

# Seeing Behind Occlusions<sup>\*</sup>

Dana H. Ballard and Rajesh P.N. Rao

Computer Science Department  
University of Rochester  
Rochester, NY 14627

**Abstract.** The location of objects in images is difficult owing to the view variance of geometric features but can be determined by developing view-insensitive descriptions of the intensities local to image points. View-insensitive descriptions are achieved in this work by describing points in terms of the responses of steerable filters at multiple scales. Owing to the use of multiple scales, the vector for each point is, for all practical purposes, unique, and thus can be easily matched to other instances of the point in other images. We show that this method can be extended to handle the case where the area near a point of interest is partially occluded. The method uses a description of the occluder in the form of a template that can be obtained easily via active vision systems using a method such as disparity filtering.

## 1 Introduction

Object recognition is a central problem of computer vision. Owing to its importance there have been a very large number of different approaches taken to solve it, which can be grouped into three different classes. The approach of one class is to find a projective invariant. This is a feature that remains invariant under imaging. For example, one such invariant is the cross-ratio, defined on four model points. Projective invariance reaches for view-insensitivity. That is, the feature variants would be a boon but for the problem of segmenting the object from the background. In a natural situation, it is extremely difficult to identify appropriate constituent points. A second main tack is to confront the view variation directly by modeling the view parameters explicitly. This results in a search process, whereby possible model-image feature correspondences are constrained to have a consistent set of viewing parameters. Examples of such approaches are Hough transforms and geometric hashing. A third class of approaches, which we are pursuing, compromises on view invariance. Instead, image features are required to be only relatively insensitive to variations in the view. Such a feature is color. Image color as a measure of surface albedo is insensitive to variations in viewing direction. Swain used color for object recognition problems by exploiting properties of the color histogram [Swa90; SB91].

---

<sup>\*</sup> This research is supported by the Human Science Frontiers Program research grant and by the National Science Foundation under NSF research grant no. CDA-882724.

Previously we have shown that geometric features can be found that behave like color [BW93]. These are the steerable filters [FA91; JM92; MP89]. Such filters are a way of describing the intensities near a given point. The filters depend on the choice of the coordinate system; however, there is a normalization procedure that makes them invariant to rotations about the view vector. This means that an index can be constructed that almost uniquely describes the local intensity variations about a point. This description is in the form of a 45 element vector or *zip-code* of filter responses at different scales. In the two-dimensional case, for all practical purposes, this vector is unique and its location can be recovered by the process of backprojection, or comparing a model response vector to the response vectors of all image locations.

For rotations about axes other than the viewing axis, the success of the descriptors depends on their view insensitivity. Our experiments using backprojection showed that the filters are insensitive to three-dimensional rotations of up to 45°.

In this paper, we show that this method can be extended to handle the case where the area near the point is partially occluded. The method uses a description of the occluder in the form of a template. This can be obtained via active vision systems.

## 2 Steerable Filters

Steerable filters are a set of oriented basis filters with the important property that the response of a filter at an arbitrary orientation can be synthesized from linear combinations of the basis filters. As shown by Freeman and Adelson [FA91], starting from a symmetric Gaussian function in Cartesian coordinates:

$$G(x, y) = e^{-(x^2+y^2)}$$

it is possible to define basis filters  $G_n^{\theta_n}$  as:

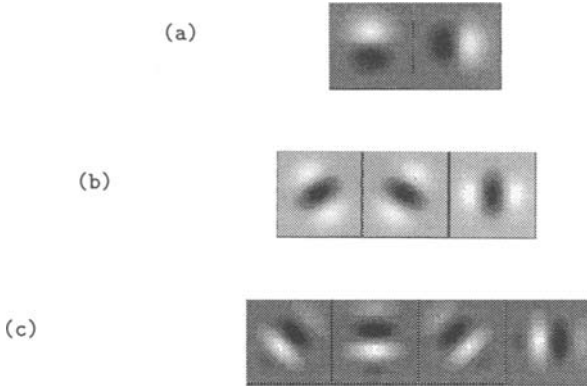
$$G_n^{\theta_n} = \frac{\partial^n}{\partial x^n} G(x, y), n = 1, 2, 3, \theta_n = 0, \dots, k\pi/(n + 1), k = 1, \dots, n.$$

Figure 1 shows these functions for a particular value of standard deviation  $\sigma$ .

### 2.1 The Interpolation Functions

As Freeman and Adelson [FA91] have also shown, different order filters are *steered* with different interpolation functions. The number of the interpolation functions that are needed for the steering is one more than the filter order. So, for example, the first-order filters can be steered with two interpolation functions given basis measurements at 0° and 90°, the second-order filters can be steered with three functions given basis measurements at 0°, 60°, and 120°, and the third-order filters can be steered with four functions oriented at 0°, 45°, 90°, and 135°. That is,

$$G_j(\theta) = \sum_{i=1}^{n_k} G_j^{\phi(i)} k_{ij}(\theta),$$



**Fig. 1.** The nine different filters that comprise the steerable filters up to third-order for  $\sigma = 6.5$ . (a)  $G_1$ ; (b)  $G_2$ ; (c)  $G_3$ .

where the first-order interpolants are given by  $j = 1$ ,

$$k_{i1}(\theta) = \frac{1}{2} [\cos(\theta - \theta_i)], i = 1, 2.$$

For  $j=2$ , we have

$$k_{i2}(\theta) = \frac{1}{3} [1 + 2\cos(2(\theta - \theta_i))], i = 1, 2, 3,$$

and for  $j=3$ ,

$$k_{i3}(\theta) = \frac{1}{4} [2\cos(\theta - \theta_i) + 2\cos(3(\theta - \theta_i))], i = 1, 2, 3, 4.$$

### 3 The Multiple-Scale Index or Zip-Code

Our goal is to create a vector that uniquely describes each point. Such a description would allow the straightforward algorithms for matching models seen as collections of points.

Combining the responses from all the filters from different orders provides nine independent measures at each image point. This can be augmented further by using the filters at different image scales. To handle variations in scale, it is best to make the scales sufficiently close together so as to be able to interpolate between scales. In practice, this means choosing image domains that are multiples of two in area, or two in linear dimension. Since there are nine measurements per scale, there are  $9 \times$  (the number of scales) total measurements. For the experiments, five different scales are used, for a total of forty-five measurements per point.

The different responses at different scales are sensitive to the width of the templates, so the responses, to be comparable across scales, have to be normalized. The easiest way to do this is to divide by the filter energy defined as:

$$e_i = \iint G_i^0(x, y)^2 dx dy, i = 1, 2, 3.$$

Now define the normalized response of a set of filters to the area surrounding a specific point in the image as the vector

$$\mathbf{r} = (r_{i,j,s}), i = 1, 2, 3; j = 1, \dots, f(i); s = s_{min}, \dots, s_{max},$$

where the index  $i$  denotes the order of the filter,  $j$  denotes the number of filters per order ( $f(i) = i + 1$ ), and  $s$  denotes the number of different scales.

### 3.1 Normalization

In two dimensions, the difference between an image point and a model point is limited to a rotation and scale. Assume that the scale is fixed. To normalize for the rotation, one strategy is to select the orientation of the first-order filters as a reference. This is a good strategy for two reasons: (1) the orientation can be computed directly from the filter responses, and (2) the filter responses are usually the most stable.

Thus the orientation is computed as

$$\alpha = \tan^{-1}(r_{1,1,s_{max}}, r_{1,2,s_{max}})$$

and then the filter responses are rotated using the steering formulae, i.e.,

$$r'_{i,j,s} = \text{Rot}(\alpha, r_{i,j,s}).$$

Note that this normalization makes the matching process more powerful than that produced with rotation invariant templates. The latter sacrifice variability in the angular direction. Instead the filters capture the variations in angle, and preserve it in their components. Another feature of the normalization process is that it can be done without additional convolutions; the interpolation properties of the existing filters allow it to be carried out with a single basis set of convolutions.

No similar normalization strategy exists for scale. This is easy to understand, since the receptive fields of the smaller scales are correspondingly smaller, and thus image data that is covered by the larger field does not affect the smaller field. Nonetheless there is a weaker method for adjusting the scale parameter. This is to establish a correspondence between the two vectors by comparing the responses, which will usually vary smoothly between scales.

In the three-dimensional case, the strategy for matching is essentially to rely on the two-dimensional match properties. Rotation about the view axis can be corrected for, leaving scale and rotations about axes in the image plane as the main difficulties. Scale can be handled by matching as in the two-dimensional case, but there is a more practical alternative in 3D. Since the approximate

distances are usually known by the perceiver, and the dimensions of the viewed object are usually small compared to the viewing distance, the scale can be pre-adjusted prior to the matching process. Rotations about an image plane axis are ameliorated in two ways. In the first place, the filter responses are dominated by a cosine envelope, so that there is a useful range of rotations for which the responses will be effectively invariant. Second, the algorithms for identification and location work as long as there is a useful subset of filter responses. All the responses do not have to be correct.

### 3.2 The Backprojection Algorithm

In order to describe the algorithm for location, we first need to describe the match between two response vectors, one from an model point and one from an image point. Denote the vector from an image point as  $\mathbf{r}^i$  and that from a model point as  $\mathbf{r}^m$ . Then the distance between them is simply the Euclidean distance  $d_{im} = \|\mathbf{r}^i - \mathbf{r}^m\|$ .

The location algorithm crucially depends on the fact that only a single model is matched to an image at any instant. Let us denote this model as

$$M = \{\mathbf{r}^m, m = 1, \dots, m_{max}\}.$$

For each model point  $m$ , create a backprojected distance image  $I_m$  defined by

$$I_m(x, y) = \min[I_{max} - \beta d_{im}, 0]$$

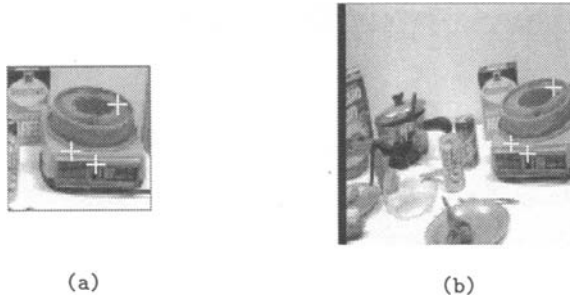
where  $\beta$  is a suitably chosen constant.

### 3.3 Results of Using Basic Backprojection

The experiments use pre-chosen model points to represent each object. These points were chosen by hand and an image containing a slide projector was used. The slide projector was imaged at two views of comparable scale that were  $22.5^\circ$  apart.

**View Insensitivity.** A measure of the algorithm's capability in the presence of three-dimensional distortion can be appreciated from the response of the algorithm to image skew. Skew can be reported in terms of the three-dimensional rotation that produces it, as is done here.

Points were selected from an image of a slide projector used as a model. These were then backprojected onto a second test image that was taken with a view vector rotated  $22.5^\circ$  from the model view vector. In the example shown in Figure 2, three points were selected on the slide projector. Two points were correctly located by the algorithm while one was located slightly off its actual position. Figure 3 shows the responses for a different point on the projector; it can be seen that the filter responses for the same point in the original image and the skewed image are nearly identical, thereby enabling our algorithm to usually succeed when the distortion is relatively small.



**Fig. 2.** The test of the localization algorithm in 3-D. Points from the model (a), shown by crosses, are backprojected onto the rotated image (b). Two points were correctly identified while one was slightly off the mark.

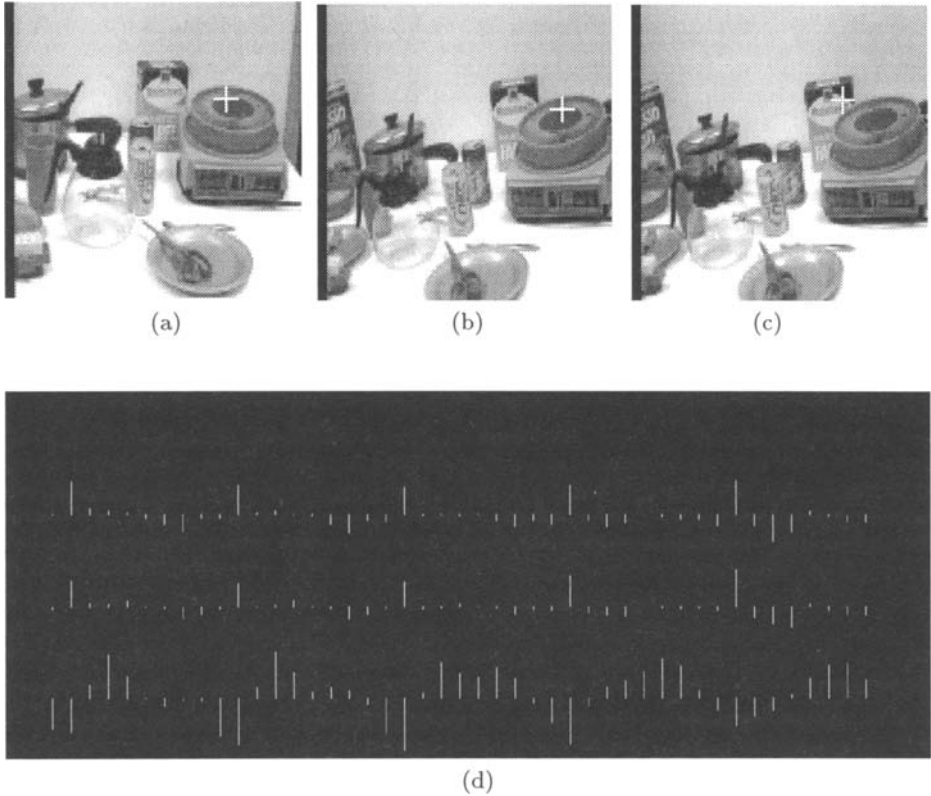
In a separate experiment we checked for scale sensitivity. As expected, the algorithm is very sensitive to scale variations. Based on similar experiments, the method tolerates about 10% variations in scale. However, we think this is not a huge factor, as in many cases scale will be known a priori. Also, there is the possibility of scale matching discussed earlier.

**The Importance of Multiple Scales.** To evaluate the uniqueness of the match, we tested the match values of corresponding points in the 2D rotated and unrotated images as a function of the number of scales used. Table 1 shows these results. With less than three scales, the matching point is not the best

<i>Number of Filters</i>	<i>Rank of Matching Point</i>	<i>Difference in Distance</i>
9	18.3	-8.1
18	4.3	-6.3
27	1.3	1.0
36	1	5.6
45	1	10.8

**Table 1.** Sensitivity of the match value to the length of the vector (= number of scales used  $\times$  nine). The figures shown are the average of the results for three pairs of corresponding points.

point selected. However, with three or more scales it is ranked the best. The third column compares the distance measures used in the match of the best and second-best point in the case where the matching point is ranked first. In the case where the matching point is not the best the distance is that of the best mi-

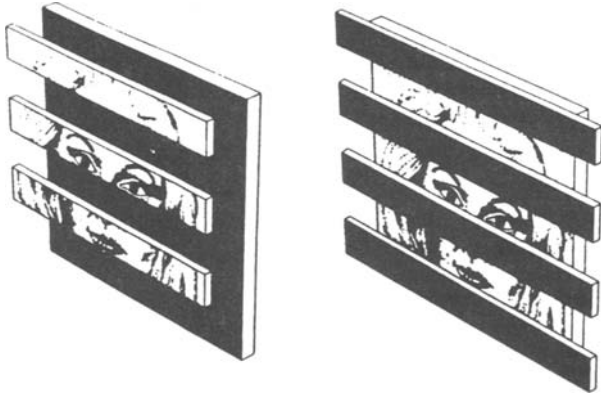


**Fig. 3.** A demonstration of view insensitivity. (a) A point on the original image; (b) the same point correctly located by the algorithm in a second image with a  $22.5^\circ$  3-D rotation; (c) an unrelated point in the rotated image for the purpose of comparison; (d) the 45 filter responses for the point in (a) (top), the point in (b) (middle) and the point in (c) (bottom).

nus that of the matching point. This column shows that even after the matching point is the best, its perspicuity continues to improve with additional scales.

#### 4 Dealing with Occlusion

The basic backprojection algorithm compares the filter responses with every point with those of a prototype. This algorithm will fail when the prototype is occluded if nothing is done as the occluder will distort the filter responses. Interestingly, humans have a similar problem. Figure 4 from Nakayama et al. [NS90] shows the experimental setup designed to test subjects' ability to identify faces in the presence of positive occlusion cues. In one instance the face is painted on a picket fence; in the other it is behind the picket fence. The results show



**Fig. 4.** Nakayama's demonstration that recognition performance depends on whether or not the occluded is positive.

that identification is improved in the latter case. This observation forms the inspiration for our solution.

Suppose that an active imaging system is used. As a consequence we can assume that the occluder can be detected by a method such as disparity filtering [CB92]. Disparity filtering is a way of creating a filter that only passes image energy in the horopter. Ideally one can create a template  $T(x, y)$  such that  $T(x, y) = 1$  for material in the horopter and  $T(x, y) = 0$  otherwise. We assume the existence of such a template for our subsequent calculations.

#### 4.1 Occlusion Algorithm

The filter responses are the responses for a set of basis functions. As a consequence the image intensities near every point can be reconstructed by appropriately combining the responses and filter functions. As the functions are not orthogonal, a pseudo-inverse must be used to do this [JM92]. This ability to reconstruct the local intensities allows the stored prototype to be made comparable to the occluded image responses. For every point, the reconstructed image intensities are appropriately masked using the occluding template. A similar process is done to the incoming image. Thus the masked reconstructed image and the masked input image are now in the same coordinate system and can be compared by differencing their filter responses. This is formalized in the following algorithm:

##### Occlusion Algorithm

1. Use the basis functions to reconstruct the local image intensities,  $I'(x, y)$  for an appropriate local domain  $D$  near a point  $(x_0, y_0)$ .
2. For every point  $(x, y)$  in the image do
  - Compute  $I''(x, y) = T(x, y)I'(x, y)$  for all  $(x, y)$  in  $D$ .
  - Compute new filter responses  $f''$  from  $I''$ .

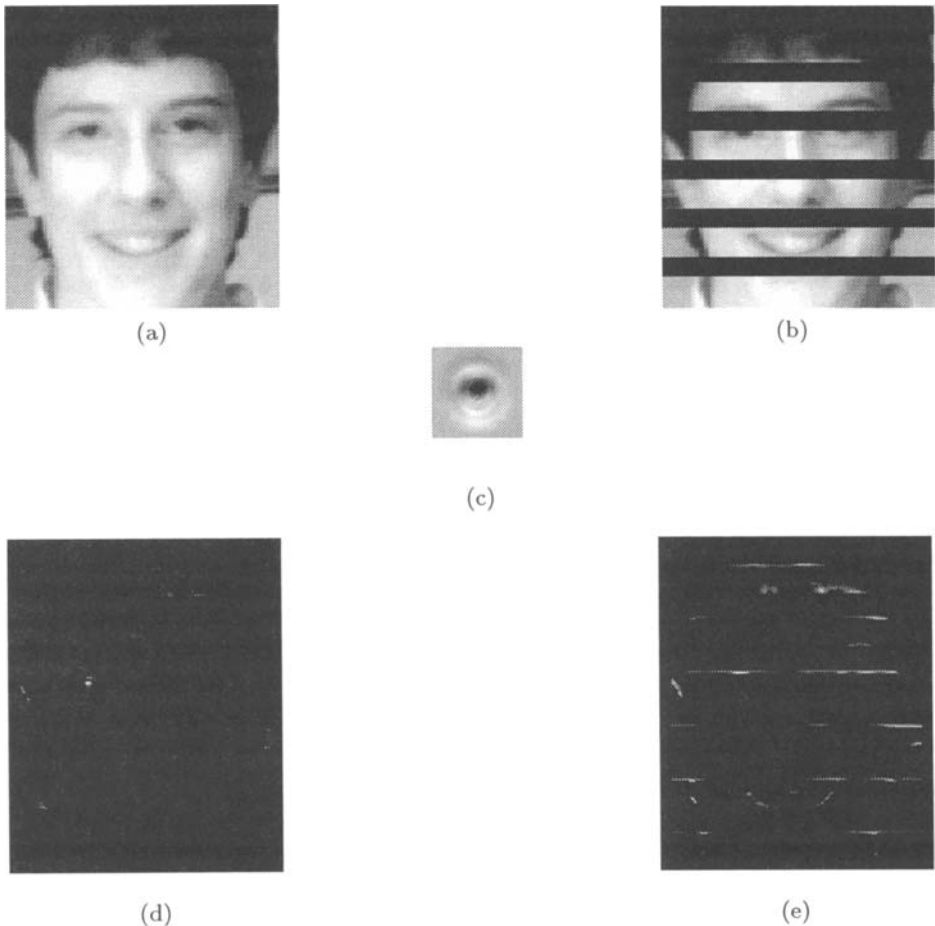


Compare those with the filter responses  $f$  computed from  $I'(x,y)T(x,y)$  to compute  $d(x',y')$ .

3. The sought after point is given by  $\operatorname{argmin} d(x',y')$ .

## 4.2 Results

To demonstrate the occlusion algorithm, we have created a face image similar to that of Figure 4. Figure 5 shows the original face and the results of picking a specific point in that image. The filter responses of the chosen point are computed



**Fig. 5.** A test of the occlusion algorithm. (a) The original image; (b) the occluded image; (c) the reconstructed patch of the left eye (unmasked); (d) the distance image showing the left eye correctly located by using responses from the masked eye patch; (e) the result of directly comparing the unoccluded responses from (c) with the responses from the occluded image.

for the unoccluded image and stored. Next the occlusion template for the image is computed. Finally we apply the rest of the steps of the occlusion algorithm. This shows that the best matching point is the correct one, as shown in Figure 5(d). Just to make the obvious point, if these operations are not done and instead the raw filter responses in the occluded image are compared to the previous point, then, as they are not comparable, the best match is not correct. This computation is shown in Figure 5(e).

**Sensitivity to degree of occlusion.** In order to test the sensitivity of the algorithm to the size and relative location of the occluder with respect to a point of interest, we ran the algorithm on a simple tabletop scene in the presence of increasing occlusion with a point near the end of the spatula's handle as the test point. As expected, recognition performance deteriorates due to the distortions in the responses as the size of the occluder increases in the area near the test point (Figure 6).

## 5 Discussion and Conclusions

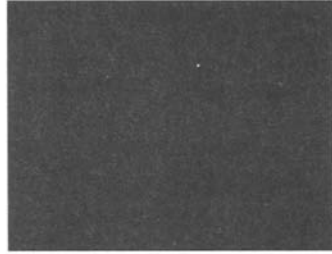
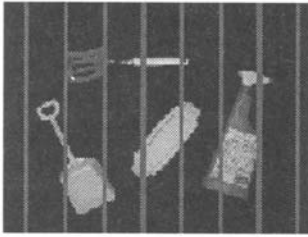
The demands of an object location method as a model of human performance and for robotic applications are that (a) it be fast and (b) it deal with varying views. Recently there has been renewed interest in correlation algorithms owing to the development of real-time signal processing hardware. However, correlation algorithms have been global, and have been sensitive to view parameters. Our algorithm succeeds owing to three principal features. First, the problem is divided into location and identification. Second, the steerability of our features allows for correction of rotations about the view vector. Third, we exploit the favorable matching properties of very long vectors.

The algorithm can deal with occlusions by using an active vision strategy [Bal91]. If one assumes that an occluding template can be obtained, then that template can be used to make the code for a stored prototype comparable to that of the image at every point. This allows the pseudo-occluded prototype to be compared to that of the occluded responses at every point as in the unoccluded case. Thus the use of the template can be seen as merely an extension of the steps in the backprojection algorithm.

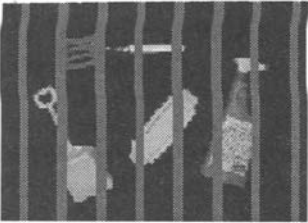
The idea of basis functions makes it suitable for instantaneously acquiring new points. This makes it different than principal components methods [MN93]. But most important, as the principal components do not have an inverse, the occlusion strategy described herein could not be used. The filters method has a further advantage over such methods in that the long feature vector formed by the steerable filter responses is robust to noise in some filter channels.

The occlusion strategy could be used in a more complicated graph-matching strategy such as that of [WvdM92], which also uses multi-resolution filters, but that would require additional computational machinery.

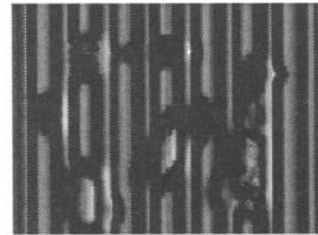
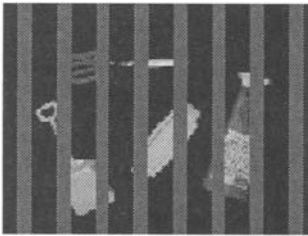
The multi-resolution structure of the filters has the additional advantage that the low-resolution components can be used in a variable resolution imaging system similar to the human retina.



(a)



(b)



(c)

**Fig. 6.** Sensitivity to degree of occlusion. (a) Partially occluded scene and distance image showing the test point correctly located. (b) and (c) Recognition performance gradually decreases as the degree of occlusion at the test point is increased.

## 6 Acknowledgements

We would like to thank the ECCV referees for their useful suggestions and criticisms. The authors would also like to thank Lambert Wixson for valuable assistance rendered during the early stages of this work.

## References

- [Bal91] D.H. Ballard. Animate vision. *Artificial Intelligence*, 48:57–86, 1991.
- [BW93] Dana H. Ballard and Lambert E. Wixson. Object recognition using steerable filters at multiple scales. In *Proceedings of the IEEE Workshop on Qualitative Vision*, 1993.
- [CB92] D.J. Coombs and C.M. Brown. Real-time smooth pursuit tracking for a moving binocular head. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 23–38, June 1992.
- [DS90] P.E. Danielsson and O. Seger. Rotation invariance in gradient and higher derivative detectors. *Computer Vision, Graphics, and Image Processing*, 49(2), February 1990.
- [FA91] William T. Freeman and Edward H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, September 1991.
- [JM92] David G. Jones and Jitendra Malik. A computational framework for determining stereo correspondence from a set of linear spatial filters. In *Proceedings of the Second European Conference on Computer Vision*, 1992.
- [MN93] H. Murase and S.K. Nayar. Learning and recognition of 3-d objects from brightness images. *Working Notes, AAAI Fall Symp. Series (Machine Learning in Computer Vision: What, Why, and How?)*, pages 25–29, October 1993.
- [MP89] Jitendra Malik and Pietro Perona. A computational model of texture segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 326–332, June 1989.
- [NS90] K. Nakayama and S. Shimojo. Towards a neural understanding of visual surface representation. *Cold Spring Harbor Symp. on Quantitative Biology, Vol. 55, The Brain*, edited by T. Sejnowski, E.R. Kandel, C.F. Stevens, and J.D. Watson, 1990.
- [PG90] T. Poggio and F. Girosi. Ai memo 1140, AI Lab, MIT, 1989; *Proc. IEEE*, 78:1481, 1990.
- [SB91] Michael J. Swain and Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, November 1991.
- [Swa90] Michael J. Swain. Color indexing. Technical Report 360, University of Rochester Computer Science Dept., 1990.
- [WT92] David Wilkes and John K. Tsotsos. Active object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 1992.
- [WvdM92] L. Wiskott and C. von der Malsburg. A neural system for the recognition of partially occluded objects in cluttered scenes. Tr, Inst. fur Neuroinformatik, Ruhr-Universitat Bochum, 1992.