

# Convex Set Reconstruction Using Prior Shape Information\*

JERRY L. PRINCE

*Image Analysis and Communications Laboratory, Department of Electrical and Computer Engineering, The Johns Hopkins University,  
Baltimore, Maryland 21218*

AND

ALAN S. WILLSKY

*Laboratory for Information and Decision Systems, Department of Electrical Engineering and Computer Science, Massachusetts Institute of  
Technology, Cambridge Massachusetts 02139*

Received September 8, 1989; accepted December 13, 1990

---

In this paper we present several algorithms for reconstructing 2D convex sets given support line measurements for which the angles are known precisely but the lateral displacements are noisy. We extend the algorithms given in a previous paper by explicitly incorporating prior information about the shape of the objects to be reconstructed. We develop the Scale-Invariant algorithms, which incorporate prior shape information by defining prior probabilities on support vectors, where a *support vector* is a vector formed from the lateral displacements of a particular set of support lines of an object. We also develop the Ellipse-Based algorithms, which either assume or jointly estimate the parameters of an ellipse, given prior distributions that favor ellipses. In order to relate the support vector prior probability to the expected shape of an object we develop a vector decomposition called the Size/Shape/Shift decomposition, which helps to provide insight into the detailed geometric relationship between support vectors and 2D convex objects. We then use the maximum a posteriori criterion to determine the specific form of the support vector estimator. The computations involve a quadratic programming optimization stage, which is used to determine one component of the decomposition, and either a line search or a conjugate gradient stage, which is used to determine the remaining components. The performance of the algorithms is demonstrated using simulated support line measurements of an ellipse. © 1991 Academic Press, Inc.

---

## 1. INTRODUCTION

This research was motivated by the problem of object reconstruction in computed tomography (CT) [2]. In CT, one makes measurements of the integrals of an object property (e.g., X-ray density) along various straight lines; the full set of line integrals for varying lateral dis-

placements at a particular angle yields a projection. The usual goal of CT is to reconstruct an image of the object given a set of measured projections obtained over a range of angles. We have found in other research [3, 4], however, that in highly noisy and limited-angle or sparse-angle situations, it is beneficial to utilize an estimate of the convex hull of the object in order to help reduce artifacts that would otherwise appear in the reconstructed imagery. This paper presents several methods, which incorporate certain prior geometric information, to aid in obtaining convex hull estimates from a collection of noisy and possibly incomplete support line measurements.

As shown in Fig. 1, an ideal projection contains information about the position of the two support lines that just graze opposite sides of the object. Given a set of such support lines, measured at many different angles, one may determine a convex polygon which circumscribes the object by intersecting all of the half-planes defined by the measurements. In the limit, as the angular spacing between projections goes to zero, one may determine precisely the convex hull of the object. This type of sensor information and reconstruction problem is also of interest in several other applications, including tactile sensing [5, 6], silhouette imaging [7, 8], robot vision [9], and chemical component analysis [10, 11].

When the projections are noisy, such as in the case of low-dose CT, and when the support lines are measured by estimating their positions from the projections, the lateral position of each measured support line will also be noisy. In this case, the set of measured lines may be inconsistent—that is, taken together, there may be no set  $S$  that has all of the measured lines as support lines. In such cases, the consistency property of support lines developed in [1] together with a precise description of the noise properties may be used to make a maximum likeli-

---

\* This research was supported by the National Science Foundation Grant ECS-87-00903 and the U.S. Army Research Office Grant DAAL03-86-K-0171. In addition, the work of the first author was partially supported by a U.S. Army Research Office Fellowship.

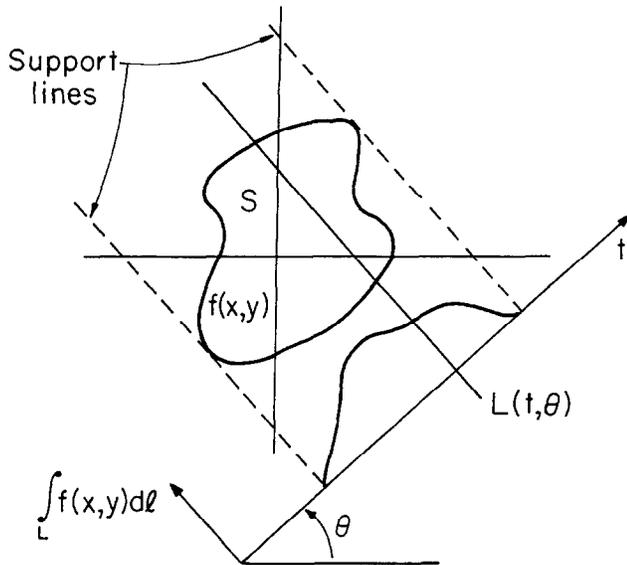


FIG. 1. The geometry of computed tomography.

hood (ML) estimate of the true collection of support lines. From this an estimate of the convex hull of the object may be determined by intersecting the resulting set of half-planes.

In this paper, we develop insights into the geometry of support lines and into the specification of prior probabilities on support lines that reflect particular prior information about the shape of the objects of interest. The major contributions are twofold: (1) the development of the Size/Shape/Shift (SSS) decomposition of support vectors, and (2) the development of algorithms for convex shape estimation that utilize prior shape information. The emphasis in this paper is on circular and elliptical shapes, but the formation is general and may be used to derive estimators for convex sets described by other prior shape information.

The fundamental problem considered in this paper is a special problem of reconstructing shape from probing [11–14], a subject considered to be within the larger area of computational geometry [15]. In our work—and this is in contrast to the usual assumptions in computational geometry—the fact that measurements are noisy and possibly incomplete is treated as a fundamental part of the problem formulation. This leads to estimation theoretic solutions which often require prior knowledge to guarantee unique solutions. In contrast to the usual goal of computational geometry, that of examining the computational complexity and data storage requirements of various algorithms, our focus is on the specific methods required to utilize noise statistics and prior knowledge optimally. Because the constraints that describe consistent sets of support lines are linear (see Section 2) and

because of the Gaussian noise models, the central optimization algorithm, which estimates object shape, is a quadratic program (QP). However, because of the requirements arising from our specification of prior knowledge (described in Section 3), the central QP is repeated iteratively to optimally estimate additional parameters describing the size and position, and in some cases the eccentricity and orientation, of an object.

The paper is organized as follows. In Section 2, we define the concept of the support vector and review the fundamental consistency conditions. Section 3 describes a particular decomposition of support vectors and determines expressions relating support vectors to the size, shape, area, position, and circumference of the objects corresponding to the support vectors. In Section 4, we develop the Scale-Invariant algorithms, which are based on a maximum a posteriori formulation with prior probabilities that favor circular objects, and the Ellipse-Based algorithms, which use prior distributions that favor ellipses. Section 5 presents experimental results, using simulated support line measurements of an ellipse, of the various estimation algorithms for different noise levels and geometric arrangements of the measurements. Finally, in Section 6 we summarize the key concepts in the paper and propose further research topics in this area.

## 2. SUPPORT LINE CONSTRAINTS

Figure 2 shows the support line  $L_S(\theta)$  of a set  $S$ . It is the line orthogonal to the unit normal  $\omega = [\cos\theta \ \sin\theta]^T$  that just “grazes”  $S$  in the positive  $\omega$  direction. The quantity  $h(\theta)$  is the value of the largest possible projection of any point in  $S$  onto the  $\omega$ -axis. One can see that  $S$  lies completely in a particular one of the two half-planes determined by  $L_S(\theta)$ . We may now define the above

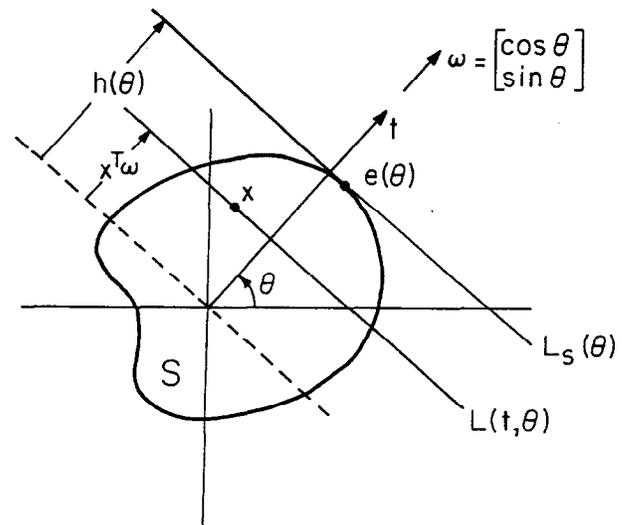


FIG. 2. The geometry of support lines.

quantities precisely. The *support line* at angle  $\theta$  for the closed and bounded 2D set  $S$  is given by

$$L_S(\theta) = \{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{x}^T \boldsymbol{\omega} = h(\theta)\}, \quad (1)$$

where

$$h(\theta) = \sup_{\mathbf{x} \in S} \mathbf{x}^T \boldsymbol{\omega}. \quad (2)$$

The function  $h(\theta)$  is called the *support function* of the set  $S$ ; for any particular value of  $\theta$  we call  $h(\theta)$  the *support value* at angle  $\theta$ . In this paper we consider a finite number  $M$  of angles  $\theta_i = 2\pi(i-1)/M$ ,  $i = 1, \dots, M$ , spaced evenly over  $[0, 2\pi)$ , and associated sets of lines  $L_i$ , orthogonal to the corresponding unit vector  $\boldsymbol{\omega}_i = [\cos \theta_i \sin \theta_i]^T$  and with lateral displacement  $h_i$ :

$$L_i = \{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{x}^T \boldsymbol{\omega}_i = h_i\}. \quad (3)$$

The vector made by organizing the  $M$  lateral displacement values of the  $M$  lines under consideration as  $\mathbf{h} = [h_1 h_2 \dots h_M]^T$  is called a *support vector* if the lines  $L_i$ , for  $i = 1, \dots, M$ , are support lines for some set  $S \subset \mathbb{R}^2$  [1]. The elements of a support vector are called *support values* and satisfy  $h_i = h(\theta_i)$ , where  $h(\theta)$  is a support function. In [1] we showed that a vector  $\mathbf{h} \in \mathbb{R}^M$  ( $M \geq 5$ ) is a support vector if and only if

$$\mathbf{h}^T \mathbf{C} \leq [0, \dots, 0], \quad (4)$$

where  $\mathbf{C}$  is an  $M$  by  $M$  matrix given by

$$\mathbf{C} = \begin{bmatrix} 1 & -k & 0 & & & -k \\ -k & 1 & -k & \cdots & \cdots & 0 \\ 0 & -k & 1 & & & \vdots \\ \vdots & 0 & -k & \cdots & \cdots & 0 \\ 0 & \vdots & & & & -k \\ -k & 0 & 0 & & & 1 \end{bmatrix} \quad (5)$$

and  $k = 1/(2 \cos(2\pi/M))$ . It follows that the convex polyhedral cone given by

$$\mathcal{C} = \{\mathbf{h} \in \mathbb{R}^M \mid \mathbf{h}^T \mathbf{C} \leq [0 \dots 0]\} \quad (6)$$

consists of all  $M$ -dimensional support vectors. We call  $\mathcal{C}$  the *support cone*.

Now suppose one obtains a support vector measurement  $\mathbf{y} \in \mathbb{R}^M$  such that  $\mathbf{y}$  does not satisfy  $\mathbf{y}^T \mathbf{C} \leq \mathbf{0}$  and is therefore not a support vector. In the algorithms described here and in [1], we estimate a convex set from

this measurement by first estimating a consistent support vector  $\hat{\mathbf{h}}$  from the measurements and then reconstructing the set estimate by intersecting the half-planes defined by  $\hat{\mathbf{h}}$ . A very basic algorithm for obtaining  $\hat{\mathbf{h}}$ , called the *Closest algorithm* [1], finds the support vector in  $\mathcal{C}$  that is closest to the measured vector, using the usual Euclidean metric. This estimator also yields the ML estimate, under the assumption that the lateral position of each support line is observed in independent zero-mean additive Gaussian noise. The closest estimate is given by

$$\hat{\mathbf{h}}_C = \underset{\mathbf{h}^T \mathbf{C} \leq \mathbf{0}}{\operatorname{argmin}} (\mathbf{y} - \mathbf{h})^T (\mathbf{y} - \mathbf{h}), \quad (7)$$

which is a quadratic program since the objective function is quadratic and the constraints are linear.

Any estimate produced by the closest algorithm must lie on the boundary of the support cone and by geometric arguments must have at least one degenerate support line (see [1] and Section 3.4). This leads to estimated sets that tend to have rather sharp boundaries. Because of this behavior and because of our desire to estimate objects that tend to have smooth boundaries, we were led to the research in this paper on the use of prior shape information or, equivalently, additional penalty terms in our optimization criterion that favor objects with smooth boundaries. Since our algorithms estimate support vectors rather than the 2D objects directly, we must describe shape information using quantities derived from the support vectors themselves. In the following section we relate several algebraic functions of support vectors to specific geometric properties of objects. These relations then lead us to the reconstruction algorithms developed in Section 4.

### 3. OBJECT AND SUPPORT CONE GEOMETRY

Given a support vector  $\mathbf{h}$ , there is, in general, an infinitely large family of sets, each of which has  $\mathbf{h}$  as its support vector. The largest of these sets, which is uniquely determined by  $\mathbf{h}$ , is the convex polygonal set

$$S_B = \{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{x}^T [\boldsymbol{\omega}_1 \boldsymbol{\omega}_2 \dots \boldsymbol{\omega}_M] \leq [h_1 h_2 \dots h_M]\}, \quad (8)$$

which we call the *basic object* of support vector  $\mathbf{h}$ . As shown in [1],  $S_B$  may also be written as

$$S_B = \operatorname{hul}(v_1, v_2, \dots, v_M), \quad (9)$$

where  $v_i$  is the intersection of the two support lines  $L_i$  and  $L_{i+1}$ , where  $i+1$  is taken modulo  $M$ , and  $\operatorname{hul}(\cdot)$  gives the convex hull of a set of points. We call the  $\{v_i\}$  *vertex points* rather than vertices since they need not be distinct. A basic object, together with its support lines and vertex points, is depicted in Fig. 3.

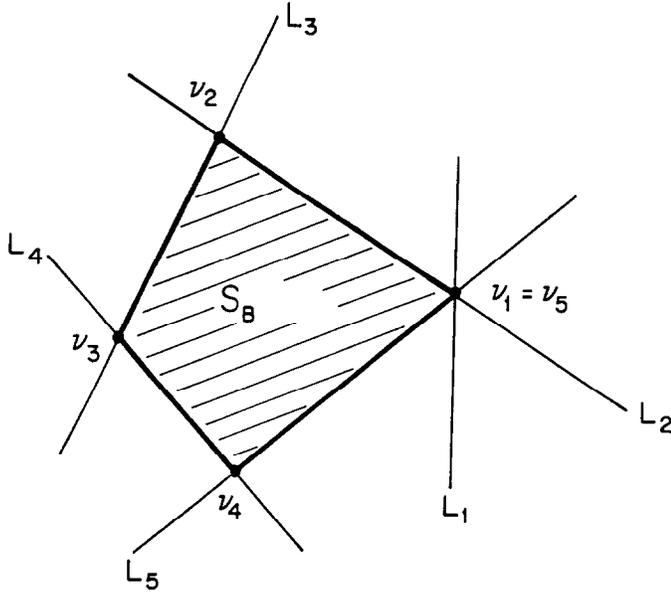


FIG. 3. Five support lines, their basic object, and vertex points.

If  $\mathbf{h}$  is an estimated support vector, then  $S_B$  is the reconstructed object, and the shape of  $S_B$  is of primary concern in this paper. A support vector  $\mathbf{h}$  may be decomposed into quantities related to the size, shape, and position of its basic object, and the area and radii of curvature may also be derived as functions of  $\mathbf{h}$ . We proceed in this section to define the SSS decomposition of a support vector, relate its components to object geometry, and then determine expressions for the radius of curvature and area of  $S_B$ .

### 3.1. Position

As shown in [1], the matrix  $\mathbf{C}$  is singular, and a basis for the null space  $\mathcal{N}$  is found to be

$$\begin{aligned} \mathbf{n}_1 &= [1 \quad \cos \theta_0 \quad \cos 2\theta_0 \quad \cdots \quad \cos(M-1)\theta_0]^T \\ \mathbf{n}_2 &= [0 \quad \sin \theta_0 \quad \sin 2\theta_0 \quad \cdots \quad \sin(M-1)\theta_0]^T, \end{aligned} \quad (10)$$

where  $\theta_0 = 2\pi/M$ . The geometrical consequence of  $\mathbf{C}$  being singular is that the support cone  $\mathcal{C}$  is not a proper cone; i.e., there is a linear subspace (of dimension 2) contained entirely in  $\mathcal{C}$ . Therefore, the support cone is composed of the Cartesian product of a proper cone,

$$\mathcal{C}_p = \{\mathbf{h} \in \mathcal{C} \mid \mathbf{h}^T \mathbf{n}_1 = 0, \mathbf{h}^T \mathbf{n}_2 = 0\}, \quad (11)$$

and  $\mathcal{N}$ , the null space of  $\mathbf{C}$ . Accordingly, any support vector may be written as the sum of two orthogonal components,  $\mathbf{h}_p$  and  $\mathbf{h}_n$ , as

$$\mathbf{h} = \mathbf{h}_p + \mathbf{h}_n, \quad (12)$$

where  $\mathbf{h}_p \in \mathcal{C}_p$  and  $\mathbf{h}_n \in \mathcal{N}$ .

In [1] we showed that adding a null vector  $\mathbf{h}_n$  to a support vector causes the basic object to shift its position in the plane. To see this we note that any null vector may be written as  $\mathbf{h}_n = \mathbf{N}\mathbf{v}$ , where  $\mathbf{N} = [\mathbf{n}_1 \mathbf{n}_2]$  and  $\mathbf{v}$  is a two-dimensional vector. Now suppose that  $\mathbf{w}$  is an element of  $S_B$ ; then  $\mathbf{w}$  satisfies  $\mathbf{h} \geq \mathbf{N}\mathbf{w}$  (see (8)). Then we must have that  $\mathbf{h} + \mathbf{h}_n = \mathbf{N}(\mathbf{w} + \mathbf{v})$ , which implies that  $\mathbf{w} + \mathbf{v}$  is an element of the basic object of  $\mathbf{h} + \mathbf{h}_n$ ; i.e., the new basic object is just a shifted version of  $S_B$ .

### 3.2. Size

Since  $\mathbf{h}_p$  lies in a proper cone  $\mathcal{C}_p$ , it may be written as

$$\mathbf{h}_p = t\mathbf{q}, \quad (13)$$

where  $t$  is a nonnegative scalar, and  $\mathbf{q} \in \mathcal{C}_p$  satisfies  $\mathbf{q}^T \mathbf{e} = \mathbf{M}$ , where  $\mathbf{e} = [1, \dots, 1]^T$ . In a later section,  $t$  is shown to be proportional to the circumference of the basic object, hence the use of the word "size" in reference to  $t$ . The vector  $\mathbf{q}$  contains the information remaining after size and position have been removed, hence the use of the word "shape" for  $\mathbf{q}$ .

To show how (13) arises, consider the truncated cone

$$\mathcal{T} = \{\mathbf{h} \in \mathbb{R}^M \mid \mathbf{h}^T \mathbf{A} \leq 0, \mathbf{h}^T \mathbf{b} \leq \mu\},$$

where  $\mathbf{A} = [\mathbf{C} : -\mathbf{N} : \mathbf{N}]$ ,  $\mu = M[(\cos \theta_0)^{-1} - 1]$ , and  $\mathbf{b} = -\sum_{j=1}^{M+4} \mathbf{a}_j$ , where  $\mathbf{a}_j$  denotes the  $j$ th column of  $\mathbf{A}$ . The polyhedron  $\mathcal{T}$  does in fact truncate the proper cone  $\mathcal{C}_p$  since it has no ray points; that is, it has no points  $\mathbf{h} \neq \mathbf{0}$  such that  $\mathbf{h}^T [\mathbf{A} : \mathbf{b}] \leq 0$ . To see this note that for nonzero  $\mathbf{h} \in \mathbb{R}^M$  we must have  $\mathbf{h}^T \mathbf{A} \neq \mathbf{0}$  since the rows of  $\mathbf{A}$  are linearly independent. Then for  $\mathbf{h}$  also satisfying  $\mathbf{h}^T \mathbf{A} \leq 0$  we must have  $\mathbf{h}^T \mathbf{a}_j < 0$  for some  $j$ . Hence,  $\mathbf{h}^T \mathbf{b} > 0$ . Now consider a vector  $\mathbf{h}_p \neq 0$  in  $\mathcal{C}_p$ . From the above argument we may conclude that  $\mathbf{h}_p^T \mathbf{b} > 0$  and therefore, for some  $\xi > 0$ , that  $\xi \mathbf{h}_p^T \mathbf{b} = \mu$  since  $\mu > 0$ . This expression simplifies to  $\xi[(\cos \theta_0)^{-1} - 1] \mathbf{h}_p^T \mathbf{e} = \mu$  and again to  $\xi \mathbf{h}_p^T \mathbf{e} = M$ , which, for  $t \equiv 1/\xi$ , gives the desired result. Note that if  $\mathbf{h}_p = \mathbf{0}$ , it may always be written as  $\mathbf{h}_p = t\mathbf{q}$ , where  $t = 0$ .

### 3.3. Size/Shape/Shift Decomposition

The results of the two previous sections give rise to the ternary decomposition, which we refer to as the SSS decomposition, stated as follows.

**THEOREM 3.1 (Size/Shape/Shift Decomposition).** *A support vector  $\mathbf{h}$  may be written as*

$$\mathbf{h} = t\mathbf{q} + \mathbf{h}_n, \quad (14)$$

where  $t \geq 0$ ,  $\mathbf{q} \in \mathcal{C}_p$ ,  $\mathbf{q}^T \mathbf{e} = M$ , and  $\mathbf{h}_n \in \mathcal{N}$ .

*Proof.* Follows from the above discussion. ■

Given an arbitrary vector  $\mathbf{h}$  we may find the components of the SSS decomposition as follows [3]:

$$\mathbf{h}_n = \frac{2}{M} \mathbf{N} \mathbf{N}^T \mathbf{h} \quad (15)$$

$$t = \frac{1}{M} \mathbf{h}^T \mathbf{e} \quad (16)$$

$$\mathbf{h}_p = \mathbf{h} - \mathbf{h}_n \quad (17)$$

$$\mathbf{q} = \mathbf{h}_p / t, \quad t \neq 0. \quad (18)$$

As with polar coordinates the  $\mathbf{0}$  support vector, with  $t = 0$ , has a nonunique SSS decomposition, since any  $\mathbf{q}$  may be chosen to satisfy the equality in (14).

### 3.4. Radius of Curvature

Now we review the idea of discrete radius of curvature, previously presented in [1], to characterize the smoothness of the boundaries of basic objects. When the boundary of  $S_B$ , as shown in Fig. 4, is traced along the  $i$ th face from  $v_{i-1}$  to  $v_i$ , the outward unit normal to the boundary changes in angle by  $\theta_0 = \theta_i - \theta_{i-1}$  over a distance  $f_i$ , the length of the  $i$ th face. In analogy to the usual radius of curvature, which is defined as the rate of change of arc length with respect to the angle the unit normal makes to the x-axis, we define the  $i$ th discrete radius of curvature as

$$r_i \equiv \frac{f_i}{\theta_0}. \quad (19)$$

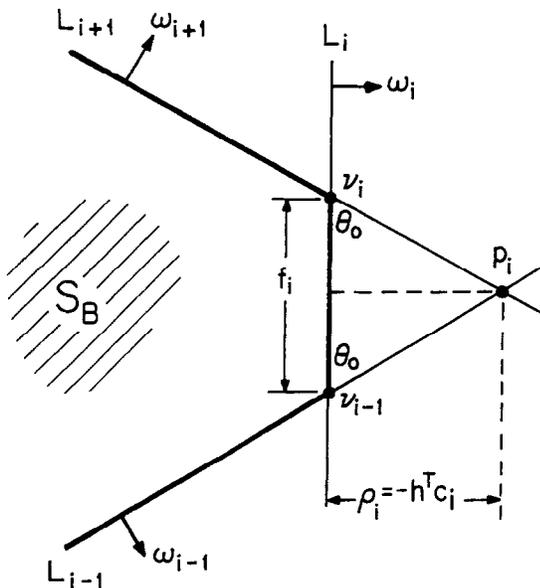


FIG. 4. Three support lines and a face of  $S_B$ .

It can be shown from the geometry [1] that the distance in Fig. 4 from  $p_i$  to  $L_i$  is  $\rho_i = -\mathbf{h}^T \mathbf{c}_i$ , where  $\mathbf{c}_i$  is the  $i$ th column of  $\mathbf{C}$ . Then, by simple trigonometry, we have

$$f_i = \frac{2\rho_i}{\tan \theta_0} \quad (20)$$

and therefore

$$r_i = \frac{-2\mathbf{h}^T \mathbf{c}_i}{\theta_0 \tan \theta_0} = \frac{-2t\mathbf{q}^T \mathbf{c}_i}{\theta_0 \tan \theta_0}. \quad (21)$$

This expression for radius of curvature gives us a description of the shape of the boundary as a function of the support vector. Sharp corners are produced when  $r_i$  is small; if  $r_i = 0$ , then support line  $L_i$  is degenerate. In Section 4, we use this type of boundary curvature information in the specification of a prior probability that gives higher probability to support vectors whose basic objects do not have small radii of curvature. It is possible, however, to use other types of boundary-specific information, derived from this expression, and incorporate them into alternate estimation algorithms.

### 3.5. Circumference

We now derive an expression for the circumference  $P(\mathbf{h})$  of a basic object as a function of its support vector  $\mathbf{h}$ . In particular, we show that the quantity  $t = \mathbf{h}^T \mathbf{e} / M$  of the SSS decomposition is proportional to  $P(\mathbf{h})$ .

The circumference of a basic object is given by the sum of the face lengths  $f_i$  as

$$P(\mathbf{h}) = \sum_{i=1}^M f_i = \sum_{i=1}^M \frac{2\rho_i}{\tan \theta_0}. \quad (22)$$

Using the definitions that  $\rho_i = -\mathbf{h}^T \mathbf{c}_i$  and  $\mathbf{e} = [1, \dots, 1]^T$  we have

$$\begin{aligned} P(\mathbf{h}) &= \frac{-2}{\tan \theta_0} \mathbf{h}^T \mathbf{C} \mathbf{e} = \frac{2}{\tan \theta_0} \mathbf{h}^T \boldsymbol{\gamma} \mathbf{e} \\ &= \frac{2\gamma}{\tan \theta_0} \mathbf{h}^T \mathbf{e} = \frac{2M\gamma}{\tan \theta_0} t, \end{aligned} \quad (23)$$

where  $\gamma$  is the sum of the elements of any row of  $\mathbf{C}$ . This expression shows the proportionality of the circumference to the  $t$ -coordinate and justifies the use of  $t$  to characterize the size of a basic object.

### 3.6. Area

We now derive an exact expression for  $S(\mathbf{h})$ , the area of the basic object of support vector  $\mathbf{h}$ . Without loss of generality let us assume that  $\mathbf{h} \in \mathcal{C}_p$ . Now consider Fig.

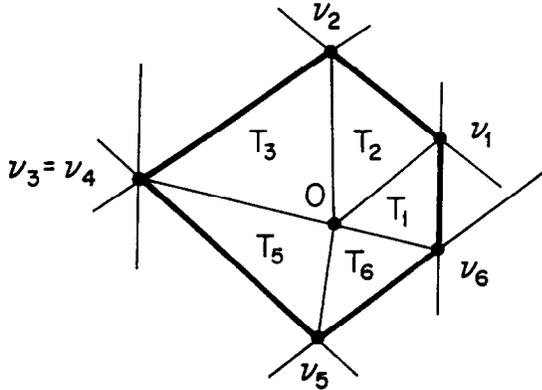


FIG. 5. Triangulated basic object for  $M = 6$ .

5, which depicts a basic object for  $M = 6$ . The area of  $S_B$  is the sum of the areas of the  $M$  triangles  $T_i = \text{hul}(O, \nu_{i-1}, \nu_i)$ ,  $i = 1, \dots, M$ , where  $O$  stands for the origin. Hence, by denoting the area of triangle  $T_i$  by  $A_i$ , we have

$$S(\mathbf{h}) = \sum_{i=1}^M A_i = \sum_{i=1}^M \frac{1}{2} f_i h_i, \quad (24)$$

using the “one-half base times height” rule for the area of a triangle. After substitution for  $f_i$  using Eq. (20) and some manipulation, we find that

$$S(\mathbf{h}) = \frac{-1}{\tan \theta_0} \mathbf{h}^T \mathbf{C} \mathbf{h}. \quad (25)$$

Several implicit assumptions were used in developing this expression:

1. The origin is contained in the basic object for a proper support vector,
2. if  $\mathbf{h} \in \mathcal{C}_p$  then  $\mathbf{h} \geq \mathbf{0}$ , and
3. the triangular regions  $T_i$  with nonzero area do not overlap.

The truth of these assumptions is shown in [3], and with these facts in hand it is apparent that the expression for  $S(\mathbf{h})$  in (25) is correct. Note that  $S(\mathbf{h})$  remains unchanged when a null vector is added to  $\mathbf{h}$  so that this formula is valid even when  $\mathbf{h}$  is not a proper support vector.

#### 4. ESTIMATION ALGORITHMS

The algorithms developed in this section use prior shape information to assist in the estimation of a consistent support vector from noisy and possibly incomplete observations. The observation model is

$$\mathbf{y} = \mathbf{S} \mathbf{h} + \mathbf{n}, \quad (26)$$

where  $\mathbf{h}$  is the true support vector,  $\mathbf{n}$  is a zero-mean jointly Gaussian vector with covariance  $\sigma^2 \mathbf{I}$ , and  $\mathbf{S}$  is a  $K \times M$  matrix ( $K < M$ ), which “selects” the elements of  $\mathbf{h}$  that are observed. For example, suppose that the first  $K$  elements of  $\mathbf{h}$  are observed; then  $\mathbf{S} = [\mathbf{I}_K \mathbf{0}]$ , where  $\mathbf{I}_K$  is the  $K \times K$  identity matrix. Different elements of  $\mathbf{h}$  may be selected by permuting the columns of  $\mathbf{S}$ , and fewer elements of  $\mathbf{h}$  may be selected by choosing a smaller  $K$ .

When  $K = M$ ,  $\mathbf{S} = \mathbf{I}_M$ , and this is a full data problem; otherwise, it is an incomplete data problem in which we are trying to estimate a support vector that is of higher dimension than the available measurements. In the incomplete case, the closest algorithm, described in Section 2, may not have a unique solution. The scale-invariant algorithms developed in the following section use additional prior knowledge in a maximum a posteriori (MAP) formulation to force a unique solution. In the full data problem, the additional prior knowledge balances the effects of noise with the prior knowledge using the MAP criterion. We present only the full data cases in the main body of the text, leaving the details of the incomplete data algorithm to Appendix A.

The observations model in (26) represents a particular model that may or may not be appropriate for a given application. In CT, where one estimates support values from measured projections, the underlying noise distribution of projections and the support value estimation algorithm together determine the distribution of  $\mathbf{n}$ . In a tactile-sensing application, unknown backlash in gears or, if a remote operation, errors in geopositioning may determine the distribution of  $\mathbf{n}$ . The model chosen here is traditional in the field of random signal analysis and therefore leads to results that may be compared to other well-known results.

One limitation of the model in (26) is that  $\mathbf{h}$  is assumed to be composed of lateral positions of support lines taken from fixed angles evenly distributed over  $[0, 2\pi)$ . In fact, many applications do not inherently satisfy this constraint. For example, most current CT machines have fan-beam projections, so that each projection yields two support values for lines that are not parallel, and the collection of angles does not yield the evenly distributed set required by our approach. Rebinning the projections [2] is one option which will yield the required projections. Another option is to derive a new  $\mathbf{C}$  matrix for non-evenly spaced angles together with a new SSS decomposition for the implied support cone. This accomplished, the estimation approaches presented in this section are still applicable, although some terms will have to be modified to reflect the new set of observation angles.

The shape estimation algorithms presented in this section are based on a prior assessment of the expected shape of objects; we concentrate on objects that are most likely to be either circles or ellipses. Applications for

which this might be relevant include medical CT imaging of the head and torso, where the convex hulls are roughly elliptical. Another relevant application is CT imaging of fuel in rocket casings, where the convex hull is often exactly circular. Moreover, the overall Bayesian approach is relevant to many situations where other prior geometric knowledge is available and can be incorporated as prior probabilities over the support cone.

#### 4.1. Scale-Invariant Algorithms

The closest algorithm can be augmented using probabilistic prior knowledge about the distribution of support vectors in the support cone to yield a MAP solution. In this section, we develop estimation algorithms, called *scale-invariant* algorithms, using the MAP criterion. The class of prior probabilities on support vectors that we consider are scale-invariant in the sense that basic objects with precisely the same shape but of different sizes have exactly the same prior probability. These probabilities are *shift-invariant* as well, so that the probability does not depend on the position of the object in the plane, either. Therefore, the scale-invariant algorithms use only prior knowledge about shape; the position and size are also estimated, but using only information that is provided in the measurements (in a maximum likelihood fashion).

The first prior that we describe, called the Scale-Invariant Close-Min (SICM) prior, favors objects whose boundaries are smooth in a mini-max sense. That is, those objects whose smallest discrete radius of curvature is large have higher prior probability. This prior is given by

$$p_{\text{SICM}}(\mathbf{h}) = \frac{1}{z} \exp \left( \frac{1}{\tau} \frac{1}{\theta_0 \tan \theta_0} \times \min\{-\mathbf{q}^T \mathbf{c}_1, \dots, -\mathbf{q}^T \mathbf{c}_M\} \right), \quad (27)$$

where  $\mathbf{c}_i$ ,  $i = 1, \dots, M$ , are the columns of the matrix  $\mathbf{C}$ . Here, the set over which the minimum is taken has elements which are proportional to the  $M$  radii of curvature of the basic object of  $\mathbf{h}$  (see Eq. (21)). If the smallest radius of curvature is zero, for example, then the exponential is equal to one. Since we restrict  $\tau$  to be positive, support vectors with larger minimum radii of curvature are more probable, the circle being most probable. Note that the probability is defined in terms of the shape vector component,  $\mathbf{q}$ , of the support vector, which implies that support vectors whose basic objects have the same shape, but are of different sizes or are at different locations, are equally probable. In this prior, as in the two that follow,  $\tau$  is a positive parameter of the probability and  $z$  is a constant chosen so that the density integrates to one.

The second prior we define, called the Scale-Invariant Closest (SIC) prior, favors objects that are more circular in their overall shape and is given by

$$p_{\text{SIC}}(\mathbf{h}) = \frac{1}{z} \exp \left( -\frac{1}{\tau} (\mathbf{q} - \mathbf{e})^T (\mathbf{q} - \mathbf{e}) \right). \quad (28)$$

When  $\mathbf{q} = \mathbf{e}$ , this density takes on its largest possible value, and since  $\mathbf{e}$  is the support vector of the unit circle, this density also favors circular objects. As  $\mathbf{q}$  moves away from  $\mathbf{e}$ , the probability grows smaller exponentially, depending only on the squared distance between  $\mathbf{q}$  and  $\mathbf{e}$ . Again, since  $\mathbf{q}$  is a shape vector, the density does not depend on the size or position of the basic object of  $\mathbf{h}$ .

Finally, we define the Scale-Invariant Max-Area (SIMA) prior, which favors objects whose area-to-squared-circumference ratio is large. This prior is given by

$$p_{\text{SIMA}}(\mathbf{h}) = \frac{1}{z} \exp \left( \frac{1}{\tau} S(\mathbf{q}) \right), \quad (29)$$

where  $S(\mathbf{q})$  is the area of the basic object of  $\mathbf{h}$ , as defined in (24). This density exploits the fact that the circle has the largest ratio of area to squared circumference over all simple plane curves.  $S(\mathbf{q})$  is zero only in the degenerate case where the basic object is a line segment, and it becomes larger as  $\mathbf{q}$  approaches  $\mathbf{e}$ . The probability increases correspondingly since the exponential is increasing. The squared circumference is implicit since  $S(\mathbf{q})$  uses  $\mathbf{q}$ , rather than  $\mathbf{h}$ , and thus already accounts for circumference normalization. Also, and as in the previous priors, the size and position do not enter into the probability.

The three scale-invariant algorithms, which solve the MAP problem for these priors, are similar in several respects. We assume the complete data case in this section, so that each observed support value  $y_i$  is given by  $y_i = h_i + n_i$ , where  $h_i$  is the true support value and  $n_i$  is a zero-mean Gaussian random variable with variance  $\sigma^2$ , which is independent of all  $h_i$  and all  $n_j$ ,  $j \neq i$ . Therefore, the observation vector  $\mathbf{y}$  is a jointly Gaussian random vector with conditional density

$$p(\mathbf{y} | \mathbf{h}) = |2\pi\sigma^2\mathbf{I}_M|^{-1/2} \exp \left[ -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{h})^T (\mathbf{y} - \mathbf{h}) \right]. \quad (30)$$

A scale-invariant (SI) estimate maximizes the a posteriori density, or equivalently the logarithm of the a posteriori density, and is given by

$$\hat{\mathbf{h}} = \underset{\mathbf{h}^T \mathbf{C} \leq \mathbf{0}}{\operatorname{argmax}} \ln p(\mathbf{h} | \mathbf{y}) \quad (31)$$

$$= \operatorname{argmax}_{\mathbf{h}^T \mathbf{C} \leq 0} \ln p(\mathbf{y} | \mathbf{h}) + \ln p_{\text{SI}}(\mathbf{h}) \quad (32)$$

$$= \operatorname{argmax}_{\mathbf{h}^T \mathbf{C} \leq 0} -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{h})^T (\mathbf{y} - \mathbf{h}) + \ln p_{\text{SI}}(\mathbf{h}), \quad (33)$$

where  $p_{\text{SI}}(\mathbf{h})$  stands for any one of the three scale-invariant priors.

Defining  $\mathbf{y}_n$  as the projection of  $\mathbf{y}$  onto the null space of  $\mathbf{C}$  and  $\mathbf{y}_p$  as  $\mathbf{y} - \mathbf{y}_n$ , we have  $(\mathbf{y} - \mathbf{h})^T (\mathbf{y} - \mathbf{h}) = (\mathbf{y}_p - \mathbf{h}_p)^T (\mathbf{y}_p - \mathbf{h}_p) + (\mathbf{y}_n - \mathbf{h}_n)^T (\mathbf{y}_n - \mathbf{h}_n)$ . Then, since  $\mathbf{h}_p = t\mathbf{q}$  and  $p_{\text{SI}}(\mathbf{h}) = p_{\text{SI}}(\mathbf{q})$ , the MAP solution is defined as  $\hat{\mathbf{h}} = t\hat{\mathbf{q}} + \hat{\mathbf{h}}_n$  such that  $t$ ,  $\hat{\mathbf{q}}$ , and  $\hat{\mathbf{h}}_n$  jointly maximize

$$f_{\text{SI}}(t, \mathbf{q}, \mathbf{h}_n) = -\frac{1}{2\sigma^2} (\mathbf{y}_p - t\mathbf{q})^T (\mathbf{y}_p - t\mathbf{q}) - \frac{1}{2\sigma^2} (\mathbf{y}_n - \mathbf{h}_n)^T (\mathbf{y}_n - \mathbf{h}_n) + \ln p_{\text{SI}}(\mathbf{q}) \quad (34)$$

subject to the support vector consistency conditions. Since  $p_{\text{SI}}(\mathbf{q})$  does not depend on  $t$  or  $\mathbf{h}_n$ , and since  $t\mathbf{q}$  is orthogonal to  $\mathbf{h}_n$ , the optimal shift component is

$$\hat{\mathbf{h}}_n = \mathbf{y}_n. \quad (35)$$

The remaining optimization problem is

$$\begin{aligned} \underset{t, \mathbf{q}}{\text{maximize}} \quad & F_{\text{SI}}(t, \mathbf{q}) = -\frac{1}{2\sigma^2} (\mathbf{y}_p - t\mathbf{q})^T (\mathbf{y}_p - t\mathbf{q}) \\ & + \ln p_{\text{SI}}(\mathbf{q}) \end{aligned} \quad (36)$$

subject to  $t \geq 0$ ,  $\mathbf{q}^T \mathbf{e} = M$ ,  $\mathbf{q}^T \mathbf{N} = 0$ , and  $\mathbf{q}^T \mathbf{C} \leq 0$ ,

which has, unfortunately, a fourth-order objective function due to the  $t\mathbf{q}$  product. For fixed  $t$ , however, this is a QP, which may be solved efficiently. Therefore, our approach to solving this optimization problem is to conduct a line search in  $t$ , solving a QP at each step in  $t$ . Since the optimal  $\mathbf{q}$  is found for each  $t$ , the globally optimal solution may be found using an exact line search in  $t$ . With minor modification, the scale-invariant algorithms can be adapted for the cases of incomplete data, as described in Appendix A.

#### 4.2. Ellipse-Based Algorithms

This section considers the use of prior knowledge related to the eccentricity and orientation of the true object. We develop three ellipse-based algorithms, each of which uses this type of prior knowledge in a different way. The first algorithm finds the closest ellipse to an observed support vector, so that under the Gaussian noise model given above, this algorithm may be used to find the ML estimates of the parameters of an ellipse, assuming that the true object is *exactly* an ellipse. The

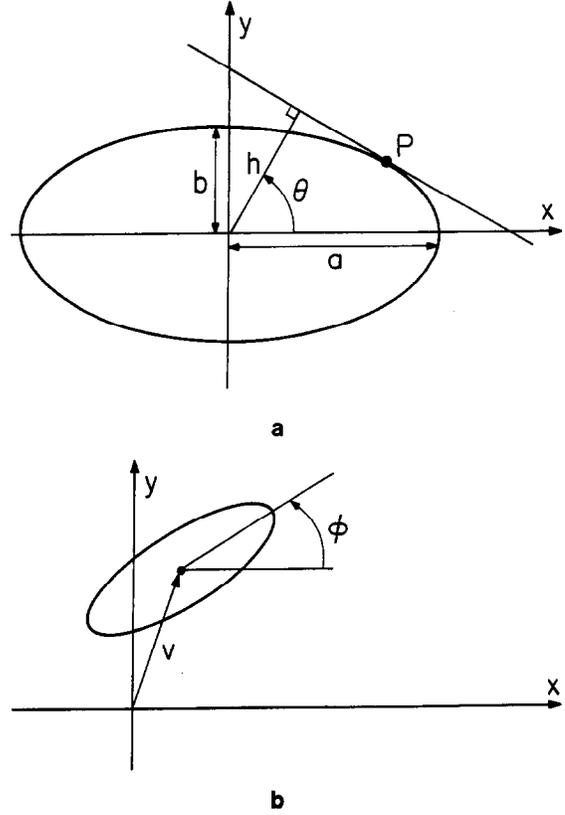


FIG. 6. Ellipse parameters.

second algorithm assumes that the true object is nearly elliptic in shape and that we have some prior knowledge of the true ellipse parameters. Finally, the third algorithm assumes that the true object is nearly elliptic in shape, but that we do *not* have any prior knowledge of the true ellipse parameters. This algorithm estimates *jointly* the support vector and the ellipse parameters.

The key to the ellipse-based algorithms is the characterization of the support vector of an ellipse  $\mathbf{h}(\mathbf{v}, t, \varepsilon, \phi)$ , where  $\mathbf{v}$  is the 2D vector indicating the center position of the ellipse,  $t$  is the size of the ellipse (proportional to its circumference),  $\varepsilon$  is the eccentricity, and  $\phi$  is the orientation as shown in Fig. 6. A formula for the  $i$ th element of  $\mathbf{h}(\mathbf{v}, t, \varepsilon, \phi)$  is given by (see Appendix B)

$$h_i = \frac{Mt\sqrt{(1/(1-\varepsilon^2))\cos^2(\theta_i - \phi) + \sin^2(\theta_i - \phi)}}{\sum_{j=1}^M \sqrt{(1/(1-\varepsilon^2))\cos^2(\theta_j - \phi) + \sin^2(\theta_j - \phi)}} + [\cos \theta_i \sin \theta_i] \mathbf{v}. \quad (37)$$

The ML estimate of the ellipse parameters, assuming the Gaussian noise model described in the previous section, is the solution of the optimization problem

$$\underset{\mathbf{v}, t, \varepsilon, \phi}{\text{minimize}} \quad \|\mathbf{y} - \mathbf{h}(\mathbf{v}, t, \varepsilon, \phi)\|^2 \quad (38)$$

subject to  $t \geq 0$  and  $0 \leq \varepsilon \leq 1$ ,

where  $\|\mathbf{x}\|^2$  denotes  $\mathbf{x}^T\mathbf{x}$ , leading to the Closest Ellipse (CE) estimate  $\hat{\mathbf{h}}_{\text{CE}} = \mathbf{h}(\hat{\mathbf{v}}, \hat{t}, \hat{\varepsilon}, \hat{\phi})$ . Because of the complicated form of  $\mathbf{h}(\mathbf{v}, t, \varepsilon, \phi)$ , this problem is highly nonlinear and must be solved iteratively. Since the objective function is differentiable and the interesting solutions do not lie on the constraints, we use the conjugate gradient method to solve this problem. The complete details are given in [3]. Rossi and Willsky [16] have presented ML methods to estimate ellipse parameters from a set of object projections; our ellipse parameter estimates, although ML estimates, are inherently different since the observations are support lines. If one starts with projection information, then Rossi and Willsky's approach should be used to give ML estimates of the ellipse parameters since it is optimal for all the measurements.

If we knew a priori that the true object shape is nearly elliptic with eccentricity  $\bar{\varepsilon}$  and orientation  $\bar{\phi}$ , then our reconstruction algorithm should favor shapes that resemble the ellipse with these parameters. This knowledge can be included using the Ellipse-Based Scale-Invariant Closest (ESIC) prior probability defined as

$$p_{\text{ESIC}}(\mathbf{q}) = \frac{1}{z} \exp \left\{ -\frac{1}{\tau} \|\mathbf{q} - \mathbf{h}(\mathbf{0}, 1, \bar{\varepsilon}, \bar{\phi})\|^2 \right\}. \quad (39)$$

This prior is identical in form to and is in fact a generalization of the SI closest prior in that the largest probabilities are concentrated around the ellipse whose support shape vector is  $\mathbf{h}(\mathbf{0}, 1, \bar{\varepsilon}, \bar{\phi})$  rather than the shape vector  $\mathbf{e}$ , which is the support vector of a circle. The resulting MAP estimation algorithm, as in the scale-invariant algorithms, performs a line search in  $t$ , solving a QP at each stage, until the jointly optimum  $(\hat{t}, \hat{\mathbf{q}})$  pair is determined. The optimum shift vector is  $\hat{\mathbf{h}}_n = \mathbf{y}_n$ .

Next we extend these two algorithms to estimate jointly a support vector and ellipse parameters. Here, the prior knowledge we utilize is that the true support vector is likely to be near to the shape of an ellipse—i.e., it has the form of (39)—but the specific ellipse parameters  $\bar{\varepsilon}$  and  $\bar{\phi}$  of the prior distribution are unknown. This problem may be written formally as

$$\begin{aligned} \underset{\mathbf{h}, \mathbf{v}, t, \varepsilon, \phi}{\text{minimize}} \quad & \alpha \|\mathbf{h} - \mathbf{y}\|^2 + (1 - \alpha) \|\mathbf{h} - \mathbf{h}(\mathbf{v}, t, \varepsilon, \phi)\|^2, \\ & 0 < \alpha \leq 1, \\ \text{subject to} \quad & t \geq 0, 0 \leq \varepsilon \leq 1, \text{ and } \mathbf{h}^T \mathbf{C} \leq \mathbf{0}, \end{aligned} \quad (40)$$

leading to the Joint Ellipse estimate  $\hat{\mathbf{h}}_{\text{JE}}$ . We note that if  $\alpha = 1$  then the objective function of (40) is independent of the ellipse parameters  $\mathbf{v}, t, \varepsilon$ , and  $\phi$ , and the optimum  $\mathbf{h}$  is found using the closest algorithm. Alternatively, as  $\alpha \rightarrow$

0, the optimum ellipse parameters approach their ML estimates, and the optimum  $\mathbf{h}$  approaches the corresponding ellipse support vector. These two extremes provide some insight to the solution in the general case when  $0 < \alpha < 1$ . For example, the optimum  $\mathbf{h}$  cannot be closer to  $\mathbf{y}$  than  $\hat{\mathbf{h}}_{\text{C}}$ , the closest estimate, since then it would be infeasible; also, it cannot be farther away from  $\mathbf{y}$  than  $\mathbf{h}(\hat{\mathbf{v}}_{\text{ML}}, \hat{t}_{\text{ML}}, \hat{\varepsilon}_{\text{ML}}, \hat{\phi}_{\text{ML}})$ .

Our numerical solution of (40) involves a type of coordinate descent algorithm which alternates between finding the optimum  $\mathbf{h}$  for fixed ellipse parameters and then finding the optimum ellipse parameters for fixed  $\mathbf{h}$ . For fixed ellipse parameters, the optimum  $\mathbf{h}$  is found using a quadratic program; for fixed  $\mathbf{h}$ , the optimum ellipse parameters are found using the CE algorithm described above. Each iteration causes the objective function to decrease (or at least not increase) so that convergence of the algorithm is guaranteed. However, as in all coordinate descent algorithms, the algorithm need not converge to either a globally or locally optimal solution. A detailed look at the convergence properties of this algorithm and of alternate computational approaches is an open problem.

## 5. RESULTS

Let us first make a few comments about computational complexity. The algorithms were coded in Fortran, and those incorporating a QP—all except the closest ellipse algorithm—uses the standard QP code ZQPCVX, an active set method due to M. J. D. Powell [17, 18]. The first step in ZQPCVX is to perform a Cholesky factorization of the constraint matrix, which involves approximately  $N^3/6$  multiplications [19]. Although a quadratic program finishes in finite time, the number of iterations required to do so is unknown (although upper bounded) [20]. However, from [17] we may conclude that each iteration will require  $O(M^2)$  multiplications, where  $M$  is the dimension of a support vector. Also, because the Cholesky decomposition is not sparse, matrix storage requirements are  $O(M^2)$ .

A further consideration is the convergence time of the line search over  $t$  in the SI algorithms. Here, for each value of  $t$ , a QP is performed, until the line search converges to the optimal  $\hat{t}$ . The line search was performed using the golden section method [21], choosing a tolerance parameter that led to adequate performance at reasonable computation times. The time and storage requirements place a constraint on the largest dimension support vectors we may consider. The largest for which we present experimental results in this paper is  $M = 60$ ; we have done experiments, however, with dimension  $M = 120$ . For  $M = 60$ , the code took between 1 and 5 min

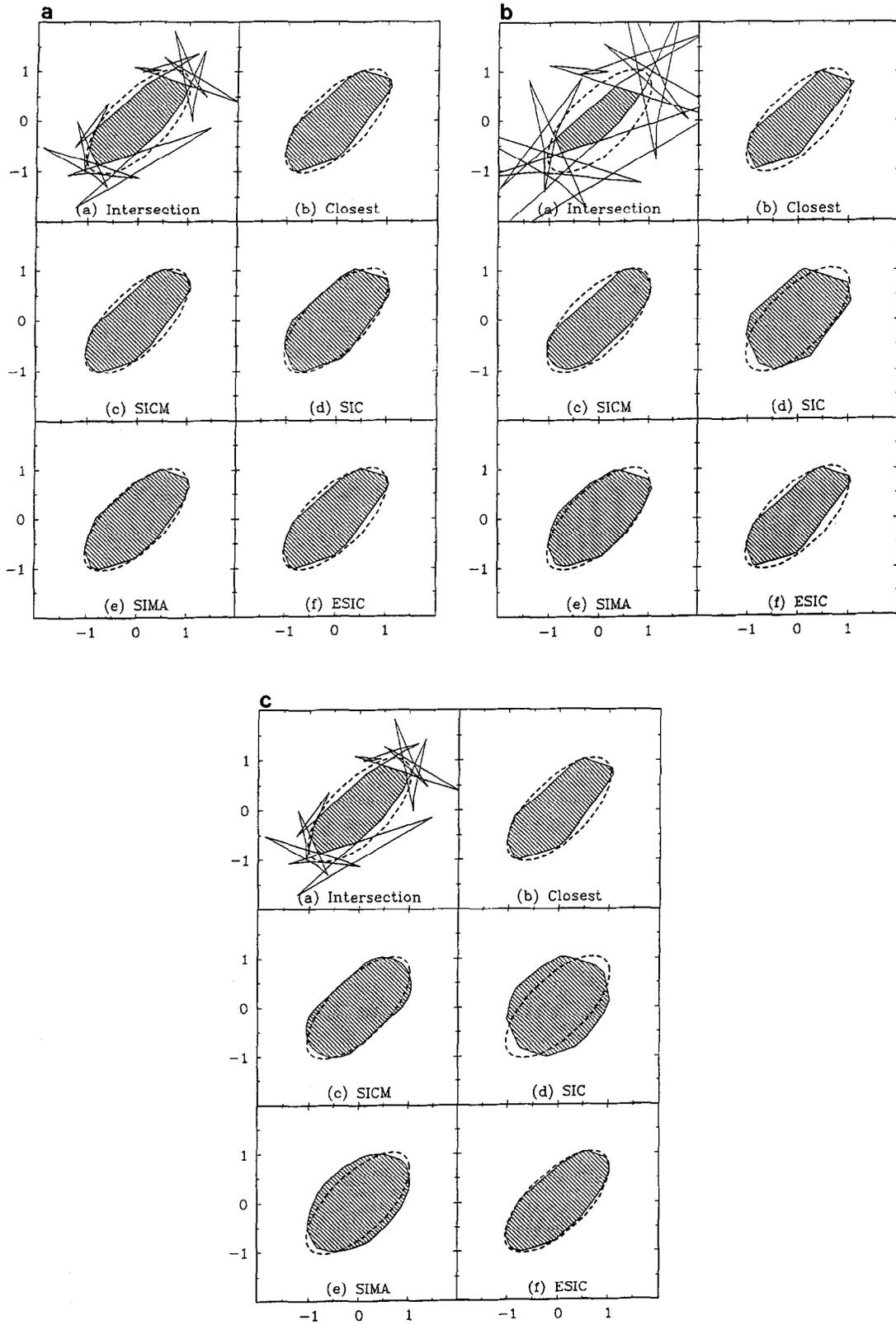


FIG. 7. Comparison of support vector estimation algorithms for  $M = 30$  and (a)  $\sigma = 0.1, \tau = 0.1$ ; (b)  $\sigma = 0.2, \tau = 0.1$ ; and (c)  $\sigma = 0.1, \tau = 0.01$ .

on a Data General MV10000 (comparable in speed to a VAX11/780).

Figure 7 compares six different reconstruction methods applied to a complete set of measurements for different noise powers and probability parameters. The true ellipse, characterized by parameters  $t = 1.0$ ,  $\epsilon = 0.9$ ,  $\phi = 45.0$ , and  $\mathbf{v} = (0.0, 0.0)$ , is shown using dashed lines in each of the panels of each figure; the support line measurements are indicated by the solid lines in panel (a) of each figure. The intersection method, which simply intersects the half-planes defined by each of the support line measurements, produces the shaded result in panel (a). The closest algorithm, yielding an ML solution, obtains the result shown in panel (b). Panels (c)–(f) show the result of the four scale-invariant algorithms (including the ellipse-based closest algorithm) described in Section 4.

It is clear that a great deal of improvement over the set intersection method and the ML method may be achieved using the scale-invariant algorithms, even in cases such as these, where the noise variance is large. The SICM estimate shown in panel (c) clearly shows that prior knowledge related to the boundary of the object can markedly improve the reconstruction. In particular, the estimate has a boundary without any of the sharp bends that are evident in the ML solution of panel (b). The SIC result shown in panel (d) has an overall shape that is more circular than any of the other estimates; this reflects the prior probability, which favors objects that are close to the shape vector  $\mathbf{e}$ . The SIMA result shown in panel (e) has few long straight boundary sections, which reflects the tendency to increase the ratio of area to circumference squared, a property of the SIMA prior. The ESIC result (f), which uses the correct  $\epsilon$  and  $\phi$ , shows a better estimate than the SIC result, which is a special case of ESIC with  $\epsilon = 0$ .

The overall difference between Fig. 7a, which has a noise standard deviation of  $\sigma = 0.1$ , and Fig. 7b, which has a noise standard deviation of  $\sigma = 0.2$ , reflects the fact that as the noise power is increased, the MAP estimator is driven to apply greater weight to the prior knowledge. Therefore, the estimated basic objects in Fig. 7b reflect more strongly the properties of their respective prior densities. The overall difference between Fig. 7a, which has a density parameter of  $\tau = 0.1$ , and Fig. 7c, which has a density parameter of  $\tau = 0.01$ , reflects the fact that one can specify prior densities that have a highly concentrated region of high probability (as in Fig. 7c), again making the estimated basic objects reflect the prior information more strongly.

Figure 8 shows the result of applying (a) the closest ellipse algorithm and (b) the joint ellipse algorithm to the measurements shown in panel (a) of Fig. 7b. In both panels, the dashed curve outlines the true ellipse. Panel (a) of

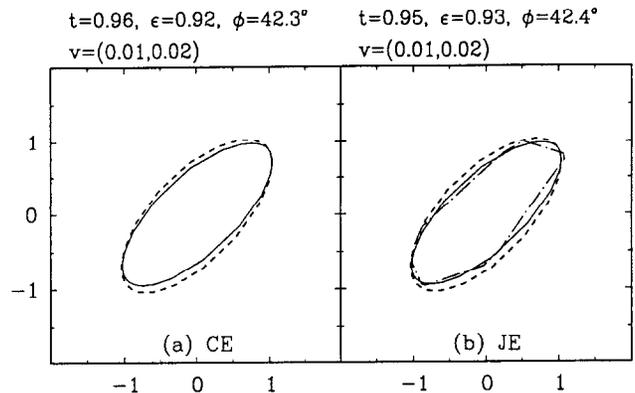


FIG. 8. Comparison of (a) the closest ellipse estimate and (b) the joint ellipse estimate for  $\alpha = 0.5$ .

Fig. 8 shows the CE estimate, solid line; the estimated ellipse parameters are shown above the panel. One can see that the estimated ellipse parameters are close to the true parameters given above. The JE result, shown in panel (b) of Fig. 8, shows the estimated ellipse, solid line, and the jointly estimated support vector, dashed/dotted curve. The parameters for the estimated ellipse are also shown above the panel; again, as in the CE estimate, the jointly estimated ellipse parameters are very close to the true ellipse parameters. What is remarkable about the JE object estimate is that it is very nearly as good as that of the ESIC algorithm shown in Fig. 7b, panel (f)—and the JE result did not require prior knowledge of the ellipse parameters. This is an important result since it is rare in applications to actually have such parameters available a priori.

Figures 9–11 show the results of applying the support vector estimation algorithms to two incomplete data cases. In the sparse-angle case, we took the 30 support

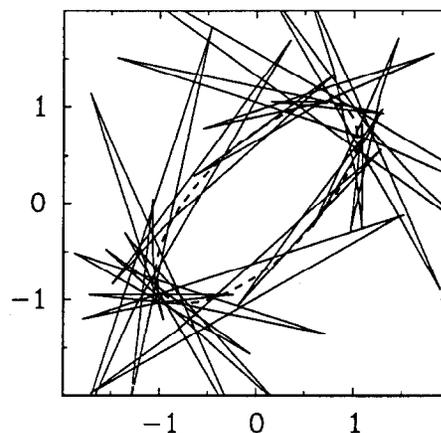


FIG. 9. Measurements and intersection estimate for incomplete data experiments where  $M = 60$  and  $\sigma = 0.1$ .

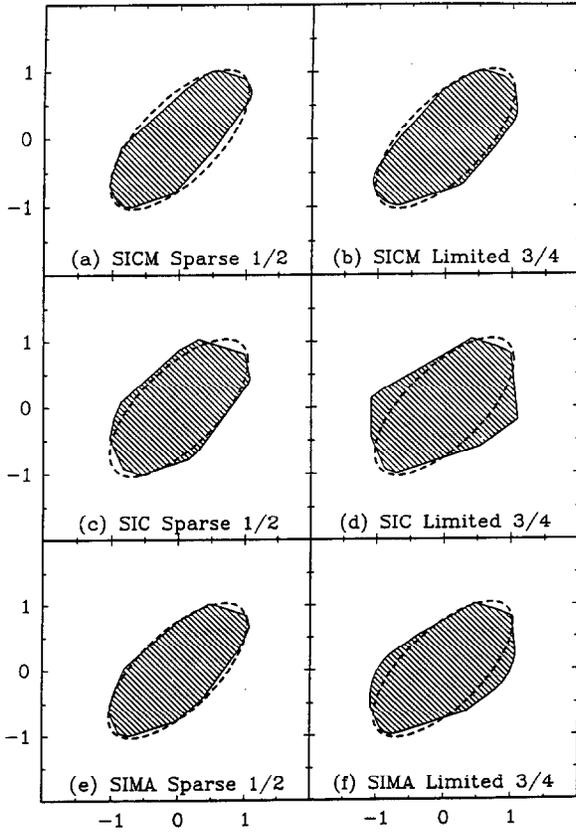


FIG. 10. Comparison of the SI support vector estimation algorithms for sparse-angle and limited-angle incomplete data measurements.

value measurements of Fig. 7a and estimated support vectors of dimension 60, essentially interpolating a support value between each observation. We label this “Sparse 1/2” since exactly  $\frac{1}{2}$  of the 60 estimated support values are observed. In the limited-angle case, our observations contain support lines whose angles are in the range of  $0-135^\circ$  and  $180-270^\circ$ . Therefore,  $\frac{3}{4}$  of the 60 support values are observed, leading to the label “Limited 3/4.” In all cases, the noise standard deviation is  $\sigma = 0.1$  and when the angles are identical, the actual measured values are equal to those shown in panel (a) of Fig. 7a. The full set of measurements and the true ellipse are shown in Fig. 9.

Figure 10 shows a comparison of the scale-invariant algorithms for the incomplete data experiments. The left-hand column shows the sparse-angle results and the right-hand column shows the limited-angle results. One should note that the sparse-angle cases look quite similar to the dimension 30 results shown in Fig. 7a. The only difference in these cases is that in this experiment 30 additional values are estimated to fill in between the observed values. With this additional freedom, the estimators are forced to rely more heavily upon prior knowl-

edge, which accounts for the fact that these estimates tend to resemble our prior expectations a bit more than those of Fig. 7a. The limited-angle experiments present a much greater challenge since the data are missing over a much greater angular range. The effect is to use prior knowledge heavily in this range, and since our prior knowledge in these scale-invariant algorithms is that the objects tend to be circular, the results are more circular than desired. One exception appears to be the SICM result, which actually gives a remarkable estimate in light of the amount of missing data and the level of noise. The reason for this is that the SICM prior awards higher probability to those objects that do not have small radii of curvature. Therefore, over the region of missing data, the best object is one that connects the available measurements without using any small radii, and this tends to create arcs of nearly constant radius over these regions. This property is not satisfied by estimates that extend outward sharply as in the corresponding SIC estimate or have abrupt transitions at the edges of the observed data as in the corresponding SIMA estimate.

Figure 11 shows a comparison of the ESIC and JE algorithms, applied to the same sparse-angle and limited-angle observations as those used in Fig. 10. The ESIC results, which again use the correct eccentricity and orientation, show a substantial improvement over the SI results of Fig. 10. This is not surprising, however, since here the probability is concentrated around the true elliptical shape and orientation. One would not expect such a good result given the wrong eccentricity and orientation. (An example of this type of behavior appears in [3].)

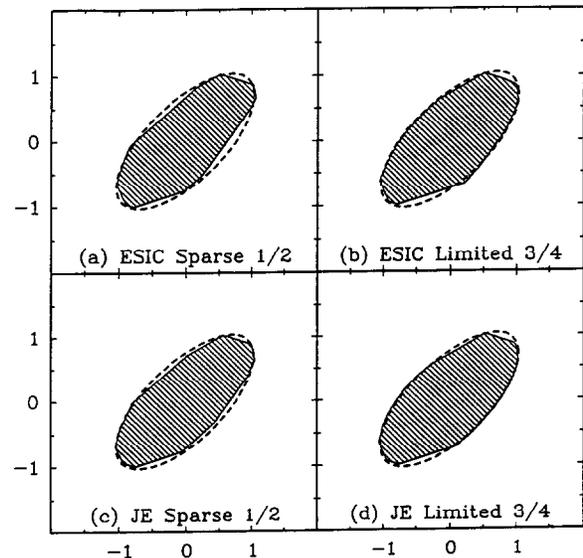


FIG. 11. Comparison of the ESIC and JE ellipse-based algorithms for incomplete data measurements.

Therefore, using the ESIC method, one risks the possibility that the assumed ellipse parameters in the ESIC prior are incorrect. The JE algorithm, on the other hand, does not suffer from this possibility since one jointly estimates the ellipse parameters; and yet, as demonstrated in panels (c) and (d) of Fig. 11, the results are still quite impressive—the ellipse is nearly perfectly reconstructed.

## 6. SUMMARY

Because of the systematic incorporation of prior knowledge, the performance of the set reconstruction methods described in this paper is better than that of the methods given in [1]. We expect this to be true in cases where the true objects are either nearly circular, or have large portions of nearly circular boundaries, or are nearly elliptic in shape. The incorporation of prior knowledge also allows us to solve sparse-angle and limited-angle incomplete data problems in which there are unobserved support values over possibly large angular ranges. In these cases, the missing support values are automatically interpolated by the algorithms using the available prior shape information. In CT, these geometric estimation algorithms may be used to estimate the convex hull of the object directly from the projections, and this estimate may be used in the reconstruction process as described in [22]. In cases where the observed angles are restricted and the observations are noisy, this procedure dramatically improves the quality of the reconstructed images over those obtained by convolution back-projection.

Geometric decomposition of support vectors is a step toward being able to develop alternate representations of prior shape knowledge. These representations, together with the fundamental support vector constraint and a full knowledge of the noise statistics, should allow one to develop estimation-theoretic algorithms that are suited to particular applications. Furthermore, extending these ideas to 3D opens up many interesting and important applications such as shape estimation from silhouettes and tactile sensing.

Many interesting theoretical and algorithmic questions remain in this area of research. For example, there is the question of performance: how well do these (and other future algorithms) perform? One can formulate several traditional measures based on the estimated parameters—the support vector and ellipse parameters, for example—but these are not measures that are related directly to the shape of the object. Another example is further decomposition—beyond just size, position, and shape—and, therefore, further parameterization of support vectors. A relatively simple further decomposition is orientation since this is simply a circular shift of the support vector, but other extensions are not as obvious. With respect to the algorithms, the QP and line search are

the workhorses of this paper, and they present a severe limitation on the workable size of future problems. A decomposition of the QP into parallel components or utilization of special symmetry properties of the constraint matrix is therefore an important future topic of research.

## APPENDIX A: INCOMPLETE DATA SI ALGORITHMS

In this Appendix we extend the results of Section 4.1 to the case where we have fewer than  $M$  (noisy) observations of a support vector but we desire an estimate of the full  $M$ -dimensional vector. In this case, the ML solution is not unique, and we must use a formulation that includes prior knowledge such as the scale-invariant MAP formulations. As we see, the main complication in this case is that the optimal shift vector cannot be solved independently of the size and shape components—but this problem is not too difficult to overcome. The resultant algorithms retain the line search over  $t$  and the embedded QP parts, as in the original SI algorithms, but they require pre- and postprocessing to account for the shift estimate. We refer to the algorithms developed in this appendix as the *Sparse Scale-Invariant* (SSI) algorithms.

The log likelihood of  $\mathbf{h}$  for the observations model of (26) is given by

$$l(\mathbf{h}) = -\frac{1}{2\sigma^2} (\tilde{\mathbf{y}} - \mathbf{S}\mathbf{h})^T (\tilde{\mathbf{y}} - \mathbf{S}\mathbf{h}) - \frac{1}{2} \ln |2\pi\sigma^2\mathbf{I}_k|, \quad (41)$$

which may be written as

$$l(\mathbf{h}) = -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{h})^T \mathbf{D} (\mathbf{y} - \mathbf{h}) - \frac{1}{2} \ln |2\pi\sigma^2\mathbf{I}_k|, \quad (42)$$

where

$$\mathbf{D} = \mathbf{S}^T \mathbf{S}, \quad (43)$$

and

$$\mathbf{y} = \mathbf{S}^T \tilde{\mathbf{y}}. \quad (44)$$

We may now form expressions for the SSI algorithms by adding the natural logarithm of the prior probability to  $l(\mathbf{h})$ . Since the SI priors depend only on the shape vector  $\mathbf{q}$  we may write a *generic* SSI problem as

$$\underset{\mathbf{h} \in \mathcal{G}}{\text{maximize}} -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{h})^T \mathbf{D} (\mathbf{y} - \mathbf{h}) + \ln p_{\text{SI}}(\mathbf{q}), \quad (45)$$

where we have dropped the constant term  $-\frac{1}{2} \ln |2\pi\sigma^2\mathbf{I}_k|$  and where  $p_{\text{SI}}(\mathbf{q})$  stands for either  $p_{\text{SICM}}(\mathbf{h})$ ,  $p_{\text{SIC}}(\mathbf{h})$ , or  $p_{\text{SIMA}}(\mathbf{h})$ , or even the ellipse-based prior  $p_{\text{ESIC}}(\mathbf{h})$ .

To solve (45) we must expand  $\mathbf{h}$  using the SSS decomposition so that  $\ln p_{\text{SI}}(\mathbf{q})$  may be combined with  $l(\mathbf{h})$ . Here is where the essential difference between the sparse case and the full-view case appears. In the sparse case we cannot completely separate the solution of  $\mathbf{h}_n$  from that of  $\mathbf{h}_p$  since they are now coupled through cross terms of the form  $\mathbf{h}_p^T \mathbf{D} \mathbf{h}_n$ . However, in what follows we show that the optimum shift vector may be calculated directly from the optimum size and shape vector, and we may use this knowledge to simplify the form of the SSI optimization problem.

To see how to find the optimum shift vector component, we focus on the expansion of  $(\mathbf{y} - \mathbf{h})^T \mathbf{D} (\mathbf{y} - \mathbf{h})$  using the SSS decomposition. First, we use the fact that  $\mathbf{h} = \mathbf{h}_p + \mathbf{h}_n$  and that  $\mathbf{h}_n = \mathbf{N} \mathbf{v}$  for some  $\mathbf{v}$  to make the manipulations

$$\begin{aligned} (\mathbf{y} - \mathbf{h})^T \mathbf{D} (\mathbf{y} - \mathbf{h}) &= \mathbf{y}^T \mathbf{D} \mathbf{y} - 2\mathbf{y}^T \mathbf{D} \mathbf{h} + \mathbf{h}^T \mathbf{D} \mathbf{h} \\ &= \mathbf{y}^T \mathbf{D} \mathbf{y} - 2\mathbf{y}^T \mathbf{D} \mathbf{h}_p - 2\mathbf{y}^T \mathbf{D} \mathbf{N} \mathbf{v} + \mathbf{h}_p^T \mathbf{D} \mathbf{h}_p \\ &\quad + 2\mathbf{h}_p^T \mathbf{D} \mathbf{N} \mathbf{v} + \mathbf{v}^T \mathbf{N}^T \mathbf{D} \mathbf{N} \mathbf{v}. \end{aligned}$$

Now, any SI objective function has a prior which does not depend on  $\mathbf{v}$ ; therefore we may determine the necessary conditions for  $\mathbf{v}$  to be a minimum by taking the (vector) partial derivative of (46) with respect to  $\mathbf{v}$  and setting it equal to zero. We get

$$-2\mathbf{N}^T \mathbf{D}^T \mathbf{y} + 2\mathbf{N}^T \mathbf{D}^T \mathbf{h}_p + 2\mathbf{N}^T \mathbf{D} \mathbf{N} \mathbf{v}^* = 0.$$

or

$$\mathbf{v}^* = (\mathbf{N}^T \mathbf{D} \mathbf{N})^{-1} \mathbf{N}^T \mathbf{D} (\mathbf{y} - \mathbf{h}_p). \quad (46)$$

Since for any choice of  $\mathbf{h}_p$ , (46) yields the optimum  $\mathbf{v}$ , we may substitute this expression back into (46) and simplify. After some algebra we find

$$\begin{aligned} (\mathbf{y} - \mathbf{h})^T \mathbf{D} (\mathbf{y} - \mathbf{h})|_{\mathbf{v}=\mathbf{v}^*} &= \mathbf{y}^T (\mathbf{D} - \mathbf{Q}) \mathbf{y} - 2\mathbf{y}^T (\mathbf{D} - \mathbf{Q}) \mathbf{h}_p \\ &\quad + \mathbf{h}_p^T (\mathbf{D} - \mathbf{Q}) \mathbf{h}_p, \end{aligned} \quad (47)$$

where

$$\mathbf{Q} = \mathbf{D} \mathbf{N} (\mathbf{N}^T \mathbf{D} \mathbf{N})^{-1} \mathbf{N}^T \mathbf{D}. \quad (48)$$

By making the substitution  $\mathbf{h}_p = t\mathbf{q}$  and adding the natural logarithm of the appropriate prior we obtain the SSI formulation

$$\begin{aligned} \underset{t, \mathbf{q}}{\text{maximize}} \quad & -\frac{t^2}{2\sigma^2} \mathbf{q}^T (\mathbf{D} - \mathbf{Q}) \mathbf{q} + \frac{t}{\sigma^2} \mathbf{y}^T (\mathbf{D} - \mathbf{Q}) \mathbf{q} \\ & + \ln p_{\text{SI}}(\mathbf{q}) \end{aligned} \quad (49)$$

$$\text{subject to } t \geq 0, \mathbf{q}^T \mathbf{e} = M, \mathbf{q}^T \mathbf{N} = 0, \text{ and } \mathbf{q}^T \mathbf{C} \leq 0. \quad (50)$$

The SSI optimization problem may be solved using a line search approach similar to that used for the SI algorithms of Section 4.1. We may see this by noting that in each case the optimization over  $\mathbf{q}$  given a fixed  $t$  is just a QP. Therefore, once the optimum size  $\hat{t}$  is found by searching the nonnegative real line, the optimum shape vector  $\hat{\mathbf{q}}$  is known. An additional step is required for the SSI algorithms, however: we must calculate the optimal shift vector  $\hat{\mathbf{h}}_n = \mathbf{N} \mathbf{v}^*$ , where  $\mathbf{v}^*$  is calculated from (46) with  $\mathbf{h}_p = \hat{t} \hat{\mathbf{q}}$ .

## APPENDIX B: THE SUPPORT FUNCTION OF AN ELLIPSE

Points  $(x, y)$  on the boundary of the ellipse shown in Fig. 6 satisfy the equation

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \quad (51)$$

where  $a$  and  $b$  are the  $x$  and  $y$  semiaxes of the ellipse. Since an ellipse is convex and has a continuously turning boundary normal, we know that each support line intersects the boundary at precisely one point—e.g., the point  $P$  in the figure. Therefore,  $P$  is on the line whose points satisfy

$$x \cos \theta + y \sin \theta = h, \quad (52)$$

where  $\theta$  is the angle (measured counterclockwise from the  $x$ -axis) of the unit outward normal to the ellipse boundary at the point  $P$  and  $h$  is the shortest distance from the origin to the tangent line at  $P$ . We recognize  $h$  to be the support distance at angle  $\theta$ . We seek in this Appendix an expression for  $h$  as a function of  $\theta$ .

The simplest approach begins by solving for the  $(x, y)$  pair that satisfies both (51) and (52). We have from (52) that

$$x^2 = \left( \frac{h - y \sin \theta}{\cos \theta} \right)^2,$$

which when substituted into (51) yields (after some manipulation)

$$\underbrace{\left( \frac{\sin^2 \theta}{a^2 \cos^2 \theta} + \frac{1}{b^2} \right)}_p y^2 + \underbrace{\left( \frac{-2h \sin \theta}{a^2 \cos^2 \theta} \right)}_q y + \underbrace{\left( \frac{h^2}{a^2 \cos^2 \theta} - 1 \right)}_r = 0. \quad (53)$$

The solution to this equation is given by

$$y = \frac{-q \pm \sqrt{q^2 - 4pr}}{2p}, \quad (54)$$

but we do not need to solve this to accomplish the goals of this section. Instead we observe that there are three possible outcomes: (1)  $y$  has two complex solutions, (2)  $y$  has two real solutions, and (3)  $y$  has a single real solution. The first possibility arises when the line  $L(h, \theta)$  does not intersect the ellipse, the second arises when the line intersects the ellipse at two points, and the third possibility occurs when the line intersects the ellipse at just one point. Only the third possibility causes the line to be a support line to the ellipse—that is the situation we require.

To ensure that the line  $L(h, \theta)$  intersects the ellipse at just one point we must have that (refer to Eq. (54))

$$q^2 - 4pr = 0, \quad (55)$$

which, when the required substitutions are made from (53), becomes

$$\frac{4h^2 \sin^2 \theta}{a^4 \cos^4 \theta} - 4 \left( \frac{\sin^2 \theta}{a^2 \cos^2 \theta} - \frac{1}{b} \right) \left( \frac{h^2}{a^2 \cos^2 \theta} - 1 \right) = 0.$$

After simplification, the above expression yields

$$h^2 = a^2 \cos^2 \theta + b^2 \sin^2 \theta,$$

from which we get

$$h(\theta) = \sqrt{a^2 \cos^2 \theta + b^2 \sin^2 \theta}, \quad (56)$$

the support function to the ellipse of Fig. 6.

The most general expression for the support function to an ellipse must include orientation and position as well as the lengths of the two semi-axes. We may produce any ellipse by rotating the ellipse of Fig. 6 in the counter-clockwise direction by  $\phi$  radians and then shifting the resulting figure so that it is centered at the point  $\mathbf{v} \in \mathbb{R}^2$ . The general expression for the support vector is then given by

$$h(\theta) = \sqrt{a^2 \cos^2(\theta - \phi) + b^2 \sin^2(\theta - \phi)} + [\cos \theta \sin \theta] \mathbf{v}, \quad (57)$$

where we have used the continuous analogue of the shift theorem to produce the last term in the equation. The support vector to an ellipse is found by sampling  $h(\theta)$  at the support angles  $\{\theta_i\}$  and arranging the samples in the require vector form.

## REFERENCES

1. J. L. Prince and A. S. Willsky, Reconstructing convex sets from support line measurements, *IEEE Trans. Pattern Anal. Mach. Intelligence* **12**(4), April 1990, 377–389.
2. G. T. Herman, *Image Reconstruction from Projections*, Academic Press, New York, 1980.
3. J. L. Prince, *Geometric Model-Based Estimation From Projections*, Ph.D. thesis, MIT, 1988.
4. J. L. Prince and A. S. Willsky, A geometric projection-space reconstruction algorithm, *Linear Algebra Appl.* **130**, 1990, 151–191 (Special Issue on Linear Algebra in Computed Tomography, G. T. Herman, Ed.).
5. P. C. Gaston and R. Lozano-Perez, Tactile recognition and localization using object models, *IEEE Trans. Pattern Anal. Mach. Intelligence* **6**, 1984, 257–266.
6. J. L. Schneiter and T. B. Sheridan, An automated tactile sensing strategy for planar object recognition and localization, *IEEE Pattern Anal. Mach. Intelligence* **12**(8), 1990, 775–786.
7. P. L. Van Hove and J. G. Verly, A silhouette-slice theorem for opaque 3-d objects, in *Proceedings, IEEE International Conference on Acoustics, Speech, and Signal Processing, March 26–29, 1985*, pp. 933–936.
8. W. C. Karl and G. C. Verghese, Curvatures of surfaces and their shadows, *Linear Algebra Appl.* **130**, 1990, 231–225.
9. B. K. P. Horn, *Robot Vision*, MIT Press, Cambridge, MA, 1986.
10. N. Ohta, Estimating absorption bands of component dyes by means of principal component analysis, *Anal. Chem.* **45**, 1973, 553–557.
11. J. P. Greshak, *Reconstructing Convex Sets*, Ph.D. thesis, Department of Electrical Engineering, MIT, 1985.
12. R. Cole and C. Yap, Shape from probing, *J. Algorithms* **8**, 1987, 19–38.
13. S. S. Skiena, *Geometric Probing*, Ph.D. thesis, Department of Computer Science, University of Illinois, 1988.
14. H. Edelsbrunner and S. S. Skiena, Probing convex polygons with X-rays, *SIAM J. Comput.* **17**(5), October 1988, 820–882.
15. F. P. Preparata and M. I. Shamos, *Computational Geometry*, Springer-Verlag, New York, 1985.
16. D. J. Rossi and A. S. Willsky, Reconstruction from projections based on detection and estimation of objects. I and II. Performance analysis and robustness analysis, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-32**(4), 1984, 886–906.
17. M. J. D. Powell, *Corrections and Extensions to the Fortran Listing of ZQPCVX*, Technical Report, University of Cambridge, February 1984.
18. M. J. D. Powell, *ZQPCVX: A Fortran subroutine for convex quadratic programming*, Technical Report DAMTP/1983/NA17, University of Cambridge, 1983.
19. D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.
20. D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed., Addison-Wesley, Reading, MA, 1984.
21. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, Cambridge Univ. Press, Cambridge, 1986.
22. J. L. Prince and A. S. Willsky, A hierarchical algorithm for limited-angle reconstruction, in *Proceedings, IEEE International Conference on Acoustics, Speech, and Signal Processing, Glasgow, May 1989*.