

Optimal Edge-Based Shape Detection

H. Moon, R. Chellappa, A. Rosenfeld

Center for Automation Research

University of Maryland

College Park, MD 20742-3275

Abstract

We propose an approach to accurately detecting two-dimensional shapes. The cross-section of the shape boundary is modeled as a step function. We first derive a one-dimensional optimal step edge operator, which minimizes both the noise power and the mean squared error between the input and the filter output. This operator is found to be the derivative of the double exponential (DODE) function, originally derived by Ben-Arie and Rao [5]. We define an operator for shape detection by extending the DODE filter along the shape's boundary contour. The responses are accumulated at the centroid of the operator to estimate the likelihood of the presence of the given shape. This method of detecting a shape is in fact a natural extension of the task of edge detection at the pixel level to the problem of global contour detection. This simple filtering scheme also provides a tool for a systematic analysis of edge-based shape detection. We investigate how the error is propagated by the shape geometry. We have found that, under general assumptions, the operator is locally linear at the peak of the response. We compute the expected shape of the response and derive some of its statistical properties. This enables us to predict both its localization and detection performance and adjust its parameters according to imaging conditions and given performance specifications. Applications to the problem of vehicle detection in aerial images, human facial feature detection, and contour tracking in video are presented.

Keywords

shape detection, edge detection, step edge, localization, boundary detection, vehicle detection, facial feature detection, contour tracking

I. INTRODUCTION

Human vision seems to make use of many sources of information to detect and recognize an object in a scene. At the lowest level of object recognition, researchers agree that edge and region information are utilized ([14]) to extract a 'perceptual unit' in the scene. Some of the possible invariant features are recognized and additional signal properties (texture or appearance) are sent to help in making the decision as to whether a point belongs to an object or not.

In many cases, boundary shape information, such as the rectangular shapes of vehicles in aerial imagery, seems to play a crucial role. Local features such as the eyes in a human face are sometimes useful. These features provide strong clues for recognition, and often they are invariant to many scene variables. Two-dimensional outlines of these features are sufficient in many cases, as humans can recognize a scene using a line drawing. Lowe [12] argued that object recognition in the human visual system doesn't necessarily require the reconstruction of depth information bottom-up from the visual input.

Identifying two-dimensional shapes has been a classical problem in computer vision. The ‘shapes’ here refer to two-dimensional (global or local) image features of an object which are either invariant to scene factors, or whose variation can be modeled easily. In most applications, it is difficult to model the intensity values of objects and their background. Therefore, it is reasonable to exploit the intensity differential along the object’s boundary. The intensity change along the boundary is usually modeled as a step edge. Once the pixels with high intensity gradient are chosen, these pixels should be examined to check if they lie on an expected shape boundary. Various methods have been used to collect local intensity changes into a global shape description.

The generalized Hough transform (GHT) [3] is a well-known approach, where the detected edge pixels vote for a shape according to a parametric representation or a table of boundary orientations with corresponding centroid positions. Since this approach depends on the orientations as well as the locations of edges, it gives poor localization performance. It is also hard to formulate the point spread function of the voting process. In [10][21], fitting polynomial curves to object boundaries to detect and classify shapes has been considered. Representing a shape using the coefficient vector of a fitted polynomial has the advantage that scale and orientation variation are easily dealt with. However, this polynomial representation depends on the order of the polynomial and provides only an approximate shape descriptor. In the case of a polygonal-shaped object, continuous polynomial representation is not efficient in general. Such a model gap is not easy to quantify, making a systematic error analysis difficult. Applying “band matching” after edge detection [8] is another approach. A band of fixed width having the expected object shape is put on the edge image, and if the number of edge pixels inside the band is above a certain threshold, the shape is considered to be detected. The width of the band gives more uncertainty for localization, and the analysis of the counting process is complicated.

These edge-based shape detection methods all suffer from the same problem: loss of information during the edge detection stage and the difficulty of statistical performance analysis. If we examine each point first as to whether it is a candidate edge pixel, then the decision is based only on local gradient information. Therefore, it is possible that a true boundary pixel, having a noisy gradient estimate, is discarded by the threshold used in edge detection. Additionally, the positions of detected edges are prone to noise. These errors are propagated by the grouping or fitting stage, and contribute to errors in detection and localization of the shape. The error

propagation of edge detection is nonlinear, and the grouping or fitting process is also nonlinear in nature [22]; the linear approximation of error propagation, when the perturbation is large, is not reliable. The relation between input and output errors, if available, can provide valuable information for adjusting parameters and predicting performance.

If we examine whether the given shape is present in a given position, by computing the intensity changes around the hypothetical shape contour at the same time, we only decide once whether the response is strong enough. We define the shape matched operator by extending the optimal step edge operator along the hypothetical object boundary contour so that responses from intensity differences along the boundary are collected simultaneously. We can prove that this filtering is in fact equivalent to collecting the gradient along the shape boundary. Moreover, since the responses are averaged over the neighborhood of the contour, our method is more robust than merely summing up the gradient magnitudes. Errors in shape position and response estimates averaged over boundary pixels are reduced proportionally to the size of the operator. More robust edge finding using an elongated edge operator has been suggested in [7]. The grouping or fitting process also gives a certain degree of averaging; however, information is lost once edge detection is performed, due to the digitization of edge position as well as the thresholding of edge strength. In our filtering approach, edges with weak responses can contribute to the shape response, so that an object having very low contrast on all or some portions of its boundary can be correctly detected. By shifting the sampling of operator values, we can estimate the positions of objects having very small numbers of pixels, with sub-pixel accuracy.

Accurate localization of the boundary edge leads to accurate estimation of geometrical parameters (size, orientation) and accurate discrimination between different shapes. In Section V, we investigate how the one-dimensional base operator affects shape discrimination performance. Our method is also more robust with respect to clutter than minimum mean squared error (MMSE) fitting, as will be explained in Section III. Detection of shapes having roof or ramp edge boundaries can also be easily implemented by replacing the base step edge operator with the corresponding matched operator. We have fewer parameters to deal with, since edge detection alone typically involves several parameters that need to be determined, and these parameters are directly related to performance predictions by our statistical analysis. There has been previous work on performance evaluation of edge detection and edge linking [17], and also on fitting of lines and circles [22]; however, it is hard to incorporate nonlinear error propagation

from edge detection to higher-level processes.

In an effort to ensure an accurate shape detection scheme, we set up a criterion for an optimal step edge operator and derive a closed-form solution, which is the derivative of the double exponential (DODE) function (Section II). We have verified that the DODE operator gives better performance than the derivative of Gaussian (DOG) operator, which is widely used for edge detection. The simplicity of this approach facilitates statistical analysis based on simple assumptions: We have found a way to compute the response profile of such an operator, using the local linearity of the response (Section IV). We are also able to formulate statistical properties of this detection procedure under the Gaussian white noise assumption, and thus predict its detection and localization performance (Section VI).

This shape detection method has a wide range of applications, in which we know the approximate shape and size of the object to be detected. We will show that operators for arbitrary shapes can be constructed automatically once we have boundary contours (not necessarily closed or even connected). We present examples involving vehicle detection, facial feature detection, and contour tracking in video in Section VII.

One weakness of our approach is that filtering with many geometrical parameters requires a large amount of computation; this can be alleviated by using multi-resolution processing according to shape complexity. In facial feature detection, for example, we first estimate the face region and orientation using a simple ellipse-shaped operator on a down-sampled image, and then proceed to detect facial features using finer operator shapes. Another way of reducing the computation is to use random sampling of parameters. While it is standard to apply all possible geometric parameters and find the set of parameters which gives the maximum response, we can find a small number of local maxima using sampled parameter vectors and look for the global maximum around the local maxima.

II. OPTIMAL EDGE OPERATOR

For many years, computer vision researchers have been trying to design good edge detectors, since edge detection is an essential step in most vision applications. A standard approach has been to define a one-dimensional edge operator and then extend it to the two-dimensional case. Since step edges are the most useful image intensity structures, most of the work on edge detection has focused on step edges.

Various forms of one-dimensional step edge operations have been investigated. In [15] and

[16], the use of difference-of-boxes operators was suggested, with the reservation that the ‘flat’ shape of boxes yields inaccurate edge localization. It is a basic result from signal theory that the optimal filter giving the best SNR for a given signal is the signal itself; the application of matched filtering to images is discussed in [18]. Therefore, the choice of an operator that has the same shape locally as a step function is reasonable, though additional discontinuities on the flanks of the operator give rise to false peaks. Argyle [2] suggested the use of a split-Gaussian operator to alleviate the problem of discontinuities. An interesting paper by Arcese et al. [1] shows that the (maximum-SNR sense) optimal filter for an intensity profile (not necessarily a step) in the presence of correlated noise with exponential correlation is a linear combination of the signal and the second derivative of the signal; the extreme case when the noise is independent corresponds to the signal only (the matched filter).

Canny [6] [7] performed a more systematic investigation incorporating an explicit localization criterion as well as a detection criterion (SNR) to find an optimal edge operator. He derived a numerical solution to the optimal edge operator using three criteria: higher SNR, more accurate localization, and fewer peaks in the response function. He suggested the use of the first derivative of a Gaussian function as a good approximation for implementation. Our noise power term is in fact equivalent to the SNR of the operator. The localization criterion in Canny’s formulation assumes the operator to be differentiable and the displacement error to be very small. However, the differentiability assumption leads to a performance limit, as will be discussed: the product of the detection and localization measures must be constant.

Edge detection is essentially finding a high intensity gradient. Ideally, the differential operator should be the optimal edge operator. In the presence of noise, we need to suppress high-frequency intensity structure while preserving the global step edge structure. Therefore, the derivative of the optimal smoothing operator should be the optimal edge operator by the following simple relation between differentiation and convolution:

$$(h * f)' = h' * f$$

Gaussian convolution has been the most popular scheme for smoothing, and this agrees with the closeness of the derivative of Gaussian to the optimal Canny operator.

In this paper, we derive a one-dimensional smoothing operator for a step function using a different criterion: minimizing the sum of the noise power and the mean squared error between input and output. Since this operator suppresses noise while preserving the step shape in an

optimal way, the derivative of the response function is less noisy and close to an impulse function, thus achieving very accurate detection and localization of the step edge.

Another difference between our approach and Canny’s is that there is no differentiability assumption in our derivation of the operator, and the minimization of the mean squared error between the true step function and the filter response ensures accurate step edge position estimation. Since the goal is to find a local maximum (Section VI), there is no simple way to incorporate the performance criterion into the criterion function. We use the squared sum of the ‘detection term’ and the ‘localization term’ rather than using the product as in [7], since the detection error is the perturbation of the edge in the y (response) direction, while the localization error is the perturbation of the edge in the x (pixel) direction; it is reasonable to take the squared sum to compute the total error. The result of this derivation turns out to be the derivative of the double exponential function, which has a discontinuity at the center. The shape of this operator is similar to that of the edge operator in [2], in that it shares the use of the central discontinuity. In fact, our closed-form solution is found to give very accurate localization, and since it is not differentiable at the center, it overcomes the trade-off between detection and localization (Section II-B) which is observed in [7]. It is also interesting that our edge operator is the second derivative of the smoothed edge profile, which is identical to one of the basis operators in [1].

There have been previous publications on deriving an optimal edge operator which takes the form of an exponential function. [5] derived the expansion filter as the optimal solution to the Discriminative Signal-to-Noise Ratio (DSNR) criterion for template matching, which emphasizes the central response. In [4], the Step Expansion Filter (SEF) is derived as an optimal step edge filter. The SEF is in fact the DODE filter that we derive here using a different criterion, and compares favorably to Canny’s step edge operator as a generic edge detection method. Optimal expansion matching filters for roof and ramp edges are also derived in [4]. While the form of the resulting operators is the same, we view edge detection as a gradient computation after regularization, and introduce the notion of optimal smoothing for achieving noise suppression for detection and accurate reconstruction of image structure for localization/recognition. We derived the DODE function using a much simpler formulation, and the same type of derivation should work for non-step edges or correlated noise, as will become obvious in the next section. Since our purpose is to design a shape detection scheme, we refer the readers to [4] for the general edge detection performance of the DODE operator.

A. Optimal smoothing operator

There has been work [19] on computing an optimal frequency domain filter for step edge detection to maximize the SNR. We also try to find an optimal edge filter in the frequency domain. Let the step edge with amplitude α be

$$X(t) = \begin{cases} 0, & x < 0 \\ \alpha, & x \geq 0 \end{cases}$$

which is corrupted by noise $N(t)$ with an autocorrelation function $R_N(t) = \mu \cdot \delta(t)$:

$$\tilde{X}(t) = X(t) + N(t)$$

We want to find an optimal smoothing filter h that minimizes the squared sum $\overline{E^2} + \overline{M^2}$, where $\overline{E^2}$ is the mean squared difference between the input and output signals, and $\overline{M^2}$ is the mean squared sum of the output noise response.

$$\tilde{Y}(t) = h * \tilde{X}(t) = h * X(t) + h * N(t)$$

Let $Y(t) = h * X(t)$, $M(t) = h * N(t)$ and $E(t) = X(t) - Y(t)$.

The mean squared errors, in terms of their frequency domain representations, are

$$\begin{aligned} \overline{E^2} &= \int |F_E(\omega)|^2 d\omega = \int F_E(\omega) F_E(-\omega) d\omega \\ &= \int S_X(\omega) (1 - H(\omega)) (1 - H(-\omega)) d\omega \end{aligned}$$

$$\begin{aligned} \overline{M^2} &= \int |F_M(\omega)|^2 d\omega = \int F_M(\omega) F_M(-\omega) d\omega \\ &= \int S_N(\omega) H(\omega) H(-\omega) d\omega \end{aligned}$$

where $S_X(\omega)$ and $S_N(\omega)$ are the spectral densities of X and N .

After simple algebraic operations, it can be proved [9] that the familiar Wiener filter

$$H(\omega) = \frac{S_X(\omega)}{S_X(\omega) + S_N(\omega)} \quad (1)$$

minimizes our criterion function.

Since we have a step edge and white noise, $S_X(\omega) = (\alpha^2/4)\delta(\omega) + (\alpha^2)/(4\pi^2\omega^2)$, and $S_N(\omega) = \mu^2$.

The optimal filter is then given by

$$H(\omega) = \frac{\frac{\alpha^2}{4}\delta(\omega) + \frac{\alpha^2}{4\pi^2\omega^2}}{\frac{\alpha^2}{4}\delta(\omega) + \frac{\alpha^2}{4\pi^2\omega^2} + \mu^2} = \frac{d^2}{d^2 + 4\pi^2\omega^2}$$

where $d = (\alpha/\mu)$; therefore

$$h(t) = \frac{d}{2} \exp(-d|t|)$$

It is interesting to observe that an exponential filter is again an optimal solution when the noise is correlated with an exponential autocorrelation function — that is, when $R_N(t) = \exp(-k|t|)$. After we put $S_N(\omega) = \frac{2k}{k^2 + 4\pi^2\omega^2}$ in (1) and rearrange, we have the following expression in the frequency domain:

$$H(\omega) = \frac{d^2 k^2}{d^2 k^2 + 4\pi^2(d^2 + 2k)\omega^2} + \frac{4\pi^2 d^2 \omega^2}{d^2 k^2 + 4\pi^2(d^2 + 2k)\omega^2} \quad (2)$$

We can find closed form expressions for the inverse Fourier transforms term by term:

$$h(t) = q(t) - \frac{d}{2k\sqrt{d^2 + 2k}} \frac{d^2 q(t)}{dt^2} \quad (3)$$

where $q(t)$ is the solution to the first term:

$$q(t) = \frac{dk}{2\sqrt{d^2 + 2k}} \exp\left(-\frac{dk}{\sqrt{d^2 + 2k}}|t|\right)$$

Now after double differentiation ($t \neq 0$), the solution is again an exponential function:

$$h(t) = \left(1 - \frac{d^3 k^2}{2k(d^2 + 2k)\sqrt{d^2 + 2k}}\right) \frac{dk}{2\sqrt{d^2 + 2k}} \exp\left(-\frac{dk}{\sqrt{d^2 + 2k}}|t|\right) \quad (4)$$

We can easily show that $0 < 1 - \frac{d^3 k^2}{2k(d^2 + 2k)\sqrt{d^2 + 2k}} < 1$. It is worthwhile to note that if we increase d and k , $\frac{dk}{\sqrt{d^2 + 2k}}$ inside the exponential grows, so that we get a sharper peak. That is, we can use a sharper smoothing function for accurate reconstruction when the noise is weak relative to the signal power and when the noise is highly uncorrelated.

The one-dimensional smoothing operator derived above can be extended to a two-dimensional operator for application to images:

$$\bar{h}(x, y) = h((x^2 + y^2)^{\frac{1}{2}})$$

The optimal edge operator is merely the (piecewise) derivative h' of the smoothing operator, by the following relation:

$$(h * I)' = h' * I \quad (5)$$

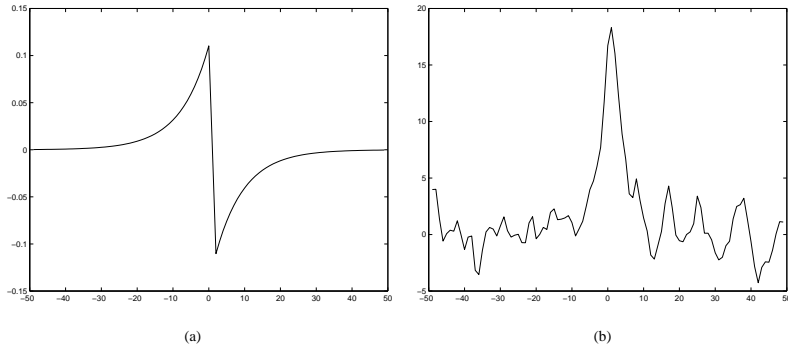


Fig. 1. The optimal edge operator and its response to a noisy step edge

The optimal step edge operator is shown in Figure 1(a). It is worthwhile to note that we can compute the optimal smoothing operator as long as we have models for the edge profile and noise.

B. Experiments

The performance of the optimal step edge operator derived in the previous section is compared with the difference-of-boxes (DOB) operator and the DOG operator. The DOB operator has a fixed width of 20, while the widths (standard deviations of the corresponding pdfs) of the DOG and DODE operators are varied from 0.2 to 20.

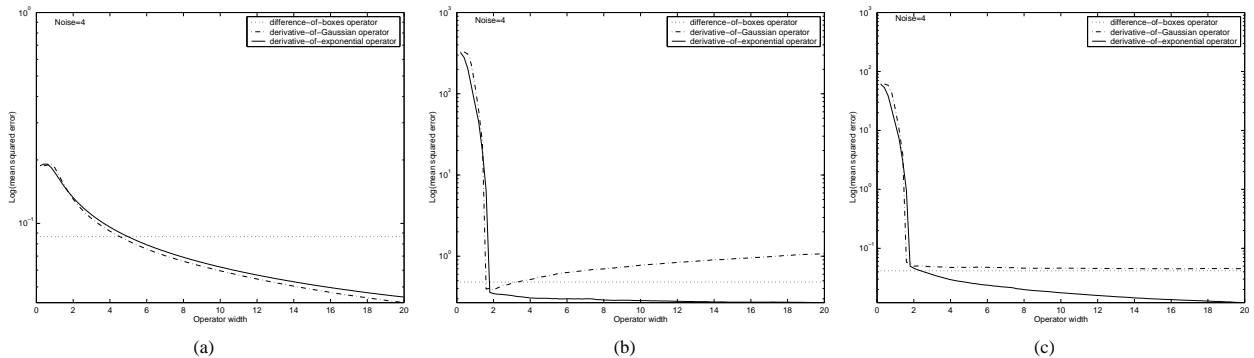


Fig. 2. Edge detection performance of DOB, DOG, and DODE operators

In this experiment, an ideal step edge is corrupted by i.i.d. Gaussian noise and convolved with the three operators. The position giving the maximum response is chosen to be the edge location and the magnitude is stored as the edge strength (Figure 1(b)). The detection performance, for input noise level $\mu = 4$ and step magnitude $\alpha = 10$, measured by the mean squared error of

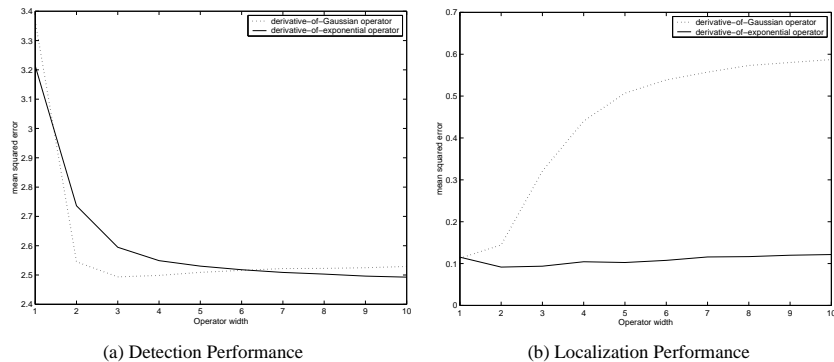


Fig. 3. Vehicle detection performance of the DOG and DODE operators

the peak response (normalized by its signal power), is given in Figure 2(a). The corresponding localization performance measured by the mean squared error is shown in Figure 2(b). We observe that the detection performance of the DODE operator is very slightly poorer than that of the DOG operator, for the same scale of the width. This poses no significant problem, since extending the width of the operator always yields better performance (as long as edges are not close together). For localization, there is a significant performance difference in favor of the DODE operator, which cannot be overcome by extending the operator width. It is interesting to observe that the invariance of the combined performance exists for DOG when the localization error is small, but not for DODE, as previously mentioned. This is shown in Figure 2(c), where the product of the detection and localization errors is plotted. For implementation, the operator should be truncated before using it for either edge detection or shape detection. Experiments using truncated operators reveal similar performance differences. The performance comparison of the DODE and DOG operators in the case of vehicle detection is shown in Figure 3. Use of the DODE operator again shows similar detection performance (Figure 3(a)) and significantly better localization performance (Figure 3(b)).

III. OPERATOR FOR ARBITRARY SHAPES

We model the intensity change at an object boundary point as a step function, and assume that the boundary is a smooth, simply connected contour. The smoothness condition can be dropped to accommodate the case of objects with piecewise smooth boundaries. The following derivation of the operator function can be easily generalized to include such shapes as polygons.

Let the object image be represented by the function $I(\mathbf{x}) = \alpha \cdot 1_D(\mathbf{x})$, where D is a simply

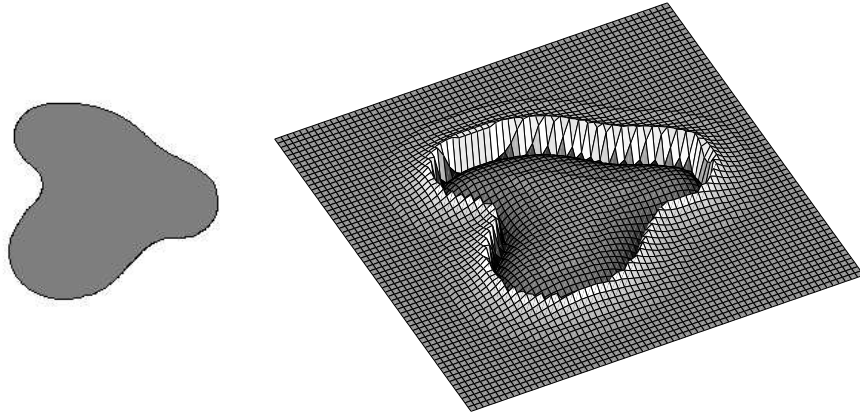


Fig. 4. An arbitrary shape and a matched operator

connected region representing the shape, and let the boundary $C = \partial D$ be parameterized by $\mathbf{c}(t), t \in J$. This assumption of uniformity of intensity is not critical in real applications, as long as there is an intensity difference on some portion of the shape boundary. We can find a level function l satisfying

$$\begin{aligned} \{\mathbf{x} : l(\mathbf{x}) = k\} &= C, \text{ for some } k > 0 \\ \tilde{l}(s) &\triangleq l(\mathbf{c}(t) + s \cdot \nabla l(\mathbf{c}(t))) = s \end{aligned} \quad (6)$$

where ∇ is the gradient operator. The second equation holds where l is well-defined.

We can construct l for implementation:

$$l(\mathbf{x}) = \begin{cases} \min_{\mathbf{z} \in C} \|\mathbf{x} - \mathbf{z}\| & \text{for } \mathbf{x} \in D \\ -\min_{\mathbf{z} \in C} \|\mathbf{x} - \mathbf{z}\| & \text{for } \mathbf{x} \in D^c \end{cases}$$

l is well-defined and continuous, and locally smooth around the boundary contour. This choice of level function facilitates the computation and implementation of the operator for any arbitrary shape. The operator employing this choice of level function is shown in Figure 4. We can even accommodate open contours by replacing [inside:outside] by [one side:the other side] in the above definition.

Let the operator function $f(\mathbf{x})$ be

$$f(\mathbf{x}) = h'(l(\mathbf{x}), \sigma)$$

where $h(t) = h(t, \sigma) = \frac{1}{\sigma} \exp(-|t|/\sigma)$, the optimal one-dimensional smoothing operator. Here, the derivative is not defined at the origin. Note that the cross-section of $f(\mathbf{x})$ along the normal

direction at a point on the contour, i.e., the gradient direction for l , is h' , since the level function l is constructed (6) so that it is the identity function along the cross-section. When the boundary contour is not simply connected, we can decompose it into a sum of simply connected closed contours to construct a shape operator by combining the operators constructed for each component.

This operator is in fact a natural two-dimensional extension of edge detection. A step edge operator applied to an image pixel is

$$\bar{h}_n = \frac{\partial \bar{h}}{\partial \mathbf{n}} = \mathbf{n}^T \nabla \bar{h}$$

where \mathbf{n} is the unit vector normal to the edge direction. Since the direction \mathbf{n} is not known in general, directional derivatives in the vertical and horizontal directions are applied to estimate both edge direction and strength. However, when we have an edge contour model, it is possible to integrate the gradient magnitude along the hypothetical shape boundary. Let p be a point on a hypothetical shape boundary C , which is put on an image I with the same shape. Assume that C is close to the object; then p picks up the maximum gradient magnitude at z_p , the closest point to P on the boundary. The gradient magnitude m_p at p depends only on the distance between p and z_p and the edge strength (assume unit strength):

$$m_p = \|\nabla(\bar{h} * I_0)(p)\| = (h * I_0)'(\|p - z_p\|) = (h' * I_0)(\|p - z_p\|),$$

where we have used (5), and I_0 is the one-dimensional step function at the origin.

If we integrate the gradient magnitude m_p along $p \in C$, the result is approximately the same as the response of the operator f to I . The sum of the gradient magnitude along C is in fact the same as the response of f to I when C is matched to the object, and the difference grows when C moves away. We can design the operator so that these two quantities match anywhere; if we wish to estimate the position of the object only along the x direction, the cross-section of f along the x direction should be h' for the above relation to hold exactly. However, we construct f using the level function l so that the cross-section along the normal direction is h' , to efficiently accommodate all geometrical features (position, orientation, size).

If we define shape detection as the process of identifying the intensity changes along the shape boundary, we can claim that our scheme is optimal; convolving the ideal image with the shape operator is equivalent to computing intensity gradients after *optimal smoothing*.

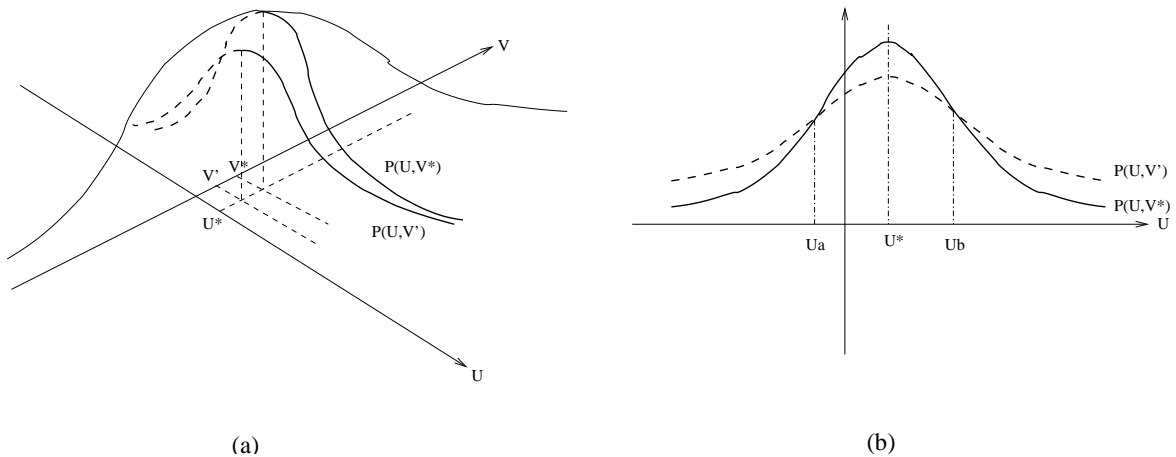


Fig. 5. Response profile and local linearity

This shape detection scheme is also robust with respect to clutter. We can interpret this as fitting a shape to edges so that the distances from the hypothetical shape boundary to the edges are minimized. Maximizing the response to the shape operator is equivalent to minimizing the sum of distances between the shape and edges, where the distance is measured using a robust error metric $e(x, y) = \text{const} \cdot (1 - \exp(-|x - y|))$. This kind of robust error metric has been used to reject the effect of outliers in an estimation problem.

IV. RESPONSE PROFILE

The shape operator is put into a position in an image, and the responses are collected at the centroid of the operator. This operation is repeated for all possible positions, and the maximum response is chosen as indicating the presence of the specified shape. If we know the scale and the orientation of the object *a priori*, we simply take the convolution. Without such information, every possible orientation and scale value should be tried.

We have found that the response profile, with or without prior information about size or orientation, is the same as the convolution response for the correctly matched operator, locally around the true centroid. We can use this local linearity property to theoretically predict the detection and localization performances. This local property doesn't hold strictly in the continuous domain; nevertheless, the response profile is usually well approximated by spatial convolution. It holds in the discrete domain due to the quantization of the parameter values.

The process of applying operators having different geometric parameters can be viewed as parameter estimation. Let X , Θ , and S denote the position, orientation, and scale parameters,

respectively. Let $f(X, \Theta, S)(\cdot)$ be the operator having these parameters, and I be the image. The estimate is given by

$$(X^*, \Theta^*, S^*) = \arg \max_{(X, \Theta, S)} f(X, \Theta, S)(I)$$

If, for example, the orientation and scale parameters are known to be (Θ_0, S_0) , then the location estimate is

$$X^* = \arg \max_X f(X, \Theta_0, S_0)(I)$$

and the operator $f(X, \Theta_0, S_0)(\cdot)$ is a convolution.

Since the location parameter X usually contains more information than the other parameters, the two-dimensional response profile

$$P(f, I)(X) = \max_{(\Theta, S)} f(X, \Theta, S)(I)$$

is stored and used to characterize the localization performance.

As noted earlier, the response profile $P(f, I)$, with or without prior information about (Θ, S) , is the same as the convolution response with the correctly matched operator, locally around the true centroid:

$$\begin{aligned} P(f, I)(X) &= \max_{(\Theta, S)} f(X, \Theta, S)(I) \\ &= \max_{\Theta} f(X, \Theta, S_0)(I) \\ &= \max_S f(X, \Theta_0, S)(I) \\ &= f(X, \Theta_0, S_0)(I) \\ &\text{for } X \in N(X_0) \end{aligned}$$

where $N(X_0)$ is a small neighborhood around the true centroid X_0 .

This local property doesn't hold strictly in the continuous domain; nevertheless, the response profile is usually well approximated locally by the spatial convolution. It holds in the discrete domain due to the quantization of the parameter values. Figure 5 shows this effect. Suppose that the response of a filter is given in terms of two parameters, U and V , as in Figure 5(a). The parameters (U^*, V^*) which give the maximum of $P(U, V)$, are picked up as the correct parameters. Consider two different situations where the correct value V^* is and is not known, respectively. Assume that we are only interested in computing the response profile for the parameter U . When

we know the correct parameter value, the response profile for U is $P(U, V^*)$ (Figure 5(b)). If, on the other hand, we don't have any information about V , then the response profile is computed as $\bar{P}(U) = \max_V P(U, V)$. Since the correct parameters are (U^*, V^*) , we have $\bar{P}(U^*) = P(U^*, V^*)$. If we move away from U^* , the function $\bar{P}(U)$ may pick up values from the other parameter value V . Let V' be the quantized parameter value closest to V^* . Then as in Figure 5(b), the profile function $P(U, V')$ is always less than or equal to $P(U, V^*)$ on some neighborhood $[U_a, U_b]$ of U^* , by the continuity of P . This should hold for other values of V . In other words, the response profile $\bar{P}(U) = \max_V P(U, V)$, without any prior information on V , always picks up values from $P(U, V^*)$, locally on $[U_a, U_b]$.

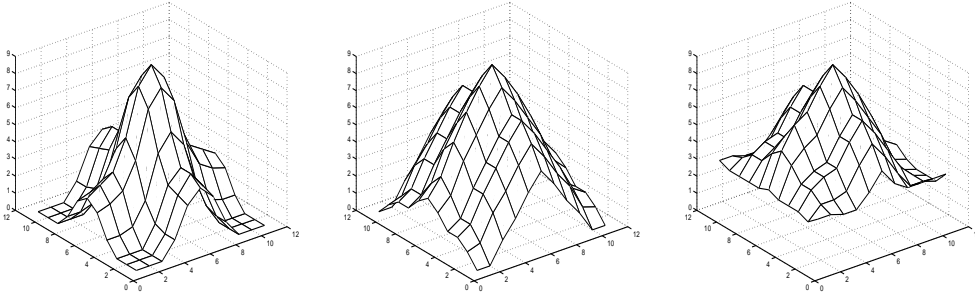


Fig. 6. Response profiles with different degrees of prior information. Left: Fixed size and orientation. Middle: Varied size, fixed orientation. Right: Varied size and orientation.

In Figure 6, the responses of the vehicle operator are shown for different degrees of availability of prior information. In the left graph, the operator size, shape and orientation are known, hence these values are fixed; in the middle, different sizes and shapes are tested; in the right, operators with different orientations are also tried. We can observe that the responses near the peaks are identical. Note also that when we have less prior information, the responses decrease slowly as we move farther from the centroid, as pointed out above.

This observation is very helpful, since the convolution profile is relatively simple to formulate because of its linearity and the known geometry of the operator, while other response profiles are much more complicated. This local property is enough to get the probability density function of the response.

It is easy to compute the convolution profile in the spatial domain. The response profile $r(\mathbf{x})$, when the true position of the object centroid is the origin and the operator is positioned at \mathbf{x} , is given by

$$r(\mathbf{x}) = \int f(\mathbf{x} - \mathbf{u})I(\mathbf{u})d\mathbf{u}$$

where f is the vehicle operator and I is the ideal vehicle image.

V. SHAPE DISCRIMINATION PERFORMANCE



Fig. 7. Simple shapes for investigating shape recognition performance

In real world scenes, a single ‘object’ class can produce many different image appearances. For example, while the human face has unique characteristics, individual faces differ. Lighting and pose can also make dramatic differences in image appearance. Therefore, two aspects of object detection/recognition should be taken into account when we design an algorithm. One aspect is to detect an object category regardless of variations in appearance. Face detection is one of many examples. The other important task is to discriminate one class of objects from another. Face recognition or gender recognition are examples. Detecting only military vehicles in aerial photos is another such task. However, the first task also requires discriminative power. For face detection, we have to eliminate other face-like objects such as dolls, volley balls, etc.

We have to design an algorithm so that it can tolerate variations inside the category of objects we want to detect, while avoiding objects that we want to rule out. Such a task should make use of good image features, and sometimes requires high-level knowledge and decision. In many problems, the shape of the boundary (for example, cars, buildings) and/or the shapes of features (for example, eyes on a face, a logo on merchandise) give strong clues about the identity, size, and pose of the object. We examined how we can manipulate the 2-D shape of the operator and the width of the base operator (cross-section) to deal with these two aspects of object detection.

We first chose four kinds of simple geometric shapes – circle, triangle, square, and hexagon, and generated images with the corresponding matched operators (Figure 7). We applied each of these operators to the images; the responses are shown in Figure 8. The first column shows

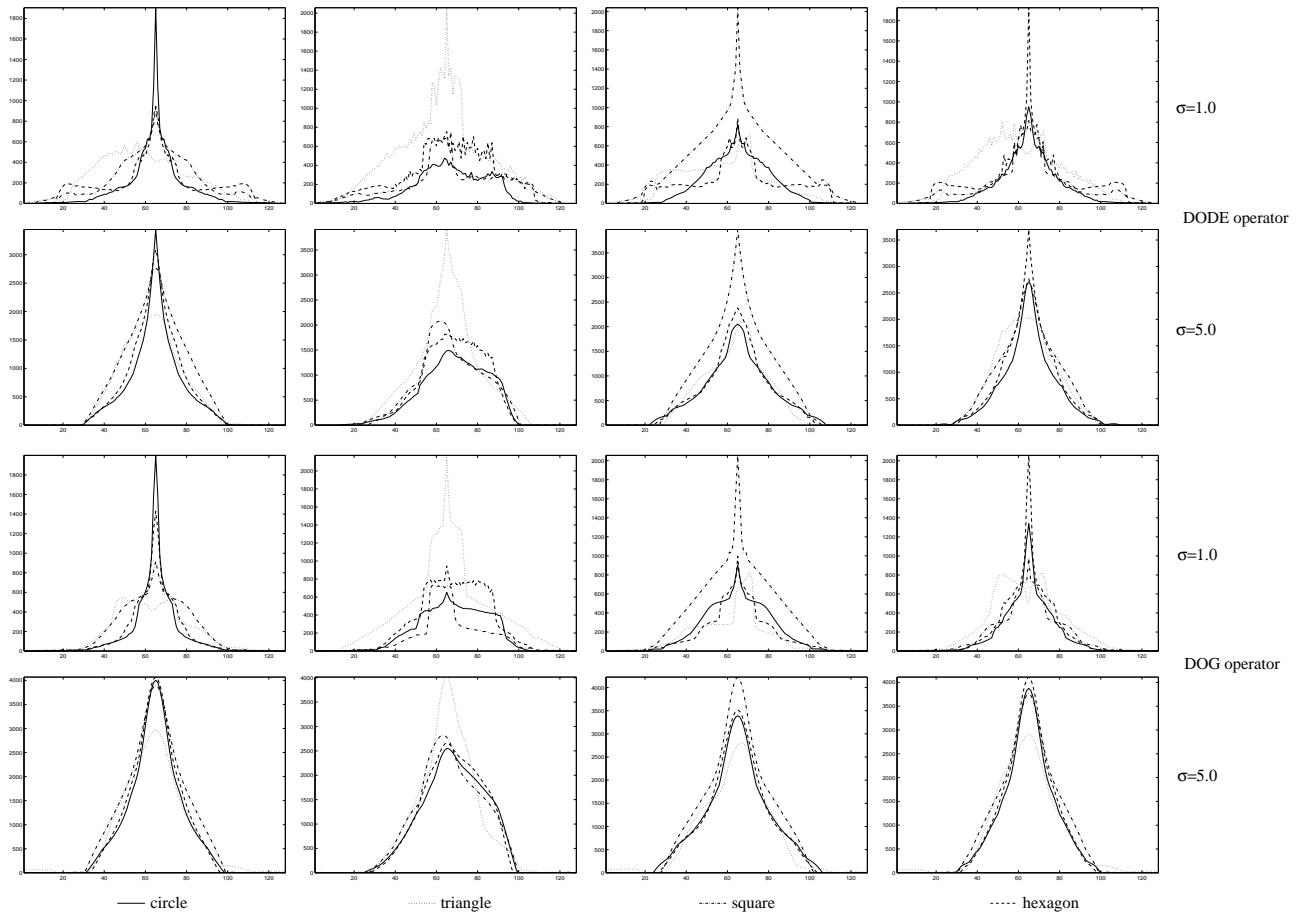


Fig. 8. Response profiles of shape operators to matching and non-matching shape images are compared. First row: responses of a sharp DODE operator ($\sigma = 1.0$). Second row: responses of a flat DODE operator ($\sigma = 5.0$). Third row: responses of a sharp DOG operator ($\sigma = 1.0$). Fourth row: responses of a flat DOG operator ($\sigma = 5.0$). Columns represent different shape images in Figure 7.

the responses of the four operators to the circle image, the second column shows their responses to the triangle image, and so on. The rows represent different operators (DODE and DOG) and different operator widths ($\sigma = 1.0$ and $\sigma = 5.0$). For each operator, we varied its size in some range and picked up the maximum response at each pixel (supposedly the likelihood of the pixel being the centroid of the given shape). To facilitate comparison, the maximum response along each column (in the response image) is collected and plotted. The plots are coded as solid or dotted lines according to the corresponding operator shapes. We observe that when the operators are sharp ($\sigma = 1.0$), they can resolve the shapes easily. For example, in the top

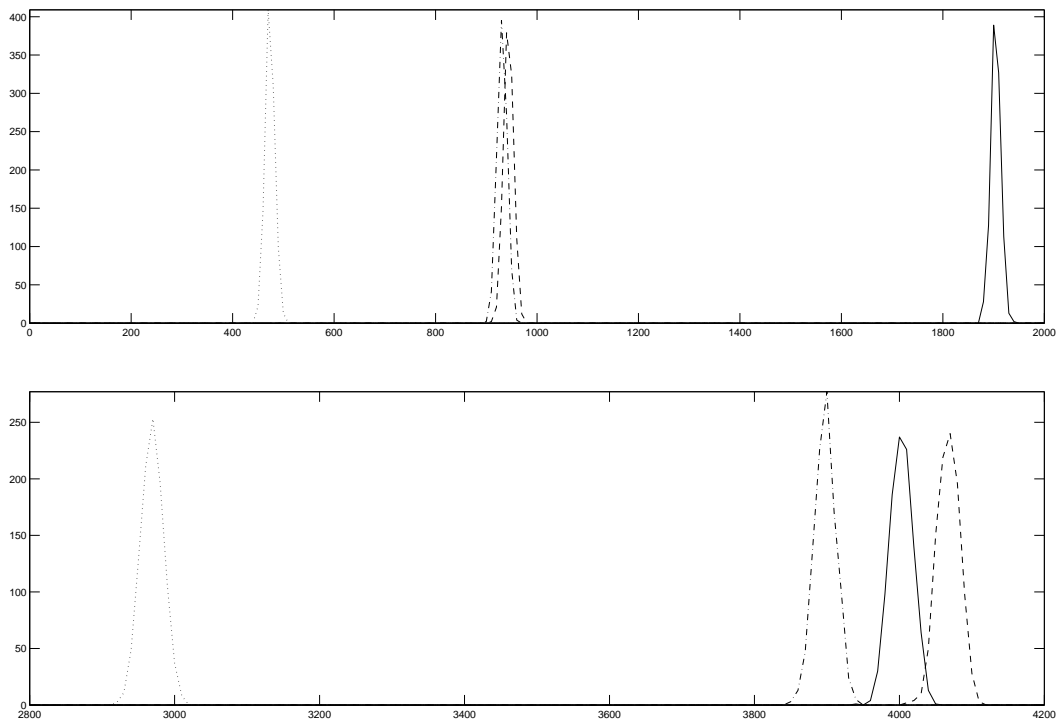


Fig. 9. Comparing the shape resolution performance using the empirical pdf of response magnitude. Top: The pdf of the response when a sharp operator is used (DODE, $\sigma = 1.0$). Bottom: The pdf of the response when a flat operator is used (DOG, $\sigma = 5.0$)

leftmost graph where the circle image is tested using sharp DODE operators, the response to the circle is prominently higher than the responses to the other shapes. When the flat ($\sigma = 5.0$) DOG operators are used ($\sigma = 5.0$) the response to the correct shape is not significantly higher than the responses to the others, so that it is hard to distinguish one shape from another. The distinction is especially weak between ‘similar’ shapes such as a circle and a hexagon (the bottom leftmost and rightmost graphs). We can observe that the DODE operator produces relatively accurate responses, even though a flat operator is used.

We investigated the shape recognition performances of two extreme operator shapes (DODE $\sigma = 1.0$ and DOG $\sigma = 5.0$), by collecting the empirical distributions of the responses. We generated a thousand noisy images of circles, using zero-mean i.i.d. Gaussian additive noise with average noise power $\mu = 20.0$. Each operator is applied, and the maximum response is recorded as the likelihood of the presence of the given shape. The top distribution in Figure 9 corresponds to the case of the top leftmost graph (DODE $\sigma = 1.0$); the bottom distribution to the case of the bottom leftmost graph. For the DODE $\sigma = 1.0$ operator, the distribution of the response to

the correct shape (circle) is far from the responses to the other shapes. For the DOG $\sigma = 5.0$ operator, the hexagon operator yields higher responses than the correct operator does, and the distributions overlap. When the scene is cluttered or the shape is less well-defined, we can expect that the shape ambiguity will be further amplified.

VI. STATISTICAL PROPERTIES

We are able to derive some of the statistical properties of the shape detection process – its detection probability and localization error, assuming additive Gaussian i.i.d. noise.

We can compute the probability density function of the responses at points around the true centroid. Since the filtering is locally linear, the responses are correlated Gaussian, and we can get the covariance matrix using the convolution profile. Let $\underline{\mathbf{x}} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$ be points around the true centroid, including the centroid; $\underline{y} = (y_1, y_2, \dots, y_N)$ be the corresponding responses; and Σ be the covariance matrix. The ideal response profile $\underline{r}(\mathbf{x}) = (r(\mathbf{x}_1), r(\mathbf{x}_2), \dots, r(\mathbf{x}_N))$ has been calculated above.

The pdf of \underline{y} is given by

$$P_Y(\underline{y}) = \frac{1}{(2\pi)^{\frac{N}{2}} \det \Sigma} \exp\left(-\frac{1}{2}(\underline{y} - \underline{r}(\mathbf{x}))^T \Sigma^{-1}(\underline{y} - \underline{r}(\mathbf{x}))\right)$$

The probability that the maximum occurs at position x , which gives the localization distribution, is given by

$$\begin{aligned} P_{\mathbf{X}_{max}}(\mathbf{x}_i) &= \text{prob}(\text{maximum occurs at } \mathbf{x}_i) \\ &= \text{prob}(y_1 \leq y_i, \dots, y_N \leq y_i) \\ &= \int \int^{y_i} \dots \int^{y_i} P(y_1, \dots, y_N) dy_N \dots dy_1 dy_i \end{aligned}$$

Note that the maximum response which is compared to the threshold is the largest order statistic $y_{(N)}$ from y_1, y_2, \dots, y_N , and the corresponding position $\mathbf{x}_{cent} = \mathbf{x}_i$ is marked as the centroid. The pdf of the maximum response is given by

$$\begin{aligned} P_{Y_{max}}(y) &= \sum_{i=1}^N P(y | \mathbf{X}_{max} = x_i) P_{\mathbf{X}_{max}}(\mathbf{x}_i) \\ &= \sum_{i=1}^N P_{y_i}(y) \int \int^y \dots \int^y P(y_1, \dots, y_N) dy_N \dots dy_1 dy_i \end{aligned}$$

where $P_{y_i}(y)$ is the marginal distribution of y_i . We conducted experiments to verify the derived theoretical performance in the case of vehicle detection. First, the probability distribution

σ	-2	-1	0	+1	+2
12	0.00001	0.04416	0.91166	0.04416	0.00001
	0.00001	0.04329	0.91246	0.04424	0.00000
16	0.00051	0.09781	0.80336	0.09781	0.00051
	0.00047	0.09732	0.81841	0.09803	0.00044
20	0.00330	0.14343	0.70654	0.14343	0.00330
	0.00292	0.14337	0.70645	0.14242	0.00325

TABLE I

THEORETICAL AND EMPIRICAL DISTRIBUTIONS OF THE CENTROID: THE ONE-DIMENSIONAL DISTRIBUTION (TOP ROWS) OF VEHICLE CENTROIDS IS COMPARED WITH THE EMPIRICAL HISTOGRAM (BOTTOM ROWS) AROUND THE TRUE CENTROID (PIXEL NO. 0) USING DIFFERENT OPERATOR WIDTHS ($\sigma = 12, 16, 20$)

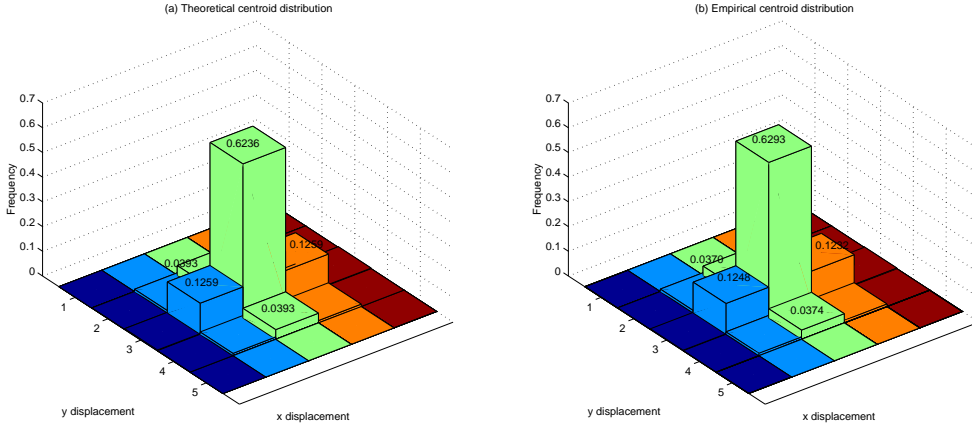


Fig. 10. Two-dimensional theoretical and empirical distributions of the centroid estimate. Left: Centroid distribution computed using equation (7). Right: Empirical centroid histogram.

$P_{\mathbf{X}_{max}}$ of the position estimate is computed by a randomized numerical integration algorithm [11] according to the above formulation. It is compared with empirical distributions obtained by the following experiments.

A rectangular shape with constant grey level 120 against a background with grey level 100 was generated. The easily available empirical response profile $r(\mathbf{x})$ was used to compute the pdf. The ideal image was perturbed by additive i.i.d. Gaussian noise with variance σ^2 . We generated many instances of images for a given σ and ran the detection algorithm on them to

get the distribution of the centroid. We tested three different noise levels: $\sigma = 12$, $\sigma = 16$, and $\sigma = 20$. 100,000 instances of perturbed vehicle images were used for each noise level. In Table I the theoretical distribution of the centroid along the vehicle length direction (top rows for each σ) and the corresponding empirical distributions (bottom rows) are summarized, and they show a close match. The two-dimensional empirical and theoretical distributions of the centroid of the rectangle around the true centroid ($\sigma = 20$) are compared in Figure 10.

Based on the above formulation, it is straightforward to predict the localization and detection performances under ideal assumptions: the localization performance is represented by the pdf of the centroid estimate p_{cent} , and the probability that the maximum response is less than the given threshold is the estimate of the misdetection probability.

A. Roof or ramp edge boundary

When the intensity change along the boundary is not a step edge, we can apply the formulation in Section II to the case of a roof or ramp edge, and construct operators to perform the corresponding task effectively. For example, the motion blur in a video sequence or the object boundary in a low-resolution image can be modeled using a ramp edge. The statistical analysis in Section VI will also hold for general cases.

VII. APPLICATIONS

In real-world problems, we have numerous issues to deal with. The range of applications is limited because our method demands that we know the expected object shape. It is not feasible to try all possible transformations and imaging conditions that the object may have gone through. Since it is impractical to expect that all the objects in an image have exactly the same shape and intensity differential from the background, it is not easy to set a threshold and assess performance according to the above formulation. However, we can incorporate domain knowledge to determine thresholds and to remove false alarms. In the case of eye detection, for example, we can make use of the distribution of flesh tone, and the positions of the eyes relative to other facial features, to determine an appropriate threshold and to remove false alarms.

We can also adjust the operator shape to make detection more robust, and we can accommodate small changes in shape and illumination. Our studies of vehicle detection [13] show that our method works under some variations in vehicle articulation and illumination change.

A. Vehicle detection

Vehicles in parking lot images have different dimensions, shapes, relative positions of windshields, and colors (Figure 11). However, they can be approximated by rectangles when the images are taken vertically, and their ranges of lengths and widths are relatively narrow. Since images can be taken from directions other than vertical, this model is generalized to a parallelogram.

The operator consists of four linearly elongated edge operators making up a parallelogram. Each operator corresponds to one of the four sides of the vehicle. Responses are collected at the center of the set of operators, which corresponds to the vehicle centroid. The algorithm tries every possible hypothesis about camera angle (along the camera axis) and vehicle size if there is no prior information. The hypothesis giving maximum response is then accepted and the corresponding response is registered as the measure of the likelihood of a vehicle being present with the chosen hypothesis (Figure 12). If we know the value or range of values of a parameter, fewer hypotheses need to be tried.

Two thresholds are used to remove spurious responses and to declare that a vehicle has been detected: If we have an empty parking lot region next to a high-contrast vehicle, the response from one side of that vehicle is high enough at the empty spot to be accepted as indicating the presence of a vehicle. We filter out this kind of response by removing candidates that don't have enough votes from all four sides. This gives rise to our first threshold. This threshold should depend on scene factors such as noise level and contrast; however, we have found that it is not very sensitive to these factors, so we have left it fixed during most of our experiments.

If the summed response at a given point from each of the four operators is above a certain value, we suspect that a vehicle may be present at that point and put it on the list of possible vehicle centroids. This step gives our second threshold. The value of this threshold can be determined to meet a given performance requirement.

Finally, since there can be spurious responses due to windshields or shadows, candidates that are inside other stronger candidates are discarded.

Figure 13 shows the final output of the vehicle detector, which gives the most probable vehicle dimensions as well as the locations of the centroids. The high-contrast vehicles again give more accurate estimates of the dimensions and centroids. The effect of contrast on the centroid estimation is not readily visible in this image; however, it is evident from equation (4) that the

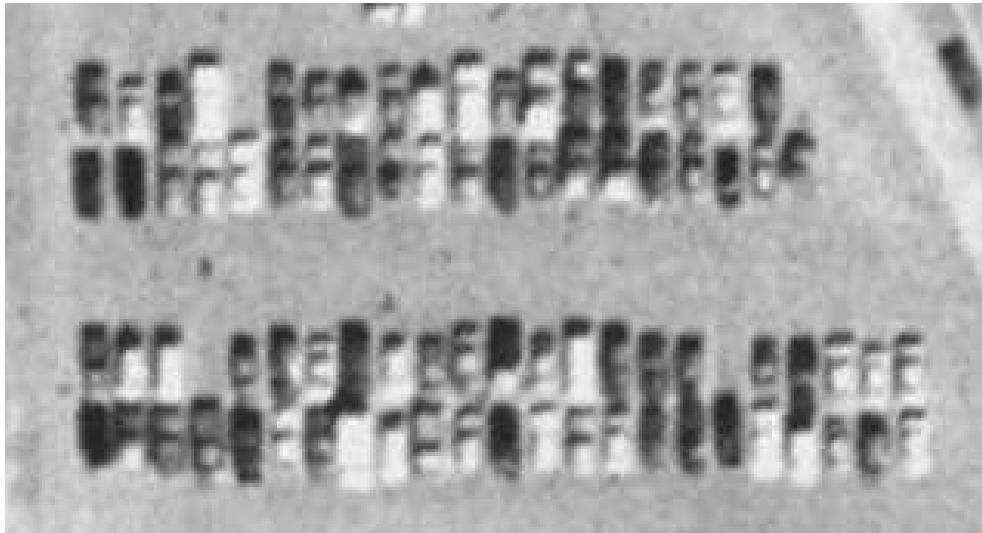


Fig. 11. Parking lot image



Fig. 12. Response image. Pixel darkness represent the likelihood of the presence of a vehicle (centered at that pixel) having a specific dimension (that yields the maximum response).

response profile would be sharper for higher contrasts, and therefore the centroid error should be smaller.

There is an important issue of prior knowledge in this application, since we can suspect that information about parking lot orientation and illumination would improve performance. The prior information is incorporated in the algorithm by testing fewer hypotheses; it has been suggested that the response profile should be the same locally, and the profiles shown in Figure

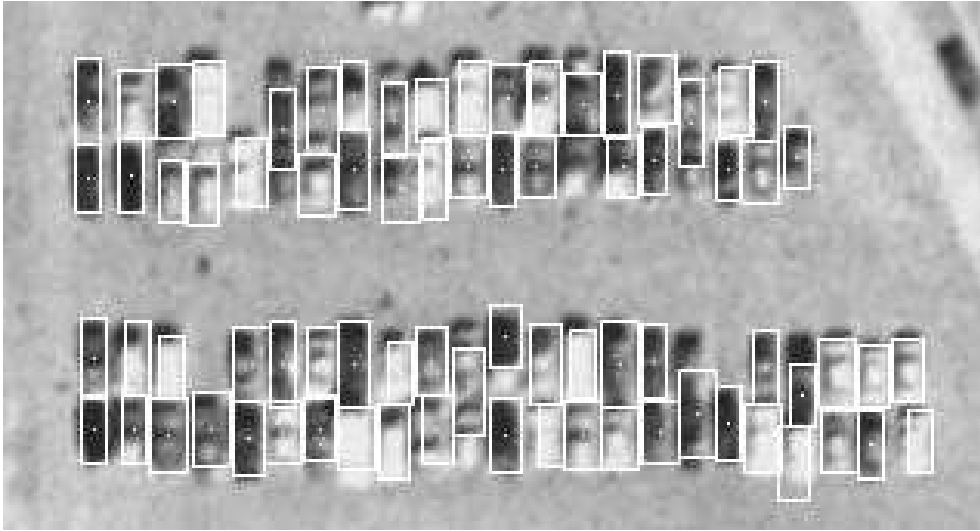


Fig. 13. Detected vehicles. The local maxima from the response image are chosen as vehicle centroids, and marked as white dots surrounded by rectangular boundaries giving the best fit. In the process, any candidate pixels inside other vehicle candidates having higher confidences are discarded.

6 give evidence for this. However, experiments on real data show that prior knowledge helps detection performance. Figure 6 also suggests that applying more operators (i.e., when there are more unknown parameter values, so that we have to try a larger set of hypotheses) gives higher responses at pixels away from the centroid; therefore we suspect that interference with adjacent vehicles will give many false alarms. Experiments showed that the localization performance is not improved by prior information, as expected from our analysis.

Table II shows the results of experiments using 15 images from the Fort Hood Image Set, containing 5073 vehicles. In parking lot images with large numbers of vehicles, it is reasonable to test some range of different vehicle dimensions; but here only the contribution of prior orientation information has been investigated. The vehicle detection output when site information (regions of interest along with parking lot orientations) is used is shown in Figure 14. The algorithm takes about 3.5 minutes using Pentium 4 1.8Ghz PC, when processing the image in Figure 14 (a 1024x1024 image). When there is no prior information about the parking lot orientation, the algorithm tests all four possible orientations ($0^\circ, 45^\circ, 90^\circ, 135^\circ$); the algorithm takes about four times (13.7 minutes) as expected.



Fig. 14. Parking lot image and detected vehicles

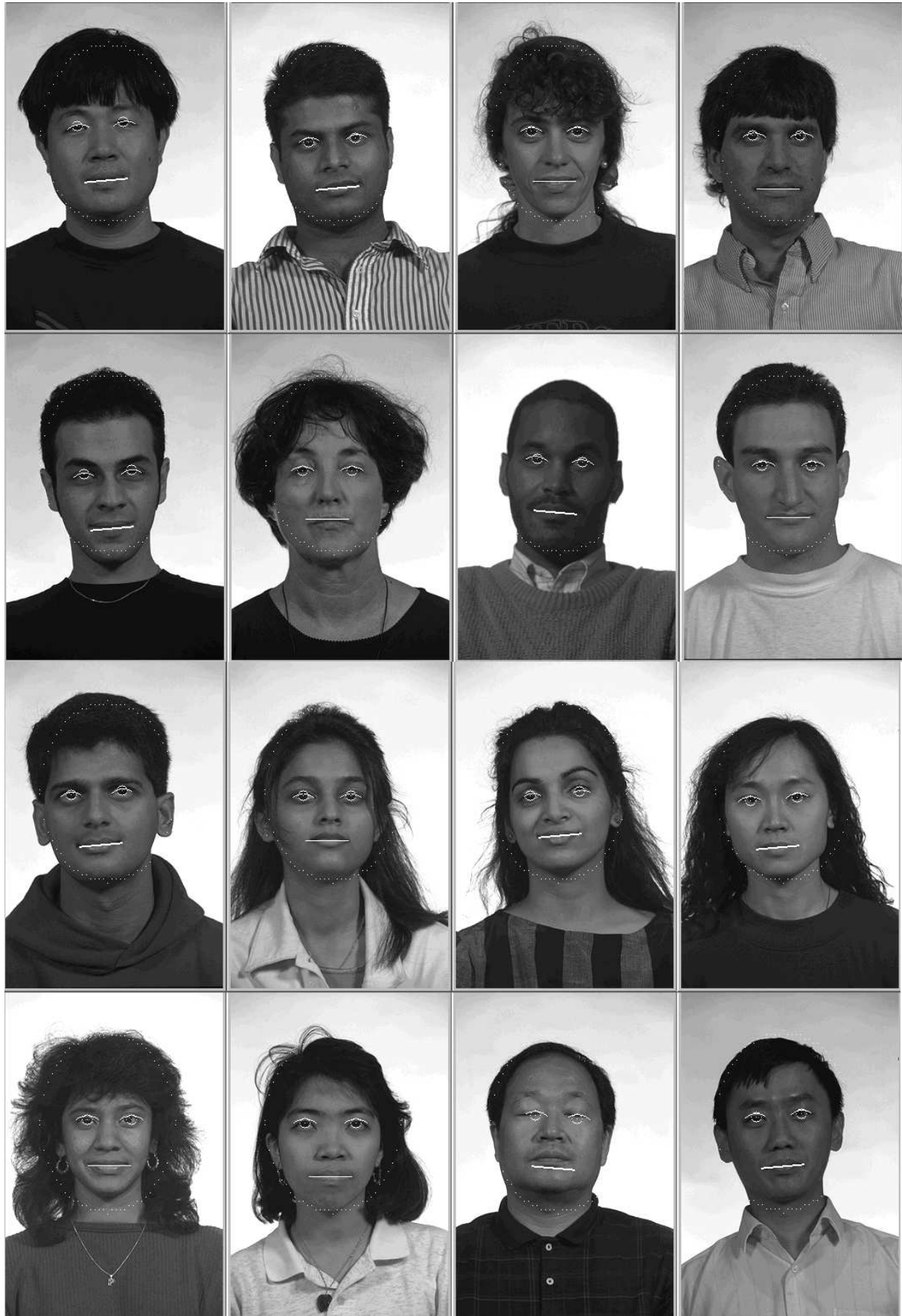


Fig. 15. Faces with detected eyes and mouths

Mask orientation	without prior	with prior
Missed detections (%) (missed/target vehicles)	36.3 (1843/5073)	20.1 (1021/5073)
False alarms (%) (false alarms/detected vehicles)	54.6 (4077/7469)	21.8 (1176/5405)
Localization error (pixels)	1.963	2.109

TABLE II

DETECTION AND LOCALIZATION PERFORMANCES WITH REAL IMAGES

B. Human facial feature detection

Different people’s eyes have different shapes, sizes, and positions in the face; however, their range of variation is small. The mouth has the same characteristics. While they are not strictly two-dimensional objects, they can be treated as patterns drawn on the face ([20]). Even with unfavorable illumination and camera angles, they have invariant features such as circular irises, intensity differences at the corners of the eyes, and the shade between the upper and lower lips. Variations due to facial expressions are well-constrained and easy to model – for example, squinting of the eyes and opening and closing of the mouth.

We designed a small set of operators for eyes and mouths, and applied them successfully to the facial feature detection problem. For eye detection, we have two operators: one for the eyelids and one for the iris. We put together two circular arcs, one on the top and the other on the bottom, that meet at sharp angles at the left and right corners. We found that, in most cases, the upper operator is sufficient to detect an eye. The lower operator is useful for a squinted eye. The operator for detecting the iris is just circle-shaped. The iris operator can be moved around inside the eyelid operators to search for the exact iris location. We used a straight line segment as the shape for detecting the mouth; it can be bent into a curve to match the different mouth shapes in images with facial expressions. This operator tries to detect the shade between the upper and lower lips, which is easier to model than the lip line. The base operator shapes for each of the above operators need to be adjusted, since each facial feature has a different intensity change.

Figure 15 shows detection results for face images from the FERET database. To limit the

search space, the face center region is estimated using an ellipse-shaped operator, and is marked by a white dotted ellipse having the matched ellipse size. The face region detection is biased because we are attempting to fit simple ellipses to faces without a precise model. Iris and eyelid detections are marked by the corresponding shapes.

We tested our algorithm on a group photo image that is more cluttered and has lower resolution than the FERET images. In this experiment we also used ellipse detection for face detection. The output is shown in Figure 16. The algorithm picked up the highest response from each detected face region; these responses are marked using the position and shape parameters corresponding to the maximum response. Every face and feature is detected correctly, except for the face on the upper right where the algorithm picks up responses due to thick eyebrows and the shadow of the nose.

Figure 17 shows eye detection results on an MPEG-7 dataset. In these images, the person wears glasses, and the image acquisition conditions are worse than in the previous images: the face is rotated or shaded. As mentioned previously, we observe that the detection is robust and accurate despite unfavorable camera angle and illumination. Since we use the operator for irises as well as the operator for eyelids, the glasses do not give rise to false detections.

C. Application to contour tracking in a video sequence

We have applied this method only to cases where we have a model of the shape to be detected. Even if an exact model is not available, an operator constructed on-line from data can be used to search for similar shapes in the same image or in different images (e.g., in video sequences).

It was shown in Section III that the shape operator can be constructed on-line by using any detected edge contour. In a video sequence, changes in an object boundary between frames can provide valuable information in estimating the motion and tracking the object. If the motion between frames is small, we can approximate the motion of the boundary as a translation. Once any portion of the object boundary contour is detected, a shape operator corresponding to this contour is constructed, and this operator searches for the same shape in the next frame. Figure 18 illustrates an example of the shape operator applied to feature-based motion estimation. Figures 18(a),(b) show a frame of the Yosemite sequence and the detected edge contours. The set of operators constructed, and the computed motion of the contours, is shown in Figures 18(c),(d). Since our detection scheme localizes and discriminates shapes accurately, feature correspondence and motion computation are potential applications.

VIII. CONCLUSION

We have presented an approach to detecting two-dimensional shapes. The scheme for shape detection has been designed from the lowest level of edge detection to guarantee optimal detection of shapes and accurate estimation of geometric parameters. It has been observed that the discontinuity of the DODE function at the center overcomes the performance limitation of the DOG operator and achieves more accurate localization without sacrificing detection performance. Another strength of our method comes from our use of a global operator, which leads to more averaging of uncertainties. This two-dimensional convolution covering various geometric parameters requires a large amount of computation, which can be reduced by multi-resolution approaches or by a random sampling scheme. By combining low-level edge detection and mid-level edge grouping, we can manipulate sensitivity to certain geometric parameters as well as to noise, by simply adjusting the shape of the operator without using any threshold. In the examples provided, detection is carried out with predetermined shape models. The shape operator can also be constructed automatically using arbitrary boundary contour segments, and can be applied to a wider range of problems, including accurate feature correspondence in video sequences. Moreover, once we have found the approximate geometric shape, a more accurate outline of an object can be computed by parameterizing the contour and searching for the maximum. For example, we constructed a family of curves using a Fourier basis, computed the corresponding shape operators, and searched for the best set of coefficients by employing simulated annealing. Figure 19 shows a structured shape (a), outlines approximated by straight lines (b), and the computed contours (c). This work is a part of our work on the shape and contour tracking problem, which is under way. There has been work involving contour fitting using parametric curves; however, our shape filtering scheme is unique, in that there is no loss of edge information from generic edge detection and we can tune the operator shape to achieve very accurate fitting. The simplicity of this approach provides comprehensive performance characterization which can be useful to higher-level vision problems.

ACKNOWLEDGMENTS

We thank the Heinrich Hertz Institute of Germany for providing the MPEG-7 content set S4.

REFERENCES

- [1] A. Arcese, P.H. Mengert, and E. Trombini, "Image Detection Through Bipolar Correlation," *IEEE Transactions on Information Theory*, Vol. 16, pp.534-541, 1970.
- [2] E. Argyle, "Techniques for Edge Detection," *Proceedings of the IEEE* Vol.59, pp. 285-287, 1971.
- [3] D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, Vol. 13, pp 111-122, 1981.
- [4] J. Ben-Arie and K.R. Rao, "A Novel Approach for Template Matching by Non-orthogonal Image Expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.3, pp. 71-84, 1993.
- [5] J. Ben-Arie and K.R. Rao, "Optimal Edge-Detection Using Expansion Matching and Restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, pp. 1169-1182, 1994.
- [6] J. Canny, "Finding Edges and Lines in Images," MIT AI TR-720, 1983.
- [7] J. Canny, "A Computational Approach to Edg seems to play a crucial rolee Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, pp.679-698, 1986.
- [8] R. Chellappa, X. Zhang, P. Burlina, C.L. Lin, Q. Zheng, L.S. Davis, and A. Rosenfeld, "An Integrated System for Site Model Supported Monitoring of Transportation Activities in Aerial Images," *Proceedings, DARPA Image Understanding Workshop*, pp.275-304, 1996.
- [9] G.R. Cooper and C.D. McGillem, "Probabilistic Methods of Signal and System Analysis," Oxford University Press, 1999.
- [10] D. Keren, D.B. Cooper, and J. Subrahmonia, "Describing Complicated Objects by Implicit Polynomials," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, pp. 38-53, 1994.
- [11] G. P. Lepage, "VEGAS: An Adaptive Multidimensional Integration Program," Publication CLNS-80/447, Cornell University, 1980.
- [12] D.G. Lowe, "Three-dimensional Object Recognition from Single Two-dimensional Images," *Artificial Intelligence*, Vol. 31, 355-395, 1987.
- [13] H. Moon, R. Chellappa, and A. Rosenfeld, "Performance Analysis of a Simple Vehicle Detection Algorithm," *Image and Vision Computing*, Vol. 20, pp. 1-13, 2002.
- [14] D. Mumford, S.M. Kosslyn, L.A. Hillger, and R.J. Hernstein, "Discriminating Figure from Ground: The Role of Edge Detection and Region Growing," *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 84, pp. 7354-7358, 1987.
- [15] A. Rosenfeld, "A Nonlinear Edge Detection Technique," *Proceedings of the IEEE*, Vol. 58, pp. 814-816, 1970.
- [16] A. Rosenfeld and M. Thurston, "Edge and Curve Detection for Visual Scene Analysis," *IEEE Transactions on Computers*, Vol. 20, pp. 562-569, 1971.
- [17] V. Ramesh and R.M. Haralick, "Performance Characterization of Edge Operators," *Proceedings, DARPA Image Understanding Workshop*, pp. 1071-1079, 1993.
- [18] A. Rosenfeld and A.C. Kak, "Digital Picture Processing," 2nd edition, Academic Press, 1982.
- [19] K. Shanmugam, F.M. Dickey, and J.A. Green, "An Optimal Frequency Domain Filter for Edge Detection in Digital Pictures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 1, pp. 37-49, 1979.

- [20] S. Sirohey and A. Rosenfeld, "Eye Detection in a Face Image Using Linear and Nonlinear Filters," *Pattern Recognition*, Vol. 34, pp. 1367-1391, 2001.
- [21] J. Subrahmonia, D.B. Cooper, and D. Keren, "Practical Reliable Bayesian Recognition of 2D and 3D Objects Using Implicit Polynomials and Algebraic Invariants," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, pp. 505-519, 1996.
- [22] G. Vosselman and R.M. Haralick, "Performance Analysis of Line and Circle Fitting in Digital Images," *Proceedings, ECCV Workshop on Performance Characteristics of Vision Algorithms*, 1996.



Fig. 16. Face detection and facial feature detection in a group photo



Fig. 17. Eye detection under unfavorable imaging conditions

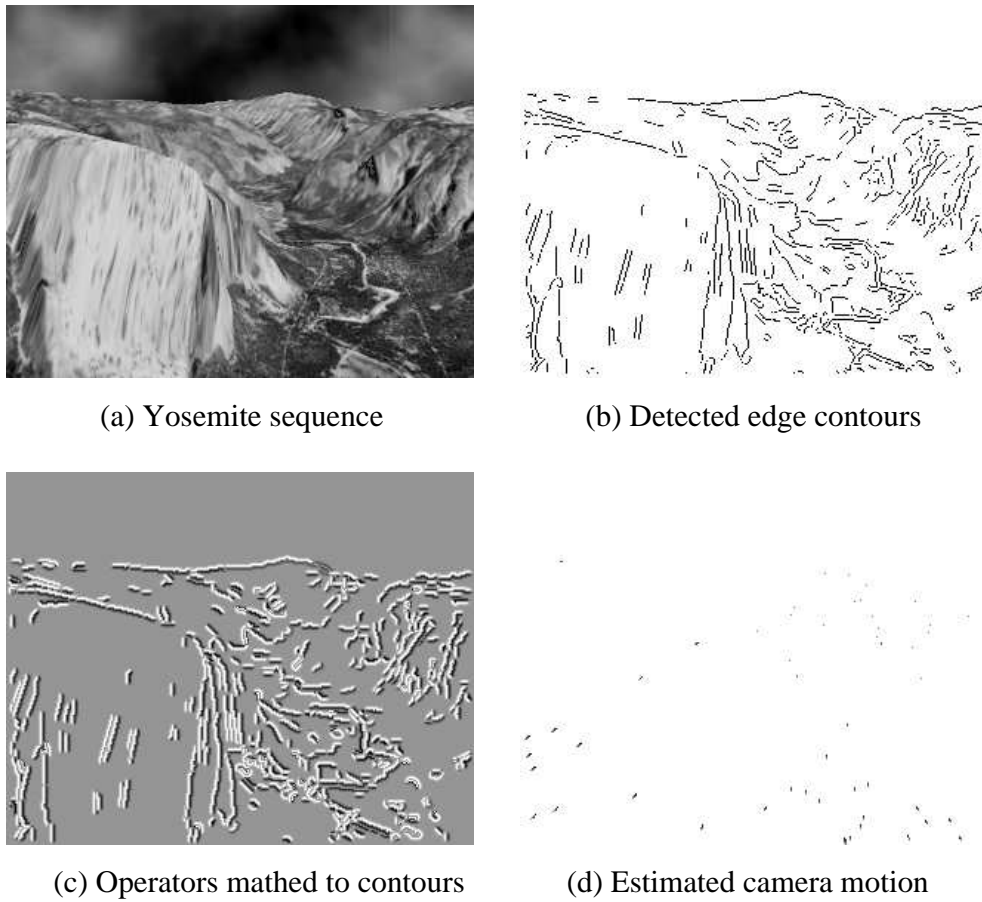


Fig. 18. Camera motion computation using shape detection

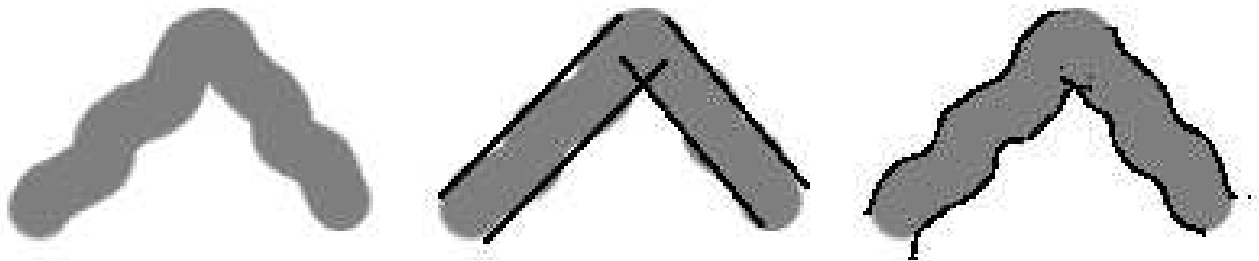


Fig. 19. An arbitrary shape and fitted outline. Given an arbitrary shape (left), an approximate structure of the shape is extracted (middle). More detailed boundary shape is computed using simple parameterization of contours and stochastic search (right).