

A Robust Background Subtraction and Shadow Detection

Thanarat Horprasert

David Harwood

Larry S. Davis

Computer Vision Laboratory
University of Maryland
College Park, MD 20742

{thanarat,harwood,lsd}@umiacs.umd.edu

ABSTRACT

This paper presents a novel algorithm for detecting moving objects from a static background scene that contains shading and shadows using color images. Although the background subtraction technique has been used for years in many vision systems as a preprocessing step for object detection and tracking, most of these algorithms are susceptible to both global and local illumination changes such as shadows and highlights. These cause the consequent processes, e.g. tracking, recognition, etc., to fail. This problem is the underlying motivation of our work. We develop a robust and efficiently computed background subtraction algorithm that is able to cope with local illumination change problems, such as shadows and highlights, as well as global illumination changes. Experimental results, which demonstrate the system’s performance, are also shown.

Keywords: segmentation, color model, background subtraction, shadow detection.

1 INTRODUCTION

The capability of extracting moving objects from a video sequence is a fundamental and crucial problem of many vision systems that include video surveillance [1, 2], traffic monitoring [3], human detection and tracking for video teleconferencing or human-machine interface [4, 5, 6], video editing, among other applications. Typically, the common approach for discriminating moving object from the background scene is *background subtraction*. The idea is to subtract the current image from a reference image, which is acquired from a static background during a period of time. The subtraction leaves only non-stationary or new objects, which include the objects’ entire silhouette region. The technique has been used for years in many vision systems as a preprocessing step for object detection and tracking [for examples, [1, 4, 5, 7, 8, 9]]. The results of the existing algorithms are fairly good; in addition, many of them run in real-time. However, many of these algorithms are susceptible to both global and local illumination changes such as shadows and highlights. These cause the consequent processes, e.g. tracking, recognition, etc., to fail. The accuracy and efficiency of the detection are very crucial to those tasks. This problem is the underlying motivation of this work. We want to

develop a robust and efficiently computed background subtraction algorithm that is able to cope with the local illumination change problems, such as shadows and highlights, as well as the global illumination changes. Being able to detect shadows is also very useful to many applications especially in “Shape from Shadow” problems [10, 11, 12, 13]. Our method must also address requirements of sensitivity, reliability, robustness, and speed of detection.

In this paper, we present a novel algorithm for detecting moving objects from a static background scene that contains shading and shadows using color images. In next section, we propose a new computational color model (brightness distortion and chromaticity distortion) that helps us to distinguish shading background from the ordinary background or moving foreground objects. Next, we propose an algorithm for pixel classification and threshold selection. Experimental results and sample applications are shown in Section 4 and 5 respectively.

2 COLOR MODEL

One of the fundamental abilities of human vision is *color constancy* [14]. Humans tend to be able to assign a constant color to an object even under changing of illumination over time or space. The perceived color of a point in a scene depends on many factors including physical properties of the point on the surface of the object. Important physical properties of the surface in color vision are surface spectral reflectance properties, which are invariant to changes of illumination, scene composition or geometry. On *Lambertian*, or perfect matte surfaces, the perceived color is the product of *illumination* and *surface spectral reflectance*.

This led to our idea of designing a color model that separates these two terms; in other words, that separates the *brightness* from the *chromaticity* component. Figure 1 illustrates the proposed color model in three-dimensional RGB space. Consider a pixel, i , in the image; let $E_i = [E_R(i), E_G(i), E_B(i)]$ represent the pixel’s *expected RGB color* in the reference or background image. The line OE_i passing through the origin and the point E_i is called *expected chromaticity line*. Next, let $I_i = [I_R(i), I_G(i), I_B(i)]$ denote the pixel’s RGB color value in a current image that we want to subtract from the background. Basically, we want to measure the distortion of I_i from E_i . We do this by decomposing the distortion measurement into two compo-

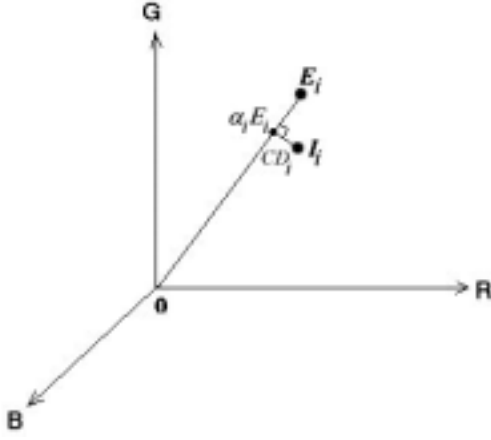


Figure 1. Our proposed color model in the three-dimensional RGB color space; the background image is statistically pixel-wise modeled. E_i represents an expected color of a given i^{th} pixel and I_i represents the color value of the pixel in a current image. The difference between I_i and E_i is decomposed into brightness (α_i) and chromaticity (CD_i) components.

nents, *brightness distortion* and *chromaticity distortion*, defined below.

Brightness Distortion (α)

The brightness distortion (α) is a scalar value that brings the observed color close to the expected chromaticity line. It is obtained by minimizing

$$\phi(\alpha_i) = (I_i - \alpha_i E_i)^2 \quad (1)$$

α_i represents the pixel's strength of brightness with respect to the expected value. α_i is 1 if the brightness of the given pixel in the current image is the same as in the reference image. α_i is less than 1 if it is darker, and greater than 1 if it becomes brighter than the expected brightness.

Color Distortion (CD)

Color distortion is defined as the orthogonal distance between the observed color and the expected chromaticity line. The color distortion of a pixel i is given by

$$CD_i = \|I_i - \alpha_i E_i\| \quad (2)$$

3 BACKGROUND SUBTRACTION

The basic scheme of background subtraction is to subtract the image from a reference image that models the background scene. Typically, the basic steps of the algorithm are as follows:

- *Background modeling* constructs a reference image representing the background.
- *Threshold selection* determines appropriate threshold values used in the subtraction operation to obtain a desired detection rate.

- *Subtraction operation or pixel classification* classifies the types of a given pixel belongs, i.e., the pixel is the part of background (including ordinary background and shaded background), or it is a moving object.

Background Modeling

In the background training process, the reference background image and some parameters associated with normalization are computed over a number of static background frames. The background is modeled statistically on a pixel by pixel basis. A pixel is modeled by a 4-tuple $\langle E_i, s_i, a_i, b_i \rangle$ where E_i is the expected color value, s_i is the standard deviation of color value, a_i is the variation of the brightness distortion, and b_i is the variation of the chromaticity distortion of the i^{th} pixel. E_i, s_i, a_i and b_i are defined explicitly later in this section.

The expected color value of pixel i is given by

$$E_i = [\mu_R(i), \mu_G(i), \mu_B(i)] \quad (3)$$

where $\mu_R(i), \mu_G(i)$, and $\mu_B(i)$ are the arithmetic means of the i^{th} pixel's red, green, blue values computed over N background frames.

In reality, we rarely observe the same value for a given pixel over a period of time due to camera noise and illumination fluctuation by light sources. This variation can be modeled by the standard deviation in each band (s_i). The standard deviation of pixel i is given by

$$s_i = [\sigma_R(i), \sigma_G(i), \sigma_B(i)] \quad (4)$$

where $\sigma_R(i), \sigma_G(i)$, and $\sigma_B(i)$ are the standard deviation of the i^{th} pixel's red, green, blue values computed over N frame of the background frames.

To normalize or balance color bands in the brightness distortion and chromaticity distortion, Equation 1 and Equation 2 become

$$\begin{aligned} \alpha_i &= \min \left[\sum_{C=R,G,B} \left(\frac{I_C(i) - \alpha_i \mu_C(i)}{\sigma_C(i)} \right)^2 \right] \\ &= \frac{\left(\frac{I_R(i) \mu_R(i)}{\sigma_R^2(i)} + \frac{I_G(i) \mu_G(i)}{\sigma_G^2(i)} + \frac{I_B(i) \mu_B(i)}{\sigma_B^2(i)} \right)}{\left(\left[\frac{\mu_R(i)}{\sigma_R(i)} \right]^2 + \left[\frac{\mu_G(i)}{\sigma_G(i)} \right]^2 + \left[\frac{\mu_B(i)}{\sigma_B(i)} \right]^2 \right)} \end{aligned} \quad (5)$$

$$CD_i = \sqrt{\sum_{C=R,G,B} \left(\frac{I_C(i) - \alpha_i \mu_C(i)}{\sigma_C(i)} \right)^2} \quad (6)$$

Next, we consider the variation of the brightness and chromaticity distortions over space and time of the training background images. We found that different pixels yield different distributions of α and CD , shown in Figure 2a, b. These variations are embedded in the background model as a_i and b_i in the 4-tuple background model for each pixel, and are used as normalization factors.

a_i represents the variation of the brightness distortion of i^{th} pixel, which is given by

$$a_i = RMS(\alpha_i) = \sqrt{\frac{\sum_{i=0}^N (\alpha_i - 1)^2}{N}} \quad (7)$$

b_i represents the variation of the chromaticity distortion of the i^{th} pixel, which is given by

$$b_i = RMS(CD_i) = \sqrt{\frac{\sum_{i=0}^N (CD_i)^2}{N}} \quad (8)$$

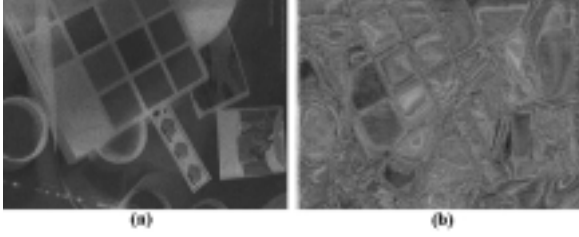


Figure 2. An illustration demonstrates the variations of brightness distortion and chromaticity distortion of different pixel colors over 100 images of a static scene. (a) is an image of a_i scaled by 2000 and (b) is an image of b_i scaled by 100.

Pixel Classification or Subtraction Operation

The difference between the reference image and the current image is evaluated in this step. The difference is decomposed into brightness and chromaticity components. Applying the suitable thresholds on the brightness distortion (α) and the chromaticity distortion (CD) of a pixel i yields an object mask $M(i)$ which indicates the type of the pixel. Our method classifies a given pixel into four categories. A pixel in the current image is

- *Original background (B)* if it has both brightness and chromaticity similar to those of the same pixel in the background image.
- *Shaded background or shadow (S)* if it has similar chromaticity but lower brightness than those of the same pixel in the background image. This is based on the notion of the shadow as a semi-transparent region in the image, which retains a representation of the underlying surface pattern, texture or color value [15].
- *Highlighted background (H)*, if it has similar chromaticity but higher brightness than the background image.
- *Moving foreground object (F)* if the pixel has chromaticity different from the expected values in the background image.

As mentioned above the different pixels yield different distributions of α_i and CD_i . In order to use a single threshold for all of the pixels, we need to rescale the α_i and CD_i . Let

$$\widehat{\alpha}_i = \frac{\alpha_i - 1}{a_i} \quad (9)$$

$$\widehat{CD}_i = \frac{CD_i}{b_i} \quad (10)$$

be *normalized brightness distortion* and *normalized chromaticity distortion* respectively.

Based on these definitions, a pixel is classified into one of the four categories B, S, H, F by the following decision procedure.

$$M(i) = \begin{cases} F & : \widehat{CD}_i > \tau_{CD}, \text{ else} \\ B & : \widehat{\alpha}_i < \tau_{\alpha 1} \text{ and } \widehat{\alpha}_i > \tau_{\alpha 2}, \text{ else} \\ S & : \widehat{\alpha}_i < 0, \text{ else} \\ H & : \text{otherwise} \end{cases} \quad (11)$$

where τ_{CD} , $\tau_{\alpha 1}$, and $\tau_{\alpha 2}$ are selected threshold values used to determine the similarities of the chromaticity and brightness between the background image and the current observed image. In next subsection, we will discuss the method to select suitable threshold values.

However, there might be a case where a pixel from a moving object in current image contains very low RGB values. This dark pixel will always be misclassified as a shadow. Because the color point of the dark pixel is close to the origin in RGB space and the fact that all chromaticity lines in RGB space meet at the origin, thus the color point is considered to be close or similar to any chromaticity line. To avoid this problem, we introduce a lower bound for the normalized brightness distortion ($\tau_{\alpha lo}$). Then, the decision procedure Equation 11 becomes

$$M(i) = \begin{cases} F & : \widehat{CD}_i > \tau_{CD} \text{ or } \widehat{\alpha}_i < \tau_{\alpha lo}, \text{ else} \\ B & : \widehat{\alpha}_i < \tau_{\alpha 1} \text{ and } \widehat{\alpha}_i > \tau_{\alpha 2}, \text{ else} \\ S & : \widehat{\alpha}_i < 0, \text{ else} \\ H & : \text{otherwise} \end{cases} \quad (12)$$

Automatic Threshold Selection

Typically, if the distortion distribution is assumed to be a Gaussian distribution, then to achieve a desired detection rate, r , we can threshold the distortion by $K\sigma$ where K is a constant determined by r and σ is the standard deviation of the distribution. However, we found from experiments that the distribution of $\widehat{\alpha}_i$ and \widehat{CD}_i are not Gaussian (see Figure 3). Thus, our method determines the appropriate thresholds by a statistical learning procedure. First, a histogram of the normalized brightness distortion, $\widehat{\alpha}_i$, and a histogram of the normalized chromaticity distortion, \widehat{CD}_i , are constructed as shown in Figure 3. The histograms are built from combined data through a long sequence captured during background learning period. The total sample would be NXY values for a histogram. (The image is $X \times Y$ and the number of trained background frames is N .) After constructing the histogram, the thresholds are now automatically selected according to the desired detection rate r . A threshold for chromaticity distortion, τ_{CD} , is the normalized chromaticity distortion value at the detection rate of r . In brightness distortion, two thresholds ($\tau_{\alpha 1}$ and $\tau_{\alpha 2}$) are needed to define the brightness range. $\tau_{\alpha 1}$ is the $\widehat{\alpha}_i$ value at that detection rate, and $\tau_{\alpha 2}$ is the $\widehat{\alpha}_i$ value at the $(1 - r)$ detection rate.

4 EXPERIMENTAL RESULTS

This section demonstrates the performance of the proposed algorithm on several image sequences of both indoor

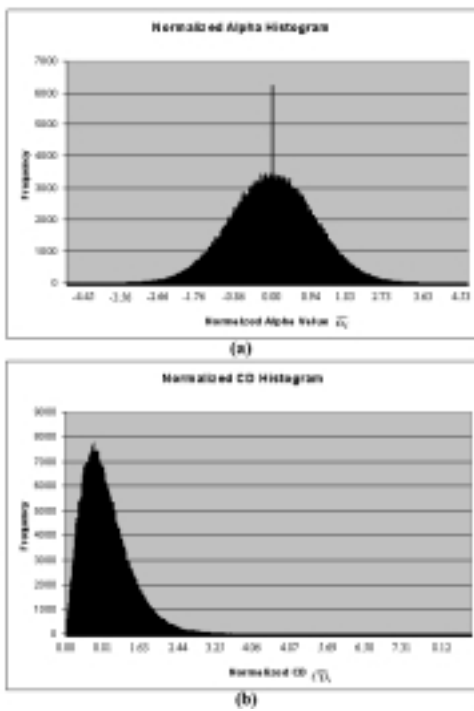


Figure 3. (a) is the normalized brightness distortion ($\hat{\alpha}_i$) histogram, and (b) is the normalized chromaticity distortion (\hat{CD}_i) histogram.

and outdoor scenes. Sequences shown here are 360x240 images. The detection rate, r , was set at 0.9999, and the lower bound of the normalized brightness distortion is set at 0.4. Figure 4 shows the result of applying the algorithm to several frames of an indoor scene containing a person walking around the room. As the person moves, he both obscures the background and casts shadows on the floor and wall. Red pixels depict the shadow, and we can easily see how the shape of the shadow changes as the person moves. Although it is difficult to see, there are green pixels, which depict the highlighted background pixels, appearing along the edge of the person's sweater. Figure 5 shows a frame of an outdoor scene containing a person walking across a street. Although there are small motions of background objects, such as the small motions of leaves and water surface, the result shows the robustness and reliability of the algorithm. Figure 6 illustrates our algorithm being able to cope with the problem of global illumination change. It shows another indoor sequence of a person moving in a room; at the middle of the sequence, the global illumination is changed by turning half of the fluorescence lamps off. The system is still able to detect the target successfully.

The detection runs at frame rate on PentiumII-400 MHz PC. The experimental results demonstrate our algorithm's sensitivity, reliability, robustness, and speed of detection.

5 SAMPLE APPLICATIONS

Motion Capture System



Figure 4. An example shows the result of our algorithm applying on a sequence of a person moving in an indoor scene. The scene, the left column is input sequence, and the middle column shows the output from our background subtraction (the foreground pixels are overlaid by light grey, the shadows are overlaid by dark grey, the highlights are overlaid by white, and the ordinary background pixels are kept as the original intensity.) The right column shows only foreground region after noise cleaning is performed.

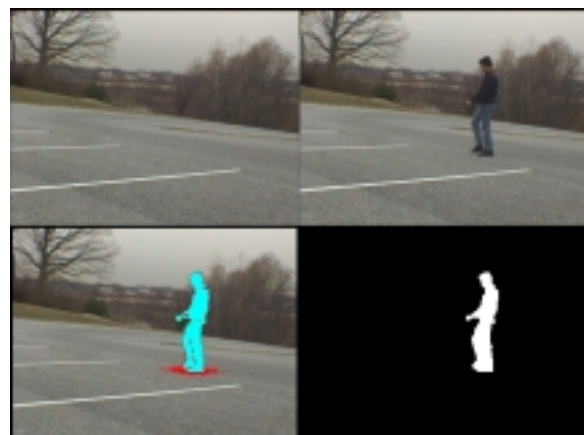


Figure 5. Another example shows the results of applying our algorithm on a sequence of an outdoor scene containing a person walking across the street.

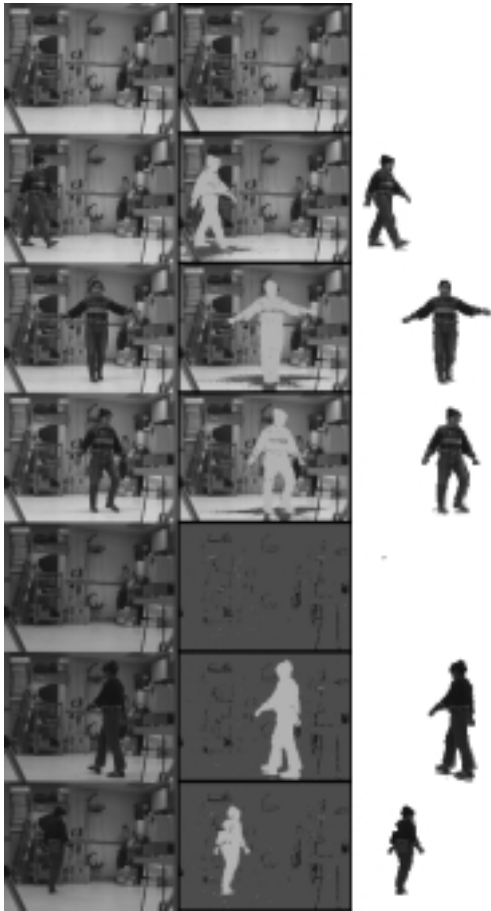


Figure 6. An illustration shows our algorithm can cope with the global illumination change. At the middle of the sequence, half of the fluorescence lamps are turned off. The result shows that the system still detects the moving object successfully.

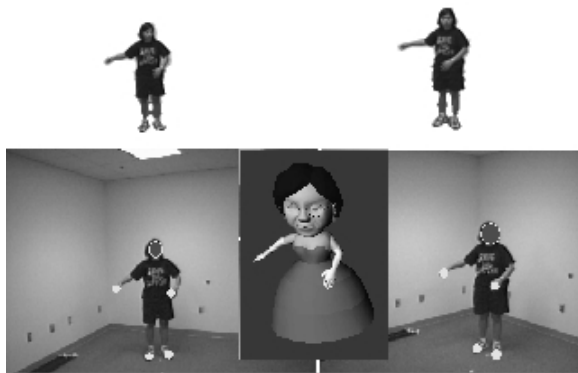


Figure 7. An illustration shows an application of the proposed background subtraction in a motion capture system.

We developed a real-time vision system for detecting and tracking human motion. The background subtraction is used as a preprocessing step to segment the moving person from the background. Then, silhouette analysis and template matching are used to locate and track the 2-D position of salient body parts. By combining 2-D body part locations from multiple views, we obtain the 3-D positions of these body parts. This information is then forwarded to a graphic reproduction system (developed by ATR's Media Integration & Communications Research Laboratory) to synthesize a CG character that has the same pose as the subject (See Figure 7).

Interactive Game

The background subtraction technique can be applied to interactive games to provide a player with control over the interaction. Figure 8 shows a snap shot of a car racing game. The movement of the player is recognized and used as a steering wheel control.

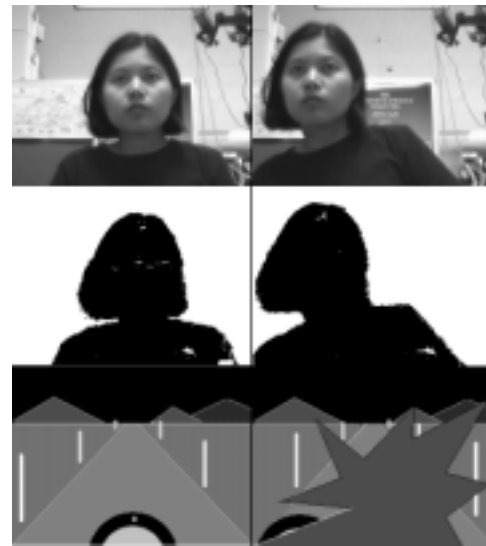


Figure 8. An illustration shows an application of moving object segmentation in an interactive game. The movement of the player is recognized and used as a steering wheel control.

Video Editing

This subsection shows the use of background subtraction in video editing. Typical video editing systems used in broadcasting are chromakey systems using either blue screen or green screen. The chromakey system extracts targets or actors, and places them into a desired scene. However, the chromakey system requires a specific background screen. Our background subtraction technique, which does not require any specific background screen or pattern, can solve this problem. Figure 9 shows a result of video editing using the proposed background subtraction technique. Note that the shadow cast by the person's hand in the captured scene also appears in the new scene. This is done by utilizing the detected shadow information from our algorithm.

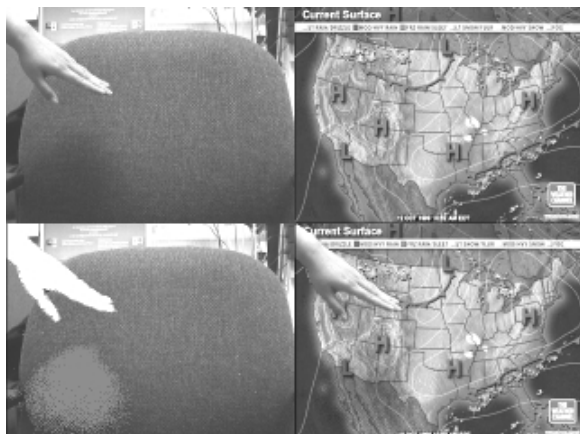


Figure 9. An application of background subtraction in video editing. The segmented moving hand is placed into a new scene. The upper left image is the captured image, and the lower left is the output from background subtraction in which the hand of the person (white) and its shadow (grey) are segmented. The upper right image is the desired scene. The lower right image is the output of our video editing system. The desired scene is overlaid by the hand of the actor as well as the shadow.

6 CONCLUSION

In this paper, we presented a novel background subtraction algorithm for detecting moving objects from a static background scene that contains shading and shadows using color images. The method is shown to be very accurate, robust, reliable and efficiently computed. Experimental results and real-time applications were also shown.

This method was designed under an assumption that the background scene is static, thus it may suffer from dynamic scene change such as an extraneous event in which there are new objects deposited into the scene and become part of the background scene. However, this problem can be coped with by allowing the system to adaptively update the background model on-the-fly, as done in [16] and [17] which is developed in our lab.

7 ACKNOWLEDGEMENTS

The support of MURI under grant NAVY N-0001-495-10521 is gratefully acknowledged.

References

- [1] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: Who? when? where? what? a real-time system for detecting and tracking people. In *Proc. the third IEEE Int'l Conf. Automatic Face and Gesture Recognition (Nara, Japan)*, pages 222–227. IEEE Computer Society Press, Los Alamitos, Calif., 1998.
- [2] P.L. Rosin. Thresholding for change detection. In *Proc. IEEE Int'l Conf. on Computer Vision*, 1998.
- [3] N. Friedman and S. Russell. Image segmentation in video sequences: A probabilistic approach. In *Proc. 13th Conf. Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1997.
- [4] C.R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfunder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1998.
- [5] J. Ohya and et al. Virtual metamorphosis. *IEEE Multimedia*, 6(2):29–39, 1999.
- [6] J. Davis and A.. Bobick. The representation and recognition of action using temporal templates. In *Proc. the Computer Vision and Pattern Recognition*, 1997.
- [7] A. Utsumi, H. Mori, J. Ohya, and M. Yachida. Multiple-human tracking using multiple cameras. In *Proc. the third IEEE Int'l Conf. Automatic Face and Gesture Recognition (Nara, Japan)*. IEEE Computer Society Press, Los Alamitos, Calif., 1998.
- [8] M. Yamada, K. Ebihara, and J. Ohya. A new robust real-time method for extracting human silhouettes from color images. In *Proc. the third IEEE Int'l Conf. Automatic Face and Gesture Recognition (Nara, Japan)*, pages 528–533. IEEE Computer Society Press, Los Alamitos, Calif., 1998.
- [9] T. Horprasert, I. Haritaoglu, C. Wren, D. Harwood, L.S. Davis, and A. Pentland. Real-time 3d motion capture. In *Proc. 1998 Workshop on Perceptual User Interface (PUI'98)*, San Francisco, 1998.
- [10] S. A. Shafer and T. Kanade. Using shadows in finding surface orientations. 22:145–176, 1983.
- [11] C. Lin and R. Nevatia. Building detection and description from a single intensity image. 72:101–121, 1998.
- [12] J. Segen and S. Kumar. Shadow gestures: 3d hand pose estimation using a single camera. In *Proc. the Computer Vision and Pattern Recognition*, 1999.
- [13] J-Y. Bouguet, M. Weber, and P. Perona. What do planar shadows tell about scene geometry? In *Proc. the Computer Vision and Pattern Recognition*, 1999.
- [14] A.C. Hurlbert. The computation of color. Technical report, MIT Artificial Intelligence Laboratory.
- [15] P.L. Rosin and T. Ellis. Image difference threshold strategies and shadow detection. In *Proc. the sixth British Machine Vision Conference*, 1994.
- [16] C. Ridder, O. Munkelt, and H. Kirchner. Adaptive background estimation and foreground detection using kalman-filtering. In *ICRAM*, 1995.
- [17] A. Elgammal, D. Harwood, and L. S. Davis. Non-parametric model for background subtraction. In *Proc. IEEE ICCV'99 FRAME-RATE Workshop*, 1999.