Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2011, Article ID 171363, 22 pages
doi:10.1155/2011/171363

Research Article

BigBackground-Based Illumination Compensation for Surveillance Video

M. Ryan Bales, Dana Forsthoefel, Brian Valentine, D. Scott Wills, and Linda M. Wills

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

Correspondence should be addressed to M. Ryan Bales, mbales3@gatech.edu

Received 25 April 2010; Revised 26 October 2010; Accepted 13 December 2010

Academic Editor: Luigi Di Stefano

Illumination changes cause challenging problems for video surveillance algorithms, as objects of interest become masked by changes in background appearance. It is desired for such algorithms to maintain a consistent perception of a scene regardless of illumination variation. This work introduces a concept we call BigBackground, which is a model for representing large, persistent scene features based on chromatic self-similarity. This model is found to comprise 50% to 90% of surveillance scenes. The large, stable regions represented by the model are used as reference points for performing illumination compensation. The presented compensation technique is demonstrated to decrease improper false-positive classification of background pixels by an average of 83% compared to the uncompensated case and by 25% to 43% compared to compensation techniques from the literature.

1. Introduction

Automated video surveillance has become increasingly attractive for monitoring environments that are difficult or dangerous for human operators to monitor. With the proper algorithms, 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. The first stages of most such 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 [1–3]. 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 may be analyzed further to recognize or track objects, or identify events.

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. The conceptual motivation behind this work is easily seen in Figure 1, which shows a scene before and after a significant lighting change. It is desired for surveillance algorithms to monitor such scenes, reliably observe foreground objects, and filter out persistent background despite such lighting changes.

In this paper, we present 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

(a) (b)

Figure 1: Example of a scene before and after a lighting change.

without requiring complete reinitialization. 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. 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 runtime tests, the proposed algorithm performs among the fastest techniques tested at 15 to 20 frames per second.

This paper is organized as follows. Section 2 describes related work in the field of illumination compensation and region modeling in video analysis. Section 3 introduces the BigBackground model, and demonstrates its stability characteristics. Section 4 describes four mathematical models for illumination compensation. Section 5 describes the addition of a color clustering step to the BigBackground algorithm and evaluates the effect of clustering on coverage and stability. Section 6 compares our technique to several illumination compensation techniques from the literature in terms of accuracy and run-time, while Section 7 provides the paper's conclusion.

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 [4] 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 3×3 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. [5] 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 approach to illumination compensation that we present 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 [6, 7] or chromaticity [8–10] 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 [6]. 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 [11] 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. In [12], 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 [13], 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. [14] first estimates 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 and Tieu [15] 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. A physics-based reflection model is used by Makihara et al. [16] 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 [17], 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 3×3 to 31×31 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 (1) and (2) 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 σ_1 and σ_2. This second-order model is also proposed by Lu et al. [18] for block-based illumination compensation in multiview 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

$$I_2(x,y)' = \frac{\overline{I_1}}{\overline{I_2}} I_2(x,y), \tag{1}$$

$$I_2(x,y)' = \left(I_2(x,y) - \overline{I_2}\right)\left(\frac{\sigma_2}{\sigma_1}\right) + \overline{I_1}. \tag{2}$$

A third method based on local statistics is proposed by Kamikura et al. [19] 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 pixelwise affine transformation is used of the type shown in (3). The gain and offset parameters (c and d, resp.) are chosen to minimize mean square error, with the optimal solution given by (4). The statistics used to calculate c and d are given in (5). 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

$$I_2'(x,y) = cI_2(x,y) + d, \tag{3}$$

$$\begin{aligned} c &= \frac{N \cdot S - P \cdot Q}{N \cdot T - P^2}, \\ d &= \frac{T \cdot Q - P \cdot S}{N \cdot T - P^2}, \end{aligned} \tag{4}$$

$$\begin{aligned} P &= \sum_{x,y \in R} I_1(x,y), \\ Q &= \sum_{x,y \in R} I_2(x,y), \\ S &= \sum_{x,y \in R} I_1(x,y) \cdot I_2(x,y), \\ T &= \sum_{x,y \in R} I_2^2(x,y). \end{aligned} \tag{5}$$

The technique described by Suau et al. [20] 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 (6), 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

$$Y' = \frac{1}{L} \sum_{k=1}^{L} \left[(Y - M_k) \cdot \frac{\sigma_o}{1 + \sqrt{V_k}} + \mu_0 \right]. \tag{6}$$

Another illumination compensation method that exploits local area statistics is given by Vijverberg et al. [21]. 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 deviations σ_k from the best-fitting model are used to derive the classification threshold of (7), where k denotes the component of a multimodal background model, $\Delta(x,y)$ is the background difference, and $T_k = \text{MAX}(T_{\min}, 1.5\sigma_k)$. This technique is intended for global illumination changes, and, as described, does not include a mechanism for handling partial changes

$$F(x,y) = \begin{cases} \text{foreground} & \text{if } \Delta(x,y) - \mu_k > T_k \ \forall k, \\ \text{background} & \text{else.} \end{cases} \tag{7}$$

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 [22–24]. Block-based histogram equalization is used in [22] 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 [23] 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 [24], 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 [13, 25, 26] focuses on correcting particular types of illumination variation. Static glare removal is considered in [25]. 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 [26] 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 [13] 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 intraframe illumination variation rather than time-varying interframe variation. The goal for intraframe 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. Interframe compensations maintain constancy during temporal lighting changes.

The concept of BB shares some philosophical similarities with the "locales" framework introduced in [27], and with the "stels" introduced in [28]. Locales provide a framework for localizing features of interest in an image and consist of a set of square tiles (called an envelope) and geometric features such as centroid or mass which those tiles have in common. These geometric features are measured in terms of the pixels within a tile, and a tile is said to have a particular feature (and is therefore a member of the corresponding locale's envelope) if a minimum density of the pixels within the tile have that feature. Stels use an unsupervised, self-learning mechanism to segment images according to regions of similar surface normal, color or texture. By basing the segmentation based on similarities within a single image, similar segmentation results can be achieved in other images that have similar features, but that were taken under different environmental conditions or from different perspectives.

3. BigBackground

The influence of an illumination change on a scene is complex. The color and intensity of the light source and the position and distance of the light source relative to the scene determine how the scene will respond. These factors—combined with the orientations of objects within the scene—determine the impact of shadows. A great deal of information and computation is needed to model a physical light source and predict how its evolution will alter the appearance of a scene. However, if the effects of a light change on reliable reference points can be observed, those effects can be quantified and applied to nearby points to estimate their appearance. This section describes a method for selecting such reference points and for deriving an illumination correction.

Within a scene, there are often large, stationary objects of relatively homogeneous color. Buildings, roads, and tree lines are examples. 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]. We choose Multimodal Mean for this study, and describe the process 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.

We choose to use maximum component differences (MCD), rather than sum of absolute differences (SAD), in most of our pixel-comparison routines. The MCD is the largest difference between two color components, as shown in

$$\mathrm{MCD} = \mathrm{MAX}(\mathrm{abs}(R1 - R2),\ \mathrm{abs}(G1 - G2),\ \mathrm{abs}(B1 - B2)). \tag{8}$$

The process for identifying BigBackground colors is as follows. The image is separated into square tiles (Rsize × 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 (see Figure 2). 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 the number of pixels that matched each color. The colors are converted from ratiometric to scalar representation. 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 top Cnum colors are chosen as the BB color palette. 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 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.

3.1. 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

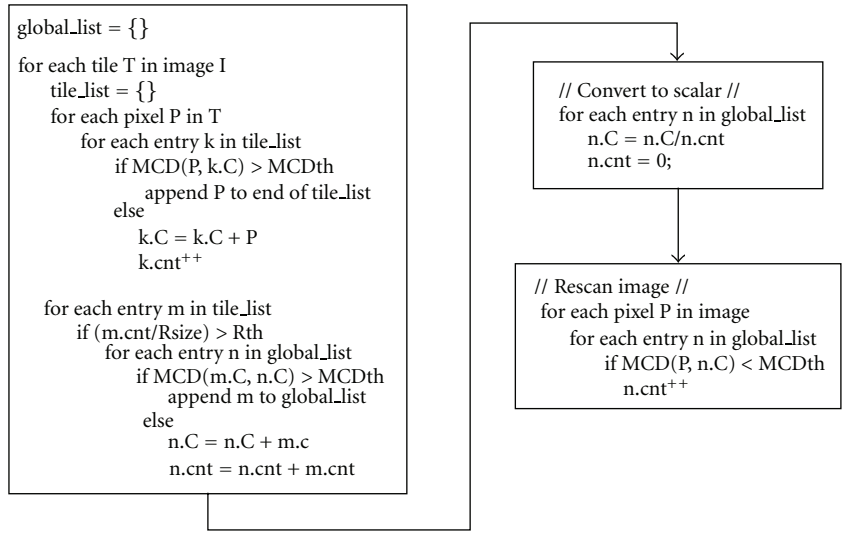


Figure 2: Process for identifying BigBackground. Color list entries such as m and n consist of an RGB triple (denoted C) and a count of the number of times a color is observed (cnt).

uses results from BB and another background model known as Multimodal Mean, which is introduced fully in [29], but 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.

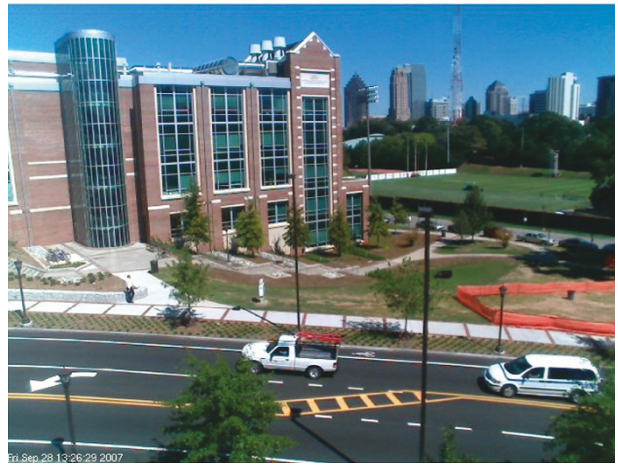
Here we evaluate the relative temporal stabilities of BB pixels and non-BB pixels. Illumination change effects are considered in the next section. Six test videos with no appreciable illumination changes are chosen to test BB stability. Sample images from these videos are shown in Figure 3. As part of the evaluation of BB's stability, we apply Multimodal Mean 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 we have computed the predominant image and the BB model, we proceed to analyze the next 100 frames in the sequence. We compare each pixel in each new frame with its corresponding pixel in the predominant image and count how many BB and non-BB pixels continue to match their original predominant pixel values within an MCD threshold.

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 serve 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.

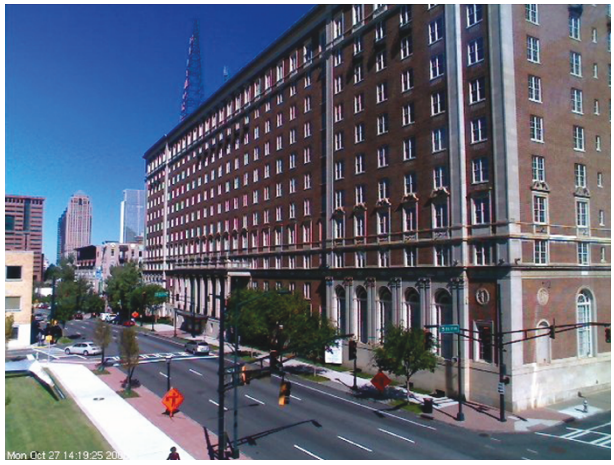
Table 1 summarizes the average BB stability statistics as observed in six video sequences. These averages are computed over 45 trials for each sequence, in which we test all combinations of the parameter choices for Cnum {5, 10, 15, 20, 25}, Rsize {8, 16, 32}, and Rth {10, 20, 30}. The precise effects of these parameters are described in the next section.

First examining the "NonBB % Match" and "BB % Match" columns, we see 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

(a)



(b)



(c)

(d)

(e)

(f)

FIGURE 3: Samples from the image sequences used in the BigBackground stability experiments: (a) shady, (b) city, (c) biltmore, (d) yard, (e) courtyard, and (f) sidewalk.

Table 1: 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	Non-BB % 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

pixels is identified as BB in each scene. In summary, Table 1 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.

3.2. Parameter Characterization of BigBackground. Next we discuss 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). There are two metrics we consider 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 4 shows the effect of different color palette sizes on BB coverage. Figure 5 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. We see from Figures 4 and 5 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

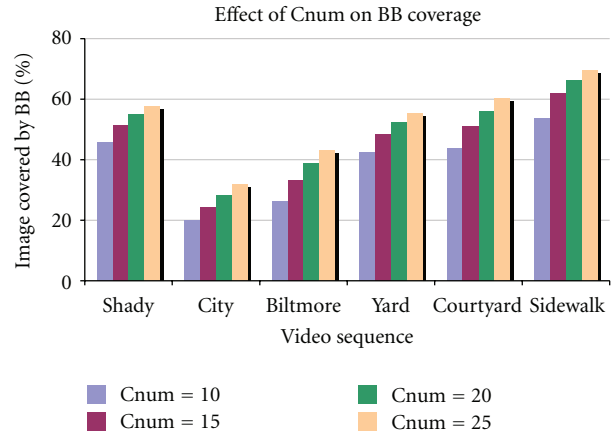


Figure 4: Increasing the size of the color palette (Cnum) increases the number of pixels belonging to BigBackground.

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 6. False colors are used to highlight the separation of BB regions. Black pixels do not map to a BB color.

4. BigBackground as an Illumination Anchor

The results in the preceding section show that BB can be relied upon as a relatively stable set of pixels. We propose 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

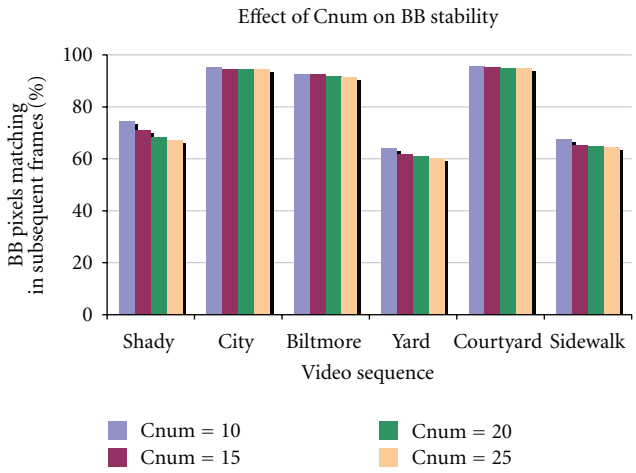


Figure 5: The additional pixels incorporated into the BigBackground model by increasing Cnum match slightly less often.

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 (9) 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

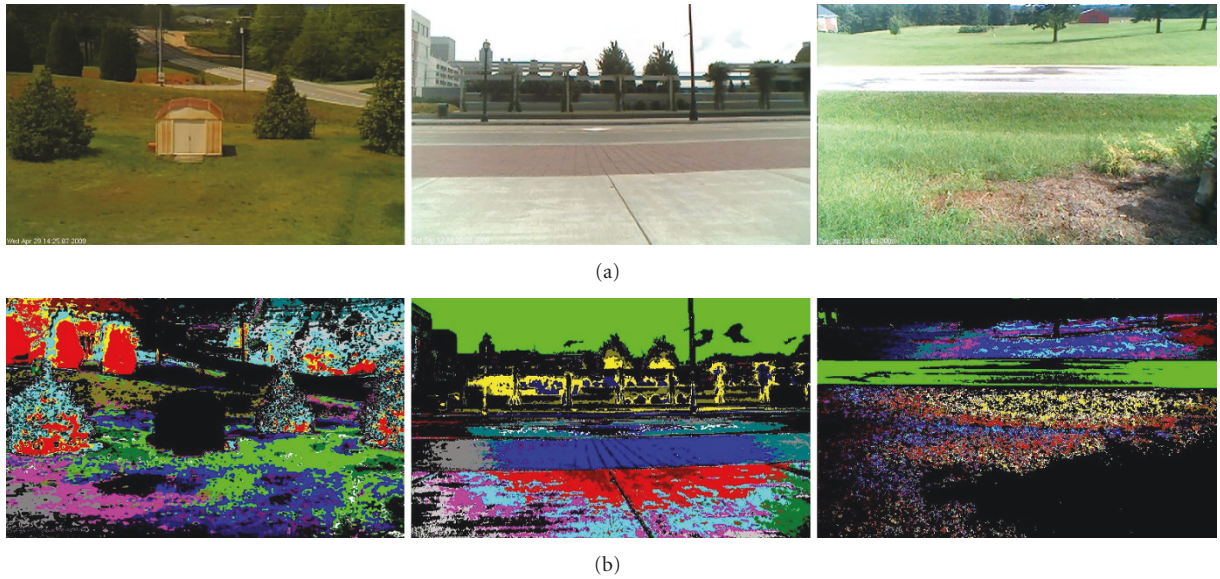
$$\begin{aligned} &S_1 > 12\% \text{ AND } S_2 > 12\% \text{ AND } |H_1 - H_2| < 8\%, \\ &S_1 > 12\% \text{ AND } S_2 > 12\% \text{ AND } |S_1 - S_2| < 8\%, \\ &S_1 < 12\% \text{ AND } S_2 < 12\%. \end{aligned} \tag{9}$$

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 (9) 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, we independently compute the parameters for a compensation model. We next present the sequences used for evaluation, address the issue of local lighting effects, and evaluate four choices for compensation models.

4.1. 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 × 480 pixel resolution. Table 2 describes the important features of the test sequences used. Samples of the video sequences are shown in Figures 9 and 10. 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.

4.2. 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 7, 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 8. 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, we have found that 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.



(a)

(b)

Figure 6: (a) Samples of multimodal mean predominant images from evaluation sequences. (b) False-color BigBackground maps. Each color represents a different BB region. For these examples, Cnum = 15, Rth = 20, and Rsize = 16.

Table 2: 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

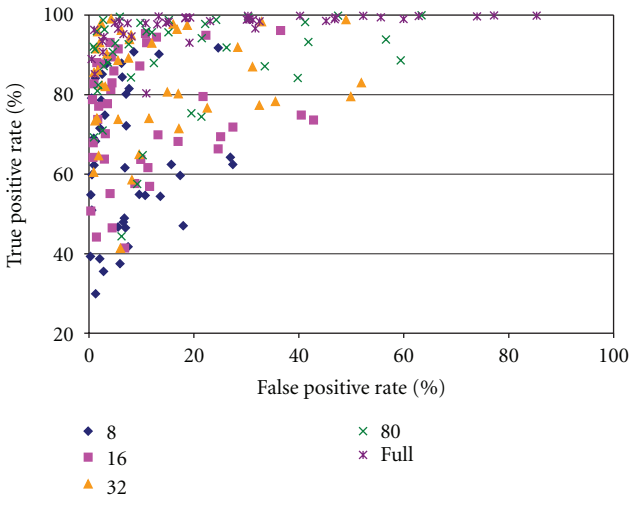


Figure 7: An ROC plot showing the effects of different tile sizes (8, 16, 32, and 80 pixels per side) on foreground/background classification accuracy. The data points shown were computed from ten video sequences using classification thresholds of 5, 7, 10, and 15.

4.3. Illumination Model Selection. As described in [4], several transformation models are available for dealing with illumination change. We determine experimentally which model is most effective for exploiting the BB model 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 [4, 30] 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 we examine 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

(a) (b)

Figure 8: Sample images from the Ford2 (a) and TechSquare1 (b) 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.

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 3, 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 4, 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 3: Computing illumination compensation models parameters.

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

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

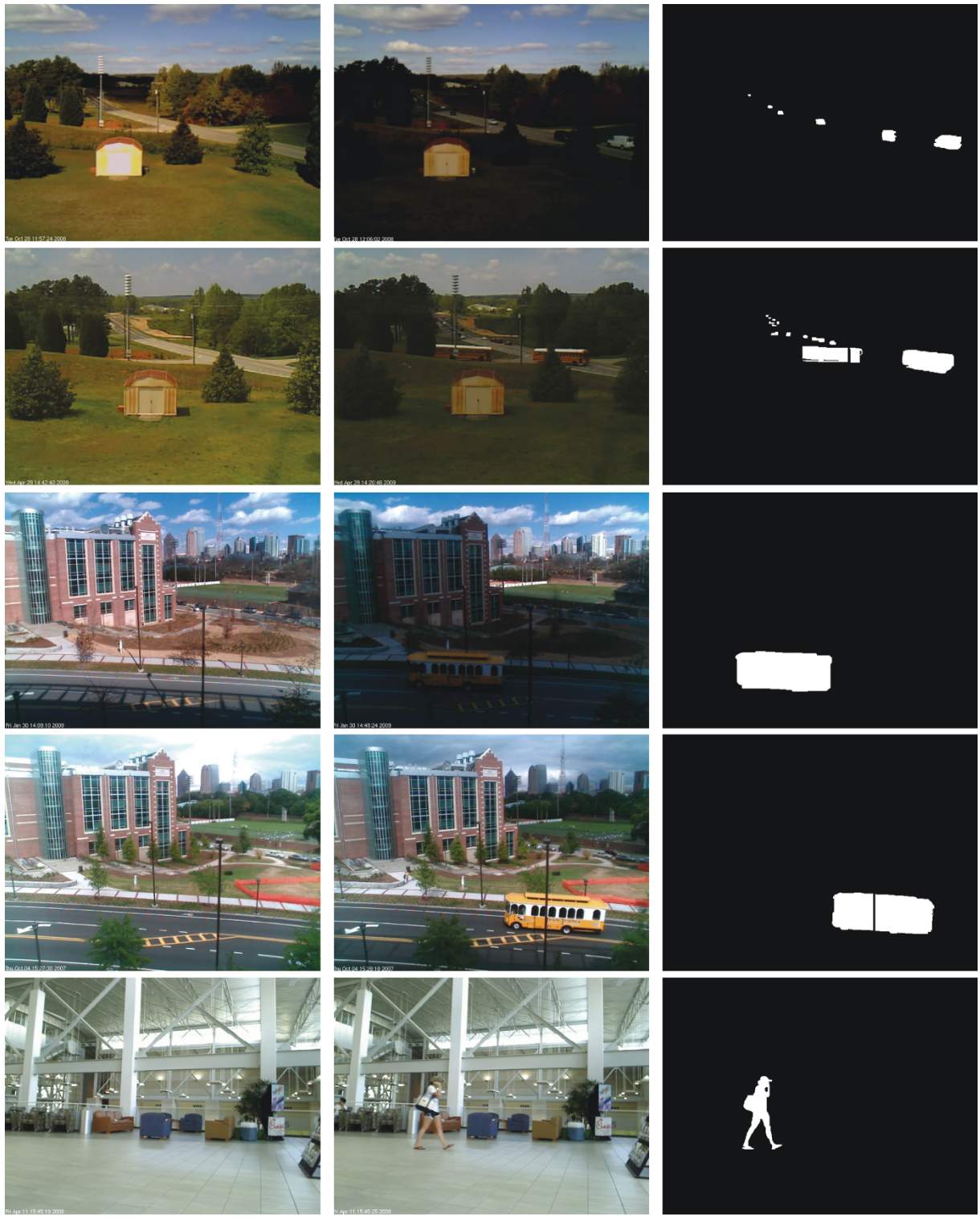


Figure 9: Left to right: Initial image, postlighting change image, and hand-marked ground-truth image from the sequences used to evaluate illumination compensation methods. Top to bottom: Backyard1, Backyard2, Ford1, Ford2, and RecCenter.

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

Figure 10: Left to right: Initial image, postlighting 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.

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 11 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

Table 4: Application of illumination compensation models to pixels belonging to BB.

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

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 12. False positive and false negative rates are shown in Figures 13 and 14, 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.

5. Chromatic Clustering in BigBackground

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 us to observe the effects of a lighting change 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. Thus we measure the separation of the colors in the BB palette and the separation between the illumination compensations of those colors and put the data in an XY scatter plot to observe any correlation (Figure 15).

It is apparent from Figure 15 that palette colors that are very close together have very similar compensation factors. A strong linear relationship can be seen between color

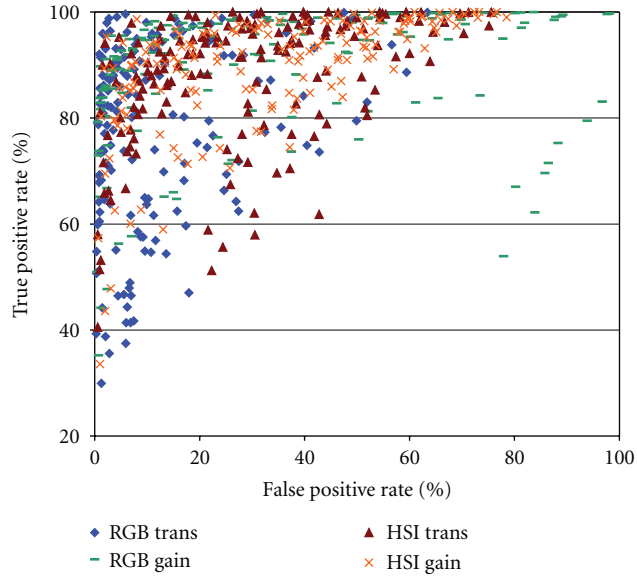


Figure 11: ROC plot for four mathematical models for illumination compensation. The data points shown were computed from ten test sequences, sweeping MCDth over {5, 7, 10, 15} and tile size over {8, 16, 32, 80}. The RGB Translation model stays more consistently concentrated in the low false positive range, which are the errors we are trying to minimize during illumination changes.

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, we examine all of the colors in the list and organize colors that match within a clustering maximum component difference (ClMCD) of each other 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.

We repeat the previous experiments to observe the effect of clustering on palette diversity and BB coverage. Figure 16 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.

We repeat the stability experiment to observe the effect of clustering on BB coverage and relative stability. We also test the effects of parameters Rsize and Rth on the algorithm with clustering. Table 5 shows that again, the pixels branded as BB match significantly more often than the remaining pixels. Compared with the values in Table 1, we see that the clustering process slightly decreases stability. However, the BB model covers significantly greater image area—generally

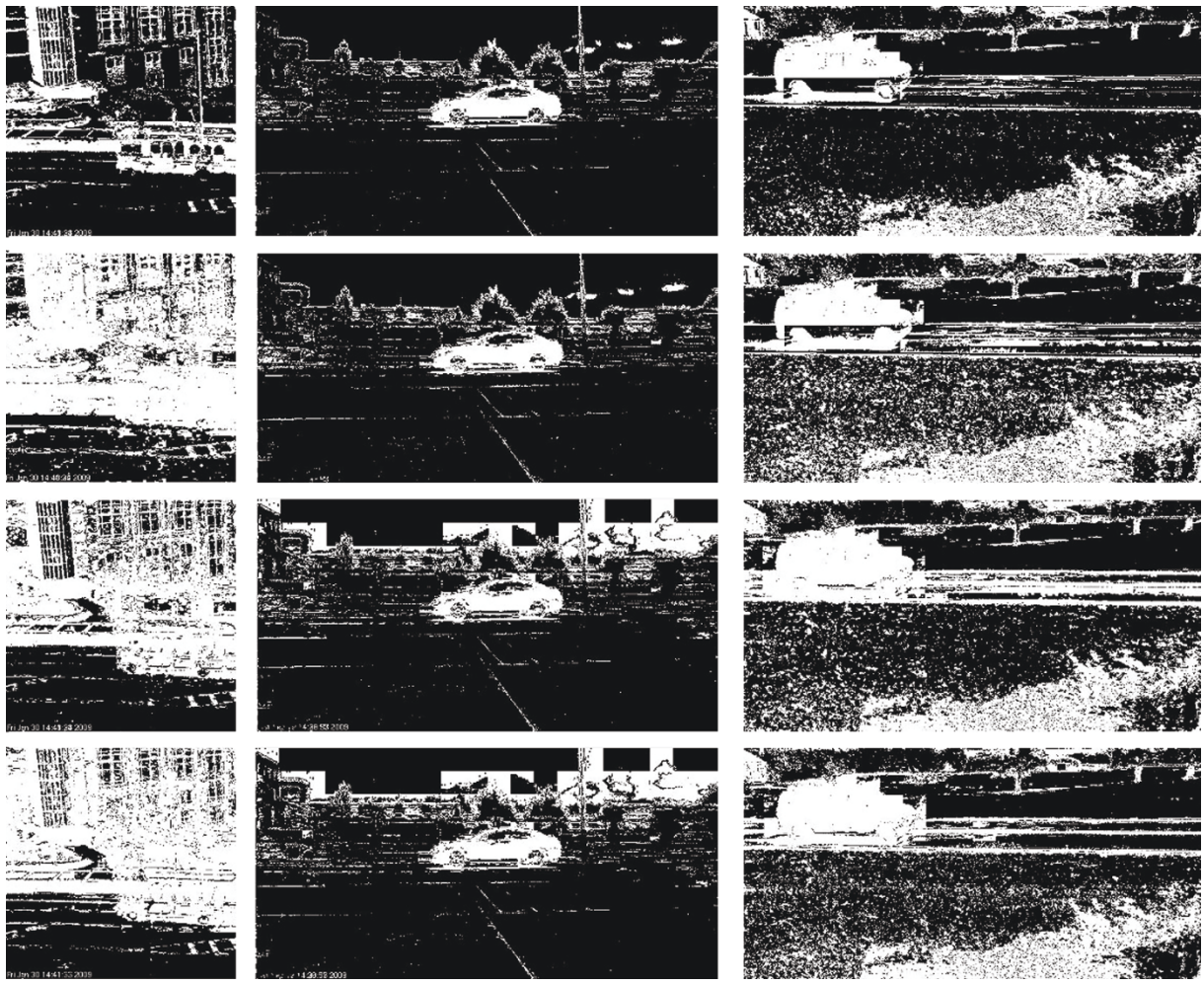


Figure 12: Sample images of foreground/background classification for four illumination compensation models. Left to right: Ford2, Techsquare2, and Roadside sequences. Top to bottom: RGB Translation, RGB Gain, HSI Translation, and HSI Gain. These samples were processed with Cnum = 15, Rsize = 16, Rth = 10, MCDth = 7.

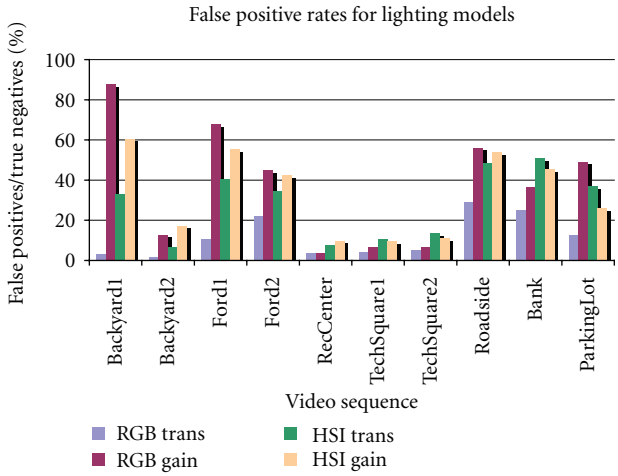


Figure 13: Comparison of four compensation models in terms of false positives/true negatives. The RGB translation technique consistently achieves the lowest false positive rate.

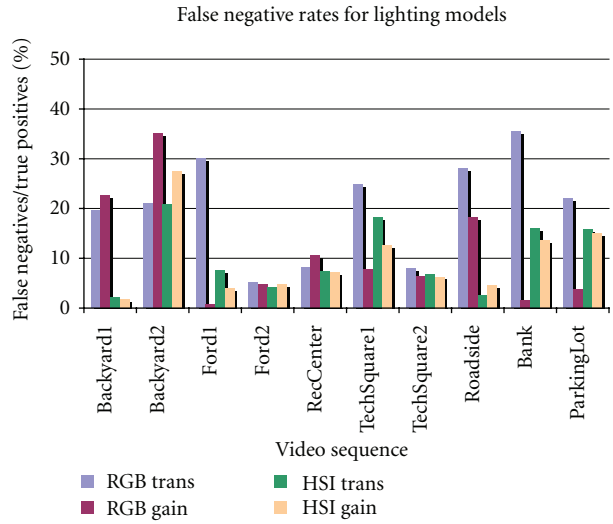


Figure 14: Comparison of four compensation models in terms of false negatives/true positives.

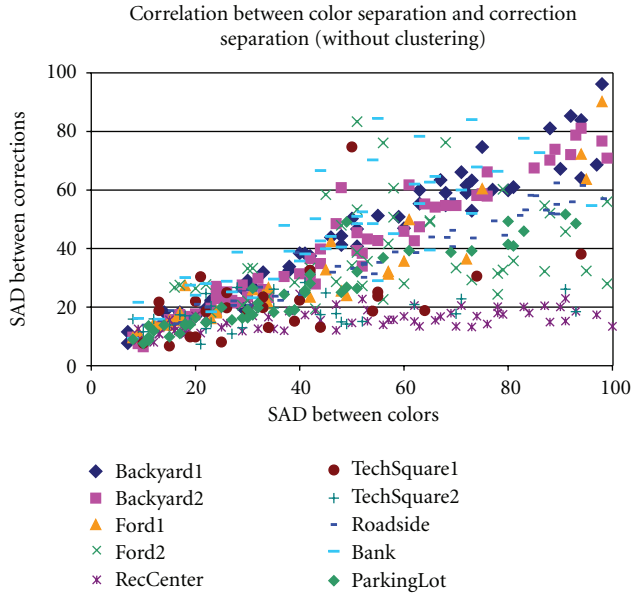


Figure 15: Correlation between color separation and correction separation before the clustering step is introduced. Pairs of colors that have a small sum of absolute difference tend to have a small SAD between their illumination corrections as well, indicating that the pair could possibly be treated as a single color for compensation purposes.

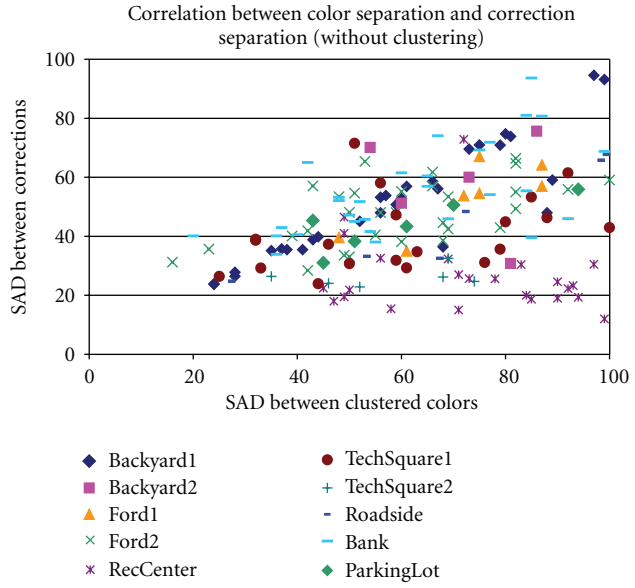


Figure 16: Correlation between color group separation and correction separation after the clustering step is introduced. Far fewer color pairs have SADs less than 100.

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 5: 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

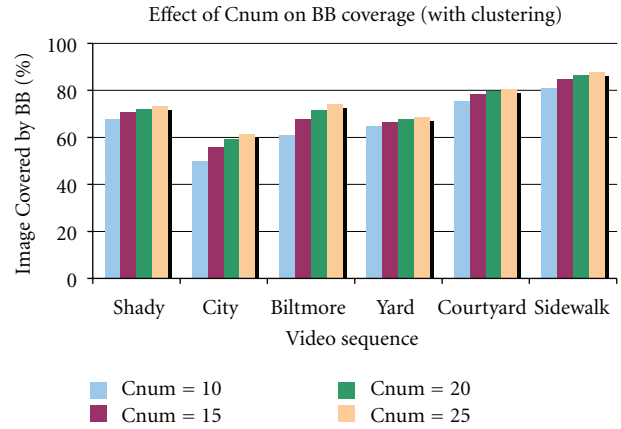


Figure 17: The effect of increasing color palette size on BB coverage after similar colors have been clustered into groups. The clustering step succeeds in increasing coverage between 20% and 40% over the nonclustering algorithm for the test sequences.

Next, we examine the effects of adjusting the BB parameters Cnum, Rsize, Rth, and the new clustering maximum component difference threshold (ClMCD), on BB coverage and stability. Figures 17 and 18 show the results for varying Cnum. The results shown are the averages for each sequence over all combinations of Rth, Rsize, and ClMCD. Again, we observe a law of diminishing returns in coverage, and 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.

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 19 and 20 show the performance of different Rth and Rsize combinations for different ClMCD values and a constant Cnum of 20 for the representative Biltmore video. 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.

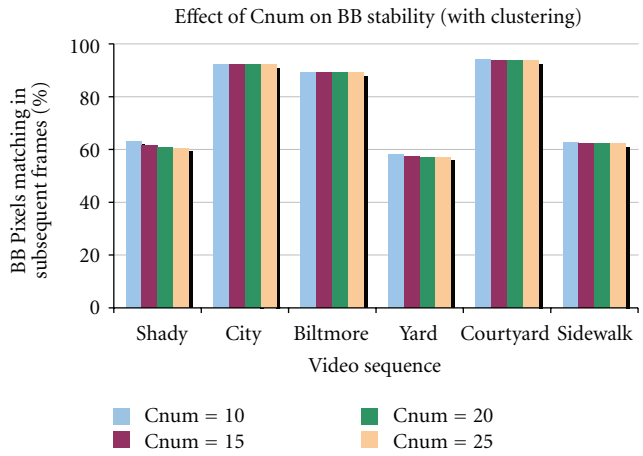


Figure 18: The effect of increasing the size of the color palette on BB pixel stability. A general decrease in stability is observed when compared to the preclustering stability measurements (in Figure 5), but the sharp increase in overall coverage more than compensates for the stability decline.

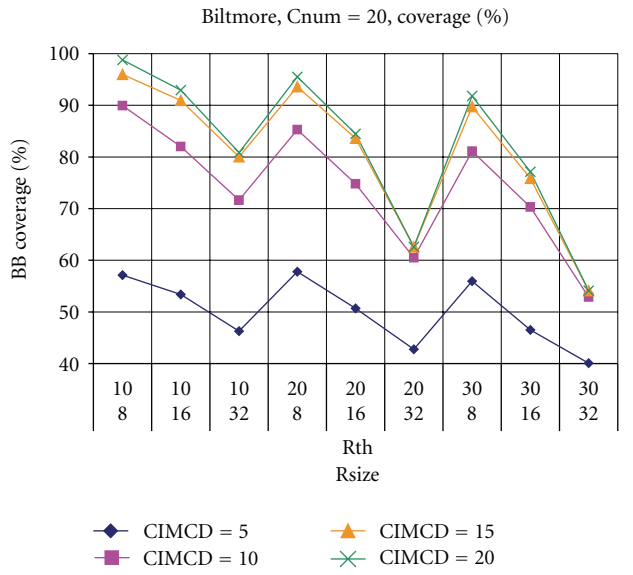


Figure 19: BigBackground coverage of the Biltmore sequence as a function of ClMCD, Rth, and Rsize.

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 are not 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 21. The sample images show the false-color BB maps from Figure 6 on top (in which no clustering

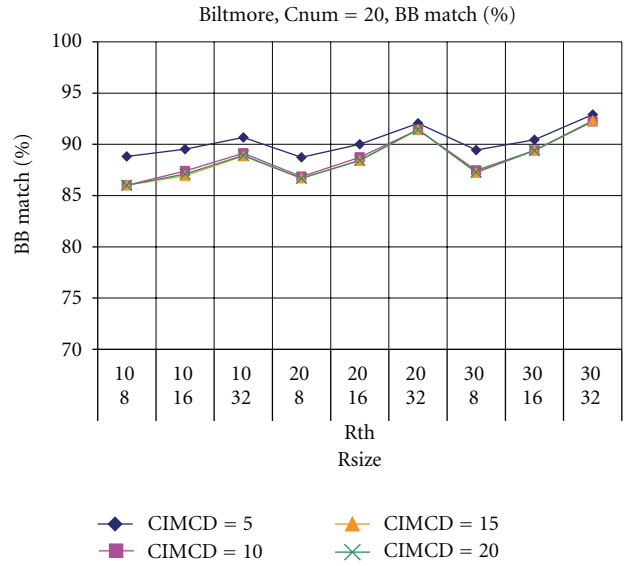


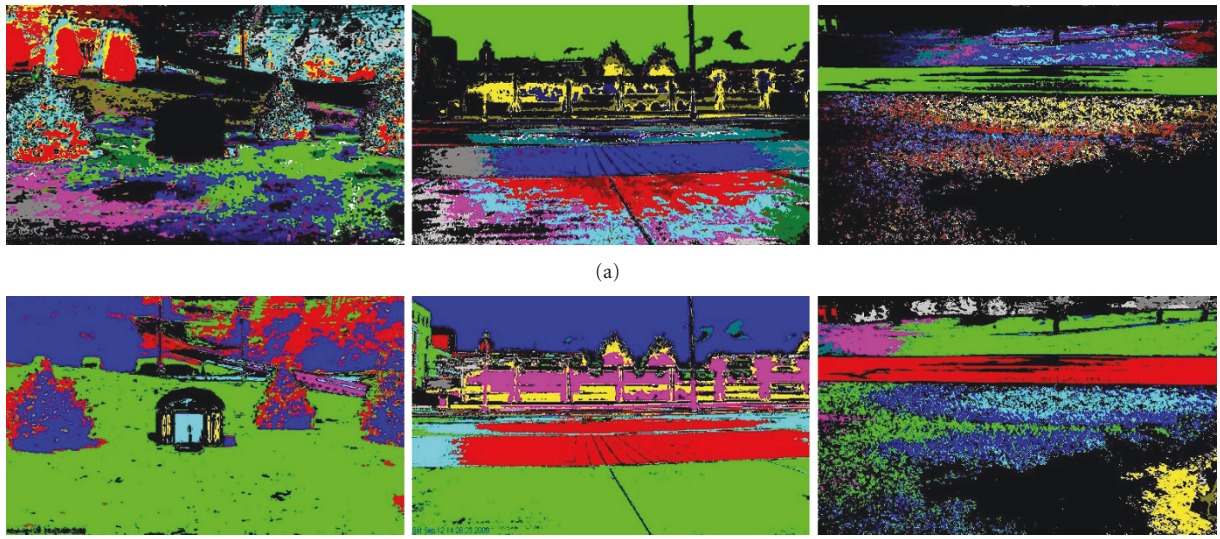
Figure 20: BigBackground stability in the Biltmore sequence as a function of ClMCD, Rth and Rsize.

was used) and new false-color maps produced by BB with clustering on bottom. 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).

6. Comparison to Other Methods

In this section, the proposed BB-based illumination compensation method is compared to the illumination compensation techniques described in [17–20] in terms of accuracy and execution time. These techniques are chosen for comparison because they are of similar structure and complexity to ours. 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.

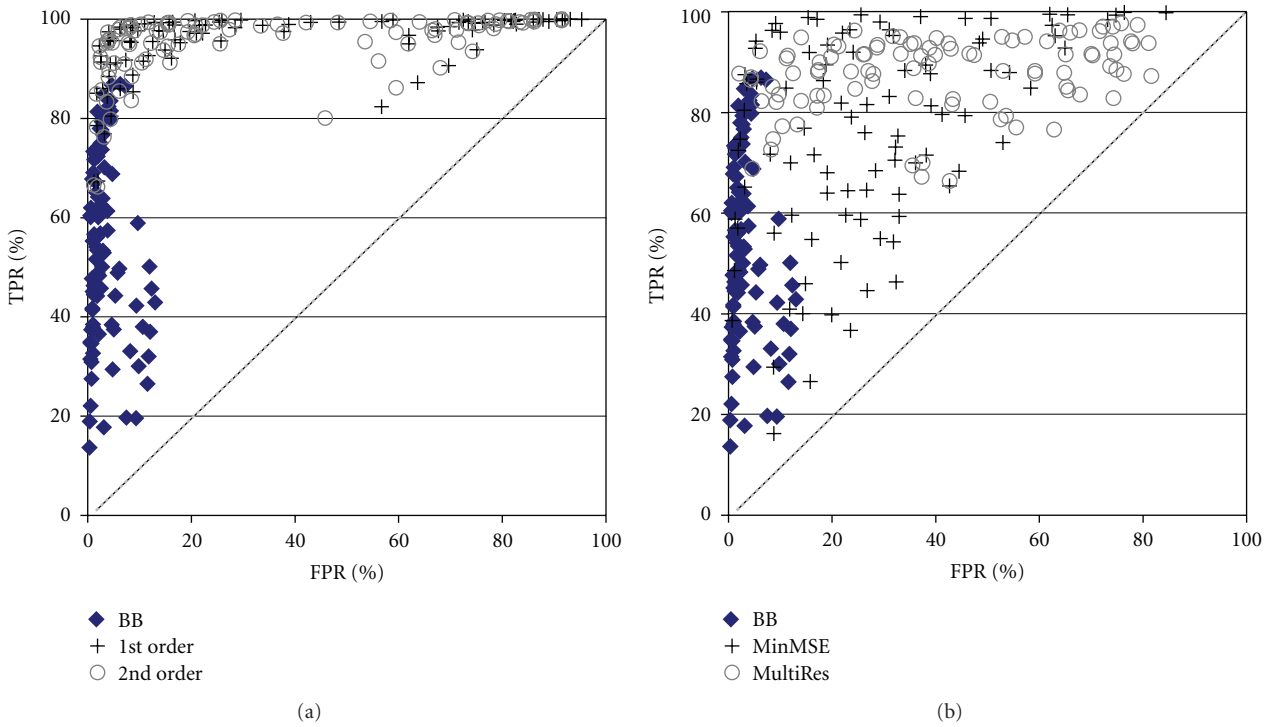
6.1. Accuracy Comparison. In this experiment, the settings shown in Table 6 are used for the BB process. To make results more comparable, the competing methods are coded to calculate correction statistics for each fixed tile, and to apply those corrections to all pixels in the tile rather than recalculating 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 22. 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). Figure 22(a) compares the proposed method with the first-order and second-order



(a)

(b)

Figure 21: (a) False color BigBackground maps without extra clustering step. (b) False-color BigBackground maps using clustering step to improve coverage. Each color represents a different BB region. For these samples, Cnum = 15, ClMCD = 15, Rth = 20, and Rsize = 16.



(a) (b)

Figure 22: 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}.

techniques. Figure 22(b) compares the proposed method with the MinMSE and multiresolution techniques.

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 [20] 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

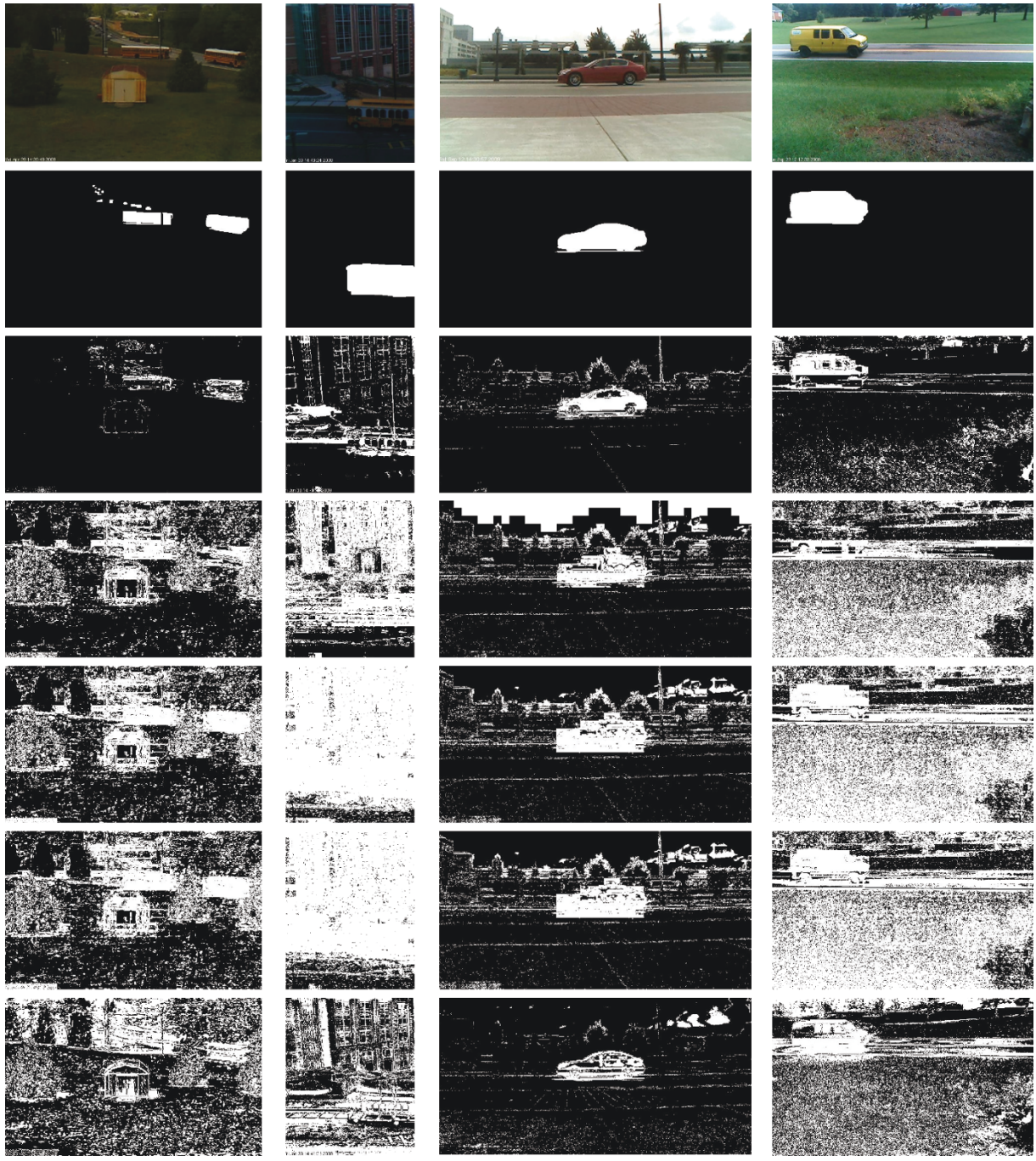


Figure 23: Segmentation results for illumination compensated scenes. Left to right: Backyard2, Ford1, TechSquare2, and Roadside sequences. Top to bottom: Original scene, Ground Truth, BB-based, MinMSE, First-order, Second-order, and Multiresolution compensation methods. For these examples, the parameters used were MCD = 7, illumination tile size = 32, Rth = 10, Rsize = 8.

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. We have found that by decreasing Rsize to 8 and Rth to 10 to improve BB coverage, our 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 23 shows examples of the foreground/background segmentation

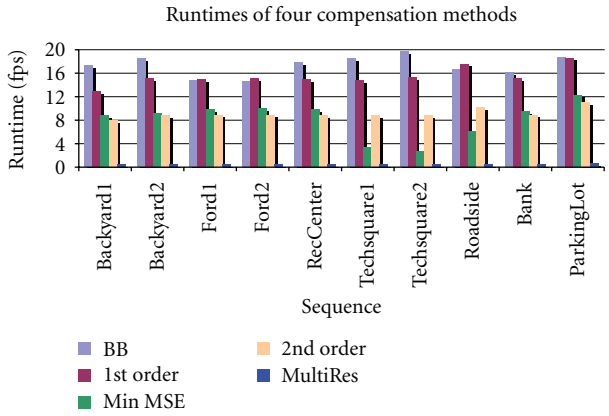


Figure 24: Runtime performance (in frames per second) of five illumination compensation techniques.

Table 6: Settings used when comparing BB compensation to other methods. These parameters do not affect the other compensation methods.

Cnum	15
ClMCD	15
Rth	20
Rsize	32

achieved using each compensation method. The most useful conclusions are drawn from the combination of Figures 22 and 23. While the local area statistic approaches presented in [17, 18] typically have high true positive rates, the unpredictable number of remaining false positives presents a challenge to subsequent processes. The techniques from [19, 20] 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 are 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.

6.2. 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 1 GB 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 24 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 proposed 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.

7. Conclusions

This paper has introduced BigBackground, a stable feature identifier based on chromatic similarity, and has presented an illumination compensation method based on the stable regions identified by BigBackground. 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. 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. Additional 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 segmentation 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 posses 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.

References

[1] M. Piccardi, "Background subtraction techniques: a review," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC '04)*, pp. 3099–3104, October 2004.

[2] 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.

[3] S.-C. S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," in *Visual Communications and Image Processing*, vol. 5308 of *Proceedings of SPIE*, pp. 881–892, San Jose, Calif, USA, January 2004.

[4] P. Gros, "Color illumination models for image matching and indexing," in *Proceedings of International Conference on Pattern Recognition (ICPR '00)*, vol. 3, pp. 576–579, 2000.

[5] M. R. Bales, D. Forsthoefel, D. S. Wills, and L. M. Wills, "Chromatic sensitivity of illumination change compensation techniques," in *Proceedings of the International Symposium on Visual Computing (ISVC '10)*, G. Bebis, R. D. Boyle, B. Parvin et al., Eds., vol. 6453 of *Lecture Notes in Computer Science*, pp. 211–220, Springer, 2010.

[6] M. J. Hossain, J. W. Lee, and O. S. Chae, "An adaptive video surveillance approach for dynamic environment," in *Proceedings of International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS '04)*, pp. 84–89, November 2004.

[7] Y. Wang, T. Tan, and K. F. Loe, "A probabilistic method for foreground and shadow segmentation," in *Proceedings of International Conference on Image Processing (ICIP '03)*, vol. 3, pp. 937–940, September 2003.

[8] M. S. Drew, J. Wei, and Z. N. Li, "Illumination-invariant color object recognition via compressed chromaticity histograms of color-channel-normalized images," in *Proceedings of the 6th IEEE International Conference on Computer Vision*, pp. 533–540, January 1998.

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

[10] M. Zhao, J. Bu, and C. Chen, "Robust background subtraction in HSV color space," in *Multimedia systems and Applications V*, vol. 4861 of *Proceedings of SPIE*, pp. 325–332, Boston, Mass, USA, July 2002.

[11] B. K. P. Horn, *Robot Vision*, McGraw-Hill, New York, NY, USA, 1986.

[12] G. D. Hager and P. N. Belhumeur, "Real-time tracking of image regions with changes in geometry and illumination," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '96)*, pp. 403–410, June 1996.

[13] Y. Matsushita, K. Nishino, K. Ikeuchi, and M. Sakauchi, "Shadow elimination for robust video surveillance," in *Proceedings of Workshop on Motion and Video Computing (WMVC '02)*, pp. 15–21, 2002.

[14] H. Wu, P. Jiang, and J. Zhu, "An illumination compensation method for images under variable lighting condition," in *Proceedings of the IEEE International Conference on Robotics, Automation and Mechatronics (CRAM '08)*, pp. 1022–1026, September 2008.

[15] E. G. Miller and K. Tieu, "Color eigenflows: statistical modeling of joint color changes," in *Proceedings of the 8th International Conference on Computer Vision (ICCV '01)*, pp. 607–614, July 2001.

[16] Y. Makihara, Y. Shirai, and N. Shimada, "Online learning of color transformation for interactive object recognition under various lighting conditions," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, vol. 3, pp. 161–164, August 2004.

[17] S. Young, M. Forshaw, and M. Hodgetts, "Image comparison methods for perimeter surveillance," in *Proceedings of the 7th International Conference on Image Processing and Its Applications (CIPA '99)*, vol. 2, pp. 799–802, July 1999.

[18] J. Lu, G. Lafruit, and F. Catthoor, "Streaming-mode MB-based integral image techniques for fast multi-view video illumination compensation," in *Proceedings of the 7th Pacific Rim Conference on Multimedia (PCM '06)*, vol. 4261 of *Lecture Notes in Computer Science*, pp. 414–423, Hangzhou, China, November 2006.

[19] 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 I*, vol. 86, no. 2, pp. 80–93, 2003.

[20] X. Suau, J. R. Casas, and J. Ruiz-Hidalgo, "Multi-resolution illumination compensation for foreground extraction," in *Proceedings of IEEE International Conference on Image Processing (ICIP '09)*, pp. 3225–3228, egy, November 2009.

[21] 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," in *Proceedings of the 6th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS '09)*, pp. 448–453, September 2009.

[22] X. Xie and K. M. Lam, "An efficient illumination compensation scheme for face recognition," in *Proceedings of the 8th International Conference on Control, Automation, Robotics and Vision (ICARCV '04)*, vol. 2, pp. 1240–1243, December 2004.

[23] K.-W. Wong, K.-M. Lam, and W.-C. Siu, "An efficient color compensation scheme for skin color segmentation," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '03)*, vol. 2, pp. 676–679, Bangkok, Thailand, May 2003.

[24] M. R. Yelal and S. Sasi, "Human tracking in real-time video for varying illumination," in *Proceedings of IEEE International Workshop on Intelligent Signal Processing (IWISP '05)*, pp. 364–369, September 2005.

[25] J. Cai, M. Shehata, W. Badawy, and M. Pervez, "An algorithm to compensate for road illumination changes for AID systems," in *Proceedings of the 10th International IEEE Conference on Intelligent Transportation Systems (ITSC '07)*, pp. 980–985, October 2007.

[26] J. Li, S. Z. Li, Q. Pan, and T. Yang, "Illumination and motion-based video enhancement for night surveillance," in *Proceedings of the 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (IWVSPETS '05)*, pp. 169–175, October 2005.

[27] Z. Tauber, Z.-N. Li, and M. S. Drew, "Locale-based visual object retrieval under illumination change," in *Proceedings of International Conference on Pattern Recognition (ICPR '00)*, vol. 4, pp. 43–46, 2000.

[28] N. Jojic, A. Perina, M. Cristani, V. Murino, and B. Frey, "Stel component analysis: modeling spatial correlations in image class structure," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 2044–2051, June 2009.

[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," in *Embedded Computer Vision*, B. Kisacanin, S. Bhattacharyya, and S. Chai, Eds., pp. 163–175, Springer, New York, NY, USA, 2009.

[30] 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," in *Proceedings of International Conference on Pattern Recognition (ICPR '02)*, vol. 1, pp. 620–623, 2002.