To appear in Proc. Computer Vision and Pattern Recognition Conference (CVPR 97), 1997.

Implicit Representation and Scene Reconstruction from Probability Density Functions

Steven M. Seitz
The Robotics Institute
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213
seitz@cs.cmu.edu

P. Anandan
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
anandan@microsoft.com

Abstract

A technique is presented for representing linear features as probability density functions in two or three dimensions. Three chief advantages of this approach are (1) a unified representation and algebra for manipulating points, lines, and planes, (2) seamless incorporation of uncertainty information, and (3) a very simple recursive solution for maximum likelihood shape estimation. Applications to uncalibrated affine scene reconstruction are presented, with results on images of an outdoor environment.

1 Introduction

Feature-based techniques have long been an active topic of research in computer vision, and have led to important advances in structure-from-motion, stereo, tracking, and recognition. However, most work to date has focused specifically on one type of feature, typically points [1, 2] *or* lines [3, 4]. Given that real-world scenes contain different types of features (e.g., corners, edges), it is important to develop integrated techniques that better exploit the information in an image sequence.

This type of integration is complicated, however, by the fact that points, lines, and planes generally require different representations and reconstruction equations. For example, Fig. 1 depicts different instances of an intersection operation, as applied to combinations of points, lines, and planes. Despite being conceptually similar, these operations typically have different mathematical realizations, yielding different equations for each combination of features. This problem is not limited to intersections; the same is true for projections and other coordinate transformations which must be reformulated for each type of feature.

In order to unify different types of features within a common mathematical framework, this paper introduces an algebra for operating on linear features represented *implicitly*, as probability density functions in two or three dimensions. In general, the distribution could be of any form – e.g., a line segment with no uncertainty could have all its density spread uniformly between two endpoints. However, in this paper, we restrict our attention to cases when we can model a feature and its uncertainty as a Gaussian density function p, represented as a matrix $\mathbf{\Omega}$ encapsulating both mean and covariance information:

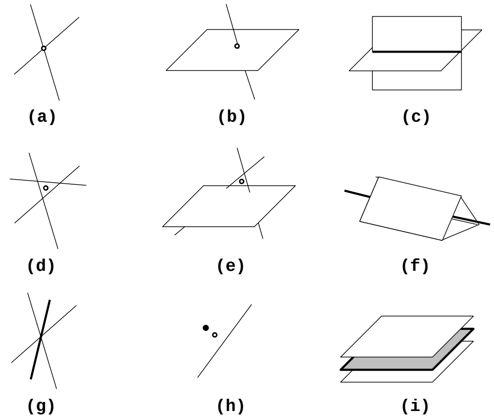

Figure 1: The Intersection Operation. Exact (a-c) and *closest* (d-f) intersections between any combination of point, line, and plane features are computed with the same formula in the implicit approach. (g-i) show other variations that are computable in the same manner.

$$p(\mathbf{X}) = k\, exp(-\frac{1}{2}\mathbf{X}^T\mathbf{\Omega}\mathbf{X})$$

The feature itself is expressed as the *maximum likelihood* contour of p, given by the locus of points $\mathbf{X}$ minimizing a set of quadratic equations:

$$argmin_{\mathbf{X}}(\mathbf{X}^T\mathbf{\Omega}\mathbf{X}) \qquad (1)$$

In the case of a point, Eq. (1) has one solution, whereas a line and plane will give rise to a one and two parameter family of solutions, respectively. In this paper, we use this implicit representation to model both image features and scene features. In particular, image features are represented by a 3×3 matrix $\mathbf{\Omega}^{2D}$ with $\mathbf{X} = [x\ y\ 1]^T$, and scene features by an 4×4 matrix $\mathbf{\Omega}^{3D}$ with $\mathbf{X} = [X\ Y\ Z\ 1]^T$.

A key property of this formulation is that points, lines, and planes are represented in exactly the same way, i.e., with the same parameters. Furthermore, intersections and transformations operate directly on the probability density function, and therefore apply equally to all types and combinations of linear features. For example, all of the operations in Fig. 1 are achieved via the same formula (matrix addition). More generally, the implicit representation offers the following advantages:

- **Uniformity**: the same representation and operations apply to all types and combinations of linear features, i.e., points, lines, and planes. Furthermore, features in the image plane and in the scene are modeled in a uniform manner, greatly simplifying 2D to 3D transformations.

- **Uncertainty**: measurement uncertainty (i.e., covariance information) is an integral part of the implicit representation, and is transformed along with the feature itself.

- **Maximum Likelihood Estimation**: the intersection operation can be recast as a maximum likelihood estimation approach, yielding the *closest* point/line/plane in a weighted least-squares sense.

- **Affine reconstruction**: the implicit representation enables 3D reconstruction from *uncalibrated* images, in which a metric 3D coordinate system is not immediately available.

As a means for integrating different types of features in the context of scene reconstruction, the implicit approach is related to previous reconstruction approaches that incorporate both points and lines. In particular, McLauchlan and Murray [5] developed an approach for reconstructing points and lines using a recursive filtering technique. While points and lines were represented and reconstructed differently, the same filter mechanism was used to integrate measurements over time. Morris and Kanade [6] demonstrated a factorization method that integrated point, line, and plane features. In their approach, lines and planes were represented by two and three points, respectively. Whereas both of these previous approaches offer a unified *computational* framework for representing points and lines, they do not provide a unified *representational* framework, i.e., points and lines are represented differently.

In contrast to the traditional approaches wherein geometric features are represented by a parameter vector (e.g., a plane in terms of its normal and offset), our proposed approach represents features as probability distributions over spatial locations. This replaces the traditional method of assigning uncertainty to the values of the parameters (e.g., a covariance matrix for the plane coefficients). This conceptual shift allows us to treat different geometric objects in a uniform fashion.

The implicit approach also relates to the Grassmann-Cayley algebra, which provides an elegant framework for representing points, lines, and planes, and their intersections via the *meet* operation, and has recently been applied to the task of scene reconstruction [7, 8]. One limitation of the Grassmann-Cayley algebra however, is that it is inherently exact—it is unclear how to incorporate uncertainty information or to perform maximum likelihood estimation in the Grassman-Cayley framework.

The remainder of the paper is structured as follows. Section 2 introduces the implicit representation and contrasts it with other feature-based representations used in computer vision. Section 3 describes how this representation may be used to compute intersections by formulating and solving a maximum likelihood estimation problem, and Section 4 describes other transformations in the implicit framework. We describe how to combine these operations in order to perform scene reconstruction in Section 5.

2 Implicit Feature Representation

A parametric, or *explicit*, representation describes an object as a function of a fixed set of variables. For example, 2D points may be represented in Euclidean (x, y) or polar (r, θ) coordinates. A 3D line can be specified by two points $(X_1, Y_1, Z_1, X_2, Y_2, Z_2)$, a point and two angles (X, Y, Z, θ, ϕ), or the intersections with two planes (U_1, V_1, U_2, V_2). Of course other representations are possible. Note that the number of parameters depends both on the dimension and the type of feature being represented.

In contrast, we propose to model features nonparametrically, by probability density functions in a 2- or 3-dimensional space. In this *implicit* representation, a feature is described by a displacement $\tilde{\mathbf{D}}$ from the origin and a symmetric, positive-semidefinite, inverse covariance matrix $\mathbf{\Lambda}$. $\mathbf{\Lambda}$ encodes the shape and orientation of the feature, as a Gaussian function centered at $\tilde{\mathbf{D}}$. More specifically, the eigenvectors (principle directions) of $\mathbf{\Lambda}$ specify the orientation and the eigenvalues the spatial extent of the feature in each principle direction.

We denote homogeneous 3D points $\mathbf{X} = [X\ Y\ Z\ 1]^T$ in uppercase, and homogeneous 2D points $\mathbf{x} = [x\ y\ 1]^T$ in lowercase. Points in Euclidean coordinates are denoted

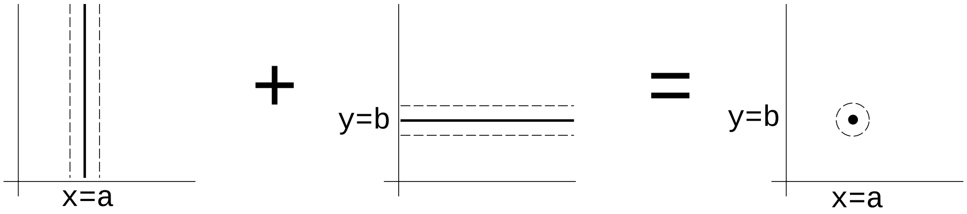

Figure 2: Features are represented implicitly as Gaussian density functions. The maximum-likelihood (0-distance) contour is shown solid, whereas the unit distance contour is dashed. The maximum-likelihood intersection of two such densities is given by the summation of inverse covariance matrices.

with a tilde: $\tilde{\mathbf{X}} = [X\ Y\ Z]^T$ and $\tilde{\mathbf{x}} = [x\ y]^T$. Given $\mathbf{\Lambda}$ and $\tilde{\mathbf{D}}$, we can write the *Mahalanobis distance* of a point to a feature:

$$\begin{aligned} M(\mathbf{X}) &= (\tilde{\mathbf{X}} - \tilde{\mathbf{D}})^T \mathbf{\Lambda} (\tilde{\mathbf{X}} - \tilde{\mathbf{D}}) \\ &= \mathbf{X}^T \begin{bmatrix} \mathbf{\Lambda} & -\mathbf{\Lambda}\tilde{\mathbf{D}} \\ -\tilde{\mathbf{D}}^T\mathbf{\Lambda} & \tilde{\mathbf{D}}^T\mathbf{\Lambda}\tilde{\mathbf{D}} \end{bmatrix} \mathbf{X} \end{aligned}$$

If we define

$$\mathbf{\Omega} = \begin{bmatrix} \mathbf{\Lambda} & -\mathbf{\Lambda}\tilde{\mathbf{D}} \\ -\tilde{\mathbf{D}}^T\mathbf{\Lambda} & \tilde{\mathbf{D}}^T\mathbf{\Lambda}\tilde{\mathbf{D}} \end{bmatrix}$$

then Eq. (1) defines the feature as the isocontour corresponding to $M(\mathbf{X}) = 0$. Under the Gaussian assumption, this contour corresponds to the *maximum likelihood* estimate of the feature's position and extent. A primary advantage of this implicit approach is that points, lines, planes, and their uncertainty are represented in exactly the same parameter space.

For instance, consider

$$\mathbf{\Omega}_1 = \begin{bmatrix} 1 & 0 & -a \\ 0 & 0 & 0 \\ -a & 0 & a^2 \end{bmatrix} \quad \mathbf{\Omega}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -b \\ 0 & -b & b^2 \end{bmatrix}$$

$\mathbf{\Omega}_1$ represents the vertical line $x = a$, as the solution space to the implicit equation $[x\ y\ 1]^T \mathbf{\Omega}_1 [x\ y\ 1] = 0$. Similarly, $\mathbf{\Omega}_2$ implicitly represents the horizontal line $y = b$. Fig. 2 pictorally shows these features and their associated density functions. Now consider the point of intersection of these two lines: $[x\ y] = [a\ b]$. This point may be represented implicitly by:

$$\mathbf{\Omega}_3 = \begin{bmatrix} 1 & 0 & -a \\ 0 & 1 & -b \\ -a & -b & a^2 + b^2 \end{bmatrix}$$

since $[x\ \ y] = [a\ \ b]$ is the sole solution to $[x\ y\ 1]^T \mathbf{\Omega}_3 [x\ y\ 1] = 0$.

Note that $\mathbf{\Omega}_3 = \mathbf{\Omega}_1 + \mathbf{\Omega}_2$. This is no coincidence—for a point $\mathbf{X}$ to be on both lines, it must satisfy both $\mathbf{X}^T\mathbf{\Omega}_1\mathbf{X} = 0$ and $\mathbf{X}^T\mathbf{\Omega}_2\mathbf{X} = 0$. When $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$ are positive-semidefinite, these two conditions are equivalent to the single constraint:

$$\begin{aligned} 0 &= \mathbf{X}^T\mathbf{\Omega}_1\mathbf{X} + \mathbf{X}^T\mathbf{\Omega}_2\mathbf{X} \\ &= \mathbf{X}^T(\mathbf{\Omega}_1 + \mathbf{\Omega}_2)\mathbf{X} \qquad (2) \end{aligned}$$

Thus, addition of matrices in the implicit representation corresponds to computing feature intersections. In the following sections, we recast the intersection operation as a maximum likelihood estimation approach and demonstrate how the implicit approach simplifies other reconstruction-oriented operations.

3 Implicit Intersection and Estimation

In this section we consider the problem of computing the intersection of a set of implicitly-represented features $\mathbf{\Omega}_1, \ldots, \mathbf{\Omega}_n$. When the features all share at least one point in common, a direct generalization of Eq. (2) specifies the point, line, or plane of intersection:

$$\bigcap_{i=1}^{n} \mathbf{\Omega}_i = \sum_{i=1}^{n} \mathbf{\Omega}_i \qquad (3)$$

Eq. (3) provides a simple, unified mechanism for computing intersections between any combination of linear features, such as those shown in Fig. 1(a-c). However, we would also like to cope with the situation in which features do not intersect exactly, e.g., Fig. 1(d-i), but there exists a point, line, and plane that are *closest* to the set of features, i.e., the maximum likelihood intersection. In other words, we wish to minimize the weighted sum of squares distance to each feature, given by:

$$\sum_{i=1}^{n} \mathbf{X}^T\mathbf{\Omega}_i\mathbf{X} = \mathbf{X}^T(\sum_{i=1}^{n} \mathbf{\Omega}_i)\mathbf{X} \qquad (4)$$

The solution to this minimization problem is given by Eq. (1), where $\mathbf{\Omega} = \sum_{i=1}^{n} \mathbf{\Omega}_i$. This result demonstrates that Eq. (3) applies not just to exact intersections, but also computes the point that is *closest* to a set of features, i.e., the maximum likelihood point of intersection. It will be shown in this section that the optimal point $\mathbf{\Omega}^{point}$, line $\mathbf{\Omega}^{line}$, and plane $\mathbf{\Omega}^{plane}$ are obtained by decomposing $\mathbf{\Omega}$ into a sum of component matrices

$$\mathbf{\Omega} = \mathbf{\Omega}^1 + \mathbf{\Omega}^2 + \mathbf{\Omega}^3 + \mathbf{C}$$

where

$$\begin{aligned} \mathbf{\Omega}^{point} &= \mathbf{\Omega}^1 + \mathbf{\Omega}^2 + \mathbf{\Omega}^3 + \mathbf{C} \\ \mathbf{\Omega}^{line} &= \mathbf{\Omega}^1 + \mathbf{\Omega}^2 + \mathbf{C} \\ \mathbf{\Omega}^{plane} &= \mathbf{\Omega}^1 + \mathbf{C} \end{aligned}$$

Observe that in combining features in the manner of Eq. (3), we are summing inverse covariance matrices of Gaussian distributions. In this respect, the implicit approach is similar to the Kalman filter and other recursive linear estimators that incrementally update a Gaussian state estimate and covariance matrix. In the present case, however, the state vector is dispensed with altogether, being represented implicitly in the covariance matrix. A key advantage of this technique, in addition to the simplified update procedure, is that we do not have to define different filters for different types of features, enabling operations on any combination of linear features.

3.1 Decomposing Ω

$\mathbf{\Omega} = \sum_{i=1}^{n} \mathbf{\Omega}_i$ implicitly describes the maximum likelihood point of intersection $\mathbf{D}$ and the cumulative covariance $\mathbf{\Lambda}$, given by

$$\begin{aligned} \mathbf{\Lambda} &= \sum_{i=1}^{n} \mathbf{\Lambda}_i \\ \tilde{\mathbf{D}} &= \mathbf{\Lambda}^{+} \sum_{i=1}^{n} \mathbf{\Lambda}_i \tilde{\mathbf{D}}_i \end{aligned}$$

where $\mathbf{\Lambda}^{+}$ denotes the pseudo-inverse of $\mathbf{\Lambda}$. This expression for $\mathbf{D}$ is obtained by expanding and differentiating the right hand side of Eq. (4) and setting the result to 0.

The line or plane closest to a set of features is determined by minimizing the integral of Eq. (4) over the space of all lines or planes. Towards this end, we rewrite $\mathbf{\Omega}$ as follows:

$$\mathbf{\Omega} = \begin{bmatrix} \mathbf{\Lambda} & -\mathbf{\Lambda}\tilde{\mathbf{D}} \\ -\tilde{\mathbf{D}}^T\mathbf{\Lambda} & \tilde{\mathbf{D}}^T\mathbf{\Lambda}\tilde{\mathbf{D}} + C \end{bmatrix}$$

where C is given by

$$C = (\sum_{i=1}^{n} \tilde{\mathbf{D}}_i^T \mathbf{\Lambda}_i \tilde{\mathbf{D}}_i) - \tilde{\mathbf{D}}^T \mathbf{\Lambda} \tilde{\mathbf{D}}$$

Eigenvector decomposition of $\mathbf{\Lambda}$ gives

$$\mathbf{\Lambda} = \sum_{i=1}^{d} \mathbf{\Lambda}_i$$

where

$$\mathbf{\Lambda}_i = \lambda_i \tilde{\mathbf{E}}_i \tilde{\mathbf{E}}_i^T$$

d is the dimension of the space (2 or 3), λ_i are the eigenvalues of $\mathbf{\Lambda}$, ordered from largest to smallest, and $\tilde{\mathbf{E}}_i$ are the corresponding eigenvectors. Given this decomposition of $\mathbf{\Lambda}$ it is convenient to express $\mathbf{\Omega}$ as follows:

$$\mathbf{\Omega} = \sum_{i=1}^{d} \mathbf{\Omega}^i + \mathbf{C} \tag{5}$$

where

$$\begin{aligned} \mathbf{\Omega}^i &= \begin{bmatrix} \mathbf{\Lambda}_i & -\mathbf{\Lambda}_i\tilde{\mathbf{D}} \\ -\tilde{\mathbf{D}}^T\mathbf{\Lambda}_i & \tilde{\mathbf{D}}^T\mathbf{\Lambda}_i\tilde{\mathbf{D}} \end{bmatrix} \\ \mathbf{C} &= \begin{bmatrix} 0 & 0 \\ 0 & C \end{bmatrix} \end{aligned}$$

In order to find the closest line and plane to a set of features, we evaluate the distance from each point on the line/plane to the features and integrate this distance over a unit area region. In particular, a line may be described by $\mathbf{X} + s\mathbf{V}_1$ and a plane by $\mathbf{X} + s\mathbf{V}_1 + t\mathbf{V}_2$, where the last coordinates of $\mathbf{V}_1$ and $\mathbf{V}_2$ are 0. For brevity, denote $\mathbf{V}_s = s\mathbf{V}_1$ and $\mathbf{V}_{s,t} = s\mathbf{V}_1 + t\mathbf{V}_2$. The respective integrals are

$$E^{\Omega}_{X,V_1} = \int_{\|\mathbf{V}_s\| \le 1} (\mathbf{X} + \mathbf{V}_s)^T \mathbf{\Omega} (\mathbf{X} + \mathbf{V}_s)\, ds \tag{6}$$

$$E^{\Omega}_{X,V_1,V_2} = \int_{\|\mathbf{V}_{s,t}\| \le 1} (\mathbf{X} + \mathbf{V}_{s,t})^T \mathbf{\Omega} (\mathbf{X} + \mathbf{V}_{s,t})\, ds\, dt \tag{7}$$

Differentiating Eqs. (6) and (7) with respect to $\mathbf{X}$ and setting the result to 0 yields the solution $\mathbf{X} = \mathbf{D}$. Substituting Eq. (5) yields:

$$E^{\Omega}_{X,V_1} = \left(\sum_{i=1}^{d} \lambda_i \int_{\|\mathbf{V}_s\| \le 1} (\mathbf{V}_s^T \tilde{\mathbf{E}}_i)^2\, ds \right) + K_1 \tag{8}$$

$$E^{\Omega}_{X,V_1,V_2} = \left(\sum_{i=1}^{d} \lambda_i \int_{\|\mathbf{V}_{s,t}\| \le 1} (\mathbf{V}_{s,t}{}^T \tilde{\mathbf{E}}_i)^2\, ds\, dt \right) + K_2 \tag{9}$$

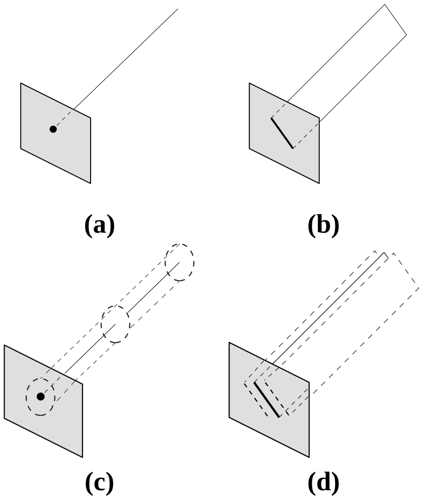

Figure 3: Inverse projection of point and line features. (a-b): the inverse projection is the set of all 3D points that project to the feature. (c-d): the covariance information is also propagated to 3D.

for constants K_1 and K_2 depending on C. It follows that Eqs. (8) and (9) are minimized when the vectors spanning the line and plane are aligned with eigenvectors of $\mathbf{\Lambda}$, i.e., $\mathbf{V}_1 = \tilde{\mathbf{E}}_d$, and $\mathbf{V}_2 = \tilde{\mathbf{E}}_{d-1}$. Note that if $\lambda_d = \lambda_{d-1}$ there is not a unique solution for the optimal line, and similarly for the plane when $\lambda_{d-1} = \lambda_{d-2}$.

4 Implicit Transformation

Observe that the intersection operation is coordinate-free; it is possible to describe the operation algebraically without introducing a specific coordinate system. However, the implicit representation itself is coordinate-specific. Consequently, we need to define how to perform coordinate transformations within the implicit framework.

Let $\mathbf{\Omega}$ represent a linear feature, and $\mathbf{T}$ an invertible affine coordinate transformation. We would like to determine the transformation $\mathbf{\Omega}'$ of $\mathbf{\Omega}$ such that $\mathbf{T}$ maps points on $\mathbf{\Omega}$ onto points on $\mathbf{\Omega}'$. Formally,

$$\mathbf{X}^T\mathbf{\Omega}\mathbf{X} = (\mathbf{T}\mathbf{X})^T\mathbf{\Omega}'(\mathbf{T}\mathbf{X})$$

Hence, the solution is: $\mathbf{\Omega}' = \mathbf{T}^{-1^T}\mathbf{\Omega}\mathbf{T}^{-1}$.

Note that projection is not an invertible transformation and therefore cannot be computed in this manner. However, *inverse projections* may be performed. The inverse projection of a 2D image feature is the 3D linear subspace that projects to that feature, as shown in Fig. 3. In particular, the inverse projection of a point is a line. Similarly, the inverse projection of a line is a plane. Here, we consider only affine projection models, including orthographic, weak-perspective, and paraperspective. In such models, projection may be represented with a 3 × 4 matrix $\mathbf{\Pi}$:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{\Pi}\mathbf{X} = \begin{bmatrix} i_1 & i_2 & i_3 & t_1 \\ j_1 & j_2 & j_3 & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Given an implicit 2D image feature $\mathbf{\Omega}^{2D}$ and a projection matrix $\mathbf{\Pi}$, the inverse projection $\mathbf{\Omega}^{3D}$ is given by:

$$\mathbf{\Omega}^{3D} = \mathbf{\Pi}^T\mathbf{\Omega}^{2D}\mathbf{\Pi} \qquad (10)$$

Importantly, the inverse projection operator transforms not just the feature itself, but the entire probability density function. In doing so, it provides a powerful mechanism for propagating uncertainty information from images to 3D scene space.

5 Implicit Reconstruction

Consider the problem of scene reconstruction from a set of point or line features. Specifically, suppose we are given m features in n images for which the correspondence is known. Furthermore, suppose the feature measurement uncertainty is Gaussian with known mean and covariance, and the projection matrices $\mathbf{\Pi}_i$ are also known up to an affine transformation.

The implicit representation gives rise to an extremely simple reconstruction formula, that applies equally to both points and lines. Specifically, if the 2D image projections of a point or line are given by $\mathbf{\Omega}_1^{2D}, \ldots, \mathbf{\Omega}_n^{2D}$, the corresponding 3D probability density function is given by:

$$\mathbf{\Omega}^{3D} = \sum_{i=1}^{n} \mathbf{\Pi}_i^T\mathbf{\Omega}_i^{2D}\mathbf{\Pi}_i \qquad (11)$$

Note that this formula may also be stated recursively, to enable incremental reconstruction algorithms in which the reconstruction is computed one frame at a time:

$$\begin{aligned} \mathbf{\Omega}_0^{3D} &= \mathbf{0} \\ \mathbf{\Omega}_i^{3D} &= \mathbf{\Omega}_{i-1}^{3D} + \mathbf{\Pi}_i^T\mathbf{\Omega}_i^{2D}\mathbf{\Pi}_i \end{aligned}$$

Given $\mathbf{\Omega}^{3D}$, an explicit representation of the reconstructed feature and its covariance may be obtained as described in Section 3.1. In particular, $\tilde{\mathbf{D}}$ gives the point, $\tilde{\mathbf{E}}_d$ the direction of the line, and $span\{\tilde{\mathbf{E}}_d, \tilde{\mathbf{E}}_{d-1}\}$ the plane of minimal Mahalanobis distance to the input features. $\mathbf{\Lambda}$ itself describes the covariance.

Eq. (11) demonstrates the attraction of the implicit approach—it provides a means for achieving powerful geometric algorithms via simple algebraic formulas.

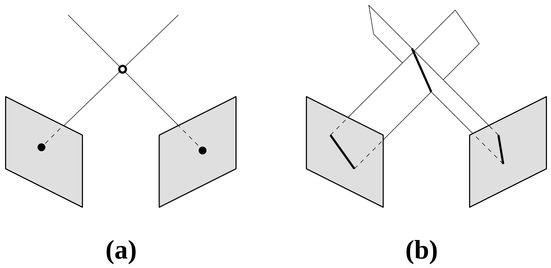

Figure 4: Reconstruction by intersecting inverse projections. In the case where the inverse projections do not intersect, the *closest* point, line, and plane of intersection will be computed, weighted by the to inverse measurement covariance.

5.1 Uncalibrated Reconstruction

Observe that Eq. (11) minimizes the *image-space* error by evaluating the projected distance from the reconstruction to each of the image-features. Consequently, this approach enables *affine* reconstruction [9], in situations when the projection matrices $\mathbf{\Pi}_i$ are known only up to an affine transformation.

In this *uncalibrated* setting, however, care must be taken when converting line and plane reconstructions to an explicit representation, since Eqs. (6) and (7) are not affine-invariant. In other words, if $\mathbf{X}$ and $\mathbf{V}$ minimize $E^{\Omega}_{X,V}$, it need not be the case that $\mathbf{TX}$ and $\mathbf{TV}$ minimize $E_{X,V}^{{T^{-1}}^{T}\Omega T^{-1}}$. It therefore follows that we cannot use Eqs. (6) and (7) to compute an explicit reconstruction without knowing the Euclidean coordinate system.

However, the implicit representation *is* valid and can be computed in an uncalibrated, affine coordinate system. This observation suggests the following *stratified* approach for 3D scene reconstruction [10]: first, an implicit affine reconstruction is computed, one frame at a time, using Eq. (11). Then, when calibration information becomes available, e.g., when a known object comes into the field of view, the scene is transformed to Euclidean coordinates and an explicit reconstruction is obtained. The reconstruction obtained from applying this procedure is identical to what would be obtained if the Euclidean coordinate system were available from the outset, but enables delaying the choice of coordinate frame until more information becomes available.

6 Results

In order to evaluate this reconstruction approach, we took eight photographs of an outdoor building scene from varying viewpoints. Feature correspondence was obtained using the technique described in [11], in which a small number of point correspondences was used to determine epipolar lines and affine camera matricies $\mathbf{\Pi}$. While four point correspondences are sufficient in principle [9], 8 points were used in our experiments in conjunction with a least-squares method to obtain more reliable epipolar lines. This information, in turn, yielded a complete line feature correspondence via a voting technique [11]. These line features were represented and reconstructed in an *uncalibrated* affine coordinate frame using the implicit approach.

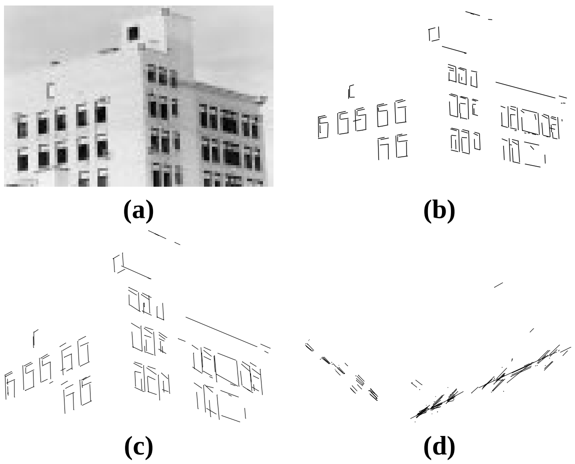

Figure 5: Implicit Reconstruction. One of 8 original images is shown in (a). Line segments were detected and reconstructed using the implicit approach. (b-d) show different views of the reconstruction.

Fig. 5(b-d) show different views of the affine reconstruction. For display in Fig. 5, Euclidean camera parameters were obtained and the features were converted to an explicit representation, as described in Section 5.1. However, the features themselves are shown in the affine reference frame (note that the walls are not perpendicular in the figure). Line segments were obtained by clipping the lines to match their projected lengths in one of the input images. These promising results demonstrate the effectiveness of the implicit approach for reconstructing real-world environments.

7 Conclusions

This paper presented a unified approach for representing and manipulating points, lines, and planes as Gaussian density functions. Rather than representing each feature explicitly, with a different set of parameters for each type of feature, we instead introduced an approach in which all features are represented implicitly, as the solution space to a set of quadratic equations. This approach was shown to have several advantages. First, the unified representation enables defining operations like intersection and coordinate transformation that apply to any type and combination of features. This avoids the need to derive separate formulae for all possible combinations of features (i.e., intersections of line-line, line-plane, plane-line-plane, etc.). Second, the ability to easily incorporate measurement uncertainty and solve weighted least-squares problems in the implicit framework makes it particularly well suited for maximum likelihood estimation. Finally, we described how the implicit approach may be applied to reconstruct 3D feature positions and orientations from uncalibrated input images. The solution is especially attractive in that it provides an extremely simple, one-line formula with the following properties: (1) it is recursive, integrating measurements incrementally in an optimal manner, (2) it applies equally to both point and line features, (3) measurement covariance is incorporated, and (4) it may be used in an uncalibrated setting, when the camera matrices are known only up to a 3D affine transformation.

There are a number of important unresolved issues with respect to both the theoretical aspects of the proposed framework and its applications. First, we have not addressed how to perform "join" operations [7, 8]. to yield the point, line, or plane that joins, i.e., interpolates, a set of features. In contrast to the feature intersection operation, which yields a feature with the same or lower dimension than that of the features being operated on, the result of a join is a feature of higher dimension. One could imagine defining a statistical join within our framework that could support regression operations to enable fitting the optimal line or plane to a set of implicit features. This paper has not addressed this problem. Second, as noted earlier, while we have developed a way to perform inverse projection, we have not addressed the problem of forward projection. Third, the current formulation using Gaussian density functions only applies to linear models and linear transformations. The extension to non-linear problems (especially perspective projection) is a subject for future work. Finally, we have not considered extensions to hierarchical features such as a plane bounded by line segments, lines with end points, etc. These are all topics that are of current interest to us and will form a part of our future work in this area.

References

[1] S. Ullman, *The Interpretation of Visual Motion*. Cambridge, MA: MIT Press, 1979.

[2] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *Int. Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[3] M. E. Spetsakis and J. Aloimonos, "Structure from motion using line correspondences," *Int. Journal of Computer Vision*, vol. 4, pp. 171–183, 1990.

[4] L. Quan, "Uncalibrated 1D projective camera and 3D affine reconstruction of lines," in *Proc. Computer Vision and Pattern Recognition Conf.*, pp. 60–65, 1997.

[5] P. F. McLauchlan and D. W. Murray, "A unifying framework for structure and motion recovery from image sequences," in *Proc. 5th International Conference on Computer Vision*, pp. 314–320, 1995.

[6] D. D. Morris and T. Kanade, "A unified factorization algorithm for points, line segments and planes with uncertainty models," in *Proc. 6th International Conference on Computer Vision*, pp. 696–702, 1998.

[7] S. Carlsson, "Multiple image invariants using the double algebra," in *Proc. Workshop on Applications of Invariants in Computer Vision*, pp. 335–350, 1993.

[8] G. Csurka and O. Faugeras, "Computing three dimensional project invariants from a pair of images using the Grassmann-Cayley algebra," *Image and Vision Computing*, vol. 16, pp. 3–12, 1998.

[9] J. J. Koenderink and A. J. van Doorn, "Affine structure from motion," *Opt. Soc. Am. A*, vol. 8, pp. 377–385, 1991.

[10] O. Faugeras, "Stratification of 3-D vision: projective, affine, and metric representations," *Journal of the Optical Society of America*, vol. 12, no. 3, pp. 465–484, 1995.

[11] S. M. Seitz and C. R. Dyer, "Complete structure from four point correspondences," in *Proc. Fifth Int. Conference on Computer Vision*, pp. 330–337, 1995.