2006 IEEE International Conference on
Multisensor Fusion and Integration for Intelligent Systems
September 3-6, 2006, Heidelberg, Germany

WeB02.3

Statistical Pattern Recognition Techniques
for Target Differentiation using Infrared Sensor

Tayfun Aytaç, Çağrı Yüzbaşıoğlu, and Billur Barshan

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 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. 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.**

I. INTRODUCTION

Target differentiation is of considerable interest for the autonomous operation of intelligent systems [1, 2, 3]. Differentiation is also important in industrial applications where different materials must be identified and separated. In this study, we achieve differentiation of commonly encountered features in indoor environments with a simple IR sensing system consisting of one emitter and one detector. These devices are inexpensive, practical, and widely available. The emitted light is reflected from the target and its intensity is measured at the detector. However, it is often not possible to make reliable distance estimates based on the value of a single intensity return because the return depends on both the geometry and surface properties of the reflecting target. Likewise, the properties of the target cannot be deduced from simple intensity returns without knowing its distance and angular location. In this paper, we consider statistical pattern recognition techniques (parametric density estimation, mixture of normals, kernel estimator, k-nearest neighbor, artificial neural network, and support vector machine classifiers) for target differentiation. We provide a comparison of these approaches based on real data acquired from simple IR sensors.

T. Aytaç and Ç. Yüzbaşıoğlu are with Havelsan Inc., TR-06520, Ankara, Turkey ryuzbasioglu@havelsan.com.tr

T. Aytaç and B. Barshan are with the Department of Electrical Engineering, Bilkent University, TR-06800, Bilkent, Ankara, Turkey {taytac,billur}@ee.bilkent.edu.tr

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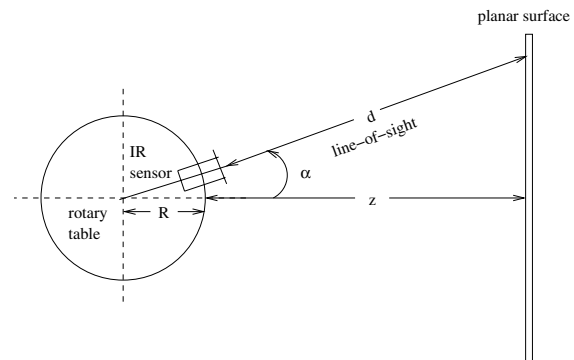

Fig. 2.   Top view of the experimental setup.

II. IR SENSING

IR sensors are used in robotics and automation, process control, remote sensing, and safety and security systems. More specifically, they have been used in simple object and proximity detection [4], counting, distance and depth monitoring, floor sensing, position measurement and control, obstacle/collision avoidance [5], and map building [6]. IR sensors are used in door detection and mapping of openings in walls [7], as well as monitoring doors/windows of buildings and vehicles, and "light curtains" for protecting an area.

The IR sensor [8] used in this study consists of an emitter and detector. The detector window is covered with an IR filter to minimize the effect of ambient light on the intensity measurements. The maximum range of operation of the sensor is about 60 cm. The IR sensor [see Fig. 1(a)] is mounted on a 12 inch rotary table [9] to obtain angular intensity scans from these surfaces. 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 target types employed are a plane, a 90° edge, and a cylinder of radius 4.8 cm, each with a height of 120 cm. The horizontal
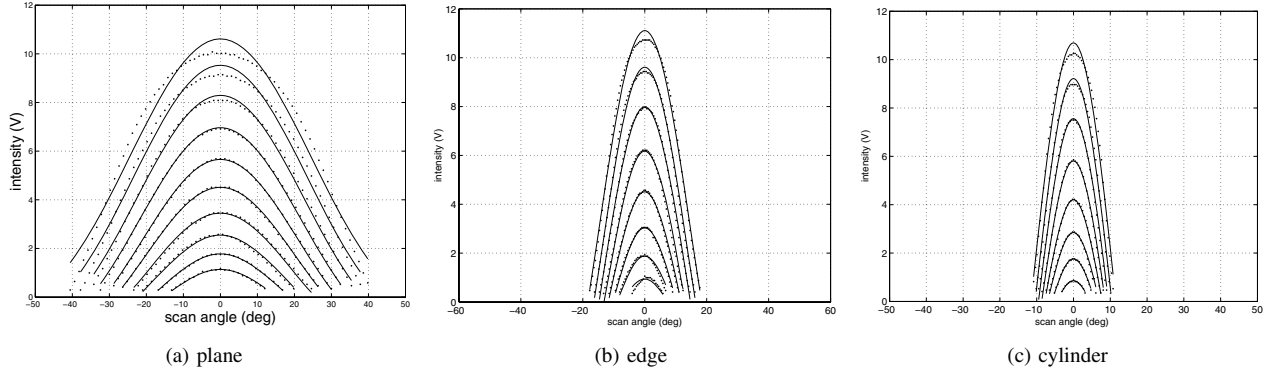
| (a) plane | (b) edge | (c) cylinder |

Fig. 3. Example intensity scans for wooden targets. Solid lines indicate the model fit and the dotted lines indicate the actual data.

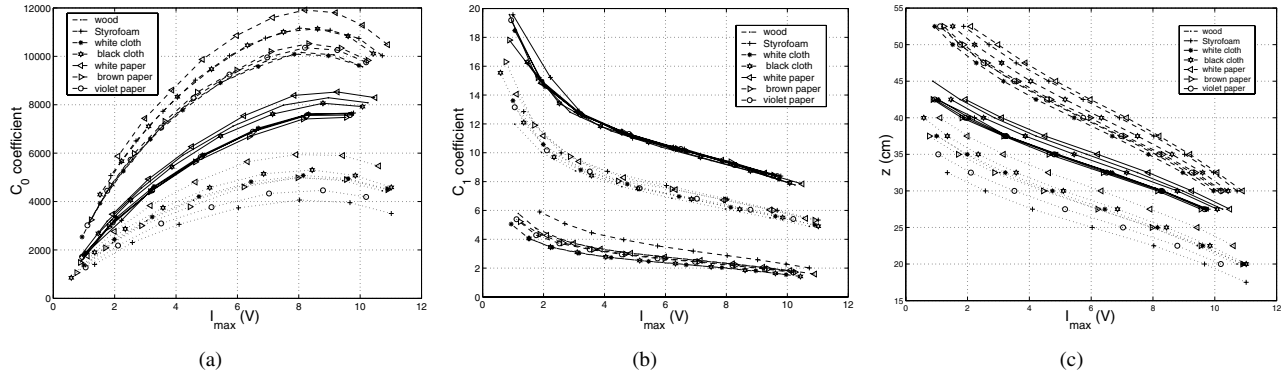| (a) | (b) | (c) |

Fig. 4. Variation of the parameters (a) C_0, (b) C_1, and (c) z with respect to maximum intensity (dashed, solid, and dotted lines are for planes, edges, and cylinders, respectively).

extent of all targets other than the cylinder is large enough that they can be considered infinite and thus edge effects need not be considered.

III. MODELING OF IR SCANS

The parametric approach is based on modeling of IR intensity scans [10]. Reference intensity scans are collected for each target type by positioning the targets over their observable ranges with 2.5 cm distance increments, at $\theta = 0°$. 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). Example reference scans for wooden targets are shown in Fig. 3 using dotted lines. These intensity scans have been modeled by approximating the targets as ideal Lambertian 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 target and is modeled with three parameters as

$$\mathcal{I} = \frac{C_0 \cos(\alpha C_1)}{\left[\frac{z}{\cos \alpha} + R(\frac{1}{\cos \alpha} - 1)\right]^2} \quad (1)$$

In Eqn. (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. 3). The z is the horizontal distance between the rotary platform and the target as shown in Fig. 2. The denominator of $\mathcal{I}$ is the square of the distance d between the IR sensor and the target. From the geometry of Fig. 2, $d + R = \frac{z+R}{\cos \alpha}$, from which we obtain d as $\frac{z}{\cos \alpha} + R(\frac{1}{\cos \alpha} - 1)$, 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 Eqn. (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^{TM} [11]. Samples of resulting curves are shown in Fig. 3 in solid lines. For the reference scans, z is not taken as a parameter since the distance between the target 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 $\mathcal{I}$ at $\alpha = 0°$, and corresponds to the product of $\mathcal{I}$ with z^2. Similarly, the initial guess for C_1 is made by evaluating C_1 from Eqn. (1) at a known angle α different than zero, with the initial guess of C_0 and the known value of z. While curve fitting, C_0 value is allowed to vary between $\pm$ 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. 4. 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 Eqn. (1).

IV. STATISTICAL PATTERN RECOGNITION TECHNIQUES

In this section, we propose the 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). PRTools [12] is used in the implementation.

After nonlinear curve fitting to the observed scan as in Section III, 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 $[C_0, I_{\max}]^T$ or the $[C_1, I_{\max}]^T$ pair, illustrated in Figs. 4 (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 $\mathbf{x} = [x_1, \ldots, x_d]^T$ falls in the region Ω_i. A rule which partitions the decision space into regions $\Omega_i, i = 1, \ldots, c$ 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 $p(\mathbf{x}|w_j)p(w_j) > p(\mathbf{x}|w_i)p(w_i)$ $\forall i \neq j \implies \mathbf{x} \in \Omega_j$ 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 $N_1 + N_2 + \ldots + N_c = N$ 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 $q_j(\mathbf{x}) > q_i(\mathbf{x})$ $\forall i \neq j \implies \mathbf{x} \in \Omega_j$ 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 [13]. 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.

A. Determination of Geometry

1) Parametric Classifiers:

a) Parameterized Density Estimation (PDE): In this method, the CCPDFs are assumed to be d-dimensional normal:

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

$i = 1, \ldots, c$, 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) [14] 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: $\sum_{i=1}^{c} \frac{N_i}{N} \hat{\mathbf{\Sigma}}_i$. 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) using the MLE, for each of the two feature vector representations described above. The training set consists of $N = 175$ data pairs for three classes: $N_1 = 50$ cylinders, $N_2 = 55$ edges, and $N_3 = 70$ planes. The test set consists of 211 data pairs for three classes: 84 cylinders, 43 edges, and 84 planes.

Since the feature vector size d is two and the number of classes c is three, three 2-D normal functions are used for classification. For the case when the $[C_0, I_{\max}]^T$ feature vector is used for differentiation, 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 differentiation rates are better than the homoscedastic case, which are 98.3% and 42.2% for the training and test sets, respectively.

For the case when the $[C_1, I_{\max}]^T$ feature vector is used for differentiation, 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, 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. These results are much better than those obtained with the classification based on the $[C_0, I_{\max}]^T$ feature vector. 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 $[C_1, I_{\max}]^T$ feature vector.

b) 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. Each normal distribution has probability density function p_j with mean vector $\boldsymbol{\mu}_j$ and covariance

matrix $\mathbf{\Sigma}_j$:

$$p_j(\mathbf{x}|\boldsymbol{\mu}_j, \mathbf{\Sigma}_j) = \frac{1}{(2\pi)^{(d/2)}|\mathbf{\Sigma}_j|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^T\mathbf{\Sigma}_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)\right],$$
(3)

$j = 1, \ldots, M$. The M normal distributions are mixed according to the following model, using the mixing coefficients α_j:

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

Here, $\mathbf{\Theta} = [\alpha_1, \ldots, \alpha_M; \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M; \mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_M]$ 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 $0 \le \alpha_j \le 1$ $\forall j$ 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 $\sum_{j=1}^{M} p(j|\mathbf{x}, \mathbf{\Theta}) = 1$. In our implementation, M takes the values two and three. For the i'th class, the parameter vector $\mathbf{\Theta}_i$ maximizing Eqn. (4) needs to be estimated, corresponding to the MLE. Since deriving an analytical expression for the MLE is not possible in this case, $\mathbf{\Theta}_i$ is estimated by using expected-maximization (E-M) clustering which is iterative [12]. The elements of the parameter vector $\mathbf{\Theta}_i$ are updated recursively as follows:

$$
\begin{aligned}
\alpha_{ijk} &= \frac{1}{N_i}\sum_{n=1}^{N_i} p(j|\mathbf{x}_n, \mathbf{\Theta}_{i,k-1})\\
\boldsymbol{\mu}_{ijk} &= \frac{\sum_{n=1}^{N_i} \mathbf{x}_n p(j|\mathbf{x}_n, \mathbf{\Theta}_{i,k-1})}{\sum_{n=1}^{N_i} p(j|\mathbf{x}_n, \mathbf{\Theta}_{i,k-1})}\\
\mathbf{\Sigma}_{ijk} &= \frac{\sum_{n=1}^{N_i}(\mathbf{x}_n-\boldsymbol{\mu}_{ijk})(\mathbf{x}_n-\boldsymbol{\mu}_{ijk})^T p(j|\mathbf{x}_n, \mathbf{\Theta}_{i,k-1})}{\sum_{n=1}^{N_i} p(j|\mathbf{x}_n, \mathbf{\Theta}_{i,k-1})}
\end{aligned}
$$
(5)

where $i = 1, \ldots, c$ and $j = 1, \ldots, M$. Here, $\mathbf{\Theta}_{i,k}$ is the parameter vector estimate of the i'th class at the k'th iteration step and N_i is the number of feature vectors in the training set representing the i'th 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.
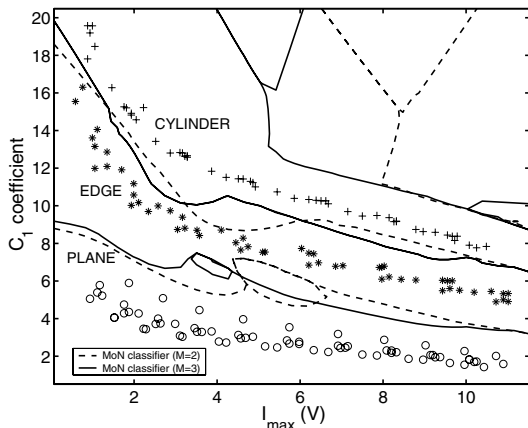

Fig. 5. Discriminant functions for the MoN classifier when the $[C_1, I_{\max}]^T$ feature vector is used.

After estimating the parameter vectors for each class based on the training set feature vectors, testing is done as follows: A target with a given test feature vector $\mathbf{x}$ is assigned to the class whose parameter vector $\mathbf{\Theta}_i$ maximizes Eqn. (4) so that $p(\mathbf{x}|\mathbf{\Theta}_i) > p(\mathbf{x}|\mathbf{\Theta}_l)$ $\forall i \ne l$. Then, the target is labeled as a member of class w_i.

The discriminant functions for classification based on $[C_1, I_{\max}]^T$ feature vector are shown in Fig. 5. For both $M = 2$ and $M = 3$, all training targets are correctly classified using the $[C_1, I_{\max}]^T$ feature vector. In the tests, for the $M = 3$ case, 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%.

2) Non-Parametric Classifiers: In this section, we consider different non-parametric classifiers, which are the kernel estimator, k-nearest neighbor, artificial neural network, and support vector machine classifiers.

a) Kernel Estimator (KE): In the KE method, the CCPDF estimates $\hat{p}(\mathbf{x}|w_i)$ are of the form

$$\hat{p}(\mathbf{x}|w_i) = \frac{1}{N_i h_i^d}\sum_{n=1}^{N_i} K\left(\frac{\mathbf{x}-\mathbf{x}_n}{h_i}\right) \quad i = 1, \ldots, c$$
(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 $K(\mathbf{z}) \ge 0$ and $\int K(\mathbf{z})d\mathbf{z} = 1$. In this method, the selection of the bandwidth h_i is important. 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 2-dimensional 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 [12]. After h_i's are computed, a test feature vector $\mathbf{x}$ is classified into that class for which the CCPDF in Eqn. (6) is maximized. This requires the training data to be stored throughout testing.

b) 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 $\hat{p}(w_i|\mathbf{x}) = \frac{k_i}{k}$, 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 nearest neighbors 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 nearest neighbors.

A major disadvantage of this method is that a pre-defined rule for the selection of the value of k does not exist. In

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	84.3	100	100
test	98.6	98.6	99.5	100	99.5	99.5	98.6	99.5	76.3	98.1	99.5	99.1

this study, the number of nearest neighbors k is determined by optimization with respect to leave-one-out error. In the implementation, k values varying between 1 and 12 have been considered. For $k = 1, 2$, and 3, the same correct differentiation rates are obtained for the training and test sets, respectively. For larger values of k, the errors start increasing. The given results correspond to $k = 1$. Again, the training data must be stored during testing. 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%.

c) Artificial Neural Network (ANN) Classifiers: Feedforward 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 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. 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. 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. However, this classifier is better than the BP method in the tests, where only one edge target is misclassified as a cylinder, resulting in a correct differentiation rate of 99.5%. 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 using LP on the training and test sets, respectively.

d) Support Vector Machine (SVM) Classifier: SVM classifier 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 is typically much higher than the original feature space. With a suitable nonlinear mapping 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. The complexity of SVMs is related to the number of resulting support vectors rather than the high dimensionality of the transformed space.

In this study, SVM 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 correct differentiation rates on the training set are 84.3%, 100%, and 100% for the SVM classifiers with polynomial, exponential, and radial basis function kernels, respectively. For the test data, these numbers, given in the same order, are 98.1%, 99.5%, and 99.1%.

To summarize the results of the statistical pattern recognition techniques for geometry classification based on the $[C_1, I_{\max}]^T$ feature vector, the overall differentiation rates are given in Table I. Best classification rate is obtained for the test scans using the MoN classifier with three components. This is followed by MoN with two components, KE, k-NN, and SVM with exponential kernel, 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, SVM with polynomial kernel, and ANN trained with LP.

The above classification approaches were applied to differentiate between surface types assuming the geometry of the targets is determined correctly beforehand. However, the results were not promising as expected from the very similar variation of the parameters for different surfaces corresponding to the same geometry (Fig. 4).

V. DISCUSSION AND CONCLUSION

We extended the parametric surface differentiation approach proposed in [10] 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 II

Overview of the differentiation techniques compared (P: plane, C: corner, E: edge, CY: cylinder; AL: Aluminum, WD: wood, WC: white cloth, BC: black cloth, WW: white wall, WP: white paper, BRP: brown paper, VP: violet paper.)

differentiation technique	type of geometry	type of surface	feature	correct diff.(%)	training data	learning	parametric
rule-based [15]	P,C,E,CY	WD	geo	91.3	used, not stored	no	no
template-based					used	no	no
[16]	P,C,E,CY	WD	geo	97			
[17]	P	AL,WW,BRP,ST	surf	87			
[18]	P,C,E	AL,WC,ST	geo	99			
"	P,C,E	"	surf	81			
"	P,C,E	"	geo+surf	80			
parametric [10]	P	ST,WW,WC(BC), WP,BRP,VP	surf	100	used, not stored	yes	yes
	"	ST,WW,WC(BC), WP,BRP,VP,WD	"	86			
	"	ST,WW,WC,BC, WP,BRP,VP	"	83			
	"	ST,WW,WC,BC, WP,BRP,VP,WD	"	73			
statistical pattern recognition	P,E,CY	ST,WC,BC, WP,BRP,VP,WD	geo				
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-E	"	"	"	99.5	used, not stored	no	no

Table II 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. 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.

References

[1] B. Barshan and R. Kuc, "Differentiating sonar reflections from corners and planes by employing an intelligent sensor," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 560–569, June 1990.

[2] L. Kleeman and R. Kuc, "Mobile robot sonar for target localization and classification. *Int. J. Robot. Res.*, vol. 14, pp. 295–318, Aug. 1995.

[3] G. Benet, F. Blanes, J. Sim, and P. Prez, "Map building using infrared sensors in mobile robots," *Book chapter in: New Developments in Robotics Research", Editor: J. X. Liu, Nova Science Publishers, Inc.*, pp. 73-103, 2006.

[4] E. Cheung and V. J. Lumelsky, "Proximity sensing in robot manipulator motion planning: system and implementation issues," *IEEE Trans. Robot. Automat.*, vol. 5, pp. 740–751, Dec. 1989.

[5] V. J. Lumelsky and E. Cheung, "Real-time collision avoidance in teleoperated whole-sensitive robot arm manipulators," *IEEE Trans. Syst. Man Cybern.*, vol. 23, pp. 194–203, Jan./Feb. 1993.

[6] H. -H. Kim, Y. -S. Ha, and G. -G. Jin, "A study on the environmental map building for a mobile robot using infrared range-finder sensors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 716–711, Las Vegas, NV, 27–31 Oct. 2003.

[7] A. M. Flynn, "Combining sonar and infrared sensors for mobile robot navigation," *Int. J. Robot. Res.*, vol. 7, pp. 5–14, Dec. 1988.

[8] Matrix Elektronik, AG, Kirchweg 24 CH-5422 Oberehrendingen, Switzerland, *IRS-U-4A Proximity Switch Datasheet*, 1995.

[9] Arrick Robotics, P.O. Box 1574, Hurst, Texas, 76053 URL: www.robotics.com/rt12.html, *RT-12 Rotary Positioning Table*, 2002.

[10] T. Aytaç and B. Barshan, "Surface differentiation and localization by parametric modeling of infrared intensity scans," *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 2294-2299, Aug. 2005, Edmonton, Alberta, Canada.

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

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

[13] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *Annals Math. Stats.*, vol. 27, no. 3, pp. 832–837, 1956.

[14] V. K. Rohatgi, *An Introduction to Probability Theory and Mathematical Statistics*. New York, John Wiley & Sons, 1976.

[15] T. Aytaç and B. Barshan, "Rule-based target differentiation and position estimation based on infrared intensity measurements," *Opt. Eng.*, vol. 42, pp. 1766–1771, June 2003.

[16] T. Aytaç and B. Barshan, "Differentiation and localization of target primitives using infrared sensors," *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol.1, pp. 105-110, Sep./Oct. 2002, Lausanne, Switzerland.

[17] B. Barshan and T. Aytaç, "Position-invariant surface recognition and localization using infrared sensors," *Opt. Eng.*, vol. 42, pp. 3589–3594, Dec. 2003.

[18] T. Aytaç and B. Barshan, "Simultaneous extraction of geometry and surface properties of targets using simple infrared sensors," *Opt. Eng.*, vol. 43, pp. 2437–2447, Oct. 2004.