Squares, which are special instances of rectangles, are harder to characterize because the length of digital edges also must be incorporated, and are left for further investigation. Triangles can also be studied on similar lines, in particular, special triangles such as right triangles, isoceles, and equilateral triangles.

References

[1] T. A. Anderson and C. E. Kim, "Representation of digital line segments and their preimages," *Comput. Vision, Graphics, Image Processing*, vol. 30, pp. 279–288, 1985.

[2] L. Dorst and A. W. M. Smeulders, "Discrete representation of straight lines," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 450–463, 1984.

[3] C. E. Kim, "Digital convexity, straightness, and convex polygons," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-4, pp. 618–626, 1982.

[4] —, "Digital disks," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 372–374, 1984.

[5] R. Krishnaswamy and C. E. Kim, "Digital parallelism, perpendicularity and rectangles," Dep. Comput. Sci., Washington State Univ., Tech. Rep. CS-84-131, 1985.

[6] C. E. Kim and A. Rosenfeld, "Digital straight lines and convexity of digital regions," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-4, pp. 149–153, 1982.

[7] A. Rosenfeld, "Digital straight line segments," *IEEE Trans. Comput.*, vol. C-23, pp. 1264–1269, 1974.

On Extensions to Fisher's Linear Discriminant Function

IAN D. LONGSTAFF

Abstract—**This correspondence describes extensions to Fisher's linear discriminant function which allow both differences in class means and covariances to be systematically included in a process for feature reduction. It is shown how the Fukunaga–Koontz transform can be combined with Fisher's method to allow a reduction of feature space from many dimensions to two. Performance is seen to be superior in general to the Foley–Sammon method. The technique is developed to show how a new radius vector (or pair of radius vectors) can be combined with Fisher's vector to produce a classifier with even more power of discrimination. Illustrations of the technique show that good discrimination can be obtained even if there is considerable overlap of classes in any one projection.**

Index Terms—**Classification, dimensionality reduction, discriminant analysis, feature selection, pattern recognition.**

I. Introduction

This correspondence is concerned with two-class discriminant analysis for the case where classes have different mean vectors and covariance matrices.

Fisher's linear discriminant function [1], [2] makes a useful classifier where the two classes have features with well separated means compared with their scatter. The method finds that vector which, when data are projected onto it, maximizes class separation. It is a many-to-one linear transform.

Manuscript received September 12, 1985; revised August 1, 1986. Recommended for acceptance by J. Kittler.

The author was with the Royal Signals and Radar Establishment (RSRE), Malvern WR14 3PS, England. He is now with Western Australia Institute of Technology, Perth, Western Australia.

IEEE Log Number 8612821.

At the other extreme, for cases where both classes have the same mean but different variances, the Fukunaga–Koontz [3] transform can be used to maximize class discrimination. This method transforms the data so the joint (sum) covariance matrix for the two classes is the identity matrix. The two classes then have the same eigenvectors, and if the class means are both equal, the vector with the biggest difference in eigenvalues is used as the discriminant function.

An advantage of linear projections such as these is that they give the system designer some appreciation of class separation and structure of the data. For example, in Fisher's original paper, clear evidence for a third class of iris can be seen when data are projected onto the Fisher vector.

An even better appreciation of data structure can be gained if data are presented as a projection onto a two-dimensional plane, where the direction of the plane is chosen to maximize class discrimination. Such projections form scatter diagrams which allow the designer to select a curved or piecewise linear decision boundary as the pattern classifier. This approach has been found to yield more productive results than the use of noninteractive methods such as the quadratic discriminant function, especially where the data are non-Gaussian.

Foley and Sammon [4] have suggested an extension to Fisher's method which gives a two- (or more)-dimensional projection for displaying data. This method is based on finding Fisher's vector first, then in the subspace normal to this vector another Fisher vector is found. Data are then displayed on the plane subtended by these vectors. It is shown later in this correspondence that the method is of doubtful value for finding the best classification subspace.

The methods proposed in this paper are based on applying a standardizing transform to the data. It is shown how this allows Fisher's method to be used in conjunction with the Fukunaga–Koontz method to select the best two-dimensional linear projection. The method is then extended to show how a nonlinear combination of features produces a two- or three-dimensional scatter diagram with better discrimination than the linear projection.

II. Observations on Fisher's Method

Fisher's method find's the vector $\boldsymbol{F}$ which gives the greatest class separation (as defined by a criterion function) to points projected onto the vector.

The vector solution to this maximizing problem is well known to be:

$$\boldsymbol{F} = [S_a + S_b]^{-1} [\boldsymbol{\mu}_a - \boldsymbol{\mu}_b]$$

where $\boldsymbol{\mu}_a$, $\boldsymbol{\mu}_b$ = means for class A, B

S_a, S_b = covariance matrices for class A, B.

It is interesting to note in passing that this does not necessarily produce the best projection for classification, as shown by Malina [5]. However, the differences are usually very small and Fisher's method is used in this correspondence because it leads to the interesting generalizations and extensions shown here.

It is clear that data can be transformed on to a new set of coordinates without loss or gain of discriminating performance provided the transform is invertible. A decision boundary in one coordinate system maps on to the other with the same number of true or false classifications on either side. Now the decision threshold on the Fisher vector corresponds to a hyperplane decision surface in the full-dimensional feature space, where the hyperplane is normal to the vector and intersecting it at the decision threshold. It can easily be shown that the same decision threshold and plane are generated by the use of Fisher's method if a linear transform is applied to the data first.

A useful transform which can be applied to the training data is that which causes the joint covariance matrix $[S_a + S_b]$ to be the

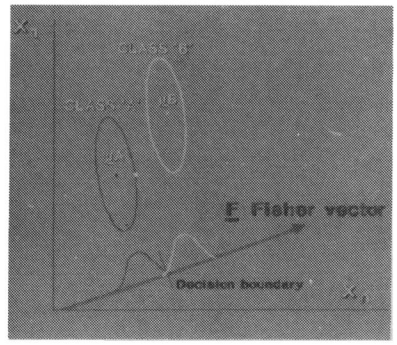

Fig. 1. Fisher's linear discriminant function.

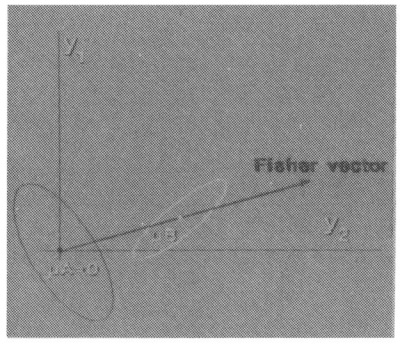

Fig. 2. Fisher's linear discriminant function after applying transform to make $(S_a + S_b) = (I)$ and $\mu_a = 0$.

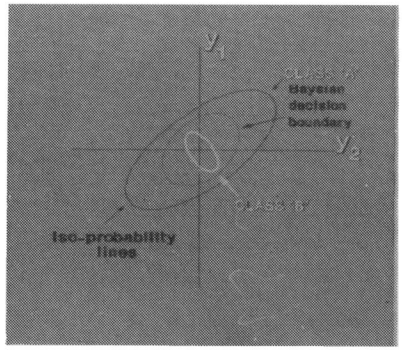

Fig. 3. Projection of data onto subspace normal to F.

identity matrix. Such a transform can be constructed by first applying a rotation of axes (orthogonal transform) so the eigenvectors of the matrix are the Cartesian coordinate system (the Karhunen–Loeve transform). Each coordinate can then be scaled to give unit variance and hence the identity covariance matrix.

In this coordinate system the Fisher axis is:

$$F' = I[\mu_a' - \mu_b'].$$

That is, F' is parallel to the axis intercepting the means of the two class distributions; see Figs. 1 and 2. This is a useful way of standardizing Fisher's method and there is no loss or gain in performance through applying the transform. The value of doing this is that if we project the data on to the subspace normal to F' then the data sets will have coincident means; see Fig. 3.

This simple procedure for causing the data to have the same means opens up a number of possibilities for extending Fisher's method, some of which are shown in this correspondence.

III. Extensions to Fisher's Method

As mentioned earlier it would be convenient if we could combine Fisher's vector with some other discriminant to give a two-

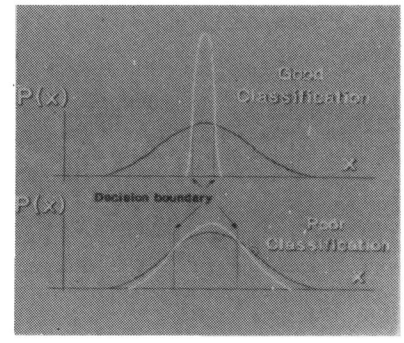

Fig. 4. Classification of data with common mean and (a) big difference in variances, (b) small difference in variances.

dimensional projection for inspecting data and constructing decision boundaries. Also two dimensions should, in general, give better discrimination than one.

If the standardizing transform described above is applied to the data the clusters in the the subspace normal to Fisher's vector will have coincident means. This makes it clear that Fisher's discriminant, which is derived from differences in class means, does not work in this subspace. Thus the selection of Fisher's vector in this subspace, as Foley and Sammon suggested, is a poor choice. It was this realization that led to the development of the methods described here which give improved discrimination over the Foley–Sammon method. Illustrative examples are given to show the improvement with these methods.

A. Fisher with Fukunaga–Koontz

In all the methods described from here onward the first step is to transform the data to give an identity joint covariance matrix (the standardizing transform). The Fisher axis is then the axis through the means. If the data are projected on to the hyperplane normal to the Fisher axis the two classes will have common means. To obtain maximum differences between the classes we can look for the projection which maximizes the differences in variances (normalized by the sum of the variances). It is then evident that the bigger the difference the better the classifier as illustrated in Fig. 4.

If two classes have covariances S_a and S_b, let T be the standardizing transform such that:

$$T[S_a + S_b]T^{-T} = I.$$

Fukunga showed that the eigenvectors of

$$TS_aT^{-T} \quad \text{and} \quad TS_bT^{-T}$$

are the same and all the eigenvalues are bounded by 0 and 1 and the sum of any pair equals one, i.e., $\lambda_{ia} + \lambda_{ib} = 1$.

It is clear from this that the axis with the biggest difference in variances is the eigenvector with the biggest difference in eigenvalues for the two classes.

Thus the Fisher projection with the Fukunaga–Koontz (F-K) projection gives a many-to-two linear transform with a performance which is usually better and never worse than the Fisher with Foley–Sammon (F-S) projection (for multivariate Gaussian data). Fig. 6 and 7 are examples of the use of this projection on data with different means and variances.

The parameters used to generate the data are shown in Fig. 5. The F-S projection with 100 data samples per class is shown in Fig. 6 and the same data with the Fisher and F-K method is shown in Fig. 7, indicating a clear improvement in separability. The decision boundaries in these examples were simply selected by eye as a smooth curve giving best separation, but of course more formal methods could be used.

B. Fisher with a Radius Vector

This use of the F-K transform suggests an even more powerful (nonlinear) transform which uses all eigenvectors with differing ei-

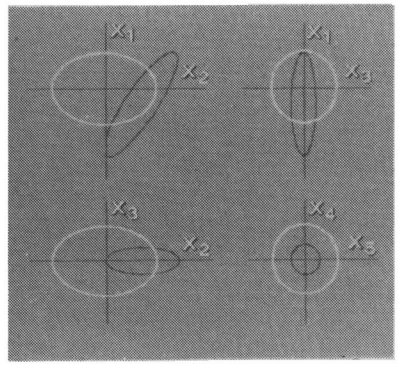

Fig. 5. Some projections of the data used in Figs. 6–8 with:

$$\mu_a = 0 \qquad \mu_b = \left(-\tfrac{1}{2}, \sqrt{3}, 0, 0, 0\right)^T$$

$$S_a = \begin{bmatrix} 1 & & & & \\ & 4 & 0 & & \\ & & 1 & & \\ & 0 & & 1 & \\ & & & & 1 \end{bmatrix}$$

$$S_b = \begin{bmatrix} 3.02 & 3.41 & 0 & 0 & 0 \\ 3.41 & 4.12 & 0 & 0 & 0 \\ 0 & 0 & 0.0625 & 0 & 0 \\ 0 & 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0 & 0.25 \end{bmatrix}$$

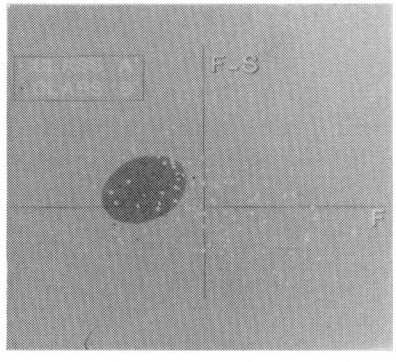

Fig. 6. Scatter in Fisher Foley–Sammon plane. Total errors ~12 percent.

genvalues. For example consider a three-feature two-class problem. After the standardizing transform and projecting on to the plane normal to the Fisher vector we might obtain distributions as shown in Fig. 3. If all the eigenvalues for class A are less than those for class B then the Bayesian decision surface can be shown to be an ellipse with the eigenvectors as axes (see Appendix).

By rotating the data and rescaling it is clear that the Bayesian surface can be made into a circle with class A inside and class B outside. The only information needed to test if a new data point lies inside or outside is to compute its radius from the common mean and test against the radius to the common threshold. In the more general multidimensional case the Bayesian surface can be made into a hypersphere if the eigenvalues for one class are all less than those for the other. Classification in this case involves assigning a new point to class A or B according to whether it lies inside (class with smaller eigenvalues) or outside (class with larger eigenvalues) hypersphere.

For either of these cases the data can be mapped from the original multidimensional space down to two, where the distance along

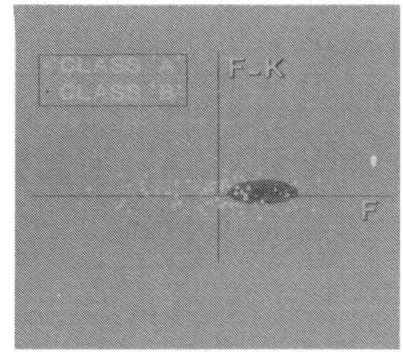

Fig. 7. Scatter in Fisher Fukanaga–Koontz plane. Total errors ~6.5 percent.

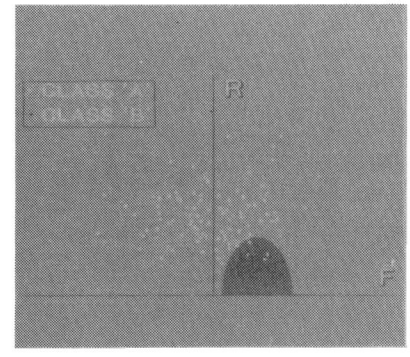

Fig. 8. Scatter in Fisher radius plane. Total errors ~2 percent.

the Fisher axis is one feature and radius (Euclidean distance) from the Fisher axis in the transformed space is the other feature.

Clearly the Fisher axis is only informative and well defined if $\mu_a' - \mu_b'$ is not small or zero and the radius axis will only be informative and well defined if $S_a' - S_b'$ is not small or zero. This method will be better than the use of Fisher with one F-K axis alone if there is more than one eigenvector with $\lambda_{ia} \neq \lambda_{ib}$. In circumstances where the class distribution functions have circular symmetry along the Fisher axis the Bayesian surface will also have circular symmetry and map on to a unique line in the Fisher–Radius plane. Hence in this case performance in the F-R space is optimal and equal to the performance of a Bayesian classifier in the full feature space.

To use this method in the more general case where all the eigenvalues of one class are not less than the other, we divide the features of the training data into two subsets of reduced dimensionality where the first subspace contains only those features where $\lambda_{ia} < \lambda_{ib}$ and the second only those where $\lambda_{ia} > \lambda_{ib}$. Any features for which $\lambda_{ia} = \lambda_{ib}$ should not be included in the radius calculation. If a two-dimensional classifier is required the subset with the best performance can be selected, otherwise a three-dimensional classifier can be constructed.

Fig. 8 shows the scatter in the F-R axes with the same data as in the previous examples. It is seen that the error rate is reduced from 6 percent with F-K to 2 percent with F-R in this example. The advantage of using the F-R method can be seen even more clearly when more features are available as in the ten-dimensional example of Fig. 9. Notice the data have the same differences in mean and variance as before, but now the error rate is approaching zero. If the F-K method had been used it would have given the same error rate as the five-dimensional F-K example. All the data used for these examples were generated from a multivariate Gaussian process with 100 samples per class. The classifiers were also tried with only 30 samples per class. Poorer estimates of mean and covariance caused performance for all methods to deteriorate but the F-R method still produced the most well separated clusters,

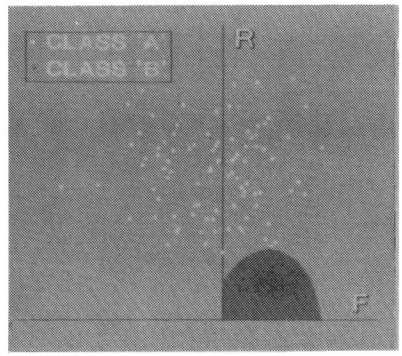

Fig. 9. Scatter in Fisher radius plane, 10 dimension data generated with

$$\mu_a = 0 \qquad \mu_b = (-\tfrac{1}{2}, \sqrt{3}, 0, 0 \cdots 0)^T$$

$$S_a = \begin{bmatrix} 1 & & & & \\ & 4 & & 0 & \\ & & 1 & & \\ & 0 & & 1 & \\ & & & & \ddots \\ & & & & & 1 \end{bmatrix}$$

$$S_b = \begin{bmatrix} 3.02 & 3.41 & 0 & & \\ 3.41 & 4.12 & 0 & & \\ 0 & 0 & 0.0625 & & \\ & & & \ddots & \\ & & & & 0.0625 \end{bmatrix}$$

showing the method to be reasonably robust to the problem of small data sets.

If data are in fact multivariate Gaussian the quadratic discriminant function will give optimum performance but the additional scope given by this method, where the designer can select the decision boundary, allows better performance to be obtained if data are non-Gaussian. For instance, data generated with a negative exponential distribution were better classified using the F-R method than the quadratic discriminant function.

IV. Conclusions

For the two-class problem, where there are differences in class means and variances, the method described in this paper allows both these differences to be used in setting up a near-optimal two-dimensional decision space.

If a standardizing transform is applied to give an identity joint covariance matrix the data have coincident means in the subspace normal to the Fisher axis. In this subspace the Fukunaga–Koontz method for centered data can be used to select the next best linear projection.

Alternatively, another scaling transform can give a hyperspherical decision boundary if all the eigenvalues for one class are less than the other. In this case the distance along the Fisher axis and distance (radius) from the Fisher axis form a powerful discriminating function.

If all the eigenvalues for one class are not less than the other the features can be divided into two groups and two radii calculated, with the best or both being used in conjunction with the Fisher distance to form a discriminant function. Both linear transforms can be combined into a single operation to simplify implementation of the classifier.

The arguments used in this correspondence are based on multivariate Gaussian distributions. In practice distributions are seldom so simple. However, the methods described here allow data to be standardized and inspected as a first approach to the classifier design problem. A good classifier may result, perhaps with the excising of pathological features or with the inclusion of special stages to include highly non-Gaussian but well discriminating features.

The methods described here are for the two-class problem only. However they are particularly well suited to the technique of reducing a many-class problem to that of many two-class problems. In this form the problem is to identify one species against the world background of all other species lumped together. This usually results in the background class having a much higher variance than the required species and the method described here takes advantage of this characteristic and can give good performance even where the means of the two classes are similar.

Appendix
Producing Hyperspherical Decision Boundaries

This Appendix shows how and when the Bayesian decision surface on multivariate Gaussian distributions with common means can be transformed to a hypersphere.

The Fukunaga–Koontz transform shows how two distributions can be transformed to have common eigenvectors. The K-L transform extracted from the covariance matrix of one class can be used to align the eigenvectors with a new coordinate system. In this system the two data sets are decorrelated and as they have common means the pdf's can be written as:

Class A:

$$P(a) = \prod_{i=1}^{L} \left(\frac{1}{(2\pi\lambda_{ia})^{1/2}} \exp - x_i^2/2\lambda_{ia} \right). \tag{1}$$

Class B:

$$P(b) = \prod_{i=1}^{L} \left(\frac{1}{(2\pi\lambda_{ib})^{1/2}} \exp - x_i^2/2\lambda_{ib} \right). \tag{2}$$

At the Bayesian decision boundary the pdf's are equal. Taking logs of (B1) and (B2) and equating gives:

$$\sum_{i=1}^{L} x_i^2/2\lambda_{ia} + \log \prod_{i=1}^{L} \lambda_{ia}^{1/2} = \sum_{i=1}^{L} x_i^2/2\lambda_{ib} + \log \prod_{i=1}^{L} \lambda_{ib}^{1/2}$$

$$\sum_{i=1}^{L} x_i^2(1/2\lambda_{ia} - 1/2\lambda_{ib}) = \frac{1}{2} \log \prod_{i=1}^{L} \left(\frac{\lambda_{ib}}{\lambda_{ia}} \right). \tag{3}$$

If all the coefficients of x have the same sign this is the equation of a hyperellipse. A simple rescaling transform can be used to reduce this to a hypersphere with the scaling factor in the ith coordinate being given by

$$k_i^2 = 1/\lambda_{ia} - 1/\lambda_{ib}.$$

If the coefficients of x in (3) have different signs some sections of the surface will be a saddle. To overcome this the data can be transformed into two subsets of similarly signed coefficients. If the variances of two features are equal

$$\lambda_{ia} = \lambda_{ib}$$

then the ith feature will give no further discrimination and can be disregarded.

Acknowledgment

The author is indebted to J. Cross and M. Hayward for valuable discussions and computing the examples.

References

[1] R. A. Fisher, "The use of multiple measures in taxonomic problems," *Ann. Eugenica*, vol. 7, part II, pp. 179–188, 1936.
[2] R. Duda and P. Harte, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973, pp. 114–128, 221–225.
[3] D. Foley and J. Sammon, "An optimal set of discriminant vectors," *IEEE Trans. Comput.*, vol. C-26, pp. 281–289, Mar. 1975.

[4] K. Fukunaga and W. L. Koontz, "Application of the Karhunen-Loeve transform to feature selection and ordering," *IEEE Trans. Comput.*, vol. C-19, pp. 311-318, Apr. 1970.
[5] W. Malina, "On an extended criterion for feature selection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, pp. 611-614, Sept. 1981.

Mobile Robot Localization Using Sonar

MICHAEL DRUMHELLER

Abstract—**This correspondence describes a method by which range data from a sonar rangefinder can be used to determine the two-dimensional position and orientation of a mobile robot inside a room. The plan of the room is modeled as a list of segments indicating the positions of walls. The algorithm works by correlating straight segments in the range data against the room model, then eliminating implausible configurations using the *sonar barrier test*, which exploits physical constraints on sonar data. The approach is extremely tolerant of noise and clutter. Transient objects such as furniture and people need not be included in the room model, and very noisy, low-resolution sensors can be used. The algorithm's performance is demonstrated using a Polaroid Ultrasonic Rangefinder.**

Index Terms—**Mobile robots, navigation, sonar.**

I. Problem Definition

This correspondence considers the problem of enabling a mobile robot to determine its position and orientation (its *configuration*) inside a building in a way that is independent of assumptions about previous movements. This ability will be called *absolute localization*, or simply *localization*. Localization is the direct measurement of vehicle position, in contrast to *dead reckoning*, or *trajectory integration*, which deduces the vehicle's position from its velocity history. Velocity is measured relative to nearby stationary objects such as the ground.

Since the errors encountered in dead-reckoning are cumulative, a robot that navigates by dead reckoning alone will eventually lose track of its position. Ultimately, this can be prevented only by periodically re-establishing the absolute position of the robot. Therefore, a means of localization is necessary for safe, reliable robot navigation.

Some solutions to this problem have been proposed that require modifications to the environment, such as triangulation from infrared beacons [4]. It would be desirable to solve the problem without modifying the environment. Most of the mobile robot navigation schemes developed so far, such as [11], are essentially dead reckoning methods, which lack any provision for periodically localizing the robot. Such schemes could benefit from localization.

In the approach to localization described in this correspondence, the robot's environment is a room or area inside a building. The environment might include the whole building. The robot's user must provide a model of the room consisting of a list of segments indicating the locations of walls. Such a model is easily constructed from an architect's drawing.

Manuscript received April 14, 1985; revised May 6, 1986. Recommended for acceptance by R. Bajcsy. This work was supported in part by a grant from the System Development Foundation, and in part by the Advanced Research Projects Agency under Office of Naval Research Contracts N00014-80-C-0505 and N00014-82-K-0334.

The author is with the Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139. Part of this work was performed while he was with Thinking Machines Corporation, 245 First Street, Cambridge, MA 02142.

IEEE Log Number 8612401.

Fig. 1. A typical room, a sonar contour obtained from inside the room, and the localization produced by the algorithm.

The rangefinding device used in this paper is a Polaroid Ultrasonic Rangefinder, but any rangefinder may be used (see, for example, [7], [8], [9], and [14]). We will call ultrasonic rangefinding *sonar* for short.

The closed contour obtained by a 360-degree sweep with a sonar beacon will be called a *sonar contour*. The lines drawn from the robot position to the individual data points in the sonar contour represent individual range readings, and are called *sonar rays*. Fig. 1 shows a typical room outline, a typical sonar contour obtained from inside the room, and the corresponding localization as determined by the algorithm.

II. Approach

The recent papers [3], [5], and [6] have introduced a new approach to object recognition and localization based on the exploitation of simple geometric constraints between sensed data and a model. Section II-D is based largely on the algorithm by Grimson and Lozano-Pérez, described in [6]. The main difference between Section II-D and these papers is the use of *sonar segments*, which are straight segments extracted from a sonar contour, instead of position/normal-vector pairs, as the primary inputs to the algorithm.

This correspondence introduces a new idea called the *sonar barrier test* in Section II-E. The sonar barrier test checks for physical consistency by determining whether the shape of a sonar contour for a proposed localization is consistent with the fact that sonar beams do not penetrate solid objects. If an inconsistency is found, the proposed localization is discarded. The sonar barrier test makes the algorithm's overall performance superior to what was obtainable using only the Grimson and Lozano-Pérez algorithm.

Miller [10] also describes an approach to robot localization using sonar, following the methods of Gaston, Grimson, and Lozano-Pérez. The method described in [10] uses single sonar rays instead of larger data features as the primary inputs to the algorithm, and it uses a different set of local geometric constraints. Nothing analogous to the sonar barrier test is presented.

Our approach is to consider the localization process to be a two-dimensional matching (including rotation) between the sonar contour and the room outline. We wish to determine the geometrical relationship between the robot and the room. The configuration of the robot relative to the sonar contour is always known, so if we determine a possible configuration for the sonar contour relative to