Performance Modeling of Vote-based Object Recognition

Edwin S. Hong\textsuperscript{a}, Bir Bhanu\textsuperscript{b}, Grinnell Jones III\textsuperscript{b} and Xiaobing Qian\textsuperscript{b}
\textsuperscript{a}University of Washington, IT/CSS Department, Tacoma, WA 98402
\textsuperscript{b}Center for Research in Intelligent Systems, University of California, Riverside, CA 92521
e-mail: edhong@u.washington.edu; (bhanu, grinnell, xqian)@vislab.ucr.edu

ABSTRACT

The focus of this paper is on predicting the bounds on performance of a vote-based object recognition system, when the features are distorted by uncertainty in both feature locations and magnitudes, by occlusion and by clutter. A method is presented to calculate lower and upper bound predictions of the probability that objects with various levels of distorted features will be recognized correctly. The prediction method takes model similarity into account, so that when models of objects are more similar to each other, then the probability of correct recognition is lower. The effectiveness of the prediction method is validated in a synthetic aperture radar (SAR) automatic target recognition (ATR) application using MSTAR public data, which are obtained under different depression angles, object configurations and object articulations. Experiments show the performance improvement that can obtained by considering the feature magnitudes, compared to a previous performance prediction method that only considered the locations of features. In addition, the predicted performance is compared with actual performance of a vote-based SAR recognition system using the same SAR scatterer location and magnitude features.

Keywords: SAR data, performance prediction, automatic target recognition

1. INTRODUCTION

The goal of our research is to develop a formal framework for predicting the fundamental performance of any given object recognition task. This particular paper focuses on predicting performance bounds of model-based object recognition systems, where the recognition system uses a feature matching criteria that is vote-based. In the prediction problem, we are given a model database as well as criteria under which features in a model can be distorted. We predict the object recognition performance assuming the system is given test data features that are generated by taking a model from the database and subjecting it to the distortion criteria. Our distortion model includes uncertainty in both feature locations and magnitudes, occlusion and clutter.

Our prediction method is general enough to represent the behavior of a wide variety of vote-based recognition systems under many different kinds of distortion. It also takes model similarity into account, meaning that when models of objects are more similar to each other, then the predicted probability of recognition is lower. It can thus be used to explore how various distortion factors and model similarity affect object recognition performance. We believe that knowing how these factors affect object recognition performance is key to understanding and improving the effectiveness of object recognition systems.

Our prediction builds upon the previous work of Boshra and Bhanu.\textsuperscript{1} We improve their previous performance prediction method by incorporating a more general feature set, namely by including magnitude as a component of the feature of a model. Although there have been other performance prediction methods, however none of them consider uncertainty, occlusion, clutter, and model similarity when generating the prediction.\textsuperscript{1,2} In comparison with Boshra and Bhanu's previous work, we also remove some of the assumptions that they needed to simplify the performance prediction computation.

Further author information: (Send correspondence to Ed Hong)

Ed Hong: Address: Computing and Software Systems, University of Washington, 1900 Commerce Street, Box 358426, Tacoma, WA 98402, USA; Telephone: 1 253 692 4659
In addition to building a general framework for performance prediction, we also validate our technique by testing it in a synthetic aperture radar (SAR) automatic target recognition (ATR) application using MSTAR public SAR data. These data are obtained under different depression angles, object configurations and object articulations. Experiments show the performance improvement that can be obtained by considering the feature magnitude, compared to a previous performance prediction method that only considered the locations of features. In addition, the predicted performance is compared with actual performance of a vote-based SAR recognition system.

Our preliminary results show that our prediction scheme may be an effective way to predict object recognition performance. We also show that the benefit of including the magnitudes of features enables us to get the same probability of correct recognition at a 5 to 10 percent higher distortion rate.

2. PREDICTION SYSTEM

In this section, we describe our system for predicting performance bounds for any vote-based object recognition system. Our prediction system takes as input a database of models each consisting of a set of features, and a distortion model. It then obtains performance bounds on the model database given the distortion model. Our features consist of a discrete $x$ and $y$ location, as well as a magnitude. Our distortion models uncertainty (in both location and magnitude), occlusion, and clutter. Our system can also handle arbitrary types of probability distribution functions (PDFs) for uncertainty and for clutter.

Note that the prediction results depend on the entire database of models given to it. When models in the database are more similar to each other, then our prediction for the probability of correct recognition (PCR) will be lower than if the models are less similar to each other. Our system generates a series of PCR bounds, one for each model $M$ in the database. The PCR bounds for $M$ represent the probability that $M$ will be recognized correctly if it undergoes the given distortion, and the recognizer must choose from among the entire database of possible models for a result.

We first start by giving a mathematical description both vote-based object recognition and our distortion model. We then show how to compute the PCR bounds based on the mathematical descriptions.

2.1. Basic Object Recognition Definitions

Our general recognition system has a set $\mathcal{M} = \{M_1, M_2, \ldots, M_n\}$ of $n$ models. Each model consists of a set of features, where each feature is a tuple of attributes. In our work, we have three attributes: a horizontal and a vertical attribute representing a discrete 2D location, and a magnitude. More formally, a feature $f$ is a tuple $(x, y, s)$, where $(x, y) \in \mathbb{Z} \times \mathbb{Z}$ is the location of the feature, and $s \in \mathbb{R}$ is the magnitude. We use $\mathcal{F}$ to denote the space of all features. Note that $M \in \mathcal{M}$ is an arbitrary set of features; each model $M$ need not have the same number of features.

The recognition system also is given a set $\mathcal{T}$ of transformations over $\mathcal{F}$. Let $\tau \in \mathcal{T} : \mathcal{F} \to \mathcal{F}$ denote one such transformation. Then $\tau(f)$ represents the feature $f'$ which results from taking feature $f$ under transformation $\tau$. In this work, we consider $\mathcal{T}$ = set of all integral 2D-translations. The magnitude $s$ of each feature remains unchanged under our transformations. We use $\tau_{i,j}$ to denote the transform that maps $(x, y, s)$ to $(x + i, y + j, s)$.

An object recognition system is given a set of features $F$, and seeks to identify $(M, \tau) \in \mathcal{M} \times \mathcal{T}$ that best describes $F$. We call the tuple $(M, \tau)$ a hypothesis. Intuitively, the input set $F$ is generated by looking at the image data (or a simulated signature) of some model object, and then extracting a set of features from the data. Since the orientation and position of the object in the image data is unknown, the transform $\tau$ representing the location of the object needs to be found, in addition to identifying the correct model $M$. Furthermore, the feature data may be distorted, so that the match between $F$ and some hypothesis $(M, \tau)$ will generally not be exact.

We consider vote-based recognition systems, where individual features that match contribute votes, and the hypothesis is picked based on the highest vote total. Consider the number of votes obtained in such a system, by some hypothesis $(M, \tau)$ on a set of features $F$. To allow for inexact matches, we consider any feature $f \in F$ to match a feature $f' \in M$ under transformation $\tau$ when $\tau(f)$ is "close enough" to $f'$. In general, there will be some region around each model feature $f'$ that represents how close $\tau(f)$ needs to be in order to be considered a matching feature for $f'$; this region is known
as voting region of \( f' \), denoted \( \text{VR}(f') \). The size of this region typically is chosen by the recognition system to improve performance.

One example definition of VR for a feature \((x, y, s)\) is

\[
\text{VR}((x, y, s)) = \{(x', y', s') : (x', y') \in 4\text{NEIGHB}(x, y) \text{ and } .9s \leq s' \leq 1.1s\},
\]

where, \( 4\text{NEIGHB}(x, y) = \{(x, y), (x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)\} \) is a set of neighbors of \((x, y)\).

Strictly speaking, the voting region for a feature \( f_1 \) of \( M \) could be a completely different size and shape than that of feature \( f_2 \) of \( M \). For this work, however, we will assume that the size and shape of each voting region of every feature of every model \( M \) in \( \mathcal{M} \) is the same. In other words, there is just one function VR that, when applied to any feature \( f' \) of model \( M \) returns a region \( \text{VR}(f') \) centered around \( f' \). Any transformed feature \( \tau(f) \) falling in \( \text{VR}(f') \) is considered a match for \( f' \) under transformation \( \tau \).

Now let \( V(F, (M, \tau)) \) denote the number of votes generated by hypothesis \((M, \tau)\) on feature set \( F \). We define this is as:

\[
V(F, (M, \tau)) = |\{ f' \in M : \exists f \in F, \tau(f) \in \text{VR}(f') \}|
\]

In other words, we are counting the number of features \( f' \) of model \( M \) that have at least one feature \( f \) in \( F \) matching it. Note that \( V \), the number of votes, clearly depends on the exact definition for the voting region. To better distinguish between features, we might expect voting regions for two features \( f'_1 \) and \( f'_2 \) from some model \( M \) to be nonoverlapping. However, note that this is not a strict requirement of the above definition. If the voting regions of \( f'_1 \) and \( f'_2 \) overlap, then one feature \( f \) in the feature set \( F \) may potentially contribute two votes if it happens to be in the union of the two regions \( \text{VR}(f'_1) \cup \text{VR}(f'_2) \).

We focus on the forced recognition problem, meaning that the object recognition algorithm \( \mathcal{A} \) running on \( F \) returns the hypothesis \((M, \tau)\) that maximizes \( V(F, (M, \tau)) \).

The performance of the recognition algorithm is measured in terms of the percentage of correct responses when recognizing a test data set consisting of many feature sets. There is a ground truth associated with each feature set in the test data that represents the correct hypothesis. In many object recognition scenarios, a hypothesis that is close enough to the correct one is acceptable. Let \( H_{acc}(F) \) be the set of acceptable hypotheses for the feature set \( F \). Then the performance of algorithm \( \mathcal{A} \) is measured by the number of feature sets \( F \) in the test data that satisfy \( \mathcal{A}(F) \in H_{acc}(F) \).

### 2.2. Distortion Model

As part of the performance prediction scheme, we assume that the feature sets \( F \) input into the recognition algorithm are generated by distorting features of some model \( M \) in the model database. Our distortion process consists of three steps: uncertainty, occlusion, and clutter. The distortion depends on the following user-defined parameters:

- **Uncertainty PDFs** over \( F \) for each feature, representing how likely each feature is to be perturbed.
- **Occlusion amount** \((O)\), for determining the number of features to occlude.
- **Clutter amount** \((C)\), for determining the number of clutter features to add.
- **Clutter Region** \((CR)\), for determining where clutter features should be added.
- **Clutter PDF**, for determining distribution of clutter over the clutter region.
For simplicity, we describe our distortion method using uniform PDFs. Using other PDFs is also possible and easy to implement.

Let \( M = \{ f_1, f_2, \ldots, f_k \} \) be the model to be distorted. Then the distortion algorithm \( D \) does the following:

1. (Uncertainty) Replace each \( f_i = ((x, y), s) \), with a new feature \( f'_i \) chosen uniformly at random from the set
   \[ \{ (x', y', s') : (x', y') \in 4\text{NEIGHB}(x, y) \text{ and } 0.9s \leq s' \leq 1.1s \} \]

2. (Occlusion) Uniformly choose \( O \) features out of the \( k \) present (so each size \( k \) subset is equally likely); remove these features.

3. (Clutter) Add \( C \) additional features, where each feature is generated by picking a feature uniformly at random from CR (the clutter region).

The clutter region typically depends upon the given model \( M \) that we are distorting. As an example clutter region, consider the bounding box on the feature locations and magnitudes in \( M \). More formally, let \( x_{\text{max}} \) (\( x_{\text{min}} \)) represent the \( x \) value of the feature with maximum (minimum) magnitude in \( M \). Similarly define \( y_{\text{max}}, y_{\text{min}}, s_{\text{max}}, \) and \( s_{\text{min}} \). Then the bounding box clutter region CR is

\[ \text{CR} = \{ (x, y, s) : x_{\text{min}} \leq x \leq x_{\text{max}}, y_{\text{min}} \leq y \leq y_{\text{max}}, s_{\text{min}} \leq s \leq s_{\text{max}} \} \] (2)

We define the distortion region of feature \( f \), denoted by \( \text{DR}(f) \), as the union of all features that could be generated as uncertain versions of \( f \). In the above algorithm, the distortion region is the set specified in step 1. Note that in Bosshard and Bhanu,\(^1\) the distortion region and the voting regions are always the same; they are both called the uncertainty region. Separating out these two regions is useful, because the recognition algorithm may not always know the size of the distortion region. Furthermore, under non-uniform distributions where the center of the distortion region is more likely to occur, having a voting region smaller than the distortion region may result in better performance.

In step 3 of this distortion model, it is possible that a clutter feature chosen is exactly the same as an occluded model feature. If this occurs, our clutter model adds less than \( C \) additional features that can be identified as clutter.

Furthermore, it is possible for two clutter features to be chosen with the same \( (x, y) \) location. In many cases, constraints on the method of generating test data prevents two features from ever sharing the same location, and possibly even from sharing in some cases may prevent features from sharing neighboring locations. In these cases, it is possible for our distortion algorithm to generate invalid distorted data, either due to overlapping distortion regions, or to clutter. We feel that adding a more sophisticated distortion model to handle these constraints will significantly increase the complexity of the prediction system without generating a noticeable improvement in results. We argue that the distortion regions are typically small enough that they rarely overlap, and when they do, the resulting effects are small. Furthermore, number of features studied is typically small compared to the size of the clutter region, making it unlikely that overlapping clutter will significantly affect our results.

### 2.3. Prediction Method Overview

In the performance prediction problem, our goal is to determine the performance of a given recognition system when the models to be recognized undergo distortion. Our goal is to determine the probability of correct recognition (PCR), which we define as follows:

Let \( \text{PCR}(M, \mathcal{M}, \mathcal{A}, D, H_{\text{acc}}) \) denote the probability that recognition algorithm \( A \) will return a hypothesis in \( H_{\text{acc}} \) when run with knowledge of the models in \( \mathcal{M} \) on an input generated by distortion algorithm \( D \) running on model \( M \in \mathcal{M} \).

Note that \( \mathcal{A} \) (described in 2.1) depends on the size of the voting region \( \mathcal{V}_R \) for each feature and the set of transforms \( \mathcal{T} \).

Also, \( \mathcal{D} \) (described in section 2.2) depends on many distortion parameters, including occlusion and clutter. Note that the
PCR implicitly takes into account the similarity between models by using its knowledge about the features of all models in \( \mathcal{M} \).

Let \( \hat{M} \) denote the probability distribution over feature sets obtained when applying \( D \) on model \( M \).

We can now define \( V(\hat{M}, (M', \tau)) \) as a random variable representing the number of votes generated by \( \mathcal{A} \) for distorted feature set \( \hat{M} \) on hypothesis \( (M', \tau) \).

The PCR can be formally defined as

\[
\text{PCR}(M, \mathcal{M}, A, D, H_{\text{acc}}) = \Pr[\forall h' \not\in H_{\text{acc}}, \exists h \in H_{\text{acc}}, V(\hat{M}, h) > V(\hat{M}, h')].
\]

We write \( \text{PCR}(M, H_{\text{acc}}) \) to denote \( \text{PCR}(M, \mathcal{M}, A, D, H_{\text{acc}}) \) when \( \mathcal{M}, A, \) and \( D \) are fixed. Similarly, we define the probability of incorrect recognition (PIR) as

\[
\text{PIR}(M, H_{\text{acc}}) = 1 - \text{PCR}(M, H_{\text{acc}})
\]

\[
= \Pr[\exists h' \not\in H_{\text{acc}}, \forall h \in H_{\text{acc}}, V(\hat{M}, h) \leq V(\hat{M}, h')]
\]

Unfortunately, the abundance of hypotheses appears to make an exact computation of the PCR computationally infeasible. Thus we approximate the PCR by computing upper and lower bounds on it. For each hypothesis \( h \) and each model \( M \), we first compute approximate PDFs for \( V(\hat{M}, h) \). We then use these PDFs to derive upper and lower bounds on the PCR. Note that in our prediction algorithm, we compute \( V(\hat{M}, h) \) only if it is necessary to compute our bounds; we omit the computation of PDFs for many infeasible hypotheses, such as those where \( V(\hat{M}, h) = 0 \) with probability 1.

### 2.4. Lower bound on PCR

When computing \( \text{PCR}(M, H_{\text{acc}}) \), we assume that the correct hypothesis \( h_c = (M, \tau_{0,0}) \) is in \( H_{\text{acc}} \). We know that

\[
\text{PIR}(M, H_{\text{acc}}) \leq \Pr[\exists h' \not\in H_{\text{acc}}, s.t. V(\hat{M}, h_c) \leq V(\hat{M}, h')],
\]

because this probability above has less stringent conditions. This is a reasonable approximation since we expect \( h_c \) to get the most votes. We can now approximate the above by summing over probabilities of incorrect hypotheses having high numbers of votes:

\[
\text{PIR}(M, H_{\text{acc}}) \leq \sum_{h' \not\in H_{\text{acc}}} \Pr[V(\hat{M}, h_c) \leq V(\hat{M}, h')],
\]

Now assuming that the voting region is bigger than or equal to the distortion region, we know \( V(\hat{M}, h_c) \geq |M| - O \), where \( O \) is the amount of occlusion. This implies

\[
\Pr[V(\hat{M}, h_c) \leq V(\hat{M}, h')] \leq \Pr[|M| - O \leq V(\hat{M}, h')].
\]

Our overall lower bound on the PCR is

\[
\text{PCR}(M, H_{\text{acc}}) \geq 1 - \sum_{h' \not\in H_{\text{acc}}} \Pr[|M| - O \leq V(\hat{M}, h')].
\]

This is easily computed once PDFs for \( V(\hat{M}, h') \) are known.
2.5. Upper bound on PCR

Let \( h_c \) represent the correct hypothesis \((M, \tau_{0,0})\). We first approximate the PCR by

\[
\text{PCR}(M, H_{\text{acc}}) \approx \Pr[\forall h \notin H_{\text{acc}}, V(\tilde{M}, h_c) > V(\tilde{M}, h)].
\]

Note that this is exact and not an approximation when \(|H_{\text{acc}}| = 1\). Now let \( H \) be a set of hypothesis, none of which are in \( H_{\text{acc}} \). Then we can bound the above approximation by:

\[
\Pr[\forall h \notin H_{\text{acc}}, V(\tilde{M}, h_c) > V(\tilde{M}, h)] \leq \Pr[\forall h \in H, V(\tilde{M}, h_c) > V(\tilde{M}, h)].
\]

Now let us assume that for any two hypothesis \( h \) and \( h' \) in \( H \), \( \Pr[V(\tilde{M}, h_c) > V(\tilde{M}, h)] \) is independent from \( \Pr[V(\tilde{M}, h_c) > V(\tilde{M}, h')] \). That means the above upper bound can be written as

\[
\prod_{h \in H} \Pr[V(\tilde{M}, h_c) > V(\tilde{M}, h)].
\]  

(3)

Similar to the method used in the lower bound, we can approximate the above upper bound as

\[
\prod_{h \in H} \Pr[|M| - O > V(\tilde{M}, h)].
\]  

(4)

Following the work of Boshra and Bhanu,\(^2\) we use the set of peak hypotheses, in the above bound. The peak hypotheses are those hypotheses whose expected number of votes (for \( M \)) are local maxima in the space of transforms. It is formally defined as

\[
H = \{(M', \tau_{ij}) : (M', \tau) \notin H_{\text{acc}} \text{ and } M' \in \mathcal{M} \text{ and } \forall (x', y') \in 8\text{NEIGHB}(i, j), (x', y') \neq (i, j), \quad E[V(\tilde{M}, (M', \tau_{ij}))] > E[V(\tilde{M}, (M', \tau_{x',y'}))]\}
\]

Here, 8NEIGHB\((i, j)\) is the set of 8 neighbors closest to \((i, j)\) including \((i, j)\), meaning \(\{(x', y') : i - 1 \leq x' \leq i + 1 \text{ and } j - 1 \leq y' \leq j + 1\}\).

We argue that assuming the probabilities above are independent for peak hypotheses is a reasonable assumption, and not too far from the truth.

2.6. Computing \( V(\tilde{M}, h) \)

Here we compute the random variable \( V(\tilde{M}, (M', \tau)) \). For conciseness, we use \( V \) to denote \( V(\tilde{M}, (M', \tau)) \); \( \tilde{M} \) and \( (M', \tau) \) are dropped, since they are assumed to be fixed for the rest of this section. There are three components to the computation of \( V \): First, we compute the PDF of \( V_s \), a random variable representing the number of votes that \((M', \tau)\) receives due to model similarity, without considering occlusion. Then we compute the conditional PDF of \( V_o \), representing the number of similar votes that are occluded given the distribution of \( V_s \). Finally, we compute \( V_c \), the number of votes contributed due to clutter. Let \( V_{so} \) denote the number of votes due to model similarity after considering occlusion. Then

\[
V = V_{so} + V_c,
\]

where \( V_{so} \) is given by

\[
\Pr[V_{so} = x] = \frac{1}{|M'|} \sum_{v_s = x} \Pr[V_s = v_s] \Pr[V_o = v_o - x | V_s = v_s].
\]
We compute $V_o$ as a sum of indicator variables $X_i$, where $X_i$ is 0 or 1 depending on whether feature $i$ of $M$, when perturbed according to the uncertainty PDF of the distortion model, will contribute a vote to hypothesis $(M', \tau)$. For the uniform distribution, $\Pr[X_i = 1]$ is very simple to calculate. Let $f_i$ be feature $i$ of $M$. Let $VR(M')$ denote the union of all voting regions of all features of $M'$. Then $\Pr[X_i = 1] = VOL(DR(f_i) \cap VR(M'))/VOL(DR(f_i))$, where VOL denotes the volume, calculated over both the location and magnitude.

For non-uniform distributions, the $\Pr[X_i = 1]$ is also fairly simple to calculate. It would simply mean calculating $\Pr[f_i \in DR(f_i) \cap VR(M')]$ according to the probability distribution given for the uncertainty in location and magnitude.

Note that we are summing over the distorted features in $\tilde{M}$ to obtain our similarity count, and not summing over all possible voting regions that may contribute a vote. If there is no feature whose voting region overlaps with more than one distortion region, than our similarity count will be exact. If there are overlaps, then our method may over count by letting one voting region contribute more than one vote. However, we do not expect much overlap to occur in practice.

The conditional PDF of $V_o$ is given by the hypergeometric distribution.

$$\Pr[V_o = x|V_s = v_s] = hg(x, |M'| - v_s, O, |M'|),$$

where

$$hg(x, n_1, n_2, N) = \binom{n_1}{x} \binom{N - n_1}{n_2 - x} / \binom{N}{n_2}.$$

Let $p_C$ denote the probability that one clutter feature will contribute a vote. We calculate $V_o$ as a sum of $C$ random indicator variables $Y_1, \ldots, Y_C$, where $Y_i$ is 1 with probability $p_C$. Given a uniform distribution for clutter (in both magnitude and location), we know

$$p_C = VOL(CR \cap VR(M'))/VOL(CR),$$

where CR is the clutter region, and $VR(M')$ is the union of the voting regions for each feature of model $M'$. Calculating $p_C$ for non-uniform distributions is also relatively straightforward, by calculating a volume weighted by the PDF of the clutter region.

For example, assume we have a clutter PDF that is non-uniform over the location of features and uniform over magnitude. Let $w_{i,j}$ represent weight assigned for the clutter PDF at location $(i, j)$. Since this is a PDF, we know $\sum_{(i,j)} w_{i,j} = 1$. Let $s_{\min}$ and $s_{\max}$ be the bounds for the clutter PDF magnitude at each location. Define $LOC = \{(x, y) : \exists s, (x, y, s) \in CR \cap VR(M')\}$, as the set of possible locations for clutter. Then we would have the probability of clutter contributing a vote as

$$p_C = \frac{\sum_{(i,j) \in LOC} w_{i,j} |\{s : s_{\min} \leq s \leq s_{\max}\} \cap \{(s : (i, j, s) \in VR(M')\}|}{s_{\max} - s_{\min}}.$$

3. PRELIMINARY RESULTS

In order to validate the above object recognition performance prediction method, we apply it to identifying target vehicles from publicly available synthetic aperture radar (SAR) data. The data consists of images of vehicles at many different azimuths, taken from a 15 degree depression angle. Each image is first preprocessed to generate scattering centers, which are local eight-neighborhood peaks in magnitude of the SAR data images. In our prediction system, these scattering centers are the model features and each image of a vehicle at one particular azimuth corresponds to one model $M$. In the data sets we used, there are typically between 50 and 80 scattering centers generated in the image preprocessing step. We limit the number of features of each model to 30 by using only 30 scattering centers with the highest magnitude in each image. The scattering centers with higher magnitude correspond more to the detailed geometry of the object than those of lower magnitude.

The model database $\mathcal{M}$ that we use consists of 582 models: 194 images of a T-72 tank in configuration #132, 194 images of a BMP2 Infantry Fighting Vehicle in configuration #c21, and 194 images of a BTR-70 Armored Personnel Carrier in configuration #c71. Note that each image depicts a vehicle at a different azimuth.
In the preliminary results reported below, we used the following parameters in all cases:

- The voting region VR for each feature is the 4 neighborhood region around the feature, along with a 10% uncertainty in magnitude. In other words, it is the sample definition given in equation (1).
- There is only one acceptable correct hypothesis for each generated distorted feature set. In other words, when feature set $F$ is a distorted version of model $M$, then the set of acceptable hypothesis is given by $H_{acc}(F) = \{(M, r_0, 0)\}$.
- Uncertainty PDFs in the distortion model are chosen uniformly at random over the uncertainty region, and the uncertainty region around each feature is the same as the voting region given in 1.
- The occlusion rate ($O$) is equal to the clutter rate ($C$). Thus, after occluding $O$ features and adding $C$ clutter features, the number of features for each distorted model (30) remains the same as for the original model. Hence forth, we use the term distortion amount to refer to both $O$ and $C$. We use distortion rate to refer to the percentage of distorted features.
- We use a slight modification of the bounding box clutter region given in equation (2). Our clutter region replaces the condition $s_{\min} \leq s \leq s_{\max}$ in equation (2) with $s_{\min} - 10 \leq s \leq s_{\max}$. Allowing features with lower magnitudes models the effect of having high-magnitude scattering centers obscured by clutter, and replaced by scattering centers that were not originally part of the top 30 scattering centers.
- The Clutter PDF in the distortion model is uniform.

Our first experiment calculates the upper and lower bound to the PCR for each model $M$, according to the method presented in sections 2.4, 2.5, and 2.6. To generate an average PCR figure for the entire database $\mathcal{M}$, we take the arithmetic mean of the PCRs for each model in the database. We compute the PCR bounds for each possible distortion amount, from 10 up to the maximum of 30. This is shown in Figure 1.

![Performance Prediction, 582 model views](image_url)

**Figure 1.** PCR results for prediction of upper and lower bounds.

In our next experiment, we examined how well our upper and lower bounds predict to an experimentally derived estimate of the actual PCR value. The actual PCR values were computed as follows: First create a set of randomly generated distorted feature sets by using the distortion algorithm on the database models. We computed exactly 4 distorted copies of each model. We then run the recognition algorithm on the distorted feature sets to compute the resulting hypothesis for each. The PCR computed is the fraction of correctly recognized feature sets, meaning the ones whose resultant hypothesis matches the model from which the feature set was created.
In this preliminary study, we report results using only the T-72 tank images from the first experiment as our model database. Results are shown in Figure 2. As can be seen, the results are not good in that the bounds are not that close to the actual value, and in fact the predicted upper bound is not even above the actual value.

![Performance Prediction, 194 model views of T72.1](image)

**Figure 2.** PCR prediction results compared to actual values.

After analyzing our method, we determined that the most significant source of error appears to be the approximation of \( \Pr[V(\bar{M}, h_c) \leq V(\bar{M}, h)] \) with \( \Pr[|M| - O \leq V(\bar{M}, h)] \), where \( h_c \) is the correct hypothesis. When the distortion rate is low, we expect \( V(\bar{M}, h_c) \) to get about \( |M| - O \) votes since without much added clutter, the correct hypothesis should not be getting many extra clutter votes incorrectly matched. However, as the distortion rates get large, the approximation gets less and less accurate as the larger amount of clutter makes it more likely to add in extra votes for the correct hypothesis. In fact, we can see that our bounds for PCR are getting worse as the distortion rate increases. It is also clear that the upper bound presented in Section 2.5 is not really an upper bound. The upper bound approximation (equation (4)) really needs to be greater than the previous upper bound equation (equation (3)), when in fact, it is less than that.

The two main problems with our above method are that the lower bound is not that close to the actual value, and the upper bound is completely inaccurate. One way of potentially fixing these problems is to simply try to approximate the actual value, rather than trying to get both upper and lower bounds. To this end, we also ran our final preliminary experiment, where we approximated \( \Pr[V(\bar{M}, h_c) > V(\bar{M}, h)] \) with \( \Pr[E(V(\bar{M}, h_c)) > V(\bar{M}, h)] \), where \( h_c \) is the correct hypothesis, and \( E() \) denotes expected value. Here, we computed \( E(V(\bar{M}, h_c)) \) as \( |M| - O + |E(V(\bar{M}, C))| + 1 \), where \( V(\bar{M}, C) \) is a random variable for the number of clutter votes that would be added when given a distorted model \( M \) and \( C \) clutter features. The overall approximate prediction value can be written as:

\[
\prod_{h \in H} \Pr[E((V \bar{M}), h_c) > V(\bar{M}, h)]
\]

This approximation result is shown in Figure 3, for both the 3D features (including the magnitude), and for 2D features where the magnitude is ignored.

As can be seen, the predicted results are close to the actual values. Also note that when we add the third dimension of magnitude, we can have 10% more distortion than before while still maintaining the same PCR level as before.

4. CONCLUSIONS

We have presented a general model for predicting the bounds on performance of a wide variety of vote-based object recognition systems. We model features as a 2D location and a magnitude, and consider uncertainty (in both magnitude and
Figure 3. The new PCR prediction results compared to actual values, for both 2D (location only) and 3D (including magnitude) feature sets.

location), occlusion, clutter, and model similarity as part of our performance prediction. We have done some preliminary evaluation of our general model to show its potential effectiveness.

Much, however, remains to be done in this work. More comprehensive tests on more sets of models need to be carried out. Furthermore, instead of only comparing to synthetically generated distorted data, performing tests on actual data distortion would help to validate our work. Furthermore, it would be interesting to use our model to predict the performance of vote-based schemes\(^3\) and\(^4\). Comparing our predictions to these implemented recognition systems would be useful in showing that the sanitized, mathematically-described algorithms of vote-based recognition that we use actually are similar enough to the actually implemented algorithms and that our prediction can in fact predict the behavior of the real algorithms.

ACKNOWLEDGMENTS

This work was supported by grant F49620-01-1-0343. The contents of the information do not necessarily reflect the position or policy of the U.S. Government.

REFERENCES