

Physical Models for Moving Shadow and Object Detection in Video

Sohail Nadimi, *Member, IEEE*, and
Bir Bhanu, *Fellow, IEEE*

Abstract—Current moving object detection systems typically detect shadows cast by the moving object as part of the moving object. In this paper, the problem of separating moving cast shadows from the moving objects in an outdoor environment is addressed. Unlike previous work, we present an approach that does not rely on any geometrical assumptions such as camera location and ground surface/object geometry. The approach is based on a new spatio-temporal albedo test and dichromatic reflection model and accounts for both the sun and the sky illuminations. Results are presented for several video sequences representing a variety of ground materials when the shadows are cast on different surface types. These results show that our approach is robust to widely different background and foreground materials, and illuminations.

Index Terms—Detecting moving objects, dichromatic reflection model, physics-based segmentation, shadows in video, spatio-temporal albedo ratio.

1 INTRODUCTION

OVER the past several decades, many approaches have been developed for detecting moving objects in indoor and outdoor scenes. Recently, statistical-based approaches using a mixture model have shown that many problematic phenomena such as repetitive motion of the background (e.g., swaying trees), sudden illumination changes (e.g., cloud cover), and sensor noise can be modeled [4], [8], [16]. Despite this success, a statistical-based approach fails to distinguish between the actual moving object and its shadow silhouette. This is due to the fact that intensity changes in the scene due to moving shadows are as high as those of the new objects appearing in a scene.

Shadows are generally divided into static and dynamic shadows. Static shadows are shadows due to static objects such as buildings, parked cars, trees, etc. Statistical-based methods for moving object detection in video do not suffer from static shadows since static shadows are modeled as part of the background and can be tracked by an adaptive procedure. Dynamic shadows in video, the subject of interest in this paper, are due to moving objects such as moving vehicles (cars, trucks), pedestrians, etc. The shadows can take on any size and shape, and can be *Umbra* (dark shadow), *Penumbra* (soft shadow), or both. Our research focuses on outdoor scenes where we have a far away point source (sun) and a diffuse source (sky) contributing to the illumination in the scene. Since the distance between the objects and the background is negligible compared to the distance of illumination sources to the objects, most or all of the shadows are umbra or strong shadow that we deal with in this paper. Note that penumbra or weak shadow exist only when the sky is very cloudy and/or illumination is highly diffuse.

2 RELATED APPROACHES AND OUR CONTRIBUTIONS

Table 1 provides a summary of approaches for shadow detection. A comparative study of selected works [1], [5], [7], [15] can be found in [11].

- The authors are with the Center for Research in Intelligent Systems, University of California, Riverside, CA 92521.
E-mail: {sohail, bhanu}@cris.ucr.edu.

Manuscript received 15 Oct. 2002; revised 10 July 2003; accepted 8 Mar. 2004.

Recommended for acceptance by R. Kumar.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 117583.

In comparison with state-of-the-art, the contributions of this paper are:

1. *Integration of different physical models*—Sound physical models (dichromatic reflection model, and reflectance-based analysis model) are used in an integrated and principled manner. Our approach is different from all the previous work (see Table 1) in that we make no assumption about surface geometries (e.g., planar versus curved, horizontal, versus vertical), surface texture (e.g., grass, brick, tiles, road, etc.), or types and shapes of shadows, objects, and background. We solely rely on models, which can represent wide classes of surface materials. In addition, unlike all previous approaches that make the assumption of one illumination source (white light), our model incorporates multiple sources (sun and sky) with different spectral power distribution (SPD). The integrated “realistic” physical models are used for shadow and object detection.
2. *Temporal improvement*—Spatial albedo ratio test [9], which is a measure of constant reflectivity of a surface, is extended temporally. The new test is called spatio-temporal albedo ratio test, and it is utilized for surface segmentation in video.
3. *Experiments and performance evaluation*—The algorithm is tested on a wide variety of video data consisting of both vehicles and people. We have provided *physically important* independent variables such as surface types and materials, surface orientation, and time of day, which are *physically linked* to the scene and are used as benchmarks [11] in our experiments.

3 TECHNICAL APPROACH

3.1 Shadow Modeling

Consider a single visible light point source (i.e., sun), a diffuse extended light source (i.e., sky), and a Lambertian surface with constant reflectance. Then, the observed surface intensity value, I_V , on a given surface can be calculated by integrating the reflectance function over the entire visible spectrum for each light source and then added.

$$I_V = \int_{\Delta\lambda_1} K_1 L_1(\lambda) f(l, e, s) d\lambda + \int_{\Delta\lambda_2} K_2 L_2(\lambda) d\lambda, \quad (1)$$

where K_1, K_2 are the coefficient of reflectances due to sun and sky; L_1, L_2 are intensity of the illumination sources sun and sky, respectively, $f(\cdot)$ is a geometric term, and the angles are l (incident angle of illumination), e (angle for viewing direction), and s (angle for specular reflection). The first term of the model in (1) is due to the point source and includes both specular and diffuse components. If the interreflections are negligible, the color of a pixel is only influenced by the light reaching it and reflected properties of the surface.

The first term in (1) can be approximated by several reflectance models. In the order of increasing sophistication, surface reflectance is modeled by: 1) Lambertian model, 2) Phong model, and 3) Dichromatic model [13]. Considering the dichromatic model for the first term in (1),

$$I = L(\lambda, l, e, s) = L_i(\lambda, l, e, s) + L_b(\lambda, l, e, s) = m_i(l, e, s) C_i(\lambda) + m_b(l, e, s) C_b(\lambda), \quad (2)$$

where L is the total radiance of the reflected light, L_i is radiance of the light reflected at the surface, and L_b is radiance of the light reflected from the body. Geometric terms for the surface and body are m_i and m_b , respectively, and the angles are described as before.

TABLE 1
Summary of Current Approaches to Shadow Detection

Approach and Principle	Advantages and Disadvantages (in comparison with this paper)
<i>Image Projection</i> – Uses vertical and horizontal image projection of binary sillouttes, finds the Loci points where feet and shadow intersect [6, 14].	Adv: Simple algorithm; can calculate the direction and length of shadows. Disadv: Need knowledge about the light source, works only for erect human objects, requires that shadows be on the ground.
<i>Geometric projection</i> , - Uses binocular vision, projects the image in one camera C1, to flat road and backprojects it to the second camera C2. Shadows in C1 and C2 occupy the same location in both images [10].	Adv: Independent of the object type, independent of road color, utilizes binocular vision. Disadv: Requires shadows to be on the flat road plane, objects and shadows must be visible to both cameras, requires manual registration, and object height.
<i>Local Intensity-based</i> – Divides the image into blocks and calculates their mean. The median of the block means is used as the threshold to separate shadows [12].	Adv: Simple algorithm; geometry and object type independent. Disadv: Image must be background dominant, background must be uniform.
<i>Knowledge-based</i> – Utilizes proximity, collinearity and convexity constraint of structured objects such as planes to separate shadow sillouttes from the actual object [2].	Adv: Utilizes geometric knowledge of objects, the knowledge provides proximity, collinearity and convexity constraints. Disadv: Requires highly structured objects (airplanes), assumes that any shadow can be represented by lines, requires camera and sun orientation and position.
<i>Hueristic-based</i> – Develops an edge detector. Discontinuities due to shadows are distinguished from the background discontinuities through heuristic rules [15].	Adv: Detects both penumbra and umbra. Disadv: Requires the background to be uniform, uses various heuristics rules in indoor scenes.
<i>Biological-based</i> – Develops a model based on human vision which is augmented with simple illumination models, assuming all light sources of the same type, for surface segmentation due to shadow and object [3].	Adv: Detects and classifies material or shadow boundaries, objects can be articulated. Disadv: Works on limited types of backgrounds, requires all illumination sources to be white.
<i>Histogram-based</i> – Statistical properties of pixels (e.g., mean and variance) are used by a maximum a porteriori classification to distinguish shadows [7].	Adv: Independent of the number of objects, outdoor scenes. Disadv: Assumes background is flat road surface, requires position of the sun with respect to camera.
<i>Color-Constancy</i> – Models the pixel luminance and chromaticity and their distortion in HSV or RGB. Several thresholds are defined to separate foreground, background, highlights and shadows [1, 5].	Adv: Works for diffuse reflections and weak, penumbra shadows on flat surfaces. Disadv: Requires all illumination sources to be white, assumes shadow and non-shadow have similar chrominance, fails on dark shadows.
<i>Physics-based, Nadimi and Bhanu (this paper)</i> – Develops an integrated approach based on sound physical models of illumination and reflectance. Develops spatio-temporal albedo segmentation used to segment detected frame into sub-regions that are verified as shadow or object based on the reflection model.	Adv: Based on sound physical principles, independent of the number and type of objects, background types/colors, and scene geometry; detects umbra in outdoor scenes, employs physically-based representation and evaluation. Disadv: Requires spectral power distribution of each illuminating source to be constant, needs to be extended to handle highlights and highly specular surfaces (e.g., puddles).

C_i and C_b are relative SPDs due to surface and body reflections due to point source (e.g., sun).

The second term of the model in (1), due to extended-diffuse illumination, is not influenced by the surface, illumination, and viewing geometries, hence, no geometrical function is required like the one in the first term of (1). Therefore, it can be replaced by an additive term C_a representing relative SPD due to ambient illumination (e.g., extended light source, sky). Using the term C_a and (2), the model in (1) can be approximated by the extended dichromatic model:

$$I_v = m_i(l, e, s)C_i(\lambda) + m_b(l, e, s)C_b(\lambda) + C_a. \quad (3)$$

The approximation model in (3) has five unknown terms, m_i , C_i , m_b , C_b , and C_a . The first two terms (m_i and C_i) are due to specular reflection and exist for surfaces with high reflectivity such as polished or smooth surfaces and conductors. Assuming surfaces have weak or no specularly and correcting for reflectance due to ambient illumination, C_a , (e.g., subtracting C_a from (3)), we are left primarily with only the reflectance due to body color C_b , which is independent of geometrical influence due to angle of illumination or viewing direction [13]. The body color C_b , an inherent property of the subsurface representing the true color of the material under white light, provides an invariant that is used in our approach to distinguish background surfaces from the foreground surfaces for shadow and object detection in video. In the existing literature,

illumination from the sky is not modeled (see Table 1). As a result, the estimation of surface color is inaccurate. In real-world outdoor scenes, the sources of illumination are both the sun and the sky and the shadows are formed when the sunlight is blocked. The SPD of the sky is more dominant (saturated) in the blue region of the spectrum. Our approach works in RGB and C_i - C_b (dichromatic space) not in HSV and the model in (2) is consistent with the RGB space. It provides a more meaningful and physically linked representation than HSV [13].

3.2 Physics-Based Shadow Detection

Our approach, depicted in Fig. 1, is a multistage approach where each stage of the algorithm (marked in numerical order) removes moving object pixels, which cannot be shadow pixels. Input video frame is passed through the system and M_i is the binary mask of potential shadow pixels updated after each step. At the end of the last stage, we obtain the moving shadow pixels as well as moving object pixels.

Step 1—Moving Object Detection: This step detects moving pixels (object, shadow, and some erroneous pixels). Sensor fusion techniques [8] are applied within a mixture model-based framework [16] in RGB space for the initial object detection. In this algorithm, each pixel is viewed as an independent process. To model the background, recent history of each pixel x , is modeled by a mixture of g Gaussians: $P(x) = \sum_{i=1}^g w_i \times \eta(x, \mu_i, \Sigma_i)$, where,

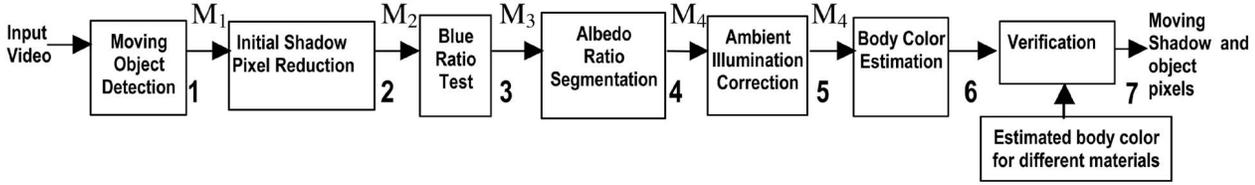


Fig. 1. Different steps of physics-based shadow detection algorithm.

for each R, G, B channel, $P(x)$ is the probability of observing pixel value x , η is the Gaussian function whose i th mixture component is characterized by the mean μ_i , covariance Σ_i , and weight w_i . The moving object detection algorithm is initialized by first collecting t initial frames, and then estimating the parameters of the mixture for each pixel by K-means clustering technique. We use the *AND* strategy [8] which specifies that an incoming pixel value must be within three standard deviations of any of its g models in all three (independent R, G, B) channels to be considered a background pixel; otherwise, it is classified as a moving pixel. This strategy provides the highest detection rate in comparison to other fusion strategies [8]. After the initial detection, the binary mask (M_1 in Fig. 1) contains the moving object, its shadow and noisy isolated pixels. Our segmentation algorithm (Step 4) needs a background image, which is simply tracked in memory.

Step 2—Initial Shadow Pixel Reduction: The test at this step states that pixels on a detected surface cannot be shadow if they have higher intensity than the actual background. Once we have the current “segmented” image and the background from Step 1, the following intensity test is applied to moving objects and shadow pixels to further reduce their number. Let $p_{\{x\}}$ be a pixel where $x \in (R, G, B)$ is background pixel and $x \in (r, g, b)$ is foreground moving object, its shadow, and some erroneous background pixels. Let $M_2 = \phi$ (initially an empty mask), then

$$\forall p \in M_1 \text{ if } (p_r < p_{Rr}) \& (p_g < p_{Gg}) \& (p_b < p_{Bb}) \Rightarrow M_2 = M_2 \cup p. \quad (4)$$

This test does not reduce the shadow areas, but may successfully reduce object areas, thus reducing the binary mask and computation at later steps. In addition, this step also eliminates pixels due to specular reflection which helps in our diffuse component estimation (Step 6).

Step 3—Blue Ratio Test: This step exploits the observation that shadow pixels falling on neutral surfaces, such as asphalt roads, tend to be more blueish. This is also true for many gray structures such as concrete buildings, walkways, etc. Shadow regions are illuminated by the sky and sky is assumed to be blue and the only source of illumination on shadowed regions. Although all RGB values are lower in the shadow region, we have observed that the amount by which this reduction occurs is not proportional. This is used to further refine the shadow segmentation. Let $p_{\{x\}}$ be defined as in Step 2, then the ratio (p_b/p_B) tends to be larger than (p_r/p_R) and (p_g/p_G) in shadow regions. We hypothesize a pixel in the image under the mask M_2 to be a shadow pixel and generate a mask M_3 (initially $M_3 = \phi$) of hypothesized shadow pixels as:

$$\forall p \in M_2 \text{ if } [(p_b/p_B) > (p_r/p_R)] \& [(p_b/p_B) > (p_g/p_G)] \Rightarrow M_3 = M_3 \cup p. \quad (5)$$

Shadow pixels tend to have lower intensity. They are more saturated toward blue, and their vectors in RGB space make smaller angle with the blue axis. Equation (5) is not applied to all the pixels in mask M_2 , but to neutral or gray surfaces that have low saturation (< 0.3). Highly saturated surfaces do not exhibit this phenomenon due to high selectivity of the reflected color. The

output of this step (M_3) provides further reduction in the number of pixels from Step 2.

Step 4—Albedo Ratio Segmentation: This step performs surface segmentation based on a new spatio-temporal albedo ratio test. Our surface color estimation algorithm (see Step 6 and Fig. 3) relies on the fact that the image has been segmented into uniform regions; where each region is potentially a shadow. Segmentation algorithm must define a uniformity test that provides the criterion for segmentation. In [9], such a criterion is introduced based on spatial albedo ratio of neighboring pixels and is given as: $(I_1/I_2 = \rho_1/\rho_2)$ where I_1, I_2 are intensity and ρ_1, ρ_2 are albedo (reflectance) of neighboring pixels. It has been shown [9] that this ratio is independent of reflectance function, illumination direction and intensity, and surface geometry (such as flat versus curved surfaces). This ratio is extended to the temporal domain as described below. First, we define the connectivity C of two neighboring pixels p_1 and p_2 (see Fig. 2) with intensities u and v as follows:

$$C(p_1, p_2) = \begin{cases} 1 & \text{if } |H(u, v)| < T \\ 0 & \text{Otherwise,} \end{cases} \quad (6)$$

where, unlike the ratio of reflectance mentioned in the previous work [9], [15], we use the following spatio-temporal relation and define the following ratios:

$$R_1 = \frac{u_{t+1} - u_t}{u_{t+1} + u_t}, R_2 = \frac{v_{t+1} - v_t}{v_{t+1} + v_t}, H(u, v) = \frac{R_1 - R_2}{R_1 + R_2}. \quad (7)$$

The first two ratios R_1 and R_2 are temporal ratios; therefore, $H(u, v)$ is the spatio-temporal albedo ratio. If two neighboring pixels belong to the same surface they will have temporal ratios that are close together; hence, the spatio-temporal relation will be close to zero. A small value for T in (6) is chosen to account for noise and other artifacts. Note that, in the spatio-temporal albedo test, the spatial constancy of the normal to the object surface is assumed. This assumption is propagated to the temporal domain where surface normals of neighboring pixels are the same. This assumption is valid for both the spatial and temporal domains for rigid bodies. It fails for deformable bodies where the surface normal can change over time.

In order to segment the image into regions of uniform reflectance (Step 4A), we spatially segment the image using the connectivity criterion (6) in a segmentation algorithm that first performs a sequential labeling [9]. Since border pixels are not reliable, the algorithm shrinks the binary mask (of the object and shadow) from the previous step by 1 pixel before applying the criterion (6). Furthermore, a size filter (S) removes spurious segments (Step 4B) and a new mask (M_4) is generated. The choices for T and the size filter (S) are discussed in Section 4.3. If the reflected light reaching

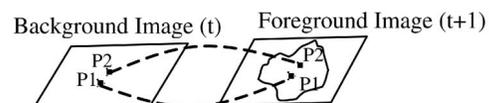


Fig. 2. Two neighboring pixels in the foreground ($t + 1$) and background (t) image. Background image contains no moving object/shadow (just the surface background).

```

Given  $m$  pixels ( $m \gg 4$ ) in RGB space
Arrange the pixels in an  $m \times 3$  matrix  $M$ 
 $(S, V) = \text{svd}(M)$ ; Apply singular value decomposition on  $M$ ,
Let  $S_1, S_2, S_3 \in \{S\}$  be singular values in descending order;
Let  $V_1, V_2, V_3 \in \{V\}$  be singular vectors corresponding to the singular values above;
If  $S_1 \gg S_2$  Then  $C_b = V_1$ ; error_term =  $V_3$ ; (Note: we use 1 order of magnitude difference between  $S_1$  &  $S_2$ )
Else if ( $S_1 \leq \delta \times S_2$ ) where  $\delta = 2$  is a constant, Then STOP. (Specular surface)

```

Fig. 3. Diffuse color estimation algorithm.

the sensor has both the specular and diffuse components, the ratio in (6) may not produce a meaningful result; therefore, to make this ratio useful, we must assume that the surfaces have little or no specular component. This assumes that there are few highlights in the image. Most matte, dielectrics, and natural surfaces have mostly diffuse reflections. This is usually not a problem in shadow regions where specular reflections do not exist.

Step 5—Ambient Illumination Correction: This step removes the effect of sky illumination. At this step, the input image under the mask has been segmented into uniform regions. Some of the segments belong to actual objects and some to the background due to shadows, but all of them have some uniform reflectance properties. Since we do not know which segment belongs to an actual object and which to a background surface, we assume that all are background surfaces.

According to (3), the shadow model developed in Section 3.1, the reflection due to sky illumination (called ambient reflection) is considered as an additive component; therefore, we subtract the foreground pixel values from the background over the masked area (M_4). The result now contains the values where the reflectance due to the sky illumination has been eliminated, so only the contribution from the sun is left. For those regions that belong to an actual object, this subtraction causes the background surface to have a very different color vector than the color vector expected for the shadow region. The hypothesis here is that all segmented surfaces are due to shadow. If our hypothesis is correct, then we should be able to correctly find the diffuse color vector due to sunlight only. On the other hand, if our hypothesis is wrong, in the case of object surfaces with very different color than the background, then we do not expect to get the correct results; hence, in our verification stage, these vectors will not be matched.

Step 6—Body Color Estimation: This step performs body color estimation (C_b) using a diffuse color estimation algorithm. At this step, the input image under the mask has been segmented into several regions and ambient reflection due to sky has been subtracted from each region. Assuming each segment is a shadow, we obtain the diffuse component (C_b) of the model (2) and then compare this to our original estimation called, the initial estimate. Two questions now remain: 1) How do we get the C_b from our current information? 2) How do we measure it in the first place so we can compare it with our new measurement? We first answer question 1), and postpone the answer of 2) to Section 4.2.

The model in (2) is represented by four unknowns, m_i , m_b , C_i , and C_b , for each pixel. This requires at least four values for a pixel.

These values can be obtained by three different methods: 1) moving the illumination source, 2) moving the camera, or 3) sampling the pixel values on larger but similar areas. Since we have no control over the illumination source and assume static camera, the first two methods do not apply. The third method requires a uniform surface that is obtained at Step 4. The algorithm for estimating the color of a surface is given in Fig. 3.

The body color (C_b) is in the form of a 3-dimensional unit vector in unit RGB space. The vector corresponding to the largest eigenvalue is estimated as the body color. Depending on the surface type (planar or curved) that matrix M represents, and the viewing geometry, the pixels in matrix M may form point-like, linear, or planar clusters. The eigenvalues and eigenvectors represent the extent and orientation of ellipsoids that fit the data.

Step 7—Verification: This step performs matching of body color of various surfaces with the stored body color of materials that we expect to see in the scene. The algorithm has training and testing phases. In the training phase, we calculate the body color of surfaces that come under shadow in the scene. This is accomplished by precomputing the body vector C_b for various surface materials such as concrete, grass, red-brick, asphalt, etc. During the testing phase, for each frame after Step 7, for each segment, we compare the body color vector of the segment with any of the body color vectors in our database. Assume that the body color of a surface patch is V_p and the true body color of the surface is V_t , then the angle θ between the two vectors is defined as: $\theta = \cos^{-1}[(V_p \cdot V_t) / (||V_p|| ||V_t||)]$. If this angle is small, then the two vectors are similar; hence, they indicate similar colors and belong to the same surface. All the pixels for the segment whose vector is just matched, are considered and collected as a shadow binary mask image. Once all the segments are verified, the result are the detected shadows. The objects are obtained by subtracting detected shadows from mask M_1 that contained both moving object and moving shadows.

4 EXPERIMENTAL RESULTS

4.1 Data and Parameters

Video data is obtained with a SONY DCR-VX1000 camera with three CCD sensors. No filter is used and all the controls for the camera (shutter speed, aperture, gain control, white balance, etc.) are set in automatic mode. The data consists of moving vehicles and people with background materials and surfaces found in urban areas, which contain structures such as buildings, houses,

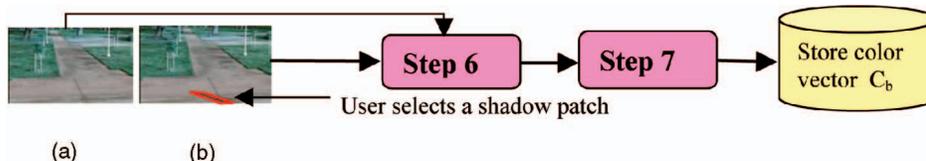


Fig. 4. (a) Background image and (b) foreground image. During the training, a user selects a patch of shadowed region from a frame. Steps 6 and 7 of our algorithm are then applied and the body color vector C_b is estimated and stored.

Example 1(a) - Date: 1-24-2003; Time: 4:00 p.m.; <i>Sun Direction: Left to right; Surface Orientation: Flat</i> <i>Surface Type: Asphalt, Concrete.</i>		Example 1(b) - Date: 1-24-2003; Time: 3:00 p.m.; <i>Sun Direction: Right to left; Surface Orientation: Flat</i> <i>Surface Type: Asphalt.</i>																																																																		
Frame # 73	92	31	117	Steps																																																																
				Input Video																																																																
				1 moving object detection																																																																
				2 initial shadow pixel reduction																																																																
				3 blue ratio test																																																																
				4 A albedo-ratio segmentation																																																																
				4 B size filter																																																																
				7 verification																																																																
				% of shadow pixels detected (green) superimposed on ground-truth shadow (white).																																																																
				% of object pixels detected (red) superimposed on ground-truth object (white).																																																																
<table border="1"> <thead> <tr> <th></th> <th>S</th> <th>O</th> <th>B</th> </tr> </thead> <tbody> <tr> <th>S</th> <td>85.8</td> <td>7.5</td> <td>6.7</td> </tr> <tr> <th>O</th> <td>4.2</td> <td>72.5</td> <td>23.3</td> </tr> <tr> <th>B</th> <td>0.2</td> <td>1.0</td> <td>98.8</td> </tr> </tbody> </table>		S	O	B	S	85.8	7.5	6.7	O	4.2	72.5	23.3	B	0.2	1.0	98.8	<table border="1"> <thead> <tr> <th></th> <th>S</th> <th>O</th> <th>B</th> </tr> </thead> <tbody> <tr> <th>S</th> <td>86.0</td> <td>14.0</td> <td>0</td> </tr> <tr> <th>O</th> <td>3.1</td> <td>91.4</td> <td>6.3</td> </tr> <tr> <th>B</th> <td>0.7</td> <td>2.8</td> <td>96.5</td> </tr> </tbody> </table>		S	O	B	S	86.0	14.0	0	O	3.1	91.4	6.3	B	0.7	2.8	96.5	<table border="1"> <thead> <tr> <th></th> <th>S</th> <th>O</th> <th>B</th> </tr> </thead> <tbody> <tr> <th>S</th> <td>51.8</td> <td>39.3</td> <td>8.9</td> </tr> <tr> <th>O</th> <td>0</td> <td>65.0</td> <td>35.0</td> </tr> <tr> <th>B</th> <td>0.1</td> <td>0.4</td> <td>99.5</td> </tr> </tbody> </table>		S	O	B	S	51.8	39.3	8.9	O	0	65.0	35.0	B	0.1	0.4	99.5	<table border="1"> <thead> <tr> <th></th> <th>S</th> <th>O</th> <th>B</th> </tr> </thead> <tbody> <tr> <th>S</th> <td>78.8</td> <td>18.6</td> <td>10.6</td> </tr> <tr> <th>O</th> <td>0.6</td> <td>51.1</td> <td>48.3</td> </tr> <tr> <th>B</th> <td>0</td> <td>4.2</td> <td>95.8</td> </tr> </tbody> </table>		S	O	B	S	78.8	18.6	10.6	O	0.6	51.1	48.3	B	0	4.2	95.8	Confusion Matrices
	S	O	B																																																																	
S	85.8	7.5	6.7																																																																	
O	4.2	72.5	23.3																																																																	
B	0.2	1.0	98.8																																																																	
	S	O	B																																																																	
S	86.0	14.0	0																																																																	
O	3.1	91.4	6.3																																																																	
B	0.7	2.8	96.5																																																																	
	S	O	B																																																																	
S	51.8	39.3	8.9																																																																	
O	0	65.0	35.0																																																																	
B	0.1	0.4	99.5																																																																	
	S	O	B																																																																	
S	78.8	18.6	10.6																																																																	
O	0.6	51.1	48.3																																																																	
B	0	4.2	95.8																																																																	

(a)

(b)

Fig. 5. Example 1, vehicles casting shadows on asphalt and walkway concrete.

walkways, roads, parks, etc. The range of physical conditions in our experiments (shown in the heading of examples) include:

1. background surface materials—grass (parks), wood (trees), concrete (buildings), asphalt (roads);
2. foreground surface materials—semipermeable (human skin), dielectric (clothing), vehicle body;
3. colors—typical uniform surface colors, textured colors, saturated and neutral;

4. surface slopes—vertical, horizontal, slopes (such as a hill);
5. sun angle—morning, high noon, early afternoon, late afternoon; and
6. distance to objects: approximately 10-200 feet.

The number of Gaussians (g at Step 1) is fixed at 3.

The three parameters that affect our algorithm are: 1) threshold $T = 0.05$ selected for surface albedo ratio segmentation (Step 4A), 2) size filter $S = 10$ pixels (Step 4B), and c) the angle threshold $\theta = 1^\circ$ selected for color equivalency (Step 7). All these thresholds

Example 2: Date: 11-01-2001; Time: 3 p.m.; Sun Direction: Left to right; Surface Orientation: Flat, Horizontal, Vertical; Surface Type: textured, concrete, brick.			Example 3: Date: 10.08.2002; Time: 4:30 p.m.; Sun Direction: left to right; Surface Orientation: Flat; Surface Type: Textured Concrete.																																																																																														
Frame #	344	705	793	86	624	642	Steps																																																																																										
							Input Video																																																																																										
							1 moving object detection																																																																																										
							2 initial shadow pixel reduction																																																																																										
							3 blue ratio test																																																																																										
							4 A albedo-ratio segmentation																																																																																										
							4 B size filter																																																																																										
							7 verification																																																																																										
							% of shadow detected (green) superimposed on ground-truth shadow (white).																																																																																										
							% of object detected (red) superimposed on ground-truth object (white).																																																																																										
	<table border="1"> <thead> <tr><th>S</th><th>O</th><th>B</th></tr> </thead> <tbody> <tr><td>S</td><td>59.0</td><td>21.3</td><td>19.7</td></tr> <tr><td>O</td><td>0</td><td>67.5</td><td>32.5</td></tr> <tr><td>B</td><td>0</td><td>2.2</td><td>97.8</td></tr> </tbody> </table>	S	O	B	S	59.0	21.3	19.7	O	0	67.5	32.5	B	0	2.2	97.8	<table border="1"> <thead> <tr><th>S</th><th>O</th><th>B</th></tr> </thead> <tbody> <tr><td>S</td><td>83.8</td><td>6.7</td><td>9.5</td></tr> <tr><td>O</td><td>2.5</td><td>62.2</td><td>35.3</td></tr> <tr><td>B</td><td>3.2</td><td>0.0</td><td>96.8</td></tr> </tbody> </table>	S	O	B	S	83.8	6.7	9.5	O	2.5	62.2	35.3	B	3.2	0.0	96.8	<table border="1"> <thead> <tr><th>S</th><th>O</th><th>B</th></tr> </thead> <tbody> <tr><td>S</td><td>77.2</td><td>8.0</td><td>14.8</td></tr> <tr><td>O</td><td>7.0</td><td>60.8</td><td>32.2</td></tr> <tr><td>B</td><td>0</td><td>1.9</td><td>98.1</td></tr> </tbody> </table>	S	O	B	S	77.2	8.0	14.8	O	7.0	60.8	32.2	B	0	1.9	98.1	<table border="1"> <thead> <tr><th>S</th><th>O</th><th>B</th></tr> </thead> <tbody> <tr><td>S</td><td>92.1</td><td>2.3</td><td>5.6</td></tr> <tr><td>O</td><td>10.1</td><td>73.0</td><td>16.9</td></tr> <tr><td>B</td><td>4.2</td><td>0.2</td><td>95.6</td></tr> </tbody> </table>	S	O	B	S	92.1	2.3	5.6	O	10.1	73.0	16.9	B	4.2	0.2	95.6	<table border="1"> <thead> <tr><th>S</th><th>O</th><th>B</th></tr> </thead> <tbody> <tr><td>S</td><td>79.6</td><td>17.0</td><td>3.4</td></tr> <tr><td>O</td><td>6.3</td><td>80.0</td><td>13.7</td></tr> <tr><td>B</td><td>0.01</td><td>7.9</td><td>92.0</td></tr> </tbody> </table>	S	O	B	S	79.6	17.0	3.4	O	6.3	80.0	13.7	B	0.01	7.9	92.0	<table border="1"> <thead> <tr><th>S</th><th>O</th><th>B</th></tr> </thead> <tbody> <tr><td>S</td><td>80.7</td><td>16.3</td><td>3.0</td></tr> <tr><td>O</td><td>8.9</td><td>74.1</td><td>17.0</td></tr> <tr><td>B</td><td>0.1</td><td>10.7</td><td>89.2</td></tr> </tbody> </table>	S	O	B	S	80.7	16.3	3.0	O	8.9	74.1	17.0	B	0.1	10.7	89.2	Confusion Matrices
S	O	B																																																																																															
S	59.0	21.3	19.7																																																																																														
O	0	67.5	32.5																																																																																														
B	0	2.2	97.8																																																																																														
S	O	B																																																																																															
S	83.8	6.7	9.5																																																																																														
O	2.5	62.2	35.3																																																																																														
B	3.2	0.0	96.8																																																																																														
S	O	B																																																																																															
S	77.2	8.0	14.8																																																																																														
O	7.0	60.8	32.2																																																																																														
B	0	1.9	98.1																																																																																														
S	O	B																																																																																															
S	92.1	2.3	5.6																																																																																														
O	10.1	73.0	16.9																																																																																														
B	4.2	0.2	95.6																																																																																														
S	O	B																																																																																															
S	79.6	17.0	3.4																																																																																														
O	6.3	80.0	13.7																																																																																														
B	0.01	7.9	92.0																																																																																														
S	O	B																																																																																															
S	80.7	16.3	3.0																																																																																														
O	8.9	74.1	17.0																																																																																														
B	0.1	10.7	89.2																																																																																														

(a)

(b)

Fig. 6. (a) Example 2 and (b) Example 3. Shadows are cast on different vertical and horizontal surfaces.

remain constant for all the results shown in the paper. The test video data are short sequences (400-4,500 frames); therefore, no background update is necessary.

4.2 Performance Evaluation

In order to evaluate the performance of our algorithm, the ground-truth data are obtained for each frame by manually drawing a contour around the moving objects and their shadows. Results are shown using a confusion matrix for shadow (S), object (O), and background (B). During the training phase, for the initial

estimation of the diffuse background color, the user selects areas of typical backgrounds when they are shadowed. As shown in Fig. 4, after user selection, Steps 6 and 7 are applied to these areas to calculate a table of diffuse color vectors for various materials. This table is used during the testing. Note that the body color of the background surface cannot be guaranteed to have the correct color vector if its reflectance is due to multiple illumination sources of different SPDs. The effect of sky illumination must first be accounted for. This is only possible if the other sources (e.g., sun) can be blocked, the surface reflectance due to sky estimated,

Example 4: Date: 11-01-2001; Time: 3:30 p.m.; Sun Direction: Right to left; Surface Orientation: Inclined, Curved; Surface Type: Grass.		Example 5: Date: 11-01-2001; Time: 4:30 p.m.; Sun Direction: Right to left; Surface Orientation: Flat, Horizontal, Vertical; Surface Type: Grass, Red Tile.		Example 6: Date: 10-26-2001; Time: 4:30 p.m.; Sun Direction: Right to left; Surface Orientation: Flat, Horizontal; Surface Type: Grass, Textured Concrete.																																																																																																		
Frame # 1076	1252	312	360	765	4300	Steps																																																																																																
						Input Video																																																																																																
						1 moving object detection																																																																																																
						2 initial shadow pixel reduction																																																																																																
						3 blue ratio test																																																																																																
						4 A albedo-ratio segmentation																																																																																																
						4 B size filter																																																																																																
						7 verification																																																																																																
						% of shadow detected (green) superimposed on ground-truth shadow (white).																																																																																																
						% of object detected (red) superimposed on ground-truth object (white).																																																																																																
<table border="1"> <tr><td></td><td>S</td><td>O</td><td>B</td></tr> <tr><td>S</td><td>5.1</td><td>92.3</td><td>2.6</td></tr> <tr><td>O</td><td>0</td><td>84.1</td><td>15.9</td></tr> <tr><td>B</td><td>0</td><td>4.1</td><td>95.9</td></tr> </table>		S	O	B	S	5.1	92.3	2.6	O	0	84.1	15.9	B	0	4.1	95.9	<table border="1"> <tr><td></td><td>S</td><td>O</td><td>B</td></tr> <tr><td>S</td><td>79.8</td><td>20.2</td><td>0</td></tr> <tr><td>O</td><td>0</td><td>95.4</td><td>4.6</td></tr> <tr><td>B</td><td>0</td><td>4.2</td><td>95.8</td></tr> </table>		S	O	B	S	79.8	20.2	0	O	0	95.4	4.6	B	0	4.2	95.8	<table border="1"> <tr><td></td><td>S</td><td>O</td><td>B</td></tr> <tr><td>S</td><td>51.2</td><td>17.4</td><td>31.4</td></tr> <tr><td>O</td><td>9.1</td><td>27.3</td><td>63.6</td></tr> <tr><td>B</td><td>0</td><td>3.5</td><td>94.1</td></tr> </table>		S	O	B	S	51.2	17.4	31.4	O	9.1	27.3	63.6	B	0	3.5	94.1	<table border="1"> <tr><td></td><td>S</td><td>O</td><td>B</td></tr> <tr><td>S</td><td>50.5</td><td>16.0</td><td>33.5</td></tr> <tr><td>O</td><td>6.8</td><td>58.1</td><td>35.1</td></tr> <tr><td>B</td><td>0</td><td>5.9</td><td>94.1</td></tr> </table>		S	O	B	S	50.5	16.0	33.5	O	6.8	58.1	35.1	B	0	5.9	94.1	<table border="1"> <tr><td></td><td>S</td><td>O</td><td>B</td></tr> <tr><td>S</td><td>63.5</td><td>31.0</td><td>5.5</td></tr> <tr><td>O</td><td>1.6</td><td>94.3</td><td>4.1</td></tr> <tr><td>B</td><td>0</td><td>2.1</td><td>97.9</td></tr> </table>		S	O	B	S	63.5	31.0	5.5	O	1.6	94.3	4.1	B	0	2.1	97.9	<table border="1"> <tr><td></td><td>S</td><td>O</td><td>B</td></tr> <tr><td>S</td><td>11.9</td><td>65.4</td><td>22.7</td></tr> <tr><td>O</td><td>6.2</td><td>82.2</td><td>11.6</td></tr> <tr><td>B</td><td>0</td><td>1.8</td><td>98.2</td></tr> </table>		S	O	B	S	11.9	65.4	22.7	O	6.2	82.2	11.6	B	0	1.8	98.2	Confusion Matrices
	S	O	B																																																																																																			
S	5.1	92.3	2.6																																																																																																			
O	0	84.1	15.9																																																																																																			
B	0	4.1	95.9																																																																																																			
	S	O	B																																																																																																			
S	79.8	20.2	0																																																																																																			
O	0	95.4	4.6																																																																																																			
B	0	4.2	95.8																																																																																																			
	S	O	B																																																																																																			
S	51.2	17.4	31.4																																																																																																			
O	9.1	27.3	63.6																																																																																																			
B	0	3.5	94.1																																																																																																			
	S	O	B																																																																																																			
S	50.5	16.0	33.5																																																																																																			
O	6.8	58.1	35.1																																																																																																			
B	0	5.9	94.1																																																																																																			
	S	O	B																																																																																																			
S	63.5	31.0	5.5																																																																																																			
O	1.6	94.3	4.1																																																																																																			
B	0	2.1	97.9																																																																																																			
	S	O	B																																																																																																			
S	11.9	65.4	22.7																																																																																																			
O	6.2	82.2	11.6																																																																																																			
B	0	1.8	98.2																																																																																																			

(a)

(b)

(c)

Fig. 7. Three examples. (a) Example 4, (b) Example 5, and (c) Example 6.

and then subtracted from the original background. For testing, once the moving pixels and associated mask M_1 are obtained in Step 1, they are used as the input to Step 2 of the algorithm. At each subsequent step of the algorithm, we only need to modify the image pixels under the mask for computational efficiency.

Example 1. Fig. 5 indicates two examples from two different scenes with moving vehicles where shadow of vehicles either follows or precedes them. The scene includes both asphalt with

different texture, and concrete. As indicated by the confusion matrix for each example, when the initial detection accuracy is reasonably high, the shadow detection performs well. In frames 31 and 117 of Example 1b, the vehicle windows are not initially detected where the pixels on the windows of the vehicles are as dark as the background. Note that the shadow detection algorithm performs better for cases where both objects and their shadows are closer to the camera.

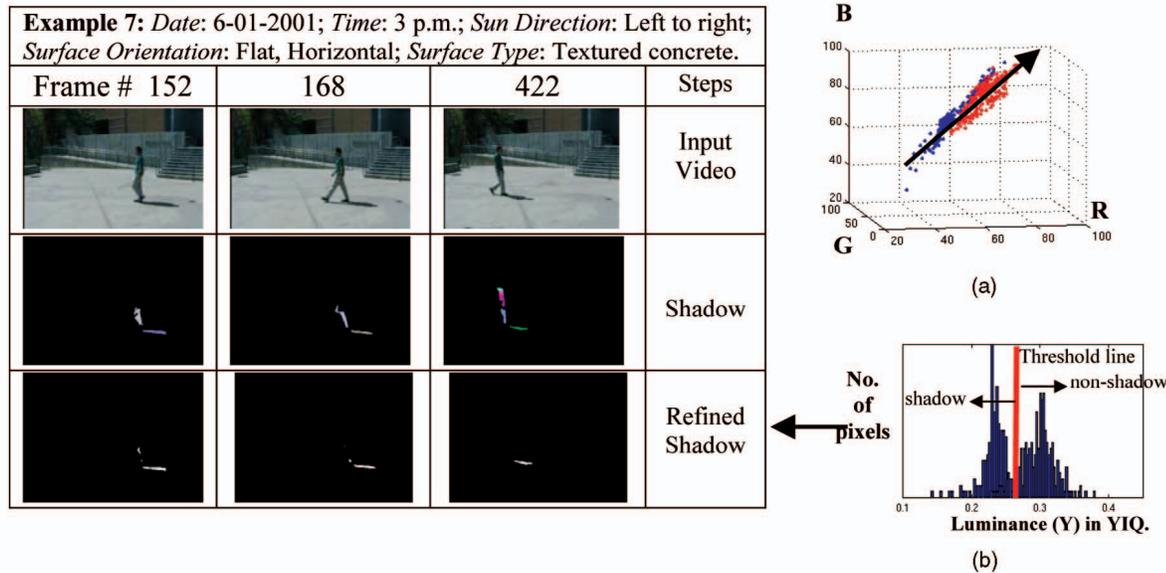


Fig. 8. Similar foreground and background colors, self-shadowing, and refined shadow. (a) Diffuse color of pant (red) and shadow (blue) for frame 152 in RGB space. (b) Histogram analysis.

Example 2. Fig. 6a is representative of different types of background surfaces including vertical, horizontal, textured, uniform, brick, and concrete. The detection algorithm performed consistently for all the geometry and surface types present in this example. As indicated in the confusion matrix, on the average 73 percent of shadow pixels are correctly detected while the percentage of object pixels classified as shadow is minimal.

Example 3. Fig. 6b is an example of shadow cast on typical textured concrete found in buildings with subject wearing a textured shirt. Despite the noise in the initial detection (Step 1), on the average over 80 percent of the shadow is correctly detected.

Example 4. Fig. 7a represents an inclined and curved grass surface. As the subject and its shadow move closer to the camera, the detection improves. This is also an example of a surface that exhibits highly saturated color and specularities due to the surface type of grass and angles of incidence. This is a challenging test since we do not account for specularities that introduce noise, which affects the estimation of C_b . Due to high saturation, Step 3 is bypassed.

Example 5. Fig. 7b shows a highly saturated color surface. The background is red polished tile and the foreground subject is wearing textured and uniformed colored clothes. The shadow cast by the moving object on the grass is not initially detected at Step 1; therefore, it is not considered by the rest of the algorithm. Due to the proximity of the surfaces to each other there exist high interreflections that are not modeled in our approach.

Example 6. Fig. 7c shows multiple moving objects at different distances from the camera. The initial moving object detection performed superbly, but shadows of objects at considerable distances are missed. This is due to the choice of the size filter used at Step 4B. Shadows of objects that are far away are segmented into smaller segments which are filtered out.

Example 7. Fig. 8 illustrates a difficult shadow and object color situation. In this case, part of the object and the shadow have the same diffuse color—the person's pant has a neutral gray color, the same as the background, and it is self-shadowed. As a result, part of the pant is labeled as shadow. The self-shadowed region of pant, however, has higher luminance than the real shadow on the ground; hence, we utilize a statistical luminance

test based on histogram analysis to further classify these two regions as shown.

4.3 Discussion of Experimental Results and Parameters

Accuracy versus Number of Shadow Pixels: Fig. 9 shows the detection of shadow for Example 4. It shows the percentage of shadow pixels detected correctly vs. distance of the moving object to camera and number of (groundtruth) shadow pixels. As the object comes closer to the camera, its shadow gets larger and the camera signal becomes better since it is inversely proportional to the distance of the reflected light. Larger areas with uniform reflectance contribute to larger segments in Steps 4 and 6, thus contributing to better diffuse color estimation.

Effect of Parameters T and S: Table 2 shows the effect of parameters T and S. It shows the initial number of segments with uniform reflectance, obtained at Step 4A, with a threshold value of $T = 0.05$. A larger value for T will merge smaller segments into a large segment whose diffuse color component may not be reliable due to inclusion of several types of surfaces, each with a different color. As indicated by our experiments, we have found that a value of the threshold $T = 0.05$ is appropriate. Table 2 also shows that the size segment threshold (S), Step 4B, greatly reduces the computational load before Step 7. By choosing large segments, we also eliminate smaller segments as well that may actually be shadows. This is indicated in Example 6 where several subjects are at different distances to the camera. The initial detection may fail to detect enough pixels on objects that are far away; hence, segmentation may result in many small surfaces that may be filtered out by the choice of S.

Effect of Parameter θ : The parameter (θ) determines the sensitivity to color vector matching. Plots in Figs. 10a and 10b are receiver operating characteristic curves (ROC) for two different

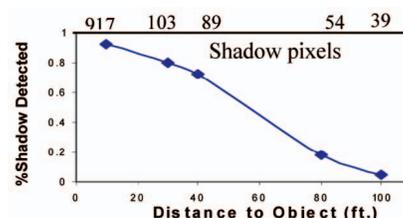


Fig. 9. Effect of shadow size.

TABLE 2
Number of Segments after Segmentation (Step 4A), Size Filter (Step 4B), and Verification (Step 7)

Example	1a		1b		2			3			4		5		6	
Frame #	73	92	31	117	86	624	642	344	705	793	1076	1252	312	360	765	4300
Step 4A	119	271	98	132	424	1156	995	1123	1845	757	626	679	631	648	1014	887
Step 4B	22	31	21	48	17	20	15	46	7	20	3	4	26	28	18	22
Step 7	5	18	6	21	15	14	10	17	4	15	1	3	12	16	15	17

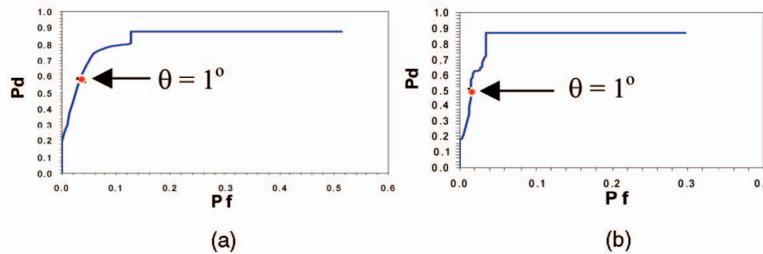


Fig. 10. ROC plots—(a) frame 344 of Example 2 and (b) frame 360 of Example 5.

scenes (Example 2 and Example 3) that indicate the sensitivity of the algorithm to color vector matching angle (θ). We use a threshold of 1° for all our examples, which corresponds to values at the lower left part of the curves where probability of false alarm is extremely low. The upper bound on the detection rate (P_d) is dependent on the initial moving object detection algorithm, the size segment threshold, the accuracy of the color estimation, and whether we consider border pixels or not. In most cases, the uniformity test causes the border pixels for object and shadow to naturally fall into smaller segments and the size filter eliminates them. We have adopted to shrink the object and its shadow by one pixel first. We note that this reduces the detection rate.

5 CONCLUSIONS

In this paper, a novel approach consisting of an original ensemble of techniques for detecting shadows and moving objects based on sound physical models was presented. Experimental results demonstrated that our approach is robust to widely different 1) background surfaces, commonly encountered in real-world urban environments, such as concrete, asphalt, bricks, tiles, and grass, 2) foreground materials such as human skin, variety of colored and textured textiles, metallic surfaces such as vehicle surfaces, and 3) illumination conditions at different times of the day and locations. Furthermore, the physical attributes such as surface type, surface roughness, and surface orientation and illumination condition were physically associated to the scene and the detection algorithm as shown in our experimental results. Steps 1, 2, and 3 of our algorithm effectively reduced the computational load for potential shadow pixels, whereas Steps 4, 5, and 6 provided a new robust method for distinguishing the shadow pixels from the moving object pixels. The spatio-temporal albedo ratio test provided a criterion for physically driven surface segmentation. The surface color estimation is based on a physical model and we have provided a robust method for estimating the body color of surfaces. The integrated system is implemented in Matlab, which is an interpreted environment, and it takes 4 seconds to complete processing for one frame. This environment is chosen for its extensive functionality, but it is inherently slow. In the future, a real-time implementation at video rate will be explored.

ACKNOWLEDGMENTS

This work was supported in part by grants F49620-02-1-0315 and DAAD19-01-0357; the contents and information do not necessarily reflect the position or policy of the US Government.

REFERENCES

- [1] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti, "Improving Shadow Suppression in Moving Object Detection with HSV Color Information," *Proc. IEEE Intelligent Transportation Systems Conf.*, pp. 334-339, Aug. 2001.
- [2] S. Das and B. Bhanu, "A System for Model-Based Object Recognition in Perspective Aerial Images," *Pattern Recognition*, vol. 3, no. 34, pp. 465-491, 1998.
- [3] R. Gershon, A.D. Jepson, and J.K. Tsotsos, "Ambient Illumination and the Determination of Material Changes," *J. OSA*, vol. 3, no. 10, pp. 1700-1707, Oct. 1986.
- [4] I. Haritaoglu, D. Harwood, and L. Davis, "W4: Real Time Surveillance of People and Their Activities," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809-830, Aug. 2000.
- [5] T. Horprasert, D. Harwood, and L.S. Davis, "A Statistical Approach for Real-Time Robust Background Subtraction and Shadow Detection," *Proc. IEEE Int'l Conf. Computer Vision, Frame Rate Workshop*, pp. 1-19, 1999.
- [6] J.W. Hsieh, W.F. Hu, C.J. Chang, and Y.S. Chen, "Shadow Elimination for Effective Moving Object Detection by Gaussian Shadow Modeling," *Int'l J. Image and Vision Computing*, vol. 21, pp. 505-516, 2003.
- [7] I. Mikic, P.C. Cosman, G.T. Kogut, and M.M. Trivedi, "Moving Shadow and Object Detection in Traffic Scenes," *Proc. Int'l Conf. Pattern Recognition*, vol. 1, no. 1, pp. 321-324, 1998.
- [8] S. Nadimi and B. Bhanu, "Multistrategy Fusion Using Mixture Model for Moving Object Detection," *Proc. Int'l Conf. Multisensor Fusion and Integration for Intelligent Systems*, pp. 317-322, 2001.
- [9] S.K. Nayar and R.M. Bolle, "Reflectance Based Object Recognition," *Int'l J. Computer Vision*, vol. 17, no. 3, pp. 219-240, 1996.
- [10] K. Onoguchi, "Shadow Elimination Method for Moving Object Detection," *Proc. Int'l Conf. Pattern Recognition*, vol. 1, no. 2, pp. 583-587, 1998.
- [11] A. Prati, I. Mikic, M.M. Trivedi, and R. Cucchiara, "Detecting Moving Shadows: Algorithms and Evaluation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 918-923, July 2003.
- [12] J.M. Scanlan, D.M. Chabries, and R.W. Christiansen, "A Shadow Detection and Removal Algorithm for 2-D Images," *Proc. Int'l Conf. Acoustics, Speech and Signal Processing*, vol. 4, pp. 2057-2060, 1990.
- [13] S.A. Shafer, "Using Color to Separate Reflection Components," *Color Research and Application*, vol. 10, no. 4, pp. 210-218, 1985.
- [14] Y. Sonoda and T. Ogata, "Separation of Moving Objects and Their Shadows, and Application to Tracking of Loci in the Monitoring Images," *Proc. Int'l Conf. Signal Processing*, vol. 2, no. 2, pp. 1261-1264, 1998.
- [15] J. Stauder, R. Mech, and J. Ostermann, "Detection of Moving Cast Shadows for Object Segmentation," *IEEE Trans. Multimedia*, vol. 1, no. 1, pp. 65-76, Mar. 1999.
- [16] C. Stauffer and W.E.L. Grimson, "Learning Patterns of Activity Using Real-Time Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747-757, Aug. 2000.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.