Automatic Target Recognition Performance Losses in the Presence of Atmospheric and Camera Effects

Xiaohan Chen, and Natalia A. Schmid,
Lane Department of Computer Science and Electrical Engineering
West Virginia University, Morgantown, WV 26506
E-mail: xchen10@mix.wvu.edu and Natalia.Schmid@mail.wvu.edu.
Phone: (304) 293-0405, Fax: (304) 293-8602.

Abstract

The importance of Networked Automatic Target Recognition systems for surveillance applications is continuously increasing. Because of the requirement of a low cost and limited payload these networks are traditionally equipped with lightweight, low-cost sensors such as Electro Optical (EO) or Infrared sensors. The quality of imagery acquired by these sensors critically depends on the environmental conditions, type and characteristics of sensors, and absence of occluding or concealing objects. In the past a large number of efficient detection, tracking, and recognition algorithms have been designed to operate on imagery of good quality. However, detection and recognition limits under non-ideal environmental and/or sensor based distortions have not been carefully evaluated.

This work introduces a fully automatic target recognition system that involves a Haar-based detector to select potential regions of interest within images; performs adjustment of detected regions; segments potential targets using a region based approach; identifies targets using Bessel K form-based encoding; and performs clutter rejection. In this work, we investigate the effects of environmental and camera conditions on target detection and recognition performance. Two databases are involved. One is a simulated database generated using a 3D tool. The other database is formed by imaging 10 die cast models of military vehicles from different elevation and orientation angles. The database contains imagery acquired both indoors and outdoors. The indoors dataset is composed of clear and distorted images. The distortions include defocus blur, sided illumination, low contrast, shadows and occlusions.

The paper was presented in part at SPIE 2007 and at SPIE 2008.
All images in this database, however, have a uniform (blue) background. The indoors database is applied to evaluate the degradations of recognition performance due to camera and illumination effects. The database collected outdoors includes a real background and is much more complex to process. The numerical results demonstrate that the complexity of the background and the presence of occlusions present a serious challenge for automatic target detection and recognition.

KEYWORDS: Automatic Target Recognition, Target Detection, EO imaging, Atmospheric Effects.

I. INTRODUCTION

Recent advances in wireless communication and signal processing made stationary and dynamic networks such as networks of unmanned ground (UGV) or aerial (UAV) vehicles equipped with various sensors and cameras a potentially most reliable machinery for surveillance missions. There is an increasing trend towards fielding swarms of UAVs operating as large-scale sensor networks in the air [1]. Because of the limited payload that a network of unmanned ground or aerial vehicles can carry, these networks are traditionally equipped with lightweight, low-cost sensors and cameras. The most commonly used type of cameras include Electro Optical (EO) and low resolution Infrared (IR) cameras. The quality of imagery acquired by these sensors critically depends on the environmental conditions, type and characteristics of sensors, and absence of occluding or concealing objects.

Over the past few years a large number of efficient detection, tracking, and recognition algorithms have been designed and tested under the assumption of almost ideal EO and IR imagery. However, detection and recognition losses under non-ideal environmental and/or man generated concealing set ups have not been carefully researched and documented. Furthermore, real world data to provide the ground for this research are not available.

In this work, we investigate the effects of environmental conditions and other atmospheric and camera induced effects on target detection and recognition performance and explore detection and recognition losses and limitations under weather, sensor, and concealing effects. For our experiments we generated a simulated Automatic Target Recognition (ATR) database using a 3D simulation tool. We also compiled a database of EO images of 10 toy targets placed onto a real background (pavement, sand, grass, etc.) under varying environmental conditions including various illumination and contrast set ups, occlusions, defocus and motion blur, and resolutions.
The toys are 1:72 die-cast copies of military and civilian targets. The images are acquired using NikonD80 camera from a distance of 3-4 meters. The main challenge in processing images is due to the presence of shadows and low contrast.

To evaluate the detection and recognition losses in the presence of weather and camera effects, we propose a fully ATR system that involves a Haar feature-based detector [2] to select potential regions of interest within images; performs adjustment and fusion of detected regions; segments potential targets using a region based approach [3]; identifies targets using Bessel K form-based encoding [4]; and performs clutter rejection. The authors do not claim an optimality of the designed system. However, the performance of the detection and recognition blocks of the system operating on high quality images is used as a baseline for evaluating the recognition limits of the entire ATR system operating on both high quality imagery and on degraded imagery.

The remainder of the paper is organized as follows. Section II reviews related work, while Section III describes the modified ATR system and the methods utilized at each step. Section IV introduces the simulated and real databases. Section V presents numerical results. Section VI summarizes the results of this work.

II. LITERATURE REVIEW

The literature contains a large number of detection and recognition algorithms applied to optical data. While it is impossible to list here all published works, we choose to characterize a few.

Detection of a possible object of interest is one of the most critical steps in object recognition problems, since the results of postprocessing depend on this step. Target detection approaches can be classified into three categories: feature invariant approaches, template matching methods and appearance-based methods. In feature invariant approaches, the algorithms aim to find structural features such as edges [5], textures, etc. that exist even when the pose, viewpoint, or lighting conditions vary, and then use the features to locate targets. In template matching methods [6] several standard patterns of a target are stored to describe the target. The correlation between an input image and the stored patterns is computed for detection. In contrast to template matching, the models (or templates) in appearance-based methods [7], [2] are learned from a set of training images, which should capture the representative variability of target appearance. These learned models are then used for detection. In this work, we will use a local Haar filter-based detection
method which is very popular in the field of face and object recognition [2] for its processing rate (30 frames/sec.) and performance.

Based on encoded information, ATR algorithms are broadly classified into three categories: shape-based, appearance-based, and Computer Aided Design (CAD)-based methods. In shape-based recognition [8], [9], the contour of the object is extracted, and then the shape templates are used to match the extracted contours. In appearance-based (or view-based) approach, the 2D intensity templates of a 3D target acquired from different viewpoints are stored as a model. Some view-based methods [10], [11] use statistical techniques to analyze the distribution of target images in a vector space, and derive an effective representation (feature space) in support of various applications. Other methods [12], [13], [14] design distortion-invariant filters to perform correlation-based matching between the model view and the input image. In CAD-based ATR [15], [16] an explicit 3D model of a target is generated and subsequently used in target recognition employing imagery acquired by a variety of sensors. The main step of CAD-based ATR is to estimate the pose of the CAD model so that the projection of a 3D model matches with the query image. However, it may not be possible to acquire CAD models of all targets. In [17] a multi-view morphing algorithm is generated to provide a 3D model using several images. Other classification methods used in ATR systems include Neural Network-based [18] and Support Vector Machine-based [7].

Most of the algorithms summarized above assume good quality data for training and performance evaluation. This is not the case in practice. The real world data are highly distorted and noisy. This work analysis performance of Haar feature-based detector and Bessel K form-based recognition algorithm when they are tested on synthetically degraded and real world data. This work further evaluates performance of a fully ATR system which is also tested on synthetically degraded and real world data.

III. AUTOMATIC TARGET RECOGNITION SYSTEM

Fig. 1 shows a block-diagram of a traditional ATR system. An ATR system can be decomposed into three major subsystems. The first subsystem performs preprocessing of acquired data. Preprocessing may involve denoising, contrast adjustment, normalization, etc. The second subsystem is a detector, which scans a preprocessed image with the purpose to locate potential objects. By locating potential subregions, we can filter out a large amount of background clutter from the
terrain scene, making object recognition feasible for large data sets. The subregions are encoded resulting in a vector of informative descriptors or features. At the last step, the classifier inputs a vector of features and returns the target’s identity by comparing query feature vector against all feature vectors in a target library.

Ideally, input to classification block should contain a detected object in the center of a selected region or window with maximally cropped background clutter. In real situations, the output of a detector may not be a complete image of an object. It may contain only parts of an object, or the object may not be located in the center of the window and/or occupy only a small part of the window. To recognize an object within the window, an additional processing has to be performed. In this paper, we propose to place a segmentation block between detector and classifier. However, as will be demonstrated later, this additional processing step results in an increased number of false alarms, which in turn leads to increased number of misclassifications, that is, an output of the detector (a window) that enters the classifier will be recognized as one of the objects in an object library. Thus, there is a possibility that clutter will be recognized as one of the objects. One approach to remove detected windows containing clutter is to perform a postprocessing, clutter rejection, after the classification step. The proposed modified structure of the ATR system is shown in Fig. 2.

In this section, we will first introduce the detection method based on Haar features, then we will perform classification using Bessel K forms. We will approach the challenge of practical issues described above by involving an object segmentation method based on B-splines and by including a clutter rejection block at the far end of the automatic recognition system.

A. Target Detection based on Haar Features

Detection of a possible object of interest is one of the most critical steps in object recognition problems, since the results of postprocessing depend on this step. In this work, we use a local Haar filter-based detection method. This rapid target detection scheme is based on the idea of a boosted classifier cascade [2].

The classifier cascade is trained on a set of positive images (targets) and a set of negative images (non-targets). For each training image, an over-complete set of Haar-like feature pool is calculated, and AdaBoost algorithm of Schapire and Singer [19] is used to build a stage classifier.

In each round of boosting, a weak learning algorithm is applied to select a single feature which
best separates the positive and negative samples. For each feature, the weak learner determines the optimal threshold classification function, such that the minimum number of examples are misclassified. Thus, a weak classifier \( h_j(\cdot) \) is a binary valued function obtained by comparing the \( j \)-th feature value \( f_j(\cdot) \) with a threshold \( \theta_j \), which is optimized by the weak learner:

\[
h_j(x) = \begin{cases} 
\alpha_j & \text{if } f_j(x) > \theta_j \\
\beta_j & \text{otherwise.}
\end{cases}
\]  

(1)

Here \( x \) is a sub-window of an image. The value of the feature is equal to weighted differences of integrals over rectangular subregions. The parameters \( \alpha_j \) and \( \beta_j \) are positive or negative votes of each feature set by AdaBoost during the learning process.

The form of the final stage classifier returned by AdaBoost is a thresholded linear combination of weak classifiers (see Fig. 3). The stage classifier is given by:

\[
C(x) = \begin{cases} 
1, & \text{if } \sum_j h_j(x) > T, \\
0, & \text{otherwise,}
\end{cases}
\]  

(2)

where \( T \) is the stage threshold set by AdaBoost during the learning process.

In order to improve computational efficiency and also reduce the rate of false positives, a sequence of increasingly more complex stage classifiers called cascade is used. At each stage, a different set of non-target images is involved for training. The training procedure is terminated if the overall false alarm rate is less than a certain small threshold value.

After the cascade of classifiers is trained, the detection algorithm is applied to a query image. A search window is sled over the query image. At each location and scale the window is passed through the cascade of classifiers. If an input window looks like an object, the algorithm passes it through a large number of classifiers, to ensure that decision it makes is reliable. This procedure may take a long time to complete in this case. Since most windows of an image do not look like objects, they are quickly discarded as non-objects. Fig. 4 illustrates a cascade of classifiers.

**B. Classification method based on Bessel K forms**

The basic idea of Bessel K based method is to decompose images into their spectral components using a set of band-pass filters and then represent images using low dimensional statistics of the filtered components. Comprehensive studies [4], [20] of natural scenes have shown that
the distributions of pixel intensities in linearly filtered images are described by a family of Bessel K distribution functions. This constitutes basis for the implemented classification algorithm.

Bessel K forms parameterized by only two parameters: (1) the shape parameter \( p, p > 0 \), and (2) the scale parameter \( c, c > 0 \), may provide a good statistical fit to empirical histogram distributions of filtered images.

Denote by \( I \) an image and by \( F \) a filter, then the filtered image \( I \) is \( I = I \ast F \), where \( \ast \) denotes the 2-dimensional convolution operation. Under the conditions stated in the work by Srivastava et al. [4], the probability density function of the intensity of \( I(\cdot) \) can be approximated by

\[
f_K(x; p, c) = \frac{2}{Z(p, c)} |x|^{p-0.5} K_{(p-0.5)}(\sqrt{2/c}|x|),
\]

where \( K_\nu(x) \) is the modified Bessel function of the second kind, and \( Z(p, c) \) is the normalization given by \( Z(p, c) = \sqrt{\pi \Gamma(p)}(2c)^{0.5p+0.25} \). Given \( J \) filters, the image \( I \) can be represented using \( 2J \) Bessel parameters.

To approximate the empirical density of the filtered image by a Bessel K form, the parameters \( p \) and \( c \) are estimated from the observed data using

\[
\hat{p} = \frac{3}{SK(I) - 3} \quad \text{and} \quad \hat{c} = \frac{SV(I)}{\hat{p}},
\]

where \( SK \) is the sample kurtosis and \( SV \) is the sample variance of the pixel values in \( I \). Since the moment-based estimate of \( p \) in (4) is sensitive with respect to outliers, in our computations we replace it with an estimate based on empirical quartiles given by

\[
\hat{p} = \frac{3}{SK(I) - 3}, \quad \text{with} \quad SK(I) = \frac{q_{0.995}(I) - q_{0.005}(I)}{q_{0.75}(I) - q_{0.25}(I)},
\]

where \( q(\cdot) \) is the quartile function that returns the \( x \) quartile of a set of samples. More information on quartile estimates can be found in the work by Freund [21]. This method provides a reasonable fit. As shown in Fig. 6, the histogram (dashed line) of images filtered by Gabor filters [22] closely follows the estimated Bessel K forms (solid line).

To quantify the difference between two filtered images based on their distributions, two distance measures: (1) a pseudo-metric introduced by Srivastava [4] and (2) the K-measure [23] between two Bessel K forms, are used. The pseudo-metric is defined as

\[
d_I(I_1, I_2) = \sqrt{\int_{-\infty}^{+\infty} (f_K(x; p_1, c_1) - f_K(x; p_2, c_2))^2 \, dx}.
\]
The closed form of $d_I(I_1, I_2)$ for the case of $p_1, p_2 > 0.25, c_1, c_2 > 0$ is given by:

$$d_I(I_1, I_2) = \left[ \frac{\Gamma(0.5)}{2\sqrt{2\pi}} \left( \frac{G(2p_1)}{\sqrt{c_1}} + \frac{G(2p_2)}{\sqrt{c_2}} - 2G(p_1 + p_2) \left( \frac{c_1}{c_2} \right) p_2 \mathcal{H} \right) \right]^{\frac{1}{2}},$$

where $G(p) = \frac{\Gamma(p-0.5)}{\Gamma(p)}$ and $\mathcal{H} = H\left(p_1 + p_2 - 0.5, p_2; p_1 + p_2; 1 - \frac{c_1}{c_2}\right)$. The function $H$ is the hypergeometric function. In cases where $\hat{p} < 0.25$ for an image-filter combination, we compute the pseudo-metric numerically using the quadrature integration. The K-measure is defined as

$$d_{KL}(I_1, I_2) = D(f_K(x; p_1, c_1)\|f_K(x; p_2, c_2)) + D(f_K(x; p_2, c_2)\|f_K(x; p_1, c_1)),$$  \hspace{1cm} (6)

where $D(f_K(x; p_1, c_1)\|f_K(x; p_2, c_2))$ is the relative entropy between two probability density functions $f_K(x; p_1, c_1)$ and $f_K(x; p_2, c_2)$ [24] given by

$$D(f_K(x; p_1, c_1)\|f_K(x; p_2, c_2)) = \int_{-\infty}^{+\infty} \log \left( \frac{f_K(x; p_1, c_1)}{f_K(x; p_2, c_2)} \right) f_K(x; p_1, c_1) dx,$$

where the ratio under the log-function is assumed to be well defined. In the above expressions $f_K(\cdot)$ is the Bessel K probability density function introduced in (3).

Given two images $\{I_1, I_2\}$ and a bank of filters $\{\mathcal{F}_j, j = 1, 2, \cdots, J\}$, we evaluate a set of filtered images $\{I_{(n,j)} = I_n * \mathcal{F}_j, n = 1, 2; j = 1, \cdots, J\}$. After estimating the parameters $p_{(n,j)}$ and $c_{(n,j)}$, each image is mapped to $J$ points in the density space. The distance between two images is calculated by $d_I(I_1, I_2) = \sum_{j=1}^{J} d_I(I_{(1,j)}, I_{(2,j)})$, and $d_{KL}(I_1, I_2) = \sum_{j=1}^{J} d_{KL}(I_{(1,j)}, I_{(2,j)})$, where $d_I(I_{(1,j)}, I_{(2,j)})$ and $d_{KL}(I_{(1,j)}, I_{(2,j)})$ are defined in (5) and (6).

The purpose of using the two distance measures is to balance accuracy and computational efficiency. K-measure is an accurate measure of similarity of two probability density functions. However, it cannot be obtained in closed form for Bessel K forms. Numerical evaluation of K-measure is computationally expensive. The pseudo-metric (5) has closed form for Bessel K forms, which means that the computational cost is relatively low. The major drawback of the pseudo-metric is its lower precision compared with the K-measure. To measure the difference between two histograms fast and with a relatively high precision, we combine these two distance measures. First, we use the fast method, the pseudo-metric, to evaluate the distance between the input image and all templates in the database. If the pseudo-metric has multiple minima close in their values, there will be a potential misclassification. The precise metric, the K-measure, is then used to re-calculate the distances and make the final decision. By setting threshold properly, we obtain reliable results.
C. Target segmentation using B-spline method

Outputs of the detector containing potential targets are further subjected to a segmentation process. The goal of segmentation is to minimize the number of pixels representing clutter within the detected window. Prior to segmentation, potential detected targets are placed in the center of the window. To design a fully automatic target segmentation algorithm, we involve deformable templates based on B-splines to describe target’s boundary. This boundary separates the entire window into two nonoverlapping regions: a target region and a background region. To identify a target region, we involve a region-based approach [3], [25]. This approach assumes that a target and a background region are described by two distinct stochastic models. Traditional models used to model pixel intensities include Gaussian model, Gaussian mixture or Poisson models. Since both the target boundary and the parameters of models are unknown, they have to be estimated using available data. This problem can be solved iteratively, i.e., fixing the parameters of models, we perform estimation of the boundary between regions; then fixing the boundary parameters, we re-estimate the models. For the sake of clarity, we first present a brief review of B-splines; for a more detailed account, see [26]. We then introduce our optimization algorithms.

1) B-splines: Given \( m + 1 \) non-decreasing real numbers \( \{t_0 \leq t_1 \leq \cdots \leq t_m\} \) called knots, a B-spline of degree \( n \) is a parametric curve \( S : [t_0, t_m] \to \mathbb{R}^2 \) composed of basic B-splines of degree \( n \)

\[
S(t) = [x(t), y(t)] = \sum_{i=0}^{m-n-1} c_i B_i^n(t), \quad t \in [t_n, t_{m-n}].
\]

The \( c_i \) are called control points. The family of basic B-splines of degrees 1, \ldots, \( n \) are defined using the recursion formula:

\[
B_i^0(t) = \begin{cases} 1 & \text{if } t_i \leq t_{i+1} \\ 0 & \text{otherwise}, \end{cases} \quad B_i^n(t) = \frac{t - t_i}{t_{i+n} - t_i} B_i^{n-1}(t) + \frac{t_{i+n+1} - t}{t_{i+n+1} - t_i} B_{i+1}^{n-1}(t).
\]

Since the basis functions are based on knot differences, the shape of basis functions depends on the knot spacing and not on specific knot values. In this work, we use the uniform B-splines with equidistant knots.

To describe closed curves, an object boundary, a periodic extension of the knot sequence, \( \{t_j, j \in \mathbb{Z}\} \) with \( t_j = t_j \mod k \) is defined. The basic functions are also extended periodically

\[
\tilde{B}_i^n(t) = \sum_{j=-\infty}^{+\infty} B_{i+j(t_k-t_0)}^n(t).
\]
Now a sampled \( m \)-knot closed curve is represented as a linear combination of \( m \) periodic basis functions \( \mathbf{S}(t) = \sum_{i=0}^{m-1} c_i \tilde{B}_i^n(t), \quad t \in \mathbb{R} \), which can be further written in matrix form: \( \mathbf{S} = \mathbf{B} \mathbf{c} \), where \( \mathbf{B} \) is a matrix and \( \mathbf{c} \) is a vector.

2) Region-based Approach and Optimization Algorithms: We use the probability model to define the coherence of different image regions to group the pixels \([3], [27]\). Given a contour, the image pixels are assumed to be independently distributed. All pixels inside or outside of the contour have a common distribution characterized by a parameter vector \( \theta_{\text{in}} \) or \( \theta_{\text{out}} \), respectively. Denote \( \mathcal{A}_{\text{in}}(\mathbf{S}) \) and \( \mathcal{A}_{\text{out}}(\mathbf{S}) \) as the inside and outside regions of the contour \( \mathbf{S} \). The likelihood function of the image, given the contour and the model parameters, is

\[
p(\mathbf{I}|\mathbf{c}, w_{\text{in}}, w_{\text{out}}, \theta_{\text{in}}, \theta_{\text{out}}) = \prod_{(i,j) \in \mathcal{A}_{\text{in}}(\mathbf{S})} w_{\text{in}} p(I_{(i,j)}|\theta_{\text{in}}) \prod_{(i,j) \in \mathcal{A}_{\text{out}}(\mathbf{S})} w_{\text{out}} p(I_{(i,j)}|\theta_{\text{out}}),
\]

with \( \mathbf{S} = \mathbf{B} \mathbf{c} \) and \( I_{(i,j)} \) denoting the pixel value at the location \((i, j)\). Two functions \( p(I_{(i,j)}|\theta_{\text{in}}) \) and \( p(I_{(i,j)}|\theta_{\text{out}}) \) are the probability density functions of the intensity values in the inner and outer regions; \( w_{\text{in}} = p((i, j) \in \mathcal{A}_{\text{in}}) \) and \( w_{\text{out}} = p((i, j) \in \mathcal{A}_{\text{out}}) \) are the priors such that \( w_{\text{in}} + w_{\text{out}} = 1 \).

Apart from the estimation of the position of the control points, \( \mathbf{c} \), the unsupervised segmentation algorithm has to estimate the parameters of the probability models \( w_{\mathcal{M}} \) and \( \theta_{\mathcal{M}} \), for \( \mathcal{M} \in \{\text{in, out}\} \). Estimation of the boundary of a smooth object, a continuous function, from a finite amount of data is an ill-posed problem, which means that an infinite number of continuous solutions may result in the same observed data. To regularize the contour estimate, that is, to restrict the estimate to a certain class of functions providing a unique solution, we use a penalty term that favors smooth contours \([28]\). Then the segmentation problem solves the following optimization problem:

\[
\hat{\mathbf{c}}, \hat{w}_{\mathcal{M}}, \hat{\theta}_{\mathcal{M}} = \arg \min_{\mathbf{c}, w_{\mathcal{M}}, \theta_{\mathcal{M}}} \left[ -\log(p(\mathbf{I}|\mathbf{c}, w_{\mathcal{M}}, \theta_{\mathcal{M}})) + \lambda \kappa \right], \tag{7}
\]

where \( \lambda \) is a regularization parameter and \( \kappa \) is the curvature of the boundary given by \( \kappa(x, y) = \frac{x'y''-x''y'}{(x'^2+y'^2)^{3/2}} \), among which \( x', y', x'' \) and \( y'' \) are the first and second order derivatives of \( x(t) \) and \( y(t) \).

The number of control points is selected such that both the accuracy of the segmentation and the computational cost are satisfied. The minimization of \((7)\) is performed iteratively. Each iteration consists of two phases. First, we solve for \( \mathbf{c} \) with \( w_{\mathcal{M}} \) and \( \theta_{\mathcal{M}} \) fixed. Second, we fix
the contour and re-estimate the parameters of the probability models.

**Phase 1: Boundary Optimization by B-spline.** To find the shape of an unknown object we use the form of a gradient projection method described in [3]. Given \( w_M \) and \( \theta_M \), for \( M \in \{ \text{in, out} \} \), and the control points \( \hat{c}^{(k)} \) at the \( k \)th iteration, the contour is estimated by minimizing the log likelihood function augmented with a regularization term.

1) Set \( p = 0 \), \( \hat{c}^{(p)} = \hat{c}^{(k)} \). Build \( B \) and compute \( B^\perp = (B^T B)^{-1} B^T \).

2) Calculate the gradient with respect to the contour \( \partial S = \nabla \left[ -\log(p(I|\hat{c}^{(p)}, w_M, \theta_M)) + \lambda_K(\hat{c}^{(p)}) \right] \), where \( \nabla \) is the gradient operation.

3) Update the contour estimate according to \( S^{(p+1)} = S^{(p)} + \epsilon \partial S \), where \( \epsilon \) controls the step size. The control points are updated by \( \hat{c}^{(p+1)} = B^\perp S^{(p+1)} \).

4) If a stopping criterion is met, stop and \( \hat{c}^{(k+1)} = \hat{c}^{(p^*)} \), where \( \hat{c}^{(p^*)} \) is a stationary point; if not, set \( p = p + 1 \), go back to step 2.

**Phase 2: Image Model Estimation.** During Phase 2, we fix the contour and minimize (7) with respect to \( w_M \) and \( \theta_M \). We use Gaussian distributions to model both the pixel values within and outside of the boundary. Then the model parameters are the mean and variance of the pixel values inside and outside of the contour, i.e., \( \theta_M = [\mu_M, \sigma_M^2] \), for \( M = \{ \text{in, out} \} \). By taking the derivatives of the log likelihood with respect to all parameters and setting the derivatives to zero, we are able to obtain the optimal parameters. The parameters \( \mu_M \) and \( \sigma_M^2 \) are sample mean and variance of image intensity in the inner and outer regions. \( w_M \) is the ratio of the number of pixels in \( A_M \) and the total number of pixels in the image.

**D. Clutter Rejection**

Bessel K forms can also be used for modeling the clutter [29]. We use this model to perform clutter rejection after classification.

Since a set of negative images are involved in the procedure of the detector training, these negative images can be used as samples of the clutter. The basic idea is to cluster sample images of clutter and each detected window in two broad classes. We involve K-mean method to perform clustering. A bank of filters is applied to each sample image of the clutter resulting in a set of filtered images. The distance between images is characterized by the summation of the distances between corresponding filtered images. We use the pseudo-metric defined in (5). The pairwise distances between sample clutter images are calculated offline. The pairwise distances between
sample clutter images and each detected image are evaluated after classification, that is, online. All the pairwise distances are used to cluster the data into two groups. If the detected window and any negative sample are clustered in one group, it is identified as clutter. Otherwise, it is concluded that the detected window contains a potential target. The index of the target is then provided by the classifier.

IV. DATA DESCRIPTION

To evaluate performance of the ATR system, we involved two databases: (1) a simulated database generated using a 3D tool provided by Augusta System, Inc. The tool generates 2D projections of 3D objects and terrain from an arbitrary view angle; (2) a real database of images of die cast vehicle models collected in our laboratory. There are other public databases used for object detection and recognition (for example, COIL-100 database [30], UIUC database [31] and UBC database [32]). However, none of existing databases contains images that would allow us to comprehensively evaluate camera and environmental effects.

A. Simulated database

An ATR Training Tool provided by Augusta Systems, Inc. was used to build a simulated database. The tool is capable of generating prospective projections of 15 distinct objects projected at different orientation and elevation angles and sampled at distinct resolutions. Fig. 7 illustrates the parameters of projections: $\theta$ is the elevation angle, $\alpha$ is the orientation angle and $d$ is the distance from the object to the camera. The objects can be manually superimposed onto a background to simulate various ground conditions. The camera parameters such as position, azimuth, declination and distance can be varied to simulate images taken by an UAV. The resolution of captured images can be adjusted from $512 \times 384$ to $1152 \times 864$. A snapshot of the Graphical User Interface (GUI) of the tool is shown in Fig. 8. Every image generated by the 3D optical tool is first processed by a target detector and then fed into a recognition system. Prior to recognition, a potential target is located and placed in a canonical (or object-centered) reference frame suitable for recognition. In our experiments, we use three target types: tank, truck, and tractor. Sample images of targets used for recognition are shown in Fig. 9. We built a dataset by projecting each 3D target into a 2D plane at discrete orientation angles spaced 5 degrees apart and elevation angles from 0 to 75 spaced 15 degrees apart.
The simulated database described above is further expanded by adding to it artificially distorted versions of images in the database. The involved distortions mimic various camera and environmental effects. The types of distortions include a Gaussian noise, a Poisson noise, an illumination effect, an effect of contrast change, motion and defocus blurs. The details of generation procedures are summarized below.

1) Images contaminated by Gaussian noise contain an additive white noise with zero mean and variance $\sigma^2$.

2) Shot noise from a Charge Coupled Device (CCD) camera is modeled as a realization of a Poisson process (see [33] for a detailed description). The intensity of the Poisson noise depends on the intensity of the underlying data. The mean of the Poisson process is equal to the square root of the image intensity.

3) The images are brightened or darkened by increasing or decreasing the intensities [34]. This procedure simulates an illumination effect. Denote by $\beta$ ($\beta \in (-1, 1)$) the parameter that controls the level of illumination. We first normalize image intensities to $(0,1)$, then brighten images by raising to the power of a number less than one, that is, $(1 - \beta, \beta \in (0,1))$ or darken images by raising to the power of a number larger than one, that is, $(\frac{1}{\beta+1}, \beta \in (-1,0))$.

4) We model a contrast change by linearly transforming the normalized histogram of an image [34]. The range of the transformation is determined by the parameter $1 - 2LF$, where $LF$ corresponds to the fraction of saturated pixels in an image.

5) A linear motion of an optical camera or of an object is simulated by convolving images with a two parameter point spread function [35]. The length $L$ in pixels and the angle $\theta$ in degrees correspond to the motion in a specific direction with a predefined camera velocity. The parameter $\theta$ follows uniform distribution on $[0, 360^\circ]$.

6) The images are filtered by a two-dimensional circular averaging filter to generate defocus blur [35]. The defocus level corresponds to the radius $r$ of the averaging filter.

By controlling the value of the parameters, different levels of noise in images can be generated. Table I lists the values of the parameter used to achieve various levels of distortions in our experiments. Distortions increase from Level 1 to Level 5. The samples of distorted images of the tractor are displayed in Fig. 10.
To compare the distortions due to various atmospheric and camera effects, we introduce the mean-square error (MSE) and peak signal-to-noise ratio (PSNR). For a given pose, MSE between a distorted image and the corresponding clear image is calculated as

$$MSE_p = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \| I_p(i, j) - K_p(i, j) \|^2,$$

(8)

where $I_p$ is the distorted image and $K_p$ is the corresponding clear image. $M \times N$ is the image size. For a given pose, PSNR for each distorted image is then defined as

$$PSNR_p = 20 \log_e \left( \frac{255}{\sqrt{MSE_p}} \right).$$

(9)

These two measures will translate the absolute distortions due to individual atmospheric and camera effect into relative distortions. The strength of effects is measured in terms of levels (Levels 1 through 5). This makes it possible to compare performance degradation due to individual and combined effects for a variety of object databases. To visualize dependence between the values of the average PSNR and absolute distortion levels, we first average conditional PSNR over all poses and then plot the dependence of the average PSNR on the value of the distortion level.

### TABLE I

The values of the parameters used to simulate various distortion levels of the camera and due to environmental effects.

<table>
<thead>
<tr>
<th>Effects</th>
<th>Parameter</th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>Level 4</th>
<th>Level 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian Noise</td>
<td>$\sigma^2$</td>
<td>0.005</td>
<td>0.01</td>
<td>0.015</td>
<td>0.02</td>
<td>0.025</td>
</tr>
<tr>
<td>Illumination (decreasing)</td>
<td>$\beta$</td>
<td>-0.1</td>
<td>-0.2</td>
<td>-0.3</td>
<td>-0.4</td>
<td>-0.5</td>
</tr>
<tr>
<td>Contrast</td>
<td>$LF$</td>
<td>0.15</td>
<td>0.2</td>
<td>0.25</td>
<td>0.3</td>
<td>0.35</td>
</tr>
<tr>
<td>Motion Blur</td>
<td>$L$</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>Defocus Blur</td>
<td>$r$</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

### B. Die Cast database

In this section, we describe a database of images of die cast models of military and civilian vehicles collected in our lab. Images taken indoors have a uniform blue background. The collected
dataset contains both high quality (referred here as clear) images and distorted images of objects. These images are utilized to evaluate the classification performance under different camera and environmental effects. Images taken outdoors involve real background. These images are treated as inputs to the modified ATR system to evaluate the system performance.

We selected 10 die cast military vehicles as targets. For consistency, all toys are 1/72 scale copies of real vehicles. The images are acquired using a NikonD80 camera from a distance of 3-4 meters. The resolution of the images is set to be $1936 \times 1296$. The camera was equipped with a Tamron AF 70-300mm f/4.0-5.6 LD Macro zoom lens. The aperture was set to f4.2. The zoom was fixed at 70mm for clear images. The camera shutter speed (intensity integration time) is set to 1/60 second. Some experiments required the use of a build-in flash.

For each die cast model, a set of clear images is captured as a function of the pose. These images are used as the ground truth. Fig. 11 illustrates these 10 objects (top view). Note that object 1 is a M1A1HA with mine plough; object 2 is a M1A2; object 3 is an UK Challenger; object 4 is a Wurfamen 40 tank; object 5 is a HMMWV M998 “gun truck;” object 6 is a Hummer; object 7 is a T-72; object 8 is an AAVP7A1; object 9 is a M113 and object 10 is a HMMWV M1025. The size of each image in Fig. 11 is from $500 \times 500$ pixels to $800 \times 800$ pixels. Each object is projected at 3 elevation angles: 0, 20.3 and 35.6 degrees. At each elevation angle, a calibrated turntable is used to control the rotation angle of the objects. The selected rotation step is 5 degree. Thus there are total 216 ($72 \times 3$) poses per object.

By adjusting the illumination and the camera settings, we can easily obtain images with shadows, low contrast, and defocus blur. The degradations are introduced at four different distortion levels. The distorted sample images of object 2 are shown in Fig. 12. The details on the illumination and the camera settings are summarized in Table II. Thus total 28,080 images taken indoors including clear and distorted images form the database used to evaluate classification performance.

V. EXPERIMENTS AND RESULTS

In this section, we first use the simulated ATR database and the Die Cast database to evaluate the influence of the illumination and camera effects on the classification performance. We then evaluate the recognition performance of the entire system displayed in Fig. 2, which processes images with automatically detected targets.
First, we test the capability of the Bessel K recognition method to deal with pose variation. All clear images from each of the two databases are divided into two nonoverlapping sets, training and testing. A bank of 38 filters including Gaussian, Laplacian of Gaussian and Gabor filters is used to process data. The recognition performance is evaluated as a function of a varied proportion of training and testing images per object. The simulated ATR database contains 864 (3 distance values, 4 elevation angles and 72 rotation angles) clear images per target. Images parameterized by a 15 degree elevation angle and a distance 10 are used as training samples. The remaining samples are used for testing. The training images are equally spaced in orientation. For the Die Cast database, the selected elevation angle is 20.3 degrees. All images are further resized to $64 \times 64$ and are preprocessed by following the equation (10):

$$I^-(x, y) = \frac{I(x, y) - \mu}{\sigma}, \ c \in \mathbb{R}^+, \mu$$

where $I(x, y)$ is the $(x, y)$-pixel value, $\mu$ and $\sigma$ are the mean and the standard deviation of the image intensity. No other processing steps are involved.

Table III summarizes the classification results using Bessel K forms. The results are presented for the combined metric described in Section III-B. For the Die Cast database when 24 images

<table>
<thead>
<tr>
<th>Mode</th>
<th>Focus Length</th>
<th>Shutter Speed</th>
<th>Flash</th>
<th>Lighting</th>
<th>Lighting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear</td>
<td>Auto</td>
<td>1/60 sec</td>
<td>ON</td>
<td>2 lamps, side</td>
<td></td>
</tr>
<tr>
<td>Dark</td>
<td>Level 1</td>
<td>Same as Clear mode</td>
<td>1/10 sec</td>
<td>OFF</td>
<td>1 lamp, top</td>
</tr>
<tr>
<td></td>
<td>Level 2</td>
<td>Same as Clear mode</td>
<td>1/20 sec</td>
<td>OFF</td>
<td>1 lamp, 24 inches from side, 55 inches high</td>
</tr>
<tr>
<td></td>
<td>Level 3</td>
<td>Same as Clear mode</td>
<td>1/40 sec</td>
<td>OFF</td>
<td>1 lamp, 24 inches from side, 20 inches high</td>
</tr>
<tr>
<td></td>
<td>Level 4</td>
<td>Same as Clear mode</td>
<td>1/60 sec</td>
<td>OFF</td>
<td>1 lamp, 48 inches from side, 55 inches high</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 lamp, 72 inches from side, 55 inches high</td>
</tr>
<tr>
<td>Shadow</td>
<td>Level 1</td>
<td>8 inches</td>
<td>1/15 sec</td>
<td>OFF</td>
<td>1 lamp, 24 inches from side, 55 inches high</td>
</tr>
<tr>
<td></td>
<td>Level 2</td>
<td>10 inches</td>
<td>1/25 sec</td>
<td>OFF</td>
<td>1 lamp, 24 inches from side, 20 inches high</td>
</tr>
<tr>
<td></td>
<td>Level 3</td>
<td>12 inches</td>
<td>1/10 sec</td>
<td>OFF</td>
<td>1 lamp, 48 inches from side, 55 inches high</td>
</tr>
<tr>
<td></td>
<td>Level 4</td>
<td>14 inches</td>
<td>1/5 sec</td>
<td>OFF</td>
<td>1 lamp, 72 inches from side, 55 inches high</td>
</tr>
<tr>
<td>Blur</td>
<td>Level 1</td>
<td>1/60 sec</td>
<td>ON</td>
<td>2 lamps, side</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Level 2</td>
<td>Same as Clear mode</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Level 3</td>
<td>Same as Clear mode</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Level 4</td>
<td>Same as Clear mode</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
per object are used to train the Bessel K-based recognition block, the correct recognition rates are 82.19% obtained with the Pseudo-metric, 85.99% obtained with the K-measure and 85.05% obtained with the combined metric. The average test time per sample is 0.013 seconds for the Pseudo-metric, 0.1 second for the K-measure and 0.02 second for the combined metric. The experiments are performed using Matlab 7.0 and Pentium 4 CPU 3.20GHz. No optimization is applied. With a proper optimization and using C/C++, the speed of the recognition method based on the Bessel K forms may be improved 15 times and up and can meet the requirement to perform in real time. We conclude that compared with the method using the Pseudo-metric only and using the K-measure only, the combined metric balances the accuracy and speed.

We further fix the training samples to be 24 clear images per target, and test the influence of the camera and illumination effects on the recognition performance. The clear images of all targets at orientations 0, 15, 30, ..., 345 form the training set. For the simulated database, the training set is from elevation 15 and distance 10. For the Die Cast database, the training set is from elevation 20.3 degree. The remaining images at all poses are used to evaluate performance. The average correct recognition rate as a function of the distortion level is shown in Fig. 13 for the simulated database and in Fig. 14 for the Die Cast database. In Fig. 13 the five plots are parameterized by the increasing levels of Gaussian noise, motion blur, defocus blur, illumination and low contrast for the simulated database. In Fig. 14 three plots are parameterized by increasing levels of blur, illumination, and the effect of the shadow for the Die Cast database. Level 0 corresponds to the case when clear images are used for testing, which is a baseline here. From the results, we

<table>
<thead>
<tr>
<th>Simulated database</th>
<th>Die Cast database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train/Test per target</td>
<td>Bessel K</td>
</tr>
<tr>
<td>36/828</td>
<td>96.86%</td>
</tr>
<tr>
<td>24/840</td>
<td>96.83%</td>
</tr>
<tr>
<td>18/846</td>
<td>96.89%</td>
</tr>
<tr>
<td>8/856</td>
<td>91.16%</td>
</tr>
<tr>
<td>4/860</td>
<td>91.32%</td>
</tr>
</tbody>
</table>
observe that the recognition performance drops dramatically at a distortion level above 0 except for the illumination and contrast effects in the case of the simulated database.

To better understand the relationship between recognition performance and the distortion level (strength of an effect) and to compare the performance degradations for the two databases, we involve PSNR as a measure of image degradation due to an applied distortion. For each distortion type at each distortion level, we further calculate the dependence of the average PSNR on the distortion level. Fig. 15 displays the average PSNR for different types of distortions as a function of the 5 distortion levels for the simulated database. Fig. 16 shows the dependence of the average PSNR on the level of distortion for the DieCast database.

We also plot the recognition performance as a function of the average PSNR in Fig. 17 and Fig. 18. We can see that the distortions in synthetic database cover a relatively broad range of PSNR values and therefore the dependence of performance on the increasing value of PSNR is clearly observed. The DieCast database with real images and real degradations due to blur, contrast and shadows covers a very narrow range of the PSNR values. Due to this narrow range of the PSNR it is hard to make any solid conclusions about the dependence of the performance on the growing values of the PSNR.

A. System Performance

1) Simulated database: To train Haar-like feature based detector, we first generate a set of positive images with targets in the center and a set of negative images, which do not include any targets. We use the 3D tool described in Section IV to generate the positive and negative samples for training the detector. The positive samples are images of the tank, truck and tractor. Each target is projected at 36 different rotation angles and 5 different elevation angles. These 180 images per target are further shifted up, down, left and right by 3 pixels. Thus, there are total 2700 positive images. All the other objects included in the 3D tool, such as trees, airplanes, balls and other objects, are combined and projected randomly to form the negative set. Few real images are also used as negative samples. The total number of negative images generated is 3342. During the training process, all the positive images are normalized to the size $24 \times 24$. At each training stage, a set of negative samples are selected from the negative images using windows of all possible sizes. A 17 stage cascade classifier with 349 weak classifiers in it is formed.
To evaluate the performance of the detector, we generated 287 images with the total of 1116 objects, including 784 tanks, 192 trucks and 140 tractors. The color imagery is converted into the grey-scale imagery. The test results are shown in Table IV. The missed targets are the occluded targets or the targets located along the boundary of images.

To keep the hit rate at a high value, the number of false alarms is allowed to be very large. In practice, we find that some targets are included in more than one window and some detected regions cover two or more targets. We use a heuristic method to filter out large windows which cover more than two small windows and combine overlapped windows. If the window size is less than a threshold, it will be removed. After the combination and removal, the number of false alarms decreases dramatically while the number of hits remains the same. The processing time per image also reduces. The Receiver Operating Characteristic curves (ROC) before and after window combination and removal are shown in Fig. 19. We can see that the number of false alarms decreases substantially after combining windows.

<table>
<thead>
<tr>
<th></th>
<th>Hits</th>
<th>Miss</th>
<th>False Alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>After detector</td>
<td>1004</td>
<td>112</td>
<td>437</td>
</tr>
<tr>
<td>After combination and removal</td>
<td>1003</td>
<td>113</td>
<td>187</td>
</tr>
</tbody>
</table>

We further apply the proposed segmentation method to each detected region to have the targets located in the center of windows. Since the targets may have some parts outside of the detected windows, we first enlarge the window by 50% and then retrieve the targets. To speed up the segmentation, the windows are normalized to the size $64 \times 64$ and the number of control points is set to 10. The detection results are shown in Fig. 20. On the top panel, two overlapped gray windows (solid line) which contain the same tank are combined using one black window (dotted line) and further adjusted based on the segmentation results to fit the white window (dashed line). On the middle panel, there are 4 targets covered by two gray windows (solid line). The larger windows are removed after the application of the window combination procedure. On the bottom panel the gray window (solid line) of a small size is removed as non-target. Overall the proposed processing after detection reduces false alarms and allows improved location of
targets. After combination and removal, we segment targets in each adjusted window. Although the segmentation increases the processing time, it results in considerable improvements in the recognition performance.

We apply Bessel K based recognition method to the detected regions with and without segmentation. 24 images per target are used as the training samples. The correct recognition rate is 85.05% without segmentation and 95.47% with segmentation. Note that the recognition performance after segmentation is considerably better than the performance without segmentation.

To reduce the number of false alarms, we manually selected 30 background textures from 6 negative images. These background textures are further filtered using the designed bank of filters. Pairwise distances are calculated between the background textures and each detected window by using the pseudo-metric of two distribution functions. K-mean clustering is then applied to place all the background windows and each detected window into two broad groups: clutter or potential target. If the detected window is clustered in the group with textures, then it is rejected as clutter. The number of false alarms after clutter rejection is 161 and the number of hits is 997.

2) Die Cast database: To train the detector, we use 3,888 positive images of 10 objects and 172 real negative images of non-targets. Positive images are obtained by applying chroma keying technique [36] to data from Indoors dataset. The keying technique removes the blue background. After the application of keying the images of objects are resized and superimposed on a set of background images. The windows from the background images are randomly selected. Sample positive images used for training the detector are shown in Fig. 21. All positive images are further resized to $24 \times 24$ pixels. A 20 stage cascade classifier with 910 weak classifiers in it is formed.

To test the performance of the detection algorithm, we used images collected outdoors. The images are taken from scenes containing the first six objects described above and three additional non-target vehicles randomly positioned in a scene. The background in scenes was a combination of the ground, grass, sand, and gravel. Images are acquired during different time of the day from a distance of 3-4 meters. We manually divided the images used for detection into 3 categories: simple background, complex background and complex background with occlusions. Sample images representing each category are shown in Fig. 22. All images for detection are resized.
to $800 \times 536$ pixels. The number of images and objects within images for each category are listed in Table V. All images listed in Table V are used to evaluate the performance of the detector. The detection results are shown in Table VI. To reduce the impact of the backgrounds, we further apply the proposed unsupervised segmentation method to each detected region and use the segmentation results to adjust the position and size of each region such that the targets are centered and occupy the most of the window.

Since the targets may have some parts outside of the detected windows, we first enlarge the window by 50% and then retrieve the targets. To speed up the segmentation, the windows are normalized to the size $128 \times 128$ and the number of control points is set to 15. We can see that after this adjustment the number of hits increases and the number of misses and false alarms decreases. The ROC curve after the adjustments is shown in Fig. 23. The detection results are further placed into three categories in Table VI. We can conclude that the complexity of the background and occlusions are serious challenges for the detector. We then apply Bessel K based

### Table V

<table>
<thead>
<tr>
<th>Background</th>
<th>Number of Images</th>
<th>Number of Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>299</td>
<td>997</td>
</tr>
<tr>
<td>Complex</td>
<td>100</td>
<td>422</td>
</tr>
<tr>
<td>Complex with occlusion</td>
<td>50</td>
<td>128</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>499</strong></td>
<td><strong>1547</strong></td>
</tr>
</tbody>
</table>

### Table VI

<table>
<thead>
<tr>
<th></th>
<th>Hits</th>
<th>Miss</th>
<th>False Alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without Adjustment</td>
<td>764</td>
<td>783</td>
<td>3620</td>
</tr>
<tr>
<td>With Adjustment</td>
<td>788</td>
<td>759</td>
<td>3504</td>
</tr>
<tr>
<td>Simple (with adjustment)</td>
<td>598</td>
<td>399</td>
<td>2066</td>
</tr>
<tr>
<td>Complex (with adjustment)</td>
<td>148</td>
<td>274</td>
<td>658</td>
</tr>
<tr>
<td>Complex with occlusion (with adjustment)</td>
<td>42</td>
<td>86</td>
<td>780</td>
</tr>
</tbody>
</table>
recognition method to the detected regions postprocessed using the region based segmentation described in Sec. III-C. 24 images per object are involved as training samples. Only 31.98% of detected objects can be recognized correctly using Bessel K method. By analyzing the detected results in Fig. 22 we observe that some detected regions represent a part of objects. The real backgrounds and illumination make the segmentation difficult. All these factors result in overall low performance of the system.

To reduce the number of false alarms, we manually select 35 background textures from 7 negative images. After using the clutter rejection method, the number of false alarms after clutter rejection is 1,392 and the number of hits is 750. This postprocessing removes most false alarms while keeping the hit rate without change.

VI. DISCUSSIONS AND CONCLUSION

In this work, we analyzed the influence of a number of environmental and camera effects on detection and recognition performance of a complete automatic target recognition system. The system is designed to operate on optical images. The detection algorithm relies on Haar-like features. The recognition algorithm is based on estimation of Bessel K forms. Additional blocks introduced into the system included adjustment of detected regions, separation of potential targets from the background and clutter rejection after recognition.

The system performance was evaluated using two databases. The first one was a simulated database generated using a 3D optical tool. Six distorted copies of a high quality (clear) image were added to mimic various camera and environmental effects. The types of distortions included a Gaussian noise, a Poisson noise, an illumination effect, an effect of contrast change, motion and defocus blurs. We separately evaluated performance of the detector and Bessel K form-based classifier. The performance of the classifier was displayed as a function of a distortion level, which was also quantified in term of PSNR. It was clearly observed that each distortion type degraded classification performance of the Bessel K form-based method when PSNR varied from its highest value (150 for an additive Gaussian noise and 225 for a contrast effect) to its lowest value (135 for an additive Gaussian noise and 205 for a contrast effect). A modified ATR system promoted in this paper was tested on 287 images with 1116 targets. The results showed that by placing the frame adjustment block and object segmentation block between Haar feature-based detector and Bessel K form classifier we substantially reduced the number of miss detections.
and by placing the clutter rejection block at the far end of the system we reduced the number of false alarms and improved the classification performance. The performance of the proposed ATR system tested on 3 targets achieved 95.74% correct recognition rate. The classification algorithm was applied to detected and segmented regions within the tested images.

The second database was a database of real images collected from 10 die cast toy models. The objects were imaged from different elevation and orientation angles. The dataset contained both high quality (clear) and distorted images. We separately evaluated performance of the detector and the classifier and then evaluated the performance of the proposed modified ATR system. The distortions included defocus blur, sided illumination, low contrast, shadows and occlusions. The classification performance of the Bessel K method was displayed as the average correct recognition rate plotted as a function of a distortion level and PSNR for the case of blur, low contrast and shadow effects. The results (see Fig. 14, Fig. 16, and Fig. 18) showed that transition from clear images (Level 0) to images distorted at Level 1 caused significant drop in the value of the average recognition rate. A further transition from distortions at Level 1 to distortions at Level 2 through Level 5 did not almost degrade the performance of the recognition block.

The detection performance using Haar-feature detector is tested on 499 images with 1549 targets. All images used for testing were placed into three categories based on the complexity of the background and processed separately. The results showed that the simple background without shadows and occlusions was relatively well managed by the Haar feature-based detector. The achieved hit rate was equal to 60%. The complex background with occlusions constituted a serious challenge for the detector. The achieved hit rate was equal to 30%. To evaluate the modified ATR system performance, a 10 target classification was performed on detected regions after segmentation. The similarity of selected targets as well as the complex background were the major cause of a high misclassification rate (only 31% of detected objects were classified correctly). In the case of real images, the involvement of the window adjustment and object segmentation blocks resulted in some performance improvement. The clutter removal block reduced the number of false alarms substantially.

To boost the performance of the modified ATR system suggested in this paper, in the future we plan to train the Haar feature based detector both on the complete and partial images of objects. For the case of real data (Die Cast dataset), we would like to refine the recognition performance curves (see Fig. 14, Fig. 16, and Fig. 18) by introducing new refined distortion
levels in the range between Level 0 and Level 1. It means that the dataset will be augmented with new images. We also plan to involve other imaging modalities such as near infrared and long wavelength infrared, which provide substantially different information about targets and can be successfully fused with the detection and recognition results described in this paper.

REFERENCES


Fig. 1. A block-diagram of a traditional ATR system.

Fig. 2. Modified ATR system.

Fig. 3. Stage classifier.

Fig. 4. Cascade of classifiers.

Fig. 5. Representation of an image $I$ using $2J$ Bessel parameters.

Fig. 6. (a) Images, (b) Gabor components of images in (a), and (c) the marginal densities using targets in the simulated database. The empirical histogram distributions are marked in dashed line. The Bessel K form approximations are shown in solid lines.

Fig. 7. The illustration of camera parameters.

Fig. 8. The GUI of the ATR training tool.

Fig. 9. Top view images of tank, truck and tractor from the simulated database.

Fig. 10. Six distorted images of the tractor shown in Fig. 9. From left to right: the image with an additive Gaussian noise, the image distorted by a Poisson noise, the image characterized by a low illumination, the image characterized by a low contrast, the motion-blurred image and the defocused image.

Fig. 11. Images (top view) of the ten objects used to compile the Die Cast database.

Fig. 12. Sample distorted images of Object 2. The first row shows defocused images. The second row shows image degradation due to a weak illumination. The third row demonstrates increasing shadows. The distortions are demonstrated at Levels 1-4.

Fig. 13. Average recognition performance as a function of the increasing distortion level for the simulated database.

Fig. 14. Average recognition performance as a function of the increasing distortion level for the Die Cast database.

Fig. 15. Average PSNR as a function of the increasing distortion level for the simulated database.
Fig. 16. Average PSNR as a function of the increasing distortion level for the DieCast database.

Fig. 17. Average recognition performance as a function of the average PSNR for the simulated database.

Fig. 18. Average recognition performance as a function of the average PSNR for the DieCast database.

Fig. 19. Detection results of the Simulated database before window combination (dashed line) and after window combination (solid line).

Fig. 20. Shown are sample images from Simulated database with detected regions. A gray solid line marks the detector outputs. A black dotted line marks the results after window combination and removal. A white dashed line marks the results after segmentation.

Fig. 21. Positive samples from Die Cast database used to train the detector.

Fig. 22. Sample images from Die Cast database with detected regions in (a) simple background, (b) complex background and (c) complex background with occlusion. A gray solid line marks the detector outputs. A black dotted line marks the results after combination of windows. A white dashed line marks the results after segmentation.

Fig. 23. Detection results after adjustment for the Die Cast database.