Available online at www.sciencedirect.com

ScienceDirect

Pattern Recognition 40 (2007) 2607–2620

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/pr

Target differentiation with simple infrared sensors using statistical pattern recognition techniques

Billur Barshan*, Tayfun Aytaç, Çağrı Yüzbaşıoğlu

Department of Electrical Engineering, Bilkent University, Bilkent, TR-06800 Ankara, Turkey

Received 22 March 2006; received in revised form 1 November 2006; accepted 3 January 2007

Abstract

This study compares the performances of various statistical pattern recognition techniques for the differentiation of commonly encountered features in indoor environments, possibly with different surface properties, using simple infrared (IR) sensors. The intensity measurements obtained from such sensors are highly dependent on the location, geometry, and surface properties of the reflecting feature in a way that cannot be represented by a simple analytical relationship, therefore complicating the differentiation process. We construct feature vectors based on the parameters of angular IR intensity scans from different targets to determine their geometry and/or surface type. Mixture of normals classifier with three components correctly differentiates three types of geometries with different surface properties, resulting in the best performance (100%) in geometry differentiation. Parametric differentiation correctly identifies six different surface types of the same planar geometry, resulting in the best surface differentiation rate (100%). However, this rate is not maintained with the inclusion of more surfaces. The results indicate that the geometrical properties of the targets are more distinctive than their surface properties, and surface recognition is the limiting factor in differentiation. The results demonstrate that simple IR sensors, when coupled with appropriate processing and recognition techniques, can be used to extract substantially more information than such devices are commonly employed for.

Keywords: Target differentiation; Geometry differentiation; Surface differentiation; Statistical pattern recognition; Feature extraction; Infrared sensors; Optical sensing

1. Introduction

Target differentiation is of considerable interest for intelligent systems that need to interact with and operate in their environment autonomously. Such systems rely on sensor modules which are often their only available source of information. Since the resources of such systems are limited, the available resources should be used in the best way possible. It is desirable to maximally exploit the capabilities of lower cost sensors before more costly and sophisticated sensors with higher resolution and higher resource requirements are employed. This can be achieved by employing better characterization and physical modeling of these sensors.

Although ultrasonic sensors have been widely used for object detection and ranging, they are limited by their large beam width and the difficulty of interpreting their readings due to specular, higher-order, and multiple reflections from surfaces. Furthermore, many readily available ultrasonic systems cannot detect objects up to 0.5 m which corresponds to their blank-out zone. Therefore, in performing tasks at short distances from objects, use of inexpensive and widely available sensors such as simple infrared detectors are preferable to employing ultrasonic sensors or more costly laser and vision systems. Furthermore, in a sensor-fusion framework, IR sensors would be perfectly complementary to these systems which are not suitable for close-range detection. Infrared detectors offer faster response times and better angular resolution than ultrasonic sensors and provide intensity readings at nearby ranges (typically from a few centimeters up to a meter). The intensity versus range characteristics are nonlinear and dependent on the properties of the surface and environmental conditions. Consequently, a major problem with the use of simple IR detectors is that it is often not possible to make accurate and reliable range estimates

* Corresponding author. Tel.: +90 312 290 2161; fax: +90 312 266 4192.
E-mail address: billur@ee.bilkent.edu.tr (B. Barshan).

0031-3203/$30.00 © 2007 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2007.01.007

based on the value of a single intensity return because the return depends on both the geometry and surface properties of the encountered object. Likewise, the surface properties and the geometry of the target cannot be deduced from simple intensity returns without knowing its position and orientation.

Due to single intensity readings not providing much information about the target properties, recognition capabilities of IR sensors have been underestimated and underused in most work. To achieve accurate results with these sensors, their nonlinear characteristics should be analyzed and modeled based on experimental data. Armed with such characterization and modeling, their potential can be more fully exploited and their usage can be extended beyond simple tasks such as counting and proximity detection. The aim of this study is to maximally realize the potential of these simple sensors so that they can be used in more complicated tasks such as differentiation, recognition, clustering, docking, perception of the environment and surroundings, and map building. For this purpose, we employ various statistical pattern recognition techniques (parametric density estimation, mixture of normals, kernel estimator, *k*-nearest neighbor (*k*-NN), artificial neural network, and support vector machine classifiers) to classify targets with different geometries, different surface properties, and the combination of the two. With the approaches considered in this paper, we can differentiate a moderate number of targets and/or surfaces commonly encountered in indoor environments, using a simple IR system consisting of one emitter and one detector. We provide a comparison of these approaches based on real data acquired from simple IR sensors. The results indicate that if the data acquired from such simple IR sensors are processed effectively through the use of suitable techniques, substantially more information about the environment can be extracted than is commonly achieved with conventional usage.

This paper is organized as follows: In Section 2, we make an introduction to IR sensing and introduce the IR sensor and the experimental set up used in this study. Section 3 summarizes our earlier work on target differentiation with IR sensors. Section 4 provides differentiation of planar surfaces based on parametric modeling of the IR intensity scans. In Section 5, statistical pattern recognition techniques based on the parameterized model are employed for geometry/surface-type determination. A comparison of the different techniques considered is provided in Section 6 together with a discussion. Concluding remarks are made and directions for future research are provided in the same section.

2. Infrared (IR) sensing

Most work on pattern recognition involving infrared deals with the recognition or detection of features or targets in conventional two-dimensional (2D) images. Examples of work in this category include face identification [1], automatic vehicle detection [2], automatic target recognition [3] and tracking [4], detection and identification of targets in background clutter [5], remote sensing, and automated terrain analysis [6]. We note that the differentiation techniques employed in this paper are different from such operations performed on conventional images [7] in that here we work not on direct "photographic" images of the targets obtained by some kind of imaging system, but rather on angular intensity scans obtained by rotating a point sensor. The targets we differentiate are not patterns in a 2D image, but rather objects in space, exhibiting depth, at different positions with respect to the sensing system.

Simple IR sensors are used in robotics and automation, process control, remote sensing, and safety and security systems. More specifically, they have been used in object and proximity detection [8], counting [9], distance and depth monitoring, floor sensing, position measurement and control [10], obstacle/collision avoidance [11], and map building [12]. Other applications include door detection and mapping of openings in walls [13], as well as monitoring doors/windows of buildings and vehicles, and "light curtains" for protecting an area. IR sensors have also been used for automated sorting of waste objects made of different materials [14].

In earlier work [15], we developed a novel range estimation technique which is independent of surface type since it is based on the position of the maximum intensity value instead of surface-dependent absolute intensity values. An intelligent feature of the system is that its operating range is made adaptive based on the maximum intensity of the detected signal.

The IR sensor [16] used in this study is employed in the active mode and consists of an emitter and detector and works with 20–28 V DC input voltage. The emitted light is back-scattered from the target and an analog output voltage, proportional to the reflected light, is measured at the detector. The detector window is covered with an IR filter to minimize the effect of ambient light on the intensity measurements. Indeed, when the emitter is turned off, the detector reading is essentially zero. The sensitivity of the device can be adjusted with a potentiometer to set the operating range of the system. The maximum range of operation of the sensor is about 60 cm. The IR sensor (see Fig. 1(a)) is mounted on a 12 in rotary table [17] to obtain angular intensity scans from the targets. A photograph of the experimental setup and its schematics can be seen in Figs. 1(b) and 2, respectively. Basically, the IR sensor, rotating on the platform, acquires angular scans from targets positioned differently. The different target types are a plane, a 90° corner, a 90° edge, and a cylinder of radius 4.8 cm, each with a height of 120 cm and with cross-sections given in Fig. 3. All targets are made of unpolished oak wood. The horizontal extent of all targets other than the cylinder is large enough that they can be considered infinite and thus edge effects can be ignored. Some example angular intensity scans acquired from these targets are provided in Fig. 4.

3. Review of our earlier work on differentiation with infrared

3.1. Rule-based target differentiation

As a first attempt in differentiation of targets with simple IR sensors, we employed a rule-based approach which is based on

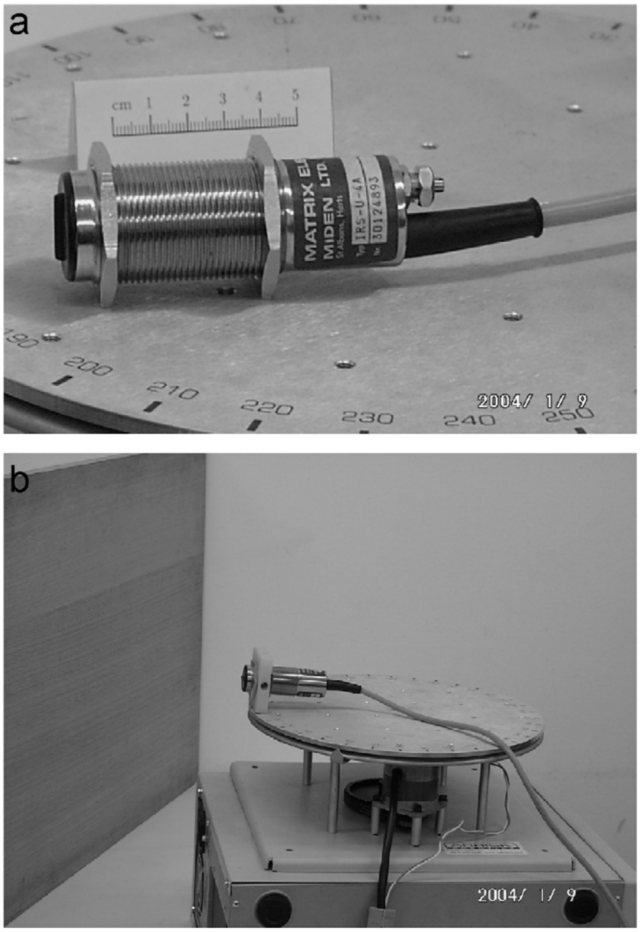

Fig. 1. (a) The IR sensor and (b) the experimental setup used in this study.

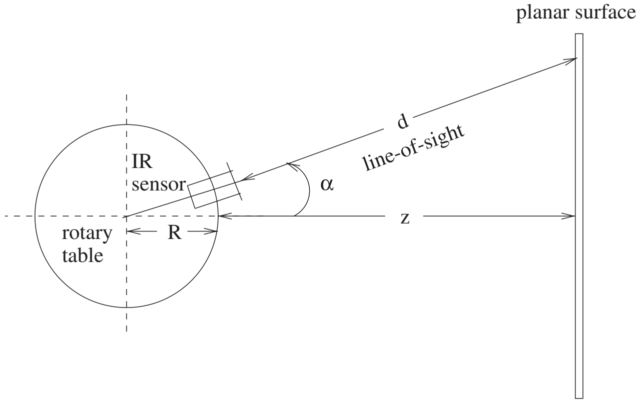

Fig. 2. Top view of the experimental setup. The emitter and detector windows are circular with 8 mm diameter and center-to-center separation 12 mm. (The emitter is above the detector.) Both the scan angle α and the surface azimuth θ are measured counter-clockwise from the horizontal axis.

extracting empirical rules by inspecting the nature of the IR intensity scans. The method can achieve position-invariant target differentiation without relying on the absolute return signal intensities of the IR sensors. Details of the differentiation rules can be found in Ref. [18]. Based on tests with experimental data from four target geometries, an average differentiation rate of 91.3% is achieved.

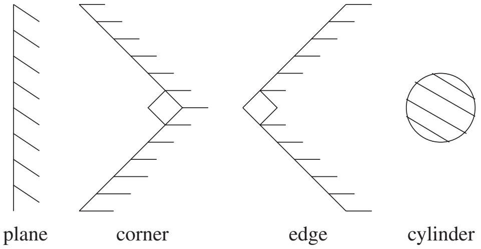

Fig. 3. The different target types considered.

3.2. Template-based target differentiation

The template-based approach is based on comparing the acquired IR intensity scans with previously stored templates acquired from targets with different properties. Hence, this approach relies on the distinctive natures of the IR intensity scans and requires the storage of a complete set of reference scans of interest. We considered the following different cases: targets with different geometrical properties but made of the same surface material [19], targets made of different surface materials but of the same planar geometry [20], and targets with both different geometry and surface properties [21], generalizing and unifying the results of Refs. [19,20].

We experimentally verified the proposed techniques by positioning the targets at randomly selected locations and collecting test scans (see Refs. [19–21] for details). For targets with different geometrical properties (plane, corner, edge, and cylinder) but made of the same surface material (wood) [19], the average correct differentiation rate over all target types is 97%. For different surface materials (aluminum, white wall, brown paper, and Styrofoam) of the same planar geometry, the average correct differentiation rate over all surfaces is 87%. For targets with both different geometry and surface properties (plane, corner, and edge covered with aluminum, white cloth, and Styrofoam), the average accuracy of differentiation over all target types is 80%.

4. Parametric surface differentiation

The parametric approach is based on modeling of IR intensity scans [22]. Light reflected from a surface depends on the wavelength, the distance and the properties of the light source (i.e., point or diffuse source), and the surface properties of the objects such as reflectivity, absorptivity, transmittivity, and orientation. Depending on the surface properties, reflectance can be modeled in different ways [23].

In parametric surface differentiation, only the reflection coefficients corresponding to the reflection model presented below are considered as parameters and used in the differentiation process, instead of using the complete IR intensity scans as in our earlier work reviewed in the previous section. The surface materials considered are unpolished oak wood,

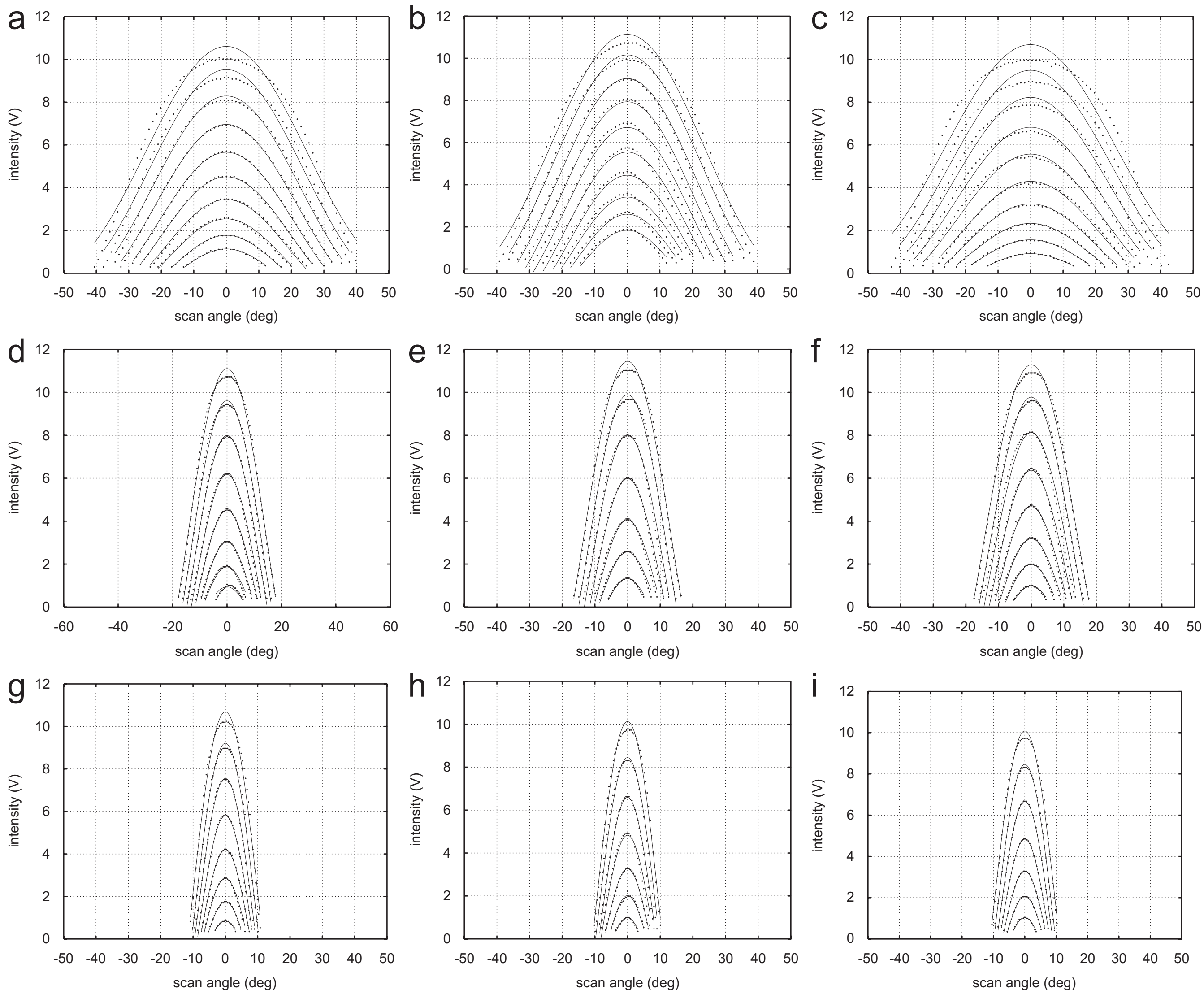

Fig. 4. Intensity scans for targets at different distances. Solid lines indicate the model fit and the dotted lines indicate the actual data. (P, plane; E, edge; CY, cylinder; WD, wood; ST, Styrofoam; WC, white cloth.) (a) P-WD, (b) P-ST, (c) P-WC, (d) E-WD, (e) E-ST, (f) E-WC, (g) CY-WD, (h) CY-ST, and (i) CY-WC.

Styrofoam packaging material, white painted matte wall, white and black cloth, and white, brown, and violet paper (matte).

Reference intensity scans are collected for each surface type by positioning the surfaces over their observable ranges with 2.5 cm distance increments, at $\theta = 0°$. Resulting reference scans for three of the eight surface types and three geometries are shown as examples in Fig. 4 using dotted lines. These intensity scans have been modeled by approximating the surfaces as ideal Lambertian (or diffusely reflecting) surfaces since all of the surface materials involved were matte. The received return signal intensity is proportional to the detector area and is inversely proportional to the square of the distance to the surface and is modeled with three parameters as

$$\mathscr{I} = \frac{C_0 \cos(\alpha C_1)}{[z/\cos\alpha + R(1/\cos\alpha - 1)]^2}. \quad (1)$$

In Eq. (1), the product of the intensity of the light emitted, the area of the detector, and the reflection coefficient of the surface is lumped into the constant C_0, and C_1 is an additional coefficient to compensate for the change in the basewidth of the intensity scans with respect to distance (Fig. 4). A similar dependence on C_1 is used in sensor modeling in Ref. [24]. The z is the horizontal distance between the rotary platform and the surface as shown in Fig. 2. The denominator of $\mathscr{I}$ is the square of the distance d between the IR sensor and the surface. From the geometry of Fig. 2, $d + R = (z + R)/\cos\alpha$, from which we obtain d as $z/\cos\alpha + R(1/\cos\alpha - 1)$, where R is the radius of the rotary platform and α is the angle between the IR sensor and the horizontal.

Using the model represented by Eq. (1), parameterized curves have been fitted to the reference intensity scans by employing a nonlinear least-squares technique based on a model-trust region method provided by MATLAB$^{\mathrm{TM}}$ [25]. Resulting curves are

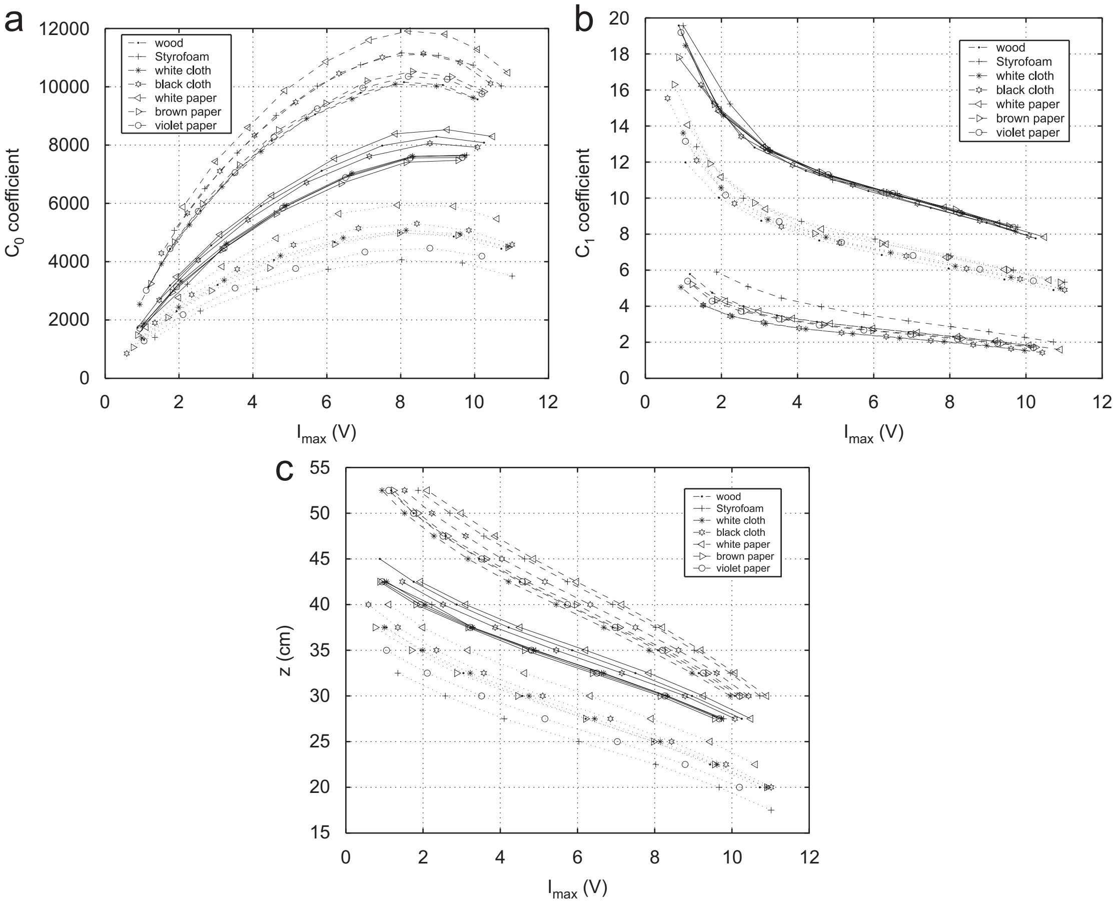

Fig. 5. Variation of the parameters (a) C_0, (b) C_1, and (c) z with respect to maximum intensity (dashed, dotted, and solid lines are for planes, edges, and cylinders, respectively).

shown in Fig. 4 in solid lines. For the reference scans, z is not taken as a parameter since the distance between the surface and the IR sensing unit is already known. The initial guesses of the parameters must be made cleverly so that the algorithm does not converge to local minima and curve fitting is achieved in a smaller number of iterations. The initial guess for C_0 is made by evaluating $\mathscr{I}$ at $\alpha = 0°$, and corresponds to the product of $\mathscr{I}$ with z^2. Similarly, the initial guess for C_1 is made by evaluating C_1 from Eq. (1) at a known angle α different than zero, with the initial guess of C_0 and the known value of z. During curve fitting, C_0 value is allowed to vary between ±2000 of its initial guess and C_1 is restricted to be positive. The variations of C_0, C_1, and z with respect to the maximum intensity of the reference scans are shown in Fig. 5. As the distance d decreases, the maximum intensity increases and C_0 first increases then decreases but C_1 and z both decrease, as expected from the model represented by Eq. (1).

The proposed method is experimentally verified with planar surfaces. In the test process, the surfaces are randomly located at range values between 30 and 52.5 cm and azimuth angles varying from −45° to 45°. First, the maximum intensity of the observed intensity scan is detected and the angular value where this maximum occurs is taken as the azimuth estimate of the surface. Then, the observed scan is shifted by the azimuth estimate and the model represented by Eq. (1) is fitted as described above for the reference scans. The initial guess for the distance z is found from Fig. 5(c) by taking the average of the minimum possible and the maximum possible range values corresponding to the maximum value of the recorded intensity scan. (Linear interpolation is used between the data points in the figure.) This results in a maximum absolute range error of approximately 2.5 cm. Therefore, the parameter z is allowed to vary between ±2.5 cm of its initial guess. Using the initial guess for z, the initial guesses for C_0 and C_1 are made in the same way as explained before for the reference scans. After nonlinear curve fitting to the observed scan, we obtain three parameters C_0, C_1, and z. In the decision process, the maximum intensity of the observed scan is used and a value of C_1 is obtained by linear interpolation between the data points in Fig. 5(b) for each surface type. In other words, Fig. 5(b) is used as a look-up table because of the more distinctive nature of the C_1 variation with respect to maximum intensity. Surface-type decisions are made by comparing the C_1 value of the observed scan with the C_1 values of the reference scans for each surface. The surface type giving the minimum absolute difference is declared to be the correct one.

For a set of six surfaces including Styrofoam packaging material, white painted matte wall, white or black cloth, and white, brown, and violet paper (also matte), we get a correct differentiation rate of 100%. We can increase the number of surfaces differentiated at the expense of a decrease in the correct differentiation rate. For example, if we add wood to our test set keeping either white or black cloth, we get a correct differentiation rate of 86% for the seven surfaces. Similarly, if we form a set of surfaces excluding wood but keeping both white and black cloth, we achieve a correct differentiation rate of 83% for seven surfaces. For all eight surfaces considered, an overall correct differentiation rate of 73% is achieved. The decrease in the differentiation rate resulting from adding new surfaces in the parametric approach does not represent an overall degradation in differentiation rates across all surface types but is almost totally explained by pairwise confusion of the newly introduced surface with a previously existing one, resulting from the similarity of the C_1 parameter of the intensity scans of the two confused surfaces. (Similar decreases in differentiation rate with increasing number of surfaces or objects are also observed with non-parametric template-based approaches.)

In the template-based approach, where we considered differentiation of surfaces by employing non-parametric approaches, a maximum correct differentiation rate of 87% over four surfaces is achieved. The parametric approach can differentiate six different surfaces with 100% accuracy. Comparing the two, we can conclude that the parametric approach is superior to non-parametric ones, in terms of the accuracy, number of surfaces differentiated, and memory requirements, since the non-parametric approaches we considered require the storage of reference scan signals. By parameterizing the intensity scans and storing only their parameters, we have eliminated the need to store complete reference scans.

5. Statistical pattern recognition techniques

In this section, we extend the parametric differentiation approach presented in the previous section to differentiation of the geometry of the target types in parameter space, using statistical pattern recognition techniques. The geometries considered are plane, edge, and cylinder made of unpolished oak wood. The surfaces are either left uncovered (plain wood) or alternatively covered with Styrofoam packaging material, white and black cloth, and white, brown, and violet paper (matte). In the implementation, PRTools [26] is used.

After nonlinear curve fitting to the observed scan as in Section 4, we get three parameters C_0, C_1, and z. We begin by constructing two alternative feature vector representations based on the parametric representation of the IR scans. The *feature vector* $\mathbf{x}$ is a 2×1 column vector comprised of either the $[C_0, I_{\max}]^{\mathrm{T}}$ or the $[C_1, I_{\max}]^{\mathrm{T}}$ pair, illustrated in Figs. 5(a) and (b), respectively. Therefore, the dimensionality d of the feature vector representations is 2.

We associate a class w_i with each target type ($i=1,\ldots,c$). An unknown target is assigned to class w_i if its feature vector $\mathbf{x}=[x_1,\ldots,x_d]^{\mathrm{T}}$ falls in the region Ω_i. A rule which partitions the decision space into regions Ω_i, $i=1,\ldots,c$ is called a *decision rule*. Each one of these regions corresponds to a different target type. Boundaries between these regions are called *decision surfaces*. Let $p(w_i)$ be the a priori probability of a target belonging to class w_i. To classify a target with feature vector $\mathbf{x}$, a posteriori probabilities $p(w_i|\mathbf{x})$ are compared and the target is classified into class w_j if $p(w_j|\mathbf{x}) > p(w_i|\mathbf{x})\ \forall i \neq j$. This is known as *Bayes minimum error rule*. However, since these a posteriori probabilities are rarely known, they need to be estimated. A more convenient formulation of this rule can be obtained by using Bayes' theorem: $p(w_i|\mathbf{x}) = p(\mathbf{x}|w_i)p(w_i)/p(\mathbf{x})$ which results in $p(\mathbf{x}|w_j)p(w_j) > p(\mathbf{x}|w_i)p(w_i)\ \forall i \neq j \Longrightarrow \mathbf{x} \in \Omega_j$ where $p(\mathbf{x}|w_i)$ are the class-conditional probability density functions (CCPDFs) which are also unknown and need to be estimated in their turn based on the training set. The training set consists of several sample feature vectors $\mathbf{x}_n$, $n=1,\ldots,N_i$ which all belong to the same class w_i, for a total of $N_1+N_2+\cdots+N_c=N$ sample feature vectors. The *test set* is then used to evaluate the performance of the decision rule used. This decision rule can be generalized as $q_j(\mathbf{x}) > q_i(\mathbf{x})\ \forall i \neq j \Longrightarrow \mathbf{x} \in \Omega_j$ where the function q_i is called a *discriminant function*.

The various statistical techniques for estimating the CCPDFs based on the training set are often categorized as non-parametric and parametric. In non-parametric methods, no assumptions on the parametric form of the CCPDFs are made; however, this requires large training sets. This is because any non-parametric PDF estimate based on a finite sample is biased [27]. In parametric methods, specific models for the CCPDFs are assumed and then the parameters of these models are estimated. These parametric methods can be categorized as normal and non-normal models.

5.1. Determination of geometry

5.1.1. Parametric classifiers

5.1.1.1. Parameterized density estimation (PDE). In this method, the CCPDFs are assumed to be d-dimensional normal:

$$p(\mathbf{x}|w_i) = \frac{1}{(2\pi)^{(d/2)}|\mathbf{\Sigma}_i|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}\mathbf{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right], \quad i=1,\ldots,c, \tag{2}$$

where the $\boldsymbol{\mu}_i$'s denote the class means, and the $\mathbf{\Sigma}_i$'s denote the class-covariance matrices, both of which must be estimated based on the training set. The most commonly used parameter estimation technique is the maximum likelihood estimator (MLE) [28] which is also used in this study.

In PDE, d-dimensional *homoscedastic* and *heteroscedastic* normal models are used for the CCPDFs. In the homoscedastic case, the covariance matrices for all classes are selected equal, usually taken as a weighted (by a priori probabilities) average of the individual class-covariance matrices: $\sum_{i=1}^{c}(N_i/N)\hat{\mathbf{\Sigma}}_i$ [29]. In the heteroscedastic case, they are individually calculated for each class.

In this study, both homoscedastic and heteroscedastic normal models have been implemented to estimate the means and the covariances of the CCPDF for each class (i.e., target type)

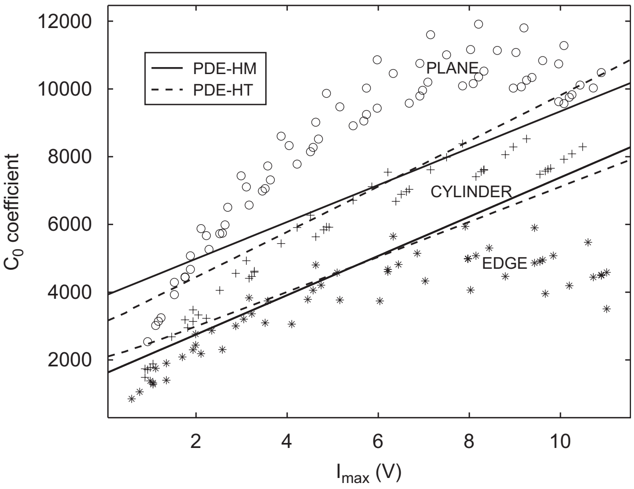

Fig. 6. Discriminant functions for PDE when the $[C_0, I_{\max}]$ feature vector is used.

Table 1
Geometry confusion matrix: homoscedastic PDE using the $[C_0, I_{\max}]^T$ feature vector

Geometry	Differentiation result			Total
	P	E	CY	
P	61(0)	0(50)	9(34)	70(84)
E	0(0)	49(43)	6(0)	55(43)
CY	4(0)	5(84)	41(0)	50(84)
Total	65(0)	54(177)	56(34)	175(211)

Numbers outside (inside) the parentheses are for the training (test) scans (P, plane; E, edge; CY, cylinder).

using the MLE, for each of the two feature vector representations described above. These are the $[C_0, I_{\max}]^T$and $[C_1, I_{\max}]^T$ feature vectors illustrated in Fig. 5(a) and (b), respectively.

The training set consists of $N = 175$ data pairs for three classes: $N_1 = 70$ planes, $N_2 = 55$ edges, and $N_3 = 50$ cylinders. The test set consists of 211 data pairs for three classes: 84 planes, 43 edges, and 84 cylinders. A given test feature vector is classified into the class for which Eq. (2) is maximum.

Since the feature vector size d is two and the number of classes c is three, three 2D normal functions are used for classification. The discriminant functions for PDE are plotted on the training set feature vectors $[C_0, I_{\max}]^T$ in Fig. 6. The classification results are given in Table 1 for both the training and test sets for homoscedastic PDE. Overall correct differentiation rates of 86.3% and 20.4% are achieved for the training and test sets, respectively. The main reason for the low differentiation rate on the test set is due to the $[C_0, I_{\max}]^T$ feature vector of the observed intensity scans not being very distinctive. For the heteroscedastic case, the geometry confusion matrix is given in Table 2. The differentiation rates for this case are the same as in the homoscedastic case.

Equal probability contours of the 2D normal functions are given in Fig. 7(a) and (b) for each case when the $[C_1, I_{\max}]^T$ feature vector is used for differentiation. The corresponding discriminant functions are shown in Fig. 8. From Table 3, the correct differentiation rates for homoscedastic PDE are 96.6% and 98.6% for the training and test sets, respectively. For the test data, only three edges are incorrectly classified as cylinders. For heteroscedastic PDE (Table 4), the differentiation rate on the training set improves to 98.3% and the correct differentiation rate on the test set is the same as in the homoscedastic case.

Table 2
Geometry confusion matrix: heteroscedastic PDE using the $[C_0, I_{\max}]^T$ feature vector

Geometry	Differentiation result			Total
	P	E	CY	
P	60(0)	0(46)	10(38)	70(84)
E	0(0)	51(43)	4(0)	55(43)
CY	4(0)	6(84)	40(0)	50(84)
Total	64(0)	57(173)	54(38)	175(211)

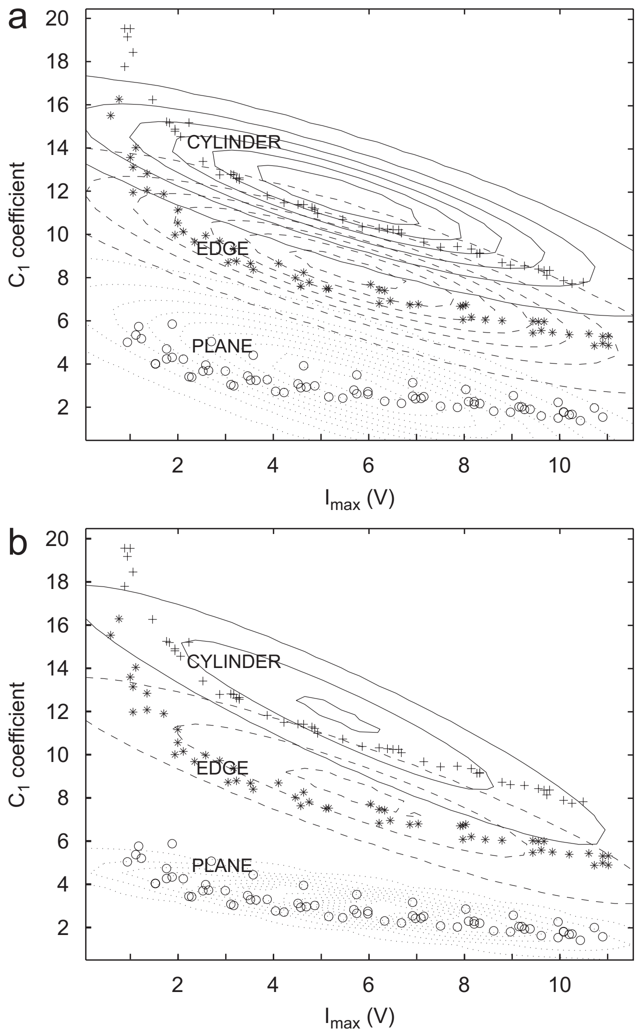

Fig. 7. 2D normal contour plots for (a) homoscedastic (b) heteroscedastic PDE when the $[C_1, I_{\max}]^T$ feature vector is used.

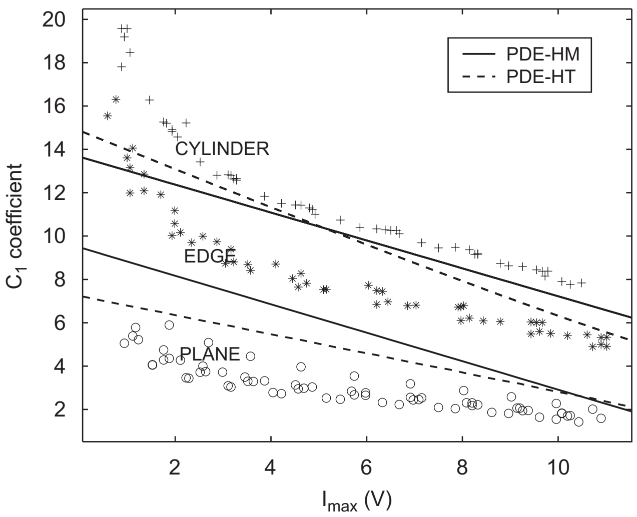

Fig. 8. Discriminant functions for PDE when the $[C_1, I_{\max}]^{\mathrm{T}}$ feature vector is used.

Table 3
Geometry confusion matrix: homoscedastic PDE using the $[C_1, I_{\max}]^{\mathrm{T}}$ feature vector

Geometry	Differentiation result			Total
	P	E	CY	
P	70(84)	0(0)	0(0)	70(84)
E	0(0)	49(40)	6(3)	55(43)
CY	0(0)	0(0)	50(84)	50(84)
Total	70(84)	49(40)	56(87)	175(211)

Table 4
Geometry confusion matrix: heteroscedastic PDE using the $[C_1, I_{\max}]^{\mathrm{T}}$ feature vector

Geometry	Differentiation result			Total
	P	E	CY	
P	70(84)	0(0)	0(0)	70(84)
E	0(0)	52(40)	3(3)	55(43)
CY	0(0)	0(0)	50(84)	50(84)
Total	70(84)	52(40)	53(87)	175(211)

These results are much better than those obtained with the classification based on the $[C_0, I_{\max}]^{\mathrm{T}}$ feature vector. We have also considered the use of the feature vectors $[C_0, C_1, I_{\max}]^{\mathrm{T}}$ and $[C_0, C_1]^{\mathrm{T}}$. However, these did not bring any improvement over those reported. Since the results indicate that C_1 parameter is more distinctive than C_0 in identifying the geometry, from now on, we concentrate on differentiation based on only the $[C_1, I_{\max}]^{\mathrm{T}}$ feature vector. These conclusions have also been verified for the other classifiers considered below.

5.1.1.2. Mixture of normals (MoN) classifier. In the MoN classifier, each feature vector in the training set is assumed to be associated with a mixture of M different and independent normal distributions [30]. Each normal distribution has probability density function p_j with mean vector $\boldsymbol{\mu}_j$ and covariance matrix $\boldsymbol{\Sigma}_j$:

$$p_j(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = \frac{1}{(2\pi)^{(d/2)}|\boldsymbol{\Sigma}_j|^{1/2}} \times \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^{\mathrm{T}}\boldsymbol{\Sigma}_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)\right], \quad j = 1, \ldots, M. \tag{3}$$

The M normal distributions are mixed according to the following model, using the mixing coefficients α_j:

$$p(\mathbf{x}|\boldsymbol{\Theta}) = \sum_{j=1}^{M} \alpha_j p_j(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j). \tag{4}$$

Here, $\boldsymbol{\Theta} = [\alpha_1, \ldots, \alpha_M;\ \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M;\ \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_M]$ is a parameter vector which consists of three sets of parameters and conveniently represents the relevant parameters for the normals to be mixed. The mixing coefficients should satisfy the normalization condition $\sum_{j=1}^{M}\alpha_j = 1$ and $0 \leqslant \alpha_j \leqslant 1\ \ \forall j$ and can be thought of as prior probabilities of each mixture component so that $\alpha_j = \text{Prob}\{j\text{th component}\} = p(j)$ and $\sum_{j=1}^{M} p(j|\mathbf{x}, \boldsymbol{\Theta}) = 1$. In our implementation, M takes the values two and three. For the ith class, the parameter vector $\boldsymbol{\Theta}_i$ maximizing Eq. (4) needs to be estimated, corresponding to the MLE. Since deriving an analytical expression for the MLE is not possible in this case, $\boldsymbol{\Theta}_i$ is estimated by using expected-maximization (E-M) clustering which is iterative [26]. The elements of the parameter vector $\boldsymbol{\Theta}_i$ are updated recursively as follows:

$$\alpha_{ijk} = \frac{1}{N_i}\sum_{n=1}^{N_i} p(j|\mathbf{x}_n, \boldsymbol{\Theta}_{i,k-1}),$$

$$\boldsymbol{\mu}_{ijk} = \frac{\sum_{n=1}^{N_i}\mathbf{x}_n p(j|\mathbf{x}_n, \boldsymbol{\Theta}_{i,k-1})}{\sum_{n=1}^{N_i} p(j|\mathbf{x}_n, \boldsymbol{\Theta}_{i,k-1})},$$

$$\boldsymbol{\Sigma}_{ijk} = \frac{\sum_{n=1}^{N_i}(\mathbf{x}_n - \boldsymbol{\mu}_{ijk})(\mathbf{x}_n - \boldsymbol{\mu}_{ijk})^{\mathrm{T}} p(j|\mathbf{x}_n, \boldsymbol{\Theta}_{i,k-1})}{\sum_{n=1}^{N_i} p(j|\mathbf{x}_n, \boldsymbol{\Theta}_{i,k-1})}$$

$$\text{where } i = 1, \ldots, c \ \text{ and } \ j = 1, \ldots, M. \tag{5}$$

Here, $\boldsymbol{\Theta}_{i,k}$ is the parameter vector estimate of the ith class at the kth iteration step and N_i is the number of feature vectors in the training set representing the ith class. The expectation and maximization steps are performed simultaneously. The algorithm proceeds by using the newly derived parameters as the guess for the next iteration. With E-M clustering, even if the dimensionality of the feature vectors increases, fast and reliable parameter estimation can be accomplished.

Each class is considered independent from the others and training is performed separately for each class. For this reason, addition of new classes can be done conveniently by adding the corresponding feature vectors to the training data set and estimating the corresponding class parameter vector.

After estimating the parameter vectors for each class based on the training set feature vectors, testing is done as follows:

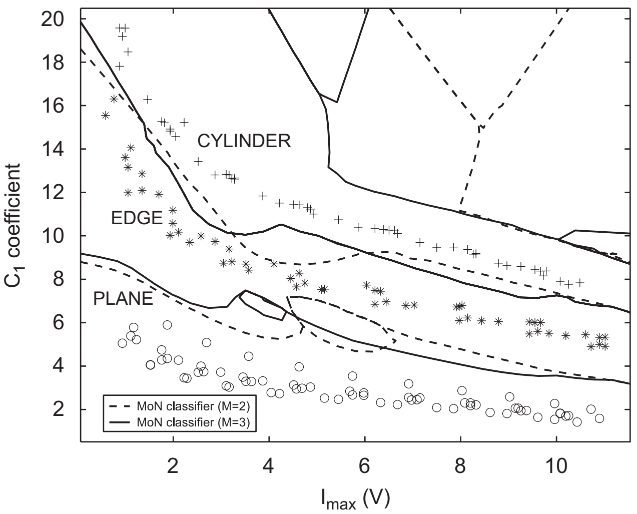

Fig. 9. Discriminant functions for the MoN classifier when the $[C_1, I_{\max}]^{\mathrm{T}}$ feature vector is used.

Table 5
Geometry confusion matrix: MoN classifier ($M = 3$) using the $[C_1, I_{\max}]^{\mathrm{T}}$ feature vector

Geometry	Differentiation result			Total
	P	E	CY	
P	70(84)	0(0)	0(0)	70(84)
E	0(0)	55(43)	0(0)	55(43)
CY	0(0)	0(0)	50(84)	50(84)
Total	70(84)	55(43)	50(84)	175(211)

a target with a given test feature vector $\mathbf{x}$ is assigned to the class whose parameter vector $\mathbf{\Theta}_i$ maximizes Eq. (4) so that $p(\mathbf{x}|\mathbf{\Theta}_i) > p(\mathbf{x}|\mathbf{\Theta}_l) \quad \forall i \neq l$. Then, the target is labeled as a member of class w_i.

The discriminant functions for classification based on $[C_1, I_{\max}]^{\mathrm{T}}$ feature vector are shown in Fig. 9. Differentiation results for $M = 3$ are given in Table 5 in the form of a confusion matrix. For both $M = 2$ and 3, all training targets are correctly classified using the $[C_1, I_{\max}]^{\mathrm{T}}$ feature vector. In the tests, for the $M = 3$ case (Table 5, in parentheses), again 100% correct differentiation rate is achieved. For the $M = 2$ case, the only difference in the test results is that one of the edges is misclassified as a cylinder so that the correct classification rate falls to 99.5%.

5.1.2. Non-parametric classifiers

In this section, we consider different non-parametric classifiers, which are the kernel estimator, k-NN, artificial neural network, and support vector machine classifiers.

5.1.2.1. Kernel estimator (KE). KE is a family of PDF estimators first proposed by Fix and Hodges in 1951 [31]. In the KE method, the CCPDF estimates $\hat{p}(\mathbf{x}|w_i)$ are of the form

$$\hat{p}(\mathbf{x}|w_i) = \frac{1}{N_i h_i^d} \sum_{n=1}^{N_i} K\left(\frac{\mathbf{x} - \mathbf{x}_n}{h_i}\right), \quad i = 1, \ldots, c, \tag{6}$$

where $\mathbf{x}$ is the d-dimensional feature vector at which the estimate is being made and $\mathbf{x}_n$, $n = 1, \ldots, N_i$ are the training set sample feature vectors associated with class w_i. Here, h_i is called the *spread* or *smoothing parameter* or the *bandwidth* of the KE, and $K(\mathbf{z})$ is a kernel function which satisfies the conditions $K(\mathbf{z}) \geqslant 0$ and $\int K(\mathbf{z})\,\mathrm{d}\mathbf{z} = 1$. In this method, the selection of the bandwidth h_i is important [32,33]. If h_i is selected too small, $\hat{p}(\mathbf{x}|w_i)$ degenerates into a collection of N_i sharp peaks, each located at a sample feature vector. On the other hand, if h_i is selected too large, the estimate is oversmoothed and an almost uniform CCPDF results. Usually, h_i is chosen as a function of N_i such that $\lim_{N_i \to \infty} h(N_i) = 0$.

In the implementation of this method, since $d = 2$, we employed a 2D normal kernel function. The bandwidth h_i for the ith class is pre-computed based on the N_i sample feature vectors available for this class by optimization with respect to *leave-one-out* error [26]. After h_i's are computed, a test feature vector $\mathbf{x}$ is classified into that class for which the CCPDF in Eq. (6) is maximized. This requires the training data to be stored throughout testing.

5.1.2.2. k-Nearest neighbor (k-NN) classifier. Consider the k-nearest neighbors of a feature vector $\mathbf{x}$ in a set of several feature vectors. Suppose k_i of these k vectors come from class w_i. Then, a k-NN estimator for class w_i can be defined as $\hat{p}(w_i|\mathbf{x}) = k_i/k$, and $\hat{p}(\mathbf{x}|w_i)$ can be obtained from $\hat{p}(\mathbf{x}|w_i)\hat{p}(w_i) = \hat{p}(w_i|\mathbf{x})\hat{p}(\mathbf{x})$. This results in a classification rule such that $\mathbf{x}$ is classified into class w_j if $k_j = \max_i(k_i)$, where $i = 1, \ldots, c$. In other words, the k-NNs of the vector $\mathbf{x}$ in the training set are considered and the vector $\mathbf{x}$ is classified into the same class as the majority of its k-NNs.

A major disadvantage of this method is that a pre-defined rule for the selection of the value of k does not exist [34]. In this study, the number of nearest neighbors k is determined by optimization with respect to leave-one-out error. In the implementation of the k-NN classifier, k values varying between 1 and 12 have been considered. For $k = 1, 2$, and 3, the same correct differentiation rates (given below) are obtained for the training and test sets. For larger values of k, the errors start increasing. Again, the training data must be stored during testing.

In Fig. 10, the discriminant functions for both the KE and the k-NN classifier are illustrated when the $[C_1, I_{\max}]^{\mathrm{T}}$ feature vector is used. For both classifiers, the training targets are correctly differentiated with 100% correct differentiation rate. For the test targets, only one edge target is incorrectly classified as a cylinder, corresponding to a correct differentiation rate of 99.5%.

5.1.2.3. Artificial neural network (ANN) classifiers. Feed-forward ANNs trained with back-propagation (BP) and Levenberg–Marquardt (LM) algorithms, and a linear perceptron (LP) are used as classifiers. The feed-forward ANN has

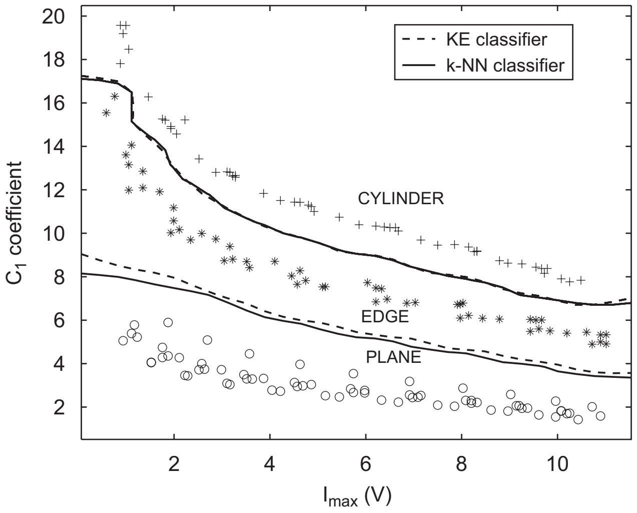

Fig. 10. Discriminant functions for the KE and the *k*-NN classifiers when the $[C_1, I_{\max}]^T$ feature vector is used.

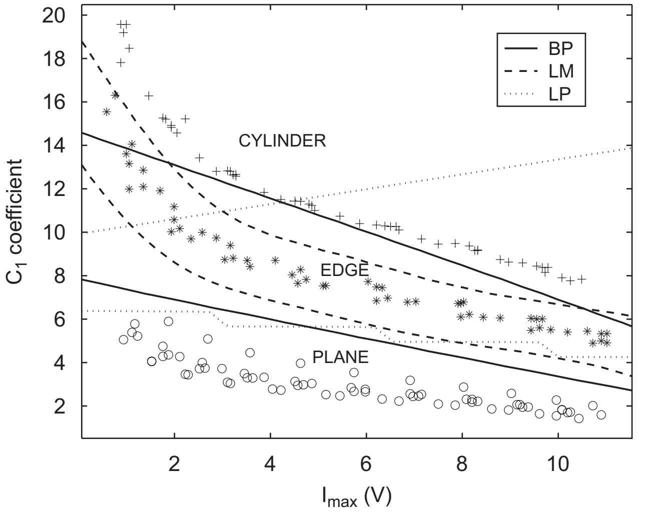

Fig. 11. Discriminant functions for ANN classifiers when the $[C_1, I_{\max}]^T$ feature vector is used.

Table 6
Geometry confusion matrix: ANN trained with BP

Geometry	Differentiation result			Total
	P	E	CY	
P	70(84)	0(0)	0(0)	70(84)
E	0(0)	52(40)	3(3)	55(43)
CY	0(0)	0(0)	50(84)	50(84)
Total	70(84)	52(40)	53(87)	175(211)

Table 7
Geometry confusion matrix: ANN trained with LM

Geometry	Differentiation result			Total
	P	E	CY	
P	70(84)	0(0)	0(0)	70(84)
E	0(0)	52(42)	3(1)	55(43)
CY	0(0)	0(0)	50(84)	50(84)
Total	70(84)	52(42)	53(85)	175(211)

Table 8
Geometry confusion matrix: LP

Geometry	Differentiation result			Total
	P	E	CY	
P	70(84)	0(0)	0(0)	70(84)
E	0(0)	45(32)	10(11)	55(43)
CY	0(0)	29(39)	21(45)	50(84)
Total	70(84)	74(71)	31(56)	175(211)

one hidden layer with four neurons. The number of neurons in the input layer is two (since the feature vector consists of two parameters) and the number of neurons in the output layer is three. LP is the simplest type of ANN, used for classification of two classes that are linearly separable. LP consists of a single neuron with adjustable input weights and a threshold value [35]. If the number of classes is greater than two, LPs are used in parallel. One perceptron is used for each output. The maximum number of epochs is chosen as 1000. The weights are initialized randomly and the learning rate is chosen as 0.1. MATLAB$^{\mathrm{TM}}$ Neural Network Toolbox is used for the implementation. The discriminant functions are given in Fig. 11 for the three classifiers. The correct differentiation rates using the BP algorithm are given in Table 6. Differentiation rates of 98.3% and 98.6% are achieved for the training and test sets, respectively. When training is done by LM, the same correct differentiation rate is obtained on the training set (see Table 7). However, this classifier is better than the BP method in the tests, where only one edge target ismisclassified as a cylinder, resulting in a correct differentiation rate of 99.5%. The results for the LP classifier are given in Table 8. As expected from the distribution of the parameters, because the classes are not linearly separable, lower correct differentiation rates of 77.7% and 76.3% are achieved on the training and test sets, respectively.

5.1.2.4. Support vector machine (SVM) classifier. SVM classifier is a machine learning technique proposed early in the eighties [36]. It has been used in applications such as object, voice, and handwritten character recognition, and text classification.

If the feature vectors in the original feature space are not linearly separable, SVMs preprocess and represent them in a space of higher dimension where they become linearly separable. The dimension of the transformed space may sometimes be much higher than the original feature space. With a suitable nonlinear mapping $\phi(.)$ to a sufficiently high dimension, data from two different classes can always be made linearly separable, and separated by a hyperplane. The choice of the nonlinear mapping depends on the prior information available to the designer. If such information is not available, one might choose to use polynomials, Gaussians, or other types of basis functions. The dimensionality of the mapped space can be arbitrarily high. However, in practice, it may be limited by computational resources. The complexity of SVMs is related to the number of resulting support vectors rather than the high dimensionality of the transformed space.

Consider SVMs in a binary classification setting. We are given the training feature vectors $\mathbf{x}_i$ that are vectors in some space $\mathscr{X} \subseteq \Re^d$ and their labels $l_i \in \{-1, 1\}$ where $i=1, \ldots, N$. The goal in training a SVM is to find the separating hyperplane with the largest margin so that the generalization of the classifier is better. All vectors lying on one side of the hyperplane are labeled as $+1$, and all vectors lying on the other side are labeled as -1. The support vectors are the (transformed) training patterns that lie closest to the hyperplane and are at equal distance from it. They correspond to the training samples that define the optimal separating hyperplane and are the most difficult patterns to classify, yet the most informative for the classification task.

More generally, SVMs allow one to project the original training data in space $\mathscr{X}$ to a higher dimensional feature space $\mathscr{F}$ via a Mercer kernel operator K [37]. We consider a set of classifiers of the form $f(\mathbf{x}) = \sum_{i=1}^{N} \beta_i K(\mathbf{x}_i, \mathbf{x})$. When $f(\mathbf{x}) \geqslant 0$, we label $\mathbf{x}$ as $+1$, otherwise as -1. When K satisfies Mercer's condition, $K(\mathbf{u}, \mathbf{v}) = \phi(\mathbf{u}) \cdot \phi(\mathbf{v})$ where $\phi(.) : \mathscr{X} \rightarrow \mathscr{F}$ is a nonlinear mapping and "·" denotes the inner product. We can then rewrite $f(\mathbf{x})$ as $f(\mathbf{x}) = \mathbf{a} \cdot \phi(\mathbf{x})$, where $\mathbf{a} = \sum_{i=1}^{N} \beta_i \; \phi(\mathbf{x}_i)$ is a weight vector. Thus, by using K, the training data is projected into a new feature space $\mathscr{F}$ which is often higher dimensional. The SVM then computes the β_i's that correspond to the maximal margin hyperplane in $\mathscr{F}$. By choosing different kernel functions, we can project the training data from $\mathscr{X}$ into spaces $\mathscr{F}$ for which hyperplanes in $\mathscr{F}$ correspond to more complex decision boundaries in the original space $\mathscr{X}$. Hence, by nonlinear mapping of the original training patterns into other spaces, decision functions can be found using a linear algorithm in the transformed space by only computing the kernel $K(\mathbf{x}_i, \mathbf{x})$.

The function $f(\mathbf{x}) = \mathbf{a} \cdot \phi(\mathbf{x})$ is a linear discriminant function in the transformed space based on the hyperplane $\mathbf{a} \cdot \phi(\mathbf{x}) = 0$. Here, both the weight vector and the transformed feature vector have been augmented by one dimension to include a bias weight so that the hyperplanes need not pass through the origin. A separating hyperplane ensures

$$l_i f(\mathbf{x}_i) = l_i \mathbf{a} \cdot \phi(\mathbf{x}_i) \geqslant 1 \quad \text{for } i = 1, \ldots, N. \tag{7}$$

It can be shown that finding the optimal hyperplane corresponds to minimizing the magnitude of the weight vector $\|\mathbf{a}\|^2$ subject to the constraint given by Eq. (7) [38]. Using the method of Lagrange multipliers, we construct the functional

$$L(\mathbf{a}, \lambda) = \frac{1}{2}\|\mathbf{a}\|^2 - \sum_{i=1}^{N} \lambda_i [l_i \mathbf{a} \cdot \phi(\mathbf{x}_i) - 1], \tag{8}$$

where the second term in the above equation expresses the goal of classifying the points correctly. To find the optimal hyperplane, we minimize $L(\mathbf{a}, \lambda)$ with respect to the weight vector $\mathbf{a}$, while maximizing with respect to the undetermined Lagrange multipliers $\lambda_i \geqslant 0$. This can be done by solving the constrained optimization problem by quadratic programming [26] or by other alternative techniques. The solution of the weight vector is $\mathbf{a}^* = \sum_{i=1}^{N} l_i \lambda_i \phi(\mathbf{x}_i)$ corresponding to $\beta_i = l_i \lambda_i$. Then the decision function is given by

$$f^*(\mathbf{x}) = \sum_{i=1}^{N} \lambda_i l_i \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}). \tag{9}$$

In this study, the method described above is applied to differentiate target feature vectors from multiple classes. Following the one-versus-rest method, c different binary classifiers are trained, where each classifier recognizes one of c target types. SVM classifiers with polynomial, exponential, and radial basis function kernels are used. The kernel functions are $K_p(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x} \cdot \mathbf{x}_i + 1)^3$, $K_e(\mathbf{x}, \mathbf{x}_i) = \mathrm{e}^{-\|\mathbf{x}-\mathbf{x}_i\|}$, $K_r(\mathbf{x}, \mathbf{x}_i) = \mathrm{e}^{-\|\mathbf{x}-\mathbf{x}_i\|^2}$, respectively [26]. The dimension of the feature space $\mathscr{F}$ is 3. Hundred percent correct differentiation rate is achieved on the training set for all of the SVM classifiers. For the test set, the correct differentiation rates are 99.5%, 99.5%, and 99.1% for SVM classifiers with polynomial, exponential, and radial basis function kernels, respectively. Therefore, the polynomial and exponential kernels result in the highest classification rates.

To summarize the results of the statistical pattern recognition techniques for geometry classification based on the $[C_1, I_{\max}]^{\mathrm{T}}$ feature vector, the overall differentiation rates are given in Table 9. Best classification rate is obtained for the test scans

Table 9
Correct differentiation percentages for different classifiers

Data set	Classification techniques											
	PDE-HM	PDE-HT	MoN-2	MoN-3	KE	k-NN	ANN-BP	ANN-LM	ANN-LP	SVM-P	SVM-E	SVM-R
Training	96.6	98.3	100	100	100	100	98.3	98.3	77.7	100	100	100
Test	98.6	98.6	99.5	100	99.5	99.5	98.6	99.5	76.3	99.5	99.5	99.1

PDE-HM, parametric density estimation-homoscedastic; PDE-HT, parametric density estimation-heteroscedastic; MoN-2, mixture of normals with two components; MoN-3, mixture of normals with three components; KE, kernel estimator; k-NN, k-nearest neighbor; ANN-BP, ANN trained with BP; ANN-LM, ANN trained with LM; ANN-LP, ANN trained with LP; SVM-P, SVM with polynomial kernel; SVM-E, SVM with exponential kernel; SVM-R, SVM with radial kernel.

Table 10
Overview of the differentiation techniques compared

Differentiation technique	Type of geometry	Type of surface	Feature	Correct diff. (%)	Training data	Learning	Parametric
Rule-based [18]	P,C,E,CY	WD	Geometry	91.3	Used, not stored	No	No
Template-based					Used	No	No
[19]	P,C,E,CY	WD	Geometry	97			
[20]	P	AL,WW,BR,ST	Surface	87			
[21]	P,C,E	AL,WC,ST	Geometry	99			
”	P,C,E	”	Surface	81			
”	P,C,E	”	Geometry + Surface	80			
Parametric [22]	P	ST,WW,WC(BC), WP,BR,VP	Surface	100	Used, not stored	Yes	Yes
	”	ST,WW,WC(BC), WP,BR,VP,WD	”	86			
	”	ST,WW,WC,BC, WP,BR,VP	”	83			
	”	ST,WW,WC,BC, WP,BR,VP,WD	”	73			
Statistical Pattern recognition	P,E,CY	ST,WC,BC, WP,BR,VP,WD	Geometry				
PDE-HM, PDE-HT	”	”	”	98.6	Used, not stored	No	Yes
MoN-3	”	”	”	100	Used, not stored	No	Yes
KE	”	”	”	99.5	Used, stored	No	No
k-NN	”	”	”	99.5	Used, stored	No	No
NN-LM	”	”	”	99.5	Used, not stored	Yes	No
SVM-P, SVM-E	”	”	”	99.5	Used, not stored	No	No

P, plane; C, corner; E, edge; CY, cylinder; AL, Aluminum; WD; wood; WC, white cloth; BC: black cloth; WW, white wall; WP, white paper; BR, brown paper; VP, violet paper.

using the MoN classifier with three components. This is followed by MoN with two components, KE, *k*-NN, and SVM with polynomial and exponential kernels, equally. Ranking according to highest classification rate continues as ANN trained with LM algorithm, SVM with radial kernel, heteroscedastic and homoscedastic PDE, ANN trained with BP, and ANN trained with LP.

5.2. Determination of surface type

Parametric surface differentiation is a more difficult problem than geometry differentiation. This is clearly seen in the very similar variation of the parameters for different surfaces corresponding to the same geometry (Fig. 5). In Ref. [22], planar surfaces covered with six different surfaces are correctly classified with 100% correct differentiation rate. Although we succeeded with surface differentiation for planar surfaces, the surface differentiation results for other geometries were not as good. The above classification approaches were applied to differentiate between surface types assuming the geometry of the targets is determined correctly beforehand. For example, for cylindrical targets, the classification error is about 85% when PDE is used. Since the results were not promising, no further attempt has been made to differentiate surface types in parametric space.

As an alternative, we extracted features from these intensity scans corresponding to different surfaces of the same geometry using *forward feature selection*. Since the magnitude and basewidth of intensity scans both change with distance, the intensity scans are first normalized before feature extraction. We experimented with different features of the intensity scans by extracting the points representing the intensity scans best and using them for differentiation. However, the differentiation results were not promising. For example, for cylindrical targets, the surfaces are correctly classified only with a correct differentiation rate of 20%. Different initialization procedures did not result in any improvement in feature extraction.

6. Discussion and conclusion

We extended the parametric surface differentiation approach proposed in Ref. [22] to differentiate both the geometry and surface type of the targets using statistical pattern recognition techniques. We compared different classifiers such as PDE, MoN, kernel estimator, *k*-NN, ANN, and SVM for geometry-type determination. Best differentiation rates (100%) are obtained for the MoN classifier with three components. MoN classifier performs better than models which associate the data with a single distribution. It is also more robust and the training set can be easily updated when new classes need to be added to the database.

Table 10 summarizes the results for all of the differentiation techniques considered in this study and in our earlier related works, allowing for their overall comparison. Only the best differentiation rates are given for the different variations of the methods considered.

In geometry classification, the greatest difficulty is encountered in the differentiation of edges of different surface types. Surface differentiation was not as successful as geometry differentiation due to the similar characteristics of the feature vectors of different surface types for non-planar geometries. The results indicate that the geometrical properties of the targets are more distinctive than their surface properties, and surface determination is the limiting factor in differentiation. Based on the data we have collected and the differentiation results, it seems possible to increase the vocabulary of different geometries, provided they are not too similar. However, the same cannot be said for the number of different surfaces. For a given total number of distinct targets, increasing the number of surfaces and decreasing the number of geometries will in general make the results worse. On the other hand, decreasing the number of surfaces and increasing the number of geometries will in general improve the results.

This work demonstrates that simple IR sensors, when coupled with appropriate processing and recognition techniques, can be used to extract substantially more information about the environment than such devices are commonly employed for. This will allow the possible applications to go beyond relatively simple tasks such as simple object and proximity detection, counting, distance and depth monitoring, floor sensing, position measurement, obstacle/collision avoidance. As such, dealing with tasks such as differentiation, classification, recognition, clustering, position estimation, map building, perception of the environment and surroundings, autonomous navigation, and target tracking will be possible.

The demonstrated system would find application in intelligent autonomous systems such as mobile robots whose task involves surveying an unknown environment made of different surface types. Industrial applications where different materials/surfaces must be identified and separated may also benefit from this approach.

Given the attractive performance-for-cost of IR-based systems, we believe that the results of this study will be useful for engineers designing or implementing IR systems and researchers investigating algorithms and performance evaluation of such systems. While we have concentrated on IR sensing, the techniques evaluated and compared in this paper may be useful for other sensing modalities and environments where the objects are characterized by complex signatures and the information from a multiplicity of partial viewpoints must be combined and resolved.

References

[1] P.J. Phillips, Matching pursuit filters applied to face identification, IEEE Trans. Image Process. 7 (8) (1998) 1150–1164.

[2] I. Pavlidis, P. Symosek, B. Fritz, M. Bazakos, N. Papanikolopoulos, Automatic detection of vehicle occupants: the imaging problem and its solution, Mach. Vision Appl. 11 (6) (2000) 313–320.

[3] H. Kwon, S.Z. Der, N.M. Nasrabadi, Adaptive multisensor target detection using feature-based fusion, Opt. Eng. 41 (1) (2002) 69–80.

[4] T. Tsao, Z.Q. Wen, Image-based target tracking through rapid sensor orientation change, Opt. Eng. 41 (3) (2002) 697–703.

[5] Z. Zalevsky, D. Mendlovic, E. Rivlin, S. Rotman, Contrasted statistical processing algorithm for obtaining improved target detection performances in infrared cluttered environment, Opt. Eng. 39 (10) (2000) 2609–2617.

[6] B. Bhanu, P. Symosek, S. Das, Analysis of terrain using multispectral images, Pattern Recognition 30 (2) (1997) 197–215.

[7] F.T.S. Yu, S. Yin, (Ed.), Selected Papers on Optical Pattern Recognition, SPIE Milestone Series, vol. MS 156, Bellingham, SPIE Optical Engineering Press, Washington, 1999.

[8] E. Cheung, V.J. Lumelsky, Proximity sensing in robot manipulator motion planning: system and implementation issues, IEEE Trans. Robotics Autom. 5 (6) (1989) 740–751.

[9] A.J. Hand, Infrared sensor counts insects, Photonics Spectra 32 (11) (1998) 30–31.

[10] B. Butkiewicz, Position control system with fuzzy microprocessor AL220, Lecture Notes in Computer Science, vol. 1226, 1997, pp. 74–81.

[11] V.J. Lumelsky, E. Cheung, Real-time collision avoidance in teleoperated whole-sensitive robot arm manipulators, IEEE Trans. Syst. Man Cybern. 23 (1) (1993) 194–203.

[12] H.-H. Kim, Y.-S. Ha, G.-G. Jin, A study on the environmental map building for a mobile robot using infrared range-finder sensors, in: Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, USA, 27–31 October 2003, pp. 711–716.

[13] A.M. Flynn, Combining sonar and infrared sensors for mobile robot navigation, Int. J. Robotics Res. 7 (6) (1988) 5–14.

[14] D.M. Scott, A 2-color near-infrared sensor for sorting recycled plastic waste, Meas. Sci. Technol. 6 (2) (1995) 156–159.

[15] Ç. Yüzbaşıoğlu, B. Barshan, Improved range estimation using simple infrared sensors without prior knowledge of surface characteristics, Meas. Sci. Technol. 16 (7) (2005) 1395–1409.

[16] Matrix Elektronik, AG, Kirchweg 24 Oberehrendingen, Switzerland, IRS-U-4A Proximity Switch Datasheet, 1995.

[17] Arrick Robotics, P.O. Box 1574, Hurst, TX, 76053 URL ⟨www.robotics.com/rt12.html⟩, RT-12 Rotary Positioning Table, 2002.

[18] T. Aytaç, B. Barshan, Rule-based target differentiation and position estimation based on infrared intensity measurements, Opt. Eng. 42 (6) (2003) 1766–1771.

[19] T. Aytaç, B. Barshan, Differentiation and localization of targets using infrared sensors, Opt. Comm. 210 (1–2) (2002) 25–35.

[20] B. Barshan, T. Aytaç, Position-invariant surface recognition and localization using infrared sensors, Opt. Eng. 42 (12) (2003) 3589–3594.

[21] T. Aytaç, B. Barshan, Simultaneous extraction of geometry and surface properties of targets using simple infrared sensors, Opt. Eng. 43 (10) (2004) 2437–2447.

[22] T. Aytaç, B. Barshan, Surface differentiation by parametric modeling of infrared intensity scans, Opt. Eng. 44(6) (2005), 067202–1–9.

[23] S.K. Nayar, K. Ikeuchi, T. Kanade, Surface reflection: physical and geometrical perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 13 (7) (1991) 611–634.

[24] G. Petryk, M. Buehler, Dynamic object localization via a proximity sensor network, in: Proceedings of IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems, Washington, DC, 8–11 December 1996, pp. 337–341.

[25] T. Coleman, M.A. Branch, A. Grace, MATLAB Optimization Toolbox, User's Guide, 1999.

[26] R.P.W. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D.M.J. Tax, A Matlab Toolbox for Pattern Recognition, PRTools4, 2004.

[27] M. Rosenblatt, Remarks on some nonparametric estimates of a density function, Ann. Math. Statist. 27 (3) (1956) 832–837.

[28] V.K. Rohatgi, An Introduction to Probability Theory and Mathematical Statistics, Wiley, New York, 1976.

[29] G.J. McLachlan, Discriminant Analysis and Statistical Pattern Recognition, Wiley, New York, 1992.

[30] J. Bilmes, A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models, Technical Report ICSI-TR-97-021, University of Berkeley, 1997.

[31] E. Fix and J.L. Hodges, Discriminatory analysis, nonparametric discrimination, consistency properties, Technical Report 21-49-004 4, United States Air Force, School of Aviation Medicine, Randolph Field, TX, 1951.

[32] M.P. Wand, M.C. Jones, Kernel Smoothing, Chapman & Hall, London, 1995.

[33] S.J. Sheather, The performance of six popular bandwidth selection methods on some real data sets, Comput. Statist. 7 (1992) 225–250.

[34] B.W. Silverman, Density Estimation for Statistics and Data Analysis. Chapman & Hall, New York, 1986.

[35] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice-Hall, Englewood Clifts, NJ, 1994.

[36] V. Vapnik, Estimation of Dependences Based on Empirical Data (in Russian), Moscow: Nauka, 1979 (English translation: Springer, New York, 1982).

[37] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, J. Mach. Learn. Res. 2 (2001) 45–66.

[38] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Wiley, New York, 2001.

About the Author—BILLUR BARSHAN received the B.S. degrees in both electrical engineering and physics from Boğaziçi University, Istanbul, Turkey and the M.S. and Ph.D. degrees in electrical engineering from Yale University, New Haven, Connecticut, in 1986, 1988, and 1991, respectively. Dr. Barshan was a research assistant at Yale University from 1987 to 1991, and a postdoctoral researcher at the Robotics Research Group at University of Oxford, UK from 1991 to 1993. In 1993, she joined Bilkent University, Ankara where she is currently a professor in the Department of Electrical Engineering. Dr. Barshan is the founder of the Robotics and Sensing Laboratory in the same department. She is the recipient of the 1994 Nakamura Prize awarded to the most outstanding paper at the 1993 IEEE/RSJ Intelligent Robots and Systems International Conference, the 1998 TÜBİTAK Young Investigator Award, and the 1999 Mustafa N. Parlar Foundation Research Award. Dr. Barshan's current research interests include intelligent sensors, sonar and inertial navigation systems, sensor-based robotics, and multi-sensor data fusion.

About the Author—TAYFUN AYTAÇ received the B.S. degree in electrical engineering from Gazi University, Ankara, Turkey in 2000 and the M.S. and Ph.D. degrees in electrical engineering from the Bilkent University, Ankara, Turkey in 2002 and 2006, respectively. He is currently a research scientist at TÜBITAK-UEKAE-ILTAREN Research Group, Ankara, Turkey. His current research interests are intelligent sensing, optical sensing, pattern recognition, sensor data fusion, target differentiation, and sensor-based robotics.

About the Author—ÇAĞRI YÜZBAŞIOĞLU received the B.S. degree in electrical engineering from Middle East Technical University, Ankara, Turkey in 2002 and the M.S. degree in electrical engineering from Bilkent University, Ankara, Turkey in 2004. His current research interests include intelligent sensing, optical sensing, pattern recognition, sensor data fusion, target differentiation, and sensor-based robotics.