

## NOTE

# Illumination Independent Change Detection for Real World Image Sequences\*

KURT SKIFSTAD AND RAMESH JAIN

*Artificial Intelligence Laboratory, Electrical Engineering and Computer Science Department,  
The University of Michigan, Ann Arbor, Michigan 48109-2122*

Received December 29, 1986; revised October 28, 1988

Change detection plays a very important role in many vision applications. Most change detection algorithms assume that the illumination on a scene will remain constant. Unfortunately, this assumption is not necessarily valid outside a well-controlled laboratory setting. The accuracy of existing algorithms diminishes significantly when confronted with image sequences in which the illumination is allowed to vary. In this note, we present two techniques for change detection that have been developed to deal with the more general scenario where illumination is not assumed to be constant. A detailed description of both new methods, the derivative model method and the shading model method, is provided. Results are presented for applying each of the techniques discussed to various image pairs. © 1989 Academic Press, Inc.

Detecting changes is fundamental to one's perception of the world. After all, the world we live in is dynamic, and the inputs to our senses are constantly changing. Many models have been proposed for detection of motion in the human visual system [13-17]. In this note our concern is only change detection; motion detection may use change detection, but it is considered beyond the scope of this discussion. It is our aim is to develop robust change detection techniques for machine vision systems.

It is not surprising that the process of change detection is fundamental to many machine vision applications. Systems that track moving objects [18], analyze cloud motion, monitor the growth of crops, or analyze traffic flow [2-4], are just a few examples of machine vision systems that use change detection algorithms. These algorithms provide the low level information that can be used by higher level algorithms to determine the information desired (the trajectory of an object, the growth of a tree, etc.). Therefore, for these systems to operate successfully, it is extremely important that change detection algorithms to accurate and robust.

Change detection may take place either at the pixel level or at a higher level—by comparing features. We will address change detection at the pixel level. Our motivation in developing robust techniques for change detection at the pixel level is the possibility of very fast change detection for robotic applications.

In this note, we present two new methods for change detection: the derivative model method and the shading model method. The derivative model method uses partial derivatives with respect to the pixel coordinates of a second order gray level surface model to compare regions and determine if a change has taken place. This

\*This work was sponsored in part by National Science Foundation Grant DCR8517251 and Department of Energy Grant DE-FG02-86NE37969.

technique is a logical extension of an existing change detection algorithm [6] and performs significantly better than the original when confronted with changes in illumination. The shading model method uses a shading model used for computer graphics applications to help determine if there has been a structural change in the scene. It is shown that the shading model method is superior to any known technique when the illumination is allowed to vary. A detailed discussion is provided for both of the new techniques.

## 1. CHANGE DETECTION ALGORITHMS

Change detection algorithms take two digitized images as input and return the locations in the field of view where differences between the images are identified. These differences may be caused by the motion of an object in the field of view, the addition or removal of an object from the scene, changes in illumination, or noise from the digitization process. The goal of such an algorithm is to locate only the changes that are due to structural changes in the scene, i.e., an object moving or the introduction or removal of an object in the scene.

Many different types of techniques for change detection exist in the literature. Changes may be detected either at the iconic or pixel intensity level or after identifying features such as lines, corners, or some other interesting entities [16, 17]. At the pixel level many diverse techniques exist for detecting changes. These range from simple differencing methods [5] to complex modeling processes [6]. Some look at individual pixels, others look at blocks of adjacent pixels [5, 6]. But, besides the specific technique used for measuring changes, this change detection process is generally the same. The input images are first divided up into defined regions. A metric  $f_i$  is then computed for a region in the field of view for both input images. By comparing the measurements computed for the two images, one determines a quantity corresponding to the difference (or similarity) between the two locations. A simple equation for this metric may appear as follows:

$$D(x, y) = f_1(x, y) - f_2(x, y) \quad (1)$$

where  $D(x, y)$  is the difference metric, and  $f_i$  is the metric computed for the particular region in question in image  $i$ . One then compares this "difference metric" to a threshold to determine if a change is to be indicated at that location.

### 1.1. Simple Differencing

Of all the change detection algorithms, simple differencing is probably the least complex and most intuitively appealing. The philosophy behind this technique is simple: if a physical change occurs at a particular location in an image, then the recorded gray level at that location will also change. The change detection involves comparing the difference in gray levels recorded at individual pixel locations in the two images to a preset threshold. We can define a binary difference picture as follows:

$$\begin{aligned} B(x, y) &= 1 && \text{if } D(x, y) > T \\ &= 0 && \text{otherwise,} \end{aligned} \quad (2)$$

where  $B(x, y)$  is the binary difference image value at pixel location  $(x, y)$ ,  $D(x, y)$

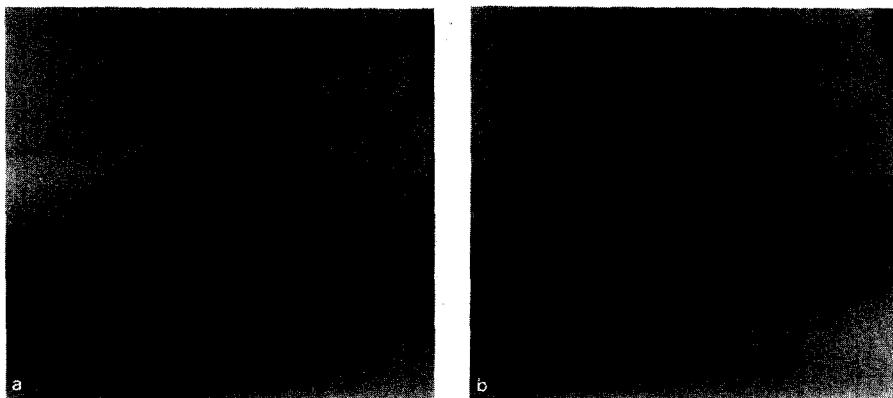


FIG. 1. Test sequence 1. Two frames of a sequence taken in the lab of a toy car rolling by some wooden blocks as a simulated cloud rolls overhead.

is the difference in intensity between frames at that same location, and  $T$  is some arbitrary threshold.

As would be expected, the simple differencing technique is not at all appropriate for applications where the illumination may vary. A change in illumination directly affects the gray levels recorded in the scene, which would be incorrectly interpreted as structural changes by this algorithm.

This technique is very fast and may be useful for applications where there is little variation in the gray levels in the scene, or where there is tolerance for error. However, considering one pixel at a time makes this approach very susceptible to noise and, thus, inappropriate for many real world applications, where high precision is required.

### 1.2. Geo-Pixel Technique

To avoid the problems associated with sensitivity to noise, several techniques have been developed that look at a block of adjacent pixels. One such method, which, for our purposes, we call the geo-pixel technique [7] looks at  $n \times n$  pixel regions in the images and computes a difference metric based on second order statistics. This difference metric is based on a likelihood ratio used by Yakimovsky [8] to determine if two neighboring test areas can be thought of as being compatible. The likelihood ratio  $L$  can be computed using the mean and variance of the two regions. It is expressed as:

$$L = \frac{\left[ (\sigma_1^2 + \sigma_2^2)/2 + ((\mu_1 - \mu_2)/2)^2 \right]^2}{\sigma_1^2 \sigma_2^2}, \quad (3)$$

where  $\mu_i$  and  $\sigma_i^2$  are the mean and variance of region  $i$ . If the likelihood ratio exceeds a given threshold, the two regions are assumed to come from different gray-level distributions and thus should not be merged. Nagel generalized this concept to compare regions from the same area of two consecutive frames in an image sequence (instead of neighboring regions in the same frame) [7].

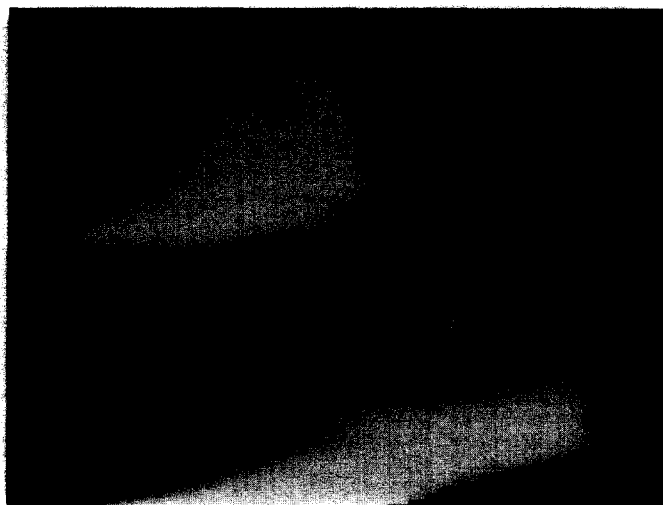


FIG. 2. The results of applying the geo-pixel algorithm to the toy car sequence. ( $T = 1$ .)

As would be expected, the geo-pixel technique works well on sequences with no change in illumination. It is much less susceptible to noise than the differencing technique and is fairly accurate. When confronted with sequences in which the illumination is allowed to change, however, it locates the places that have changed, but portions of the background are often also incorrectly indicated.

The image pair shown in Fig. 1 was staged in the laboratory and shows a toy car as it rolls by some wooden blocks. A simulated "cloud" is rolling between the light source and the scene, thus changing the illumination in a portion of the field of view.

Figure 2 shows the results of applying the geo-pixel technique to this sequence. As can be seen, this technique fails to distinguish between changes due to the car moving and changes due only to variation in illumination. The moving car is located quite accurately, however, a significant portion of the background which did not change is also indicated.

### 1.3. *The Quadratic Picture Function Model Approach*

Hsu, Nagel, and Rekers [6] tried modeling the gray level distributions in a given region to obtain more accurate results. These techniques compared models of gray level surface to create the difference metric on which change detection is determined.

Zero order, first order, and second order surface models have all been used for this application. Hsu, Nagel, and Rekers [6] show that a second order bivariate polynomial in the pixel coordinates models the gray value variation in a region of the image with such an accuracy that any remaining variation in gray level is assumed to be attributable to noise related to the sensing and digitizing devices.

It is shown in [6] that, under constant illumination, this technique is superior to any previously mentioned methods. However, as can be seen in Fig. 3, the QPF technique is quite sensitive to changes in illumination and fails to properly identify

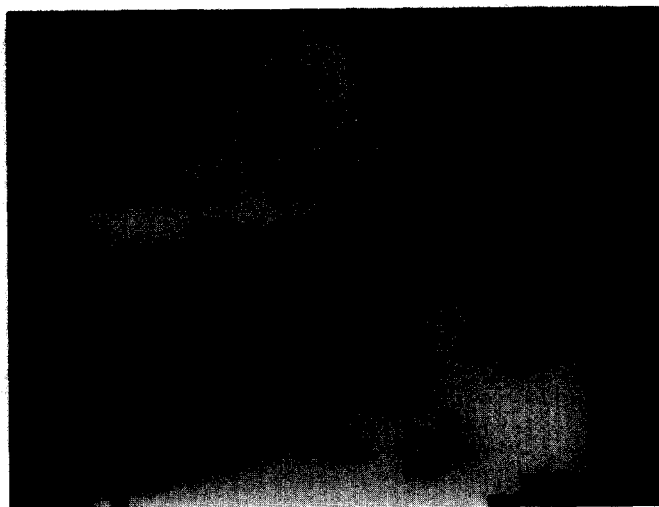


FIG. 3. The results of applying the QPF algorithm to the toy car sequence. ( $T = 65$ .)

the object that moves. This is expected however, because, at best, a change in illumination introduces a constant displacement term from one model to the next.

#### 1.4. Gray Scale Normalization Techniques

Recent change detection algorithms designed for gray scale images have not addressed the problem of varying illumination. However, there do exist change detection techniques addressing this problem which were originally developed for side looking radar (SLR) image analysis [9, 10]. These techniques look at corresponding regions in two images and normalize one with respect to the other in such a manner that the gray level distribution in both regions are of the same mean and variance. That is, given region  $i$  in image  $A$  and region  $i$  in image  $B$ , the pixel values in the corresponding region of the new normalized image  $N$  are given by:

$$\text{for } (x, y) \in \text{region } i \quad I_N(x, y) = \frac{\sigma_A(i)}{\sigma_B(i)} \{I_B(x, y) - \mu_B(i)\} + \mu_A(i), \quad (4)$$

where  $I_R(x, y)$  is the intensity recorded in image  $R$  at location  $(x, y)$  and  $\mu_R(i)$  and  $\sigma_R(i)$  are the mean and standard deviation of the intensity values recorded in region  $i$  in image  $R$ . After the normalization is performed, simple differencing is used to detect changes.

As with the geo-pixel technique, second-order statistics do not prove to be robust enough. Consider the second test pair, shown in Fig. 4. This shows two images digitized in a laboratory setting containing two blocks sitting on a wooden table with a new object (the screwdriver) added to the scene in the second frame. As can be seen, there is a significant change in the illumination on the scene. Figure 5 shows the results of applying this technique to this image pair. As can be seen, this technique is unable to locate the screwdriver and incorrectly indicates a large portion of the scene as changing.

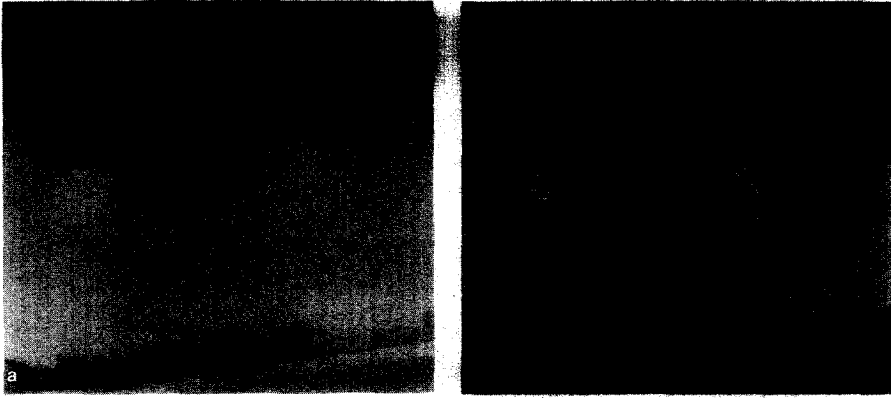


FIG. 4. Test sequence 2. Two frames of a sequence taken in the lab showing two blocks sitting on a wooden table with a new object (the screwdriver) introduced into the scene in the second frame. (This sequence contains a profound change in illumination.)

## 2. NEW TECHNIQUES

Since, in the most general case it is not necessarily reasonable to assume constant illumination, it would be desirable for a change detection algorithm be insensitive to changes in illumination. Two new techniques are presented in this paper which have been developed with the following goal: we want a technique that is both accurate and insensitive to changes in illumination. Our first attempt at such a technique, the derivative model method (D-method), is a logical extension of the QPF technique described above and performs significantly better than the original when confronted with changes in illumination. The D-method, however, does not prove to be adequate for image pairs with a profound change in illumination.

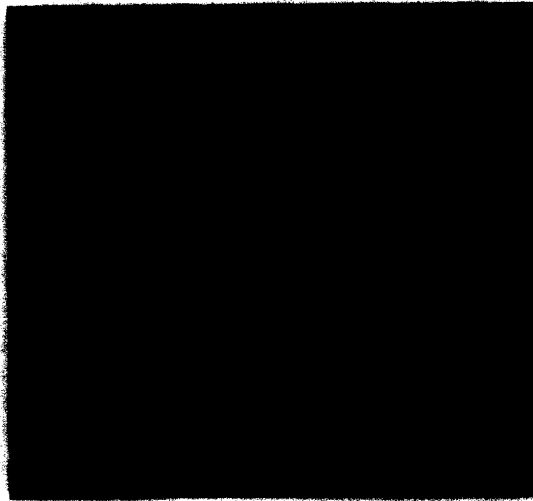


FIG. 5. The results of applying the gray scale normalization technique to the screwdriver sequence. ( $T = 5$ .)

The second technique developed (the shading model method) uses a shading model used for computer graphics applications to determine if there has been a structural change in the scene. It is shown that the shading model method is relatively insensitive to variations in the illumination levels in a scene and is superior to any known technique when the illumination is allowed to vary.

### 2.1. The Derivative Model Method

The success of the QPF method on image sequences with constant illumination indicates that looking at the gray level surface model is a good place to begin when looking for a technique that is accurate *and* insensitive to changes in illumination. The problem, however, is to find a surface model that does not change (significantly) as illumination changes. The D-method uses the  $x$  and  $y$  partial derivatives of the second-order surface model from [6].

The difference metric is defined for a region  $A$  as

$$D = \sum_{x, y \in A} f_1(x, y) - f_2(x, y) \quad (5)$$

with  $f$  being defined as

$$f_i(x, y) = \frac{\partial g_i(x, y)}{\partial x} + \frac{\partial g_i(x, y)}{\partial y}, \quad (6)$$

where  $g_i(x, y)$  is the model for the gray level surface in the given region in the  $i$ th frame [6],

$$g_i(x, y) = a_{00} + a_{10}x + a_{01}y + a_{11}xy + a_{20}x^2 + a_{02}y^2. \quad (7)$$

Figure 6 shows the results of applying the D-method to the toy car sequence. One can see that this technique is able to quite effectively isolate the moving object with a relatively small portion of the background also indicated.

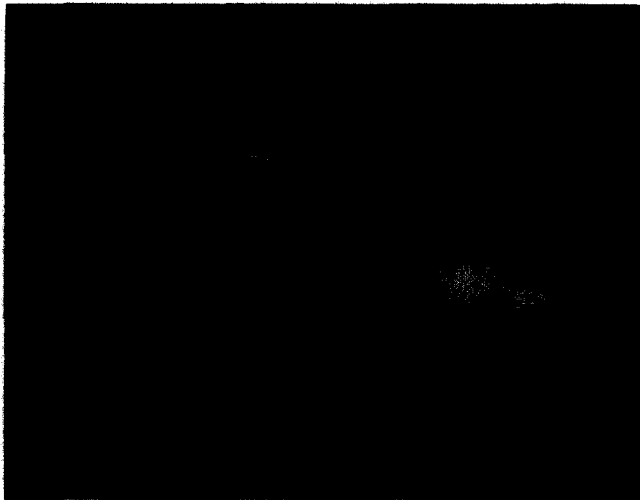


FIG. 6. The results of applying the derivative model method to the toy car sequence. ( $T = 87$ .)

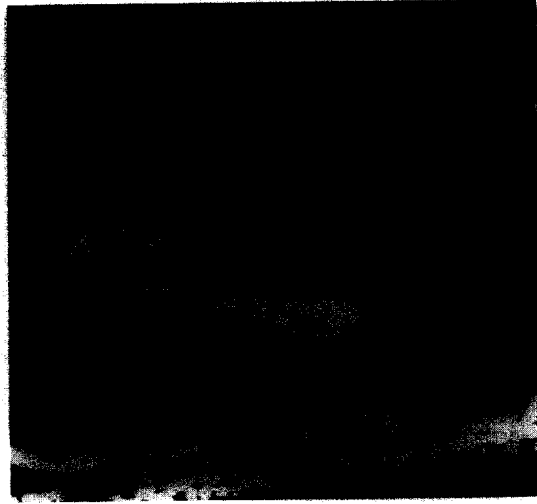


FIG. 7. The results of applying the derivative model method to the screwdriver sequence. ( $T = 40$ .)

The results of applying this technique to the second block sequence are shown in Fig. 7. As can be seen, this technique is able to isolate the screwdriver; however, a large portion of the background is also indicated.

### 2.2. The Shading Model Method

Shading models for computer graphics model the intensity at a given point ( $I_p$ ) as the product of the illumination ( $I_i$ ) and a shading coefficient ( $S_p$ ) which is calculated for each point,

$$I_p = I_i S_p. \quad (8)$$

Phong's shading model [11], is typical of the shading models used and is defined as (see also Fig. 8)

$$S_p = C_p [\cos(i)(1 - d) + d] + W(i) [\cos(s)]^n, \quad (9)$$

where

$C_p$  is the reflection coefficient of the object at point  $P$  for a certain wavelength.

$i$  is the incident angle.

$d$  is the environmental diffuse reflection coefficient.

$W(i)$  is a function which gives the ratio of the specular reflected light and the incident light as a function of the incident angle.

$s$  is the angle between the direction of the reflected light and the line of sight.

$n$  is a power which models the specular reflected light for each material.



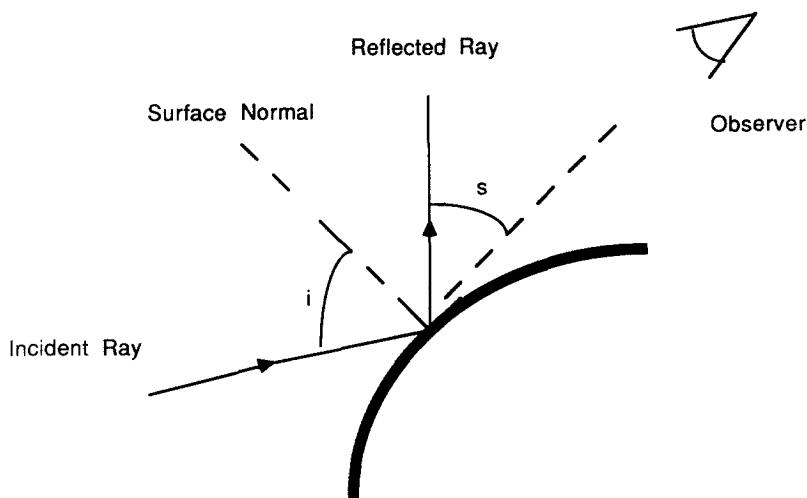


FIG. 8. Light striking the surface of an object.

As can be seen, the shading coefficient is defined uniquely by the physical surface structure of the object and the reflectance of the surface material. Thus, if the physical surface structure at a given location in an image changed (i.e., an object moved), the shading coefficient for that location would also change.

Since the shading coefficient is dependent on the physical surface of the object, it seems likely that a change detection algorithm based on this concept would be quite accurate, and, since the shading coefficient is independent of illumination, such an algorithm would also have the desirable property of being insensitive to changes in illumination. Unfortunately, it is not possible to calculate the shading coefficient for a given point without *a priori* knowledge of the surface structure. However, the exact value of the shading coefficient need not be known; one need only detect a change in the shading coefficient to be able to indicate a change between the frames of the sequence. The shading model change detection algorithm uses the ratio of intensities recorded in a region of the two frames to detect this change.

The ratio of intensities at a given location over two frames is simply the ratio of the intensities of the illuminations on the two frames times the ratio of the two shading coefficients,

$$\frac{I_{p1}}{I_{p2}} = \frac{I_{i1}}{I_{i2}} \times \frac{S_{p1}}{S_{p2}}. \tag{10}$$

If there is no change in the physical structure,

$$S_{p1} = S_{p2}, \tag{11}$$

so

$$\frac{I_{p1}}{I_{p2}} = \frac{I_{i1}}{I_{i2}}. \tag{12}$$

And if the physical structure does change,

$$S_{p1} \neq S_{p2}; \quad (13)$$

therefore,

$$\frac{I_{p1}}{I_{p2}} \neq \frac{I_{i1}}{I_{i2}}. \quad (14)$$

Thus, for any region  $A_i$ , if there were no changes in the physical surface in that region, all the intensity ratios in that region would be equal:

$$\frac{I_{m1}}{I_{m2}} = \frac{I_{n1}}{I_{n2}} = \frac{I_{i1}}{I_{i2}} \quad \text{for all points } n, m \in A_i. \quad (15)$$

Since the ratios are all equal, the variance of this ratio is zero in a region where no changes take place;

$$E\{\sigma_s^2\} = E\left\{\frac{1}{N} \sum_{x \in A_i} \left(\frac{I_{x1}}{I_{x2}} - \mu_i\right)^2\right\} \quad (16)$$

$$= \frac{1}{N} \sum_{x \in A_i} \left(E\left\{\frac{I_{x1}}{I_{x2}}\right\} - E\{\mu_i\}\right)^2, \quad (17)$$

where  $E\{\}$  is the expectation operator and  $\sigma_s^2$  is the second central moment (variance). Since  $\mu_i$  is the average value of the ratio of intensities,

$$\begin{aligned} E\{\sigma_s^2\} &= \frac{1}{N} \sum_{x \in A_i} \left(\frac{I_{i1}}{I_{i2}} - \frac{1}{N} \sum_{y \in A_i} E\left\{\frac{I_{y1}}{I_{y2}}\right\}\right)^2 \\ &= \frac{1}{N} \sum_{x \in A_i} \left(\frac{I_{i1}}{I_{i2}} - \frac{I_{i1}}{I_{i2}}\right)^2 = 0. \end{aligned} \quad (18)$$

And if there were changes in the physical surface in that region (and all the shading coefficients do not change in exactly the same manner),

$$E\{\sigma_s^2\} = E\left\{\frac{1}{N} \sum_{x \in A_i} \left(\frac{I_{x1}}{I_{x2}} - \mu_i\right)^2\right\} \geq 0. \quad (19)$$

To determine if a change has taken place in a given region, one simply calculates the variance of the intensity ratios in that region. If it is close to zero, no change is perceived to have taken place; otherwise one may assume a change has occurred. As for the size of the region, it should contain a large enough sample so that the statistics will be indicative of the nature of the region. For our experiments, we used  $5 \times 5$  regions.

When using this technique, one must keep in mind that it detects changes in physical surface *structure* and is (roughly) independent of illumination. This implies

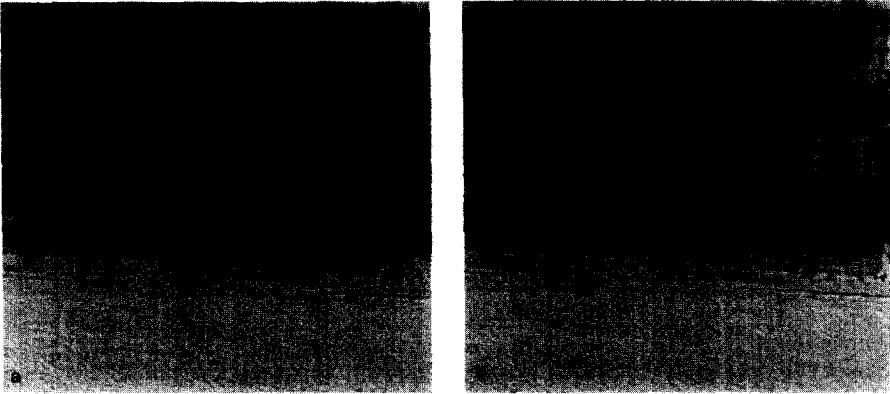


FIG. 9. Test Sequence 3. Two frames of a sequence showing action on a sidewalk.

that this technique would indicate no change if a light gray ball was replaced by a dark gray ball of the exact same size and shape under conditions with constant illumination. Whereas a simpler intensity-based approach would have no problem locating the change.

An image pair obtained with no change in illumination between frames is shown in Fig. 9. Results from applying the shading model method to this sequence are shown in Fig. 10. As can be seen, this technique identifies the object which moved quite accurately. Note that this technique is also rather insensitive to noise.

Figure 11 shows the results of applying the shading model method to the toy car sequence. As can be seen the moving car is isolated quite accurately from the background. Very little of the background is indicated as having changed.



FIG. 10. The results of applying the shading model method to the sidewalk sequence. ( $T = 0.10$ .)

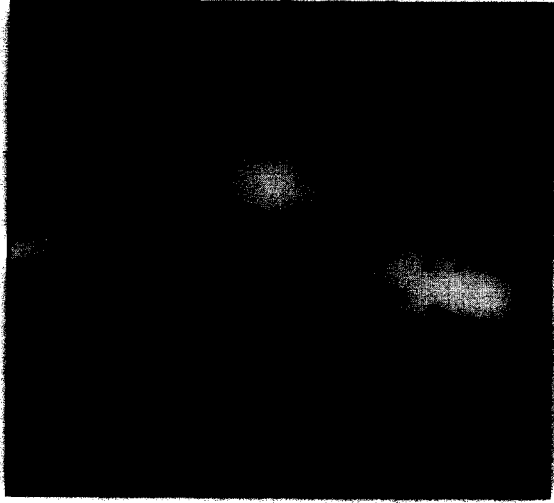


FIG. 11. The results of applying the shading model method to the toy car sequence. ( $T = 0.008$ .)

The results of applying this method to the blocks sequence are shown in Fig. 12. This technique isolates the screwdriver quite well, with little of the background being shown as changing.

The performance of the shading model method on sequences with no change in illumination is quite good. Our results have shown it to be comparable to any of the other techniques presented. When confronted with a change in illumination, this technique performs better than the D-method and the gray scale equalization

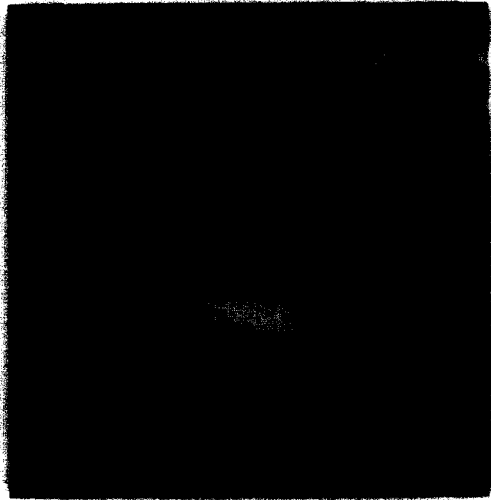


FIG. 12. The results of applying the shading model method to the screwdriver sequence. ( $T = 0.25$ .)

technique and, certainly, significantly better than any of the other techniques. On the sequence where the change in illumination is profound, it performs remarkably well and markedly better than any of the other techniques.

### 3. CONCLUSION

When designing a dynamic vision system, there are a significant number of change detection algorithms to choose from. Certainly, in the most general of applications where illumination cannot be controlled, the shading model method should be considered. However, it may be the case that a simplistic quick approach, such as simple differencing may suffice in many applications. It may also be the case that information from several different change detection algorithms may be integrated to accurately detect changes.

### REFERENCES

1. I. Rock, *Perception*, Sci. Amer. Books, New York, 1984.
2. H. H. Nagel, Recent advances in motion interpretation based on image sequences, in *Proceedings, ICASSP 1982*, pp. 1179–1186.
3. M. Onoe and K. Ohba, Digital image analysis of traffic flow, in *Proceedings Third International Joint Conference on Pattern Recognition, 1976*, pp. 803–808.
4. R. Jain, *Dynamic Scene Analysis*, University of Michigan Computing Research Laboratory CRL-TR-6-84.
5. R. Jain, Difference and accumulative difference pictures in dynamic scene analysis, *Image Vision Comput.* **2**, 1984, 99–107.
6. Y. Z. Hsu, H. H. Nagel, and G. Rekers, New likelihood test methods for change detection in image sequences, *Comput. Vision Graphics Image Process.* **26**, 1984, 73–106.
7. R. Jain, D. Militzer, and H. H. Nagel, Separating non-stationary from stationary scene components in a sequence of real world TV-images, in *Proceedings, IJCAI 1977*, pp. 612–618.
8. Y. Yakimovsky, Boundary and object detection in real world images, *J. Assoc. Comput. Mach.* **23**, 1976, 599–618.
9. R. Lillestrand, Techniques for change detection, *IEEE Trans. Comput.* **21**, No. 7, 1972, 654–659.
10. M. S. Ulstad, An algorithm for estimating small scale differences between two digital images, *Pattern Recognit.* **5**, 1973, 323–333.
11. B. T. Phong, Illumination for computer generated pictures, *Commun. ACM* **18**, 311–317.
12. J. D. Foley and A. Van Dam, *Fundamentals of Interactive Computer Graphics*, Addison-Wesley, Reading, MA, 1982.
13. D. Marr and S. Ullman, Directional selectivity and its use in early visual processing, *Proc. Roy. Soc. London B* **211**, 1984, 151–180.
14. E. C. Hildreth, *The Measurement of Visual Motion*, Ph.D. thesis, Dept. of Electrical Engineering and Computer Science, MIT, MIT Press, Cambridge, MA, 1983.
15. C. Longuet-Higgins and K. Prazdny, The interpretation of a moving retinal image, *Proc. Roy. Soc. London B* **208**, 1981, 385–397.
16. S. Anstis, The perception of apparent movement, *Phil. Trans. Roy. Soc. B* **290**, 1980, 153–168.
17. A. B. Watson and A. J. Ahumada, A model of human visual motion sensing, *J. Opt. Soc. Amer.* **2**, No. 22, 1985.
18. J. W. Roach and J. K. Aggarwal, Determining the movement of objects from a sequence of images, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-2**, No. 6, 1980, 554–562.
19. M. A. Shah and R. Jain, Detecting time-varying corners, *Comput. Vision Graphics Image Process.* **28**, No. 3, 1984, 345–355.