J. Bu, and C. Chen, “Robust background subtraction in HSV color space,” *Proceedings of SPIE,* vol. 4861, pp. 325–332, 2002.

[34] B. Chen and Y. Lei, “Indoor and outdoor people detection and shadow suppression by exploiting HSV color information,” *International Conference on Computer and Information Technology (CIT 2004),* pp. 137 – 142, 2004.

[35] B.K.P. Horn, Robot Vision. (McGraw-Hill, New York, NY, U.S., 1986), pp. 185–216.

[36] Y. Weiss, “Deriving intrinsic images from image sequences,” *IEEE International Conference on Computer Vision (ICCV 2001),* vol. 2. pp. 68–75, 2001.

[37] G.D. Hager and P.N. Belhumeur, “Real-time tracking of image regions with changes in geometry and illumination,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 1996),* pp. 403 – 410, 1996.

[38] Y. Matsushita, K. Nishino, K. Ikeuchi, and M. Sakauchi, “Shadow elimination for robust video surveillance,” *Workshop on Motion and Video Computing (WMVC 2002),* pp. 15 – 21, 2002.

[39] H. Wu, P. Jiang and J. Zhu, “An Illumination Compensation Method for Images under Variable Lighting Condition,” *IEEE Conference on Robotics, Automation and Mechatronics (CRAM 2008),* pp. 1022–1026, 2008.

[40] E.G. Miller and K. Tieu, “Color eigenflows: statistical modeling of joint color changes,” *IEEE Int’l Conference on Computer Vision (ICCV 2001),* pp. 607–614, 2001.

[41] G. Finlayson, C. Fredembach, and M.S. Drew, “Detecting Illumination in Images,” *IEEE Int’l Conference on Computer Vision (ICCV 2007),* vol. 1, 1–8, 2007.

[42] Y. Makihara, Y. Shirai, and N. Shimada, “Online learning of color transformation for interactive object recognition under various lighting conditions,” *Int’l Conference on Pattern Recognition (ICPR 2004),* vol. 3, pp. 161–164, 2004.

[43] S. Young, M. Forshaw, and M. Hodgetts, “Image comparison methods for perimeter surveillance,” *Int’l Conference on Image Processing and Its Applications (CIPA 1999),* vol. 2, pp. 799–802, 1999.

[44] J. Lu, G. Lafruit, F. Catthoor, “Streaming-Mode MB-Based Integral Image Techniques for Fast Multi-view Video Illumination Compensation,” *Pacific Rim Conference on Multimedia (PCM 2006),* (Lecture Notes in Computer Science vol. 4261), pp. 414-423, 2006.

[45] K. Kamikura, H. Watanabe, N. Kobayashi, S. Ichinose, and H. Yasuda, “Video coding using global motion and brightness-variation compensation with complexity reduction,” *Electronics and Communications in Japan (Part I: Communications)*, vol. 86, pp. 80–93, 2003.

[46] X. Suau, J.R. Casas, and J. Ruiz-Hidalgo, “Multi-Resolution Illumination Compensation for foreground extraction,” *IEEE Int’l Conference on Image Processing: 2009*. Proc. ICIP 2009. pp. 3225–3228, 2009.

[47] J.A. Vijverberg, M.J.H. Loomans, C.J. Koeleman, and P.H.N. de With, “Global Illumination Compensation for Background Subtraction Using Gaussian-Based Background Difference Modeling,” *IEEE Int’l Conference on Advanced Video and Signal Based Surveillance: 2009*. Proc. AVSS 2009. pp. 448–453 2009.

[48] X. Xie and K. Lam, “An Efficient Illumination Compensation Scheme for Face Recognition,” *Control, Automation, Robotics and Vision Conference: 2004.* Proc. ICARCV 2004, vol. 2, pp. 1240–1243, 2004.

[49] K. Wong, K. Lam, and W. Siu, “An efficient color compensation scheme for skin color segmentation,” *Int’l Symposium on Circuits and Systems: 2003.* Proc. ISCAS 2003, vol. 2, pp. 676–679, 2003.

[50] M.R. Yelal and S. Sasi, “Human tracking in real-time video for varying illumination,” *IEEE Int’l Workshop on Intelligent Signal Processing: 2005.* Proc. IWISP 2005, pp. 364–369, 2005.

[51] J. Cai, M. Shehata, W. Badawy, and M. Pervez, “An Algorithm to Compensate for Road Illumination Changes for AID Systems,” *IEEE Intelligent Transportation Systems Conference: 2007.* Proc. ITSC 2007. pp. 980–985, 2007.

[52] J. Li, S.Z. Li, Q. Pan, and T. Yang, “Illumination and motion-based video enhancement for night surveillance,” *IEEE Int’l Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance: 2005.* Proc. IWVSPETS 2005. pp.169–175, 2005.

[53] A. Yilmaz, O. Javed, and M. Shah, “Object Tracking: A Survey,” *ACM Computing Surveys,* vol. 38, no. 4, pp. 1–45, 2006.

[54] D.G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int’l Journal of Computer Vision,* vol. 60, no. 2, pp. 91–110, 2004.

[55] M.S. Rahman, A. Saha, and S. Khanum, “Multi-object Tracking in Video Sequences Based on Background Subtraction and SIFT Feature Matching,” *Proc. Int’l Conference on Computer Sciences and Convergence Information Technology 2009,* pp. 457–462, 2009.

[56] S. Zhang, H. Yao, and P. Gao, “Robust object tracking combining color and scale invariant features,” *Visual Communications and Image Processing 2010, Proceedings of the SPIE*, vol. 7744, 8 pages, 2010.

[57] J. Durand and S. Hutchinson, “Real-time object tracking using multiresolution critical point filters,” *IEEE Int’l Conference on Robotics and Automation. Proc. ICRA 2003,* vol. 2, pp. 1682–1687, 2003.

[58] T. Dharamadhat, K. Thanasoontornlerk, and P. Kanongchaiyos, “Tracking object in video pictures based on background subtraction and image matching,” *IEEE Int’l Conference on Robotics and Biomimetics 2008. Proc. ROBIO 2009,* pp. 1255–1260, 2009.

[59] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 25, no. 5, pp. 564–577, 2003.

[60] F. Porikli and O. Tuzel, "Multi-kernel object tracking," *IEEE Int'l Con-ference on Multimedia and Expo. Proc. ICME,* 4 pages, 2005.

[61] J. Jeyakar, R.V. Babu, and K.R. Ramakrishnan, "Robust Object Tracking Using Local Kernels and Background Information," *IEEE Int'l Conference on Image Processing. Proc. ICIP 2007,* pp. 49–52, 2007.

[62] L. Zhang and R. Wu, "Tracking Object Using Object-strips Color Feature," *WASE Int'l Conference on Information Engineering. Proc. ICIE 2010,* pp. 212–215, 2010.

[63] A. Tavakkoli, M. Nicolescu, and G. Bebis, "A Spatio-Spectral Algorithm for Robust and Scalable Object Tracking in Videos," *Int'l Symposium on Visual Computing: 2010.* G. Bebis et al. (Eds.): ISVC 2010, Part III, LNCS 6455, pp. 161–170. Springer, Heidelberg, 2010.

[64] K. Rangarajan and M. Shah, "Establishing motion correspondence," *Proc. IEEE Conference on CVPR,* pp 103–108, 1991.

[65] O. Masoud and N.P. Papanikolopoulos, "A novel method for tracking and counting pedestrians in real-time using a single camera," *IEEE Transactions on Vehicular Technology,* vol. 50, no. 5, pp. 1267–1278, 2001.

[66] PETS, "The PETS 2001 Dataset 3," 2001, ftp://ftp.pets.rdg.ac.uk/pub/PETS2001/DATASET3/.

VITA

Michael Ryan Bales

M. Ryan Bales was born on March 31, 1982 in Houston, Texas. He received both his primary and secondary education in De Soto, Missouri, graduating valedictorian from De Soto High School in 2000. He attended the University of Missouri – Rolla beginning in 2000. In May 2004 he graduated *magna cum laude* with a Bachelors of Science in Computer Engineering as an Honors Academy Fellow, with minors in physics and writing. He began his Masters program in June 2004, and worked as a teaching assistant in ECE laboratories. In May 2006 he earned a Masters of Science in Electrical Engineering from UMR with emphases in VLSI, video compression, and embedded systems. In January 2007, he started at Georgia Tech to pursue a doctorate in embedded vision systems. From April 2008 to May 2011, he worked as a research assistant for GTRI in the Sensors and Electromagnetic Applications Lab, where he worked on FPGA designs for radar and data recording systems.

Ryan has been a member of the Institute of Electrical and Electronics Engineers (IEEE) since 2002. He was inducted into Phi Eta Sigma and Eta Kappa Nu honor societies in 2001 and 2003, respectively. In Eta Kappa Nu, he has served in several officer positions including President, Vice President, and Bridge Correspondent.