Sensor Modeling, Probabilistic Hypothesis Generation, and Robust Localization for Object Recognition

Mark D. Wheeler and Katsushi Ikeuchi

Abstract— **In an effort to make object recognition efficient and accurate enough for real applications, we have developed three probabilistic techniques—sensor modeling, probabilistic hypothesis generation, and robust localization—which form the basis of a promising paradigm for object recognition. Our techniques effectively exploit prior knowledge to reduce the number of hypotheses that must be tested during recognition. Our recognition approach utilizes statistical constraints on the matches between image and model features. These statistical constraints are computed using a model of the entire sensing process—resulting in more realistic and tighter constraints on matches. The candidate hypotheses are pruned by probabilistic constraint satisfaction to select likely matches based on the image evidence and prior statistical constraints. The resulting hypotheses are ordered most-likely first for verification, thus minimizing unnecessary verifications. The reliability of the verification decision is significantly increased by the use of a robust localization algorithm. Our localization algorithm reliably locates objects despite partial occlusion and significant errors in initial location estimates. We have implemented these techniques in a system that recognizes polyhedral objects in range images. Our results demonstrate accurate recognition while greatly limiting the number of verifications.**

Index Terms— **Computer vision, 3D object recognition, probabilistic recognition, Markov random fields, robust pose estimation.**

I. Introduction

RECOGNIZING known objects in an image has been a principal goal of computer vision since its inception. A practical solution to this problem has numerous applications and will greatly impact the field of intelligent robotics. Over the past two decades, many researchers have studied the object recognition problem and have built experimental systems. The most successful recognition systems (that identify and localize) rely on matching object features to features in the image. The usual strategy is known as hypothesize-and-verify. This formulation divides the problem into first hypothesizing correspondences and then testing/verifying the hypotheses. The verification process requires that the pose of the hypothesized object be computed. Techniques have been developed to make the correspondence search more efficient by applying geometric and photometric constraints. Many researchers have independently studied the problem of pose estimation from feature correspondences. Despite these efforts, current systems lack the efficiency and accuracy to be widely applied in practice.

Our approach to object recognition strives to exploit our prior knowledge of the sensing process to improve recognition performance and accuracy. Improvements are necessary for object recognition systems to begin to approach the requirements for real applications. We briefly discuss a few areas where opportunities for improvement exist through effective application of prior knowledge.

Inaccurate Constraints: Hypothesis-generation procedures must compensate for inaccurate prior models by relaxing constraints, thus increasing the number of incorrect hypotheses that are generated. Most systems only model the geometry of the object and do not account for the other factors in the sensing process (e.g., the sensor, and feature extraction). The discrepancy between the simplified model and the complete model results in constraint inaccuracies which ultimately increases the recognition time and the likelihood for recognition errors.

Random Search Order: Optimizing the order in which the hypotheses are verified can reduce the number of verifications performed and, thus, the recognition time. Systems that do not optimize the ordering of hypotheses are prone to testing more hypotheses than should be necessary to identify known objects in the image. Few systems make proper use of prior knowledge to guide the order of the search so that the most likely hypotheses are verified first. At best, heuristics are used to order the search.

Inaccurate Pose Estimates: Correct verification of a hypothesis relies on accurate localization of the hypothesized object. Current polynomial-time recognition systems use correspondences of minimal sets of features to locate a hypothesized object before verification. Unfortunately, using the minimal number of features often results in inaccurate location estimates. One remedy is to increase the tolerances for the verification decision; however, this results in more recognition errors. The other option is to refine the pose estimate using more of the available image data; unfortunately, typical approaches to pose refinement are sensitive to missing object

Manuscript received Aug. 11, 1993; revised Sept. 15, 1994.

The authors are with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213-3891. email: mdwheel@cs.cmu.edu, and ki@cs.cmu.edu.

IEEECS Log Number P95028.

features in the image (e.g., from partial occlusion). This means that the refined pose may not be an improvement unless the number of missing image features is insignificant.

Our approach to improve efficiency and accuracy is based on maximizing the use of our prior knowledge to constrain and order the search. We represent prior knowledge using statistical constraints which model the relationship between object features and their corresponding image features. The statistical constraints are generated by modeling the entire sensing process—in an effort to make them as accurate as possible. For efficient recognition, the statistical constraints are applied to minimize the number of unnecessary verifications of hypotheses for recognition. Probabilistic measures based on statistical priors are used to select and order hypotheses for verification; they help remove relatively unlikely hypotheses from consideration and enable us to verify the most likely hypotheses first. Robust pose refinement, by increasing the accuracy of the location estimate, enhances the robustness and accuracy of the verification test as well.

Our object recognition strategy comprises three primary techniques: sensor modeling, probabilistic hypothesis generation, and robust localization.

Sensor Modeling: We use sensor modeling to build constraints on the relationship between object features and their corresponding image features. Our approach accounts for the important factors of the sensing process: sensor characteristics, feature-extraction-algorithm behavior, and model geometry. Statistical information about image features arising from the object is computed from a large set of ray-traced images of the object—simulating the entire sensing process from image formation to feature extraction.

Probabilistic Hypothesis Generation: In this work, we are only interested in verifying the most likely hypotheses according to our prior knowledge. This is in contrast to the typical constrained combinatorial search which tests all possible hypotheses that satisfy the prior constraints. Our sensor-modeling technique generates statistical constraints that are used to compute the relative likelihoods of the hypotheses. Hypotheses are pruned by probabilistic constraint satisfaction. The likelihoods of the hypotheses, their interdependencies, and the image evidence are used to select the hypotheses to verify—eliminating hypotheses with little supporting evidence while selecting those that have strong supporting evidence. The resulting hypotheses are ordered by their likelihoods so that the most likely are verified first.

Robust Localization: Our localization method refines the pose estimate obtained from a minimal set of correspondences between image and model features. An energy minimization approach—similar to active contours [2]—is used to estimate the object's precise position. Our algorithm uses a robust estimator for the energy function being minimized. The robust estimator is relatively insensitive to outliers occurring due to partial occlusion.

These methods were implemented and tested in a system that recognizes polyhedral models in range images. We present results of several experiments to demonstrate the ability of our system to recognize objects while greatly limiting the number of verifications performed.

II. Related Work

Presently, the most prominent recognition paradigm is the interpretation tree (IT) search made famous by Grimson and Lozano-Perez [3]. The idea of IT search is to explore the combinations of matches between image features and model features. These combinations effectively form a tree of labelings where each path from a leaf to the root represents an interpretation. The basic algorithm, searching all paths, is exponential in the number of image features. Several researchers [4], [5], [6], [7] have observed that at a certain level in the IT, the pose of the object can be computed. The idea is to first find a sufficient set of matches to determine the pose and then to use the rigidity constraint to efficiently determine the other correspondences. The result is a recognition algorithm that is polynomial in the number of image features. Its simplest form is the approach of Huttenlocher and Ullman [8] known as *alignment*.

Several other researchers independently developed IT/Alignment algorithms. The general goal of these algorithms is to minimize search by applying prior knowledge to prune and order the search. If the first few selected matches are correct, IT/Alignment algorithms can be very efficient. The 3DPO system of Bolles and Horaud [4] started with the most obvious possible match and grows the matches by searching for feature matches that will add the most information to the current interpretation, thereby reducing the degrees of freedom in the interpretation. Similarly, Faugeras and Hebert [5] used the rigidity constraint to select subsequent matches in the IT—performing recognition and localization simultaneously. Grimson and Lozano-Perez [3] explored the use of geometric constraints to prune the search. Ikeuchi [9] presented a technique for precompiling the order of comparisons of an interpretation tree. Flynn and Jain [7] used heuristic knowledge of the model database to order and prune the tree for efficient search. Using probabilistic evidence (perceptual grouping) to order the IT search was introduced by Lowe [6]. Camps, Haralick and Shapiro [10] took the approach of using probabilistic evidence to cut-off or prune the IT. IT search is a conservative approach since it considers all possibilities. This is both a strength and weakness; it is robust at the expense of efficiency.

One of the oldest object recognition paradigms frames the labeling problem in terms of constraint satisfaction networks (CSN). Typically, each node in the CSN represents an image feature and its label represents its matching model feature. An energy function that accounts for the model constraints is specified over the CSN such that the best labeling of the image features produces the minimum energy value. Optimization techniques are then applied to solve for the minimum energy state of the CSN. Bhanu [11] used relaxation labeling to determine the labeling of image regions to model regions that is most consistent with the 3D model. Bolle, Califano, and Kjeldsen [12] described a paradigm for recognition which uses networks of constraints between each level of representation

from extracted image features to object hypotheses. Cooper [13] modeled object and image primitives in a Markov random field (MRF) and used optimization to find good interpretations of the scene. Wells [14] formulated the object recognition problem as a maximum a posteriori (MAP) estimation problem over the correspondence and pose spaces. Ben-Arie [15] used relaxation techniques with statistical constraints on interpretations. Constraint satisfaction search ventures that the labeling metric can correctly distinguish the correct from incorrect labelings—sacrificing robustness for efficiency.

A recent development that has implications for all model-based vision systems is the use of sensor models. Sensor models were first used in a model-based vision system by Ikeuchi and Kanade [16]. Prior to their work, model-based recognition systems relied solely on geometric constraints of the objects. Ikeuchi and Kanade developed an analytic model of feature detectability and stability with respect to the sensor. This model was used to select features for aspect classification. Camps et al. [10] also developed an analytic model to predict the prior probability of the appearance of intensity edges in a grey level image.

A required capability of object recognition systems is to locate identified objects in the image. Several researchers have developed techniques to compute the location of the object given the correspondences between model and image features (pose estimation) or given a rough estimate of the object's location (pose refinement).

Pose estimation techniques can be used by an alignment search to generate the initial pose estimate. The general problem is to compute the pose of the object given a set of correspondences between image and model features. Typically, the techniques involve minimization of some error function over the free model parameters. Lowe's [17] method minimizes the least-squared error using a technique based on Newton-Raphson root-finding and Levenberg-Marquardt minimization to iteratively compute the model parameters. Kriegman and Ponce [18] used numerical techniques to solve large algebraic systems representing the location constraints and the error of the match between the image and the model. Bolle and Cooper [19] derived equations to compute the maximum likelihood estimate of the pose of objects composed of planar and conic surface patches matched to patches extracted from range data. Haralick et al. [20] investigated the use of robust weight functions with weighted least-squares estimation for point-based pose estimation. They focus on the situation in which some of the given correspondences may be incorrect.

Pose refinement is essential (as will be discussed later) for alignment-based [8] techniques. Typically, a few correspondences are used to compute the initial pose estimate; other correspondences are then found by local search in the image. The additional correspondences overconstrain the pose. The best example of this approach is active contours, introduced by Kass, Witkin, and Terzopoulos [2]. Active contours are models that interact with a "physical" system in which the equilibrium state determines the model parameters that bring about the best match between the image and the model. Besl and McKay [21] described a method which iteratively computes the closest data point to each model point and solves for the optimal rigid transform.

Our approach to recognition combines ideas from IT/Alignment search and constraint satisfaction techniques and extends the concept of sensor models. Our pose refinement technique is based on the ideas of active contours and robust estimation. The next three sections detail the main contributions of this paper: sensor modeling, probabilistic hypothesis generation, and robust localization.

III. Sensor Modeling

The constraints used by the hypothesis-generation process of most model-based vision systems are based solely on the geometric models of the objects and do not account for sensor or feature-extraction characteristics. Without an accurate model of these characteristics, the hypothesis-generation procedure must compensate for the inaccuracies by loosening the constraints and, thus, increasing the number of incorrect hypotheses that are generated. Our solution is to use *sensor modeling*, modeling the sensing process from imaging to feature extraction, to build accurate constraints due to the sensor and feature-extraction characteristics in addition to model geometry.

We use the Vantage solid modeler [22] to model our objects using constructive solid geometry. Our current system's sensor modality is range data, and the image features are planar regions extracted from the range image. Our recognition system searches for matches between planar regions R_i of the image and model faces M_j. Each region R_i is described by a vector of n_1 attribute values, $f_{R_i} = (f_{R_i}^1, f_{R_i}^2, \ldots, f_{R_i}^{n_1})$. The attributes are specified over 3D surfaces corresponding to planar regions extracted from range images. For this application, we use the following attributes: region area, maximum second moment, minimum second moment, and maximum diameter. Relationships between pairs of regions R_i and R_j are described by a vector of n_2 attributes,

$$\bar{f}_{R_i,R_j} = \left(f_{R_i,R_j}^1, f_{R_i,R_j}^2, \ldots, f_{R_i,R_j}^{n_2} \right).$$

These relational attributes include simultaneous visibility, relative orientation, and maximum distance between surfaces. The constraints used by our hypothesis generation algorithm are in the form of statistical distributions of the observed attributes of model faces represented by conditional distributions, $P\left(f_{R_i}^k \middle| M_j\right)$ and $P\left(f_{R_i,R_j}^k \middle| M_l, M_m\right)$, and prior probabilities $P(M_j)$. These distributions are approximated by generating and analyzing many sample images (320 images) of our object models.

Our sensor model comprises an appearance simulator and a feature-extraction algorithm. Sample range images of each object over varying poses are generated using an appearance simulator developed by Fujiwara et al. [23].[1] Regions are ex-

[1] No noise was added to the simulated images since the segmentations of the synthetic images were qualitatively similar to real images. A quantitative comparison of our sensor model to the real sensor is presented in Section VI.

tracted from the simulated images using a best-first region growing algorithm, and the attributes of each region are calculated. Fig. 1 shows an example iteration of the sensor modeling process. Since this is a simulation, we know the correspondence between the model surfaces and the image regions. Thus, we can build a list of the sampled attribute values for each model surface. With the sampled attribute values, prior distributions $P\left(f_{R_i}^k \middle| M_j\right)$ are computed for each attribute $f_{R_i}^k$ with respect to each model face M_j, and likewise for the distributions $P\left(f_{R_i,R_j}^k \middle| M_l, M_m\right)$ of second-order attributes. The prior probability $P(M_i)$ is simply the percentage of times that M_i is found by feature extraction over the set of simulated images. In this work, the viewing directions are "uniformly" distributed on the unit sphere; distribution of a model face's area value as computed using our sensor model. The figure also shows some other constraint approximations that have been used in other object recognition systems: a normal distribution fitted to the observed data, a normal distribution centered at the actual model area value, and a bounded error threshold centered around the actual model area value. These distributions are shown to demonstrate the difference between the usual assumptions of performance and the actual performance of the feature-extraction program. With simplified models, a large fraction of incorrect hypotheses will not be filtered by the constraints. The simulated distributions are not necessarily Gaussian, and attributes can be biased due to inherent characteristics; however, the distribution can be modified to reflect real world constraints.

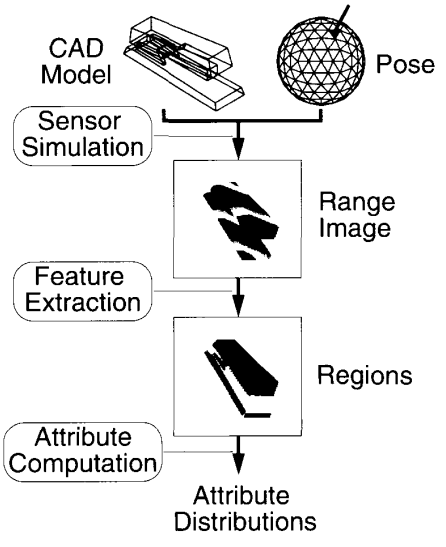


Fig. 1. An iteration of sensor modeling.

Fig. 2 shows a sample of the sensing process. Additional bias occurs from self-occlusion when viewing some objects from certain directions. The effect of self-occlusion is implicit in the statistics since it is an inherent property of the object geometry. The sensor-modeling approach builds constraints on the sensory information that will be used for recognition. Model features that are not detectable by the feature-extraction program will not affect the hypothesis generation. The inclusion of imaging and processing effects is the essential difference between our prior models and the constraints used by Burns and Riseman [24] and Ben-Arie [15].

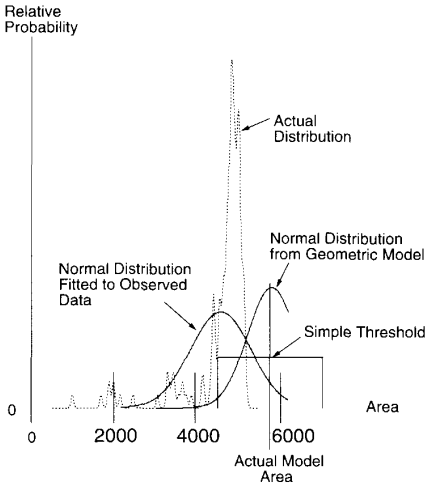


Fig. 2. An example distribution [1] of a given attribute value (area) over model face.

IV. Probabilistic Hypothesis Generation

Given a set of primitive features (e.g., regions or edges) extracted from the input image by a feature-extraction algorithm (e.g., surface segmentation or edge detection), the hypothesis-generation procedure produces a set of *possible* model features to image feature matches (hence referred to simply as hypotheses). Optimally, the generated hypotheses include all of the correct correspondences and exclude as many incorrect ones as possible. To exclude incorrect matches, we must apply constraints derived from our prior knowledge (e.g., the sensor and object models).

During recognition, we are considering many hypotheses simultaneously and wish to choose the most likely subset of these. We can think of the hypotheses as a set of variables, each of which can be assigned a discrete value of **on** or **off**. A hypothesis labeled **on** indicates that the hypothesis is assumed to be correct. A hypothesis labeled **off** indicates that the hypothesis is assumed to be incorrect. The hypotheses display Markovian characteristics. For example, if two hypotheses provide mutual support for each other, and one of them is correct, then the other is likely to be correct as well. A similar dependency exists between contradicting hypotheses. These dependencies can be represented by conditional probability distributions in the Markov random field (MRF) framework. The reason for using the MRF is that it can conveniently represent our probabilistic constraints on the matches and the dependencies between matches. The minimum energy state of the MRF represents the state that best satisfies the constraints—balancing the weight of supporting evidence and contradictory

evidence. The reader can think of the MRF representation as a probabilistic constraint satisfaction network with constraints defined by statistical priors. For a review of MRFs and their applications to computer vision, we refer the reader to the description found in [25].

A. Formulation of Hypothesis Generation using Markov Random Fields

MRFs are used to represent the probability distribution of the values of a set of random variables. A MRF can represent the conditional dependencies of each variable's value on the values of its *neighbor* variables. For our application, each MRF variable, X_{R_i,M_j}, represents the hypothesis that region R_i in the image arose from model face M_j. Each X_{R_i,M_j} has an associated value $\omega_{R_i,M_j} \in \{\text{ON, OFF}\}$. The values *ON* and *OFF* indicate our belief or disbelief, respectively, in the hypothesis. We use X to denote the set of MRF variables X_{R_i,M_j} and ω to denote the current assignment ω_{R_i,M_j} to those variables.

To represent conditional dependencies between the MRF variables, we define two *neighborhood relations* between pairs of hypotheses: N^- for contradictory hypotheses and N^+ for supporting hypotheses. The N^- neighborhood is defined as

$$N^- = \left\{ \left(X_{R_i,M_j}, X_{R_i,M_k} \right) \middle| \left(M_j \neq M_k \right) \right\}. \quad (1)$$

The above rule essentially states that two hypotheses corresponding to the same image region are contradictory. The N^+ neighborhood is defined as

$$N^+ = \left\{ \left(X_{R_i,M_j}, X_{R_k,M_l} \right) \middle| P\left(\vec{f}_{R_i,R_k} \middle| M_j, M_l \right) > 0 \right\}. \quad (2)$$

The N^+ rule specifies that if two hypotheses are consistent with respect to our prior constraints, then they provide mutual support for each other.

Given a set of independent observations, the extracted image regions $R_1, R_2, \ldots, R_n$, the most likely state of the MRF variables can be found by minimizing its posterior energy function

$$U\left(\omega \middle| R_1, R_2, \ldots, R_n\right) = \sum_{c \in C} V_c(\omega) - \sum_{X_{R_i,M_j} \in X} \log P\left(R_i \middle| \omega_{R_i,M_j}\right) \quad (3)$$

where C is the set of cliques of neighbor variables in the MRF, and $V_c(\omega)$ measures the potential (energy) of clique c under assignment ω. The energy function is in terms of things we can calculate or specify: clique potentials $V_c(\omega)$ (represent higher-order, prior constraints among related variables) and prior distributions for our observations $P\left(R_i \middle| \omega_{R_i,M_j}\right)$. By defining our constraints in terms of clique potentials and likelihoods in the MRF framework, we formulate the search for the most likely hypotheses as a *maximum a posteriori* (MAP) estimate of the MRF's state (i.e., the minimum energy state). The result is the set of hypotheses with the highest probability of occurring based on our prior constraints.

We first describe the computation of the prior probabilities $\left(P\left(R_i \middle| \omega_{R_i,M_j}\right)\right)$ from our compiled statistical distributions. As described in Section III, each region, R_i, is described by a vector of attribute values $\vec{f}_{R_i} = \left(f_{R_i}^1, f_{R_i}^2, \ldots, f_{R_i}^{n_1}\right)$. These attributes are assumed to be independent for a given model face. This assumption is required to efficiently represent and compute the probabilities. If the attributes are not independent, then we have redundant attributes which should be removed. The independence assumption gives us

$$P\left(\vec{f}_{R_i} \middle| M_j\right) = \Pi_{k=1}^{n_1} P\left(f_{R_i}^k \middle| M_j\right).$$

We need to determine the likelihood that an image region, R_i, arose from the presence of a model face, M_j, in the scene. This is the probability of observing R_i assuming that the match hypothesis $\left(R_i, M_j\right)$ is correct. In terms of our label set, we equate

$$P\left(R_i \middle| \omega_{R_i,M_j} = ON\right) = P\left(R_i \middle| M_j\right) = P\left(\vec{f}_{R_i} \middle| M_j\right). \quad (4)$$

To calculate the probability that R_i was observed given that hypothesis $\left(R_i, M_j\right)$ is incorrect (i.e., R_i resulted from some other face or background), we use

$$P\left(R_i \middle| \omega_{R_i,M_j} = OFF\right) = P\left(R_i \middle| \neg M_j\right) = \frac{\Sigma_{k \neq j} P\left(R_i \middle| M_k\right) P\left(M_k\right) + P\left(R_i \middle| B\right) P(B)}{1 - P\left(M_j\right)} \quad (5)$$

where $P\left(M_j\right)$ is the normalized probability of detecting $M_j \left(\Sigma_i P\left(M_i\right) + P(B) = 1\right)$, and B represents the possibility of background or no label. $P\left(R_i \middle| B\right)$ and $P(B)$ are set to constant values (0.000003 and 0.3, respectively).

Equations (4) and (5) provide us with the prior probabilities of the observations $\left(P\left(R_i \middle| \omega_{R_i,M_j}\right)\right)$ required for the energy function of (3). Next, we need to specify the clique potentials. For efficiency concerns, we only consider 1- and 2-cliques when computing the energy function.[2] In other words, $V_c(\omega) = 0$ when $|c| > 2$. For 1-cliques, clique c consists of a single MRF variable. The 1-clique energies of $c = \{X_{R_i,M_j}\}$ correspond to the prior probabilities, $P\left(M_j\right)$, of the hypothesized model face. When the hypothesis $\left(R_i, M_j\right)$ is *ON*, the 1-clique energy is $V_c(\omega_{R_i,M_j} = ON) = \log P\left(M_j\right)$, and when the hypothesis $\left(R_i, M_j\right)$ is *OFF*, the 1-clique energy is $V_c(\omega_{R_i,M_j} = OFF) = \log\left(1 - P\left(M_j\right)\right)$.

For 2-cliques, clique c consists of two related MRF variables: $c = \{X_{R_i,M_j}, X_{R_k,M_l}\}$ such that $\{X_{R_i,M_j}, X_{R_k,M_l}\} \in$

[2] The restriction to 1- and 2-cliques is only for computing the energy function. As explained later in this section, 3-cliques are used to form hypotheses for verification.

$N^+ \cup N^-$. The 2-clique potentials in (3) used in our experiments are

$$V_{c\in N^+}\left(\omega_{R_i,M_j} = ON, \omega_{R_k,M_l} = ON\right) = -60.0,$$
$$V_{c\in N^+}\left(\omega_{R_i,M_j} = OFF, \omega_{R_k,M_l} = OFF\right) = 40.0,$$
$$V_{c\in N^-}\left(\omega_{R_i,M_j} = ON, \omega_{R_k,M_l} = ON\right) = 60.0,$$
$$V_{c\in N^-}\left(\omega_{R_i,M_j} = ON, \omega_{R_k,M_l} = OFF\right) = -10.0$$

(other combinations are given zero energy). For example, if a hypothesis is *ON* and a consistent (N^+) neighbor hypothesis is *ON*, –60.0 is the potential of that 2-clique. The choice of clique potentials ultimately determines the amount of supporting evidence that a hypothesis requires to survive the pruning (MRF energy minimization). The potentials used here were determined experimentally to conform with our sense of consistency and mutual support among hypotheses and to provide a reasonable level of pruning.

At run time, the recognition program extracts regions from the image and computes the attributes over all regions and relational attributes over all pairs of regions. The extracted regions are used to build the MRF representing the hypotheses and prior constraints. Using (4) and (5), we first compute the log-likelihoods of the observation of R_i, given that hypothesis (R_i, M_m) is correct, $\log P\left(R_i \middle| \omega_{R_i,M_m} = ON\right)$, and incorrect, $\log P\left(R_i \middle| \omega_{R_i,M_m} = OFF\right)$. When computing $P(R_i|M_m)$, we throw out hypotheses that have zero probability. With the remaining hypotheses, we then determine the neighborhood relations N^- and N^+ using the rules specified in (1) and (2). This information is sufficient to form the MRF.

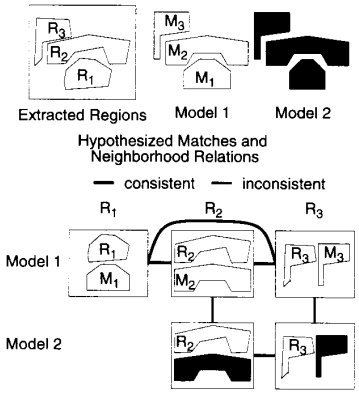


Fig. 3. An example MRF produced from a simple scene containing three regions with a model database containing two similar geometric objects.

To help the reader visualize a typical resulting MRF, a very simple example is shown in Fig. 3. In this example, the model database contains two similar geometric models, model 1 and model 2. A MRF is constructed for this model database and an image of model 1 from which three regions were extracted. The matches shown in Fig. 3 are the pairs of regions R_i and model faces M_j that have nonzero conditional probabilities $P\left(R_i \middle| M_j\right)$. Notice that obvious mismatches such as R_3 and M_2 are not generated since they have zero prior probability. The neighborhood relation between the generated hypotheses is shown using grey and black arcs representing inconsistent and consistent relations, respectively. Hypotheses for the same region are inconsistent (in N^-). In this example, all of the hypotheses for model 1 are consistent (in N^+) with each other, as are all of the hypotheses for model 2.

B. Selecting and Ordering Hypotheses for Verification

Once the MRF is created, we wish to find the most likely set of hypotheses based on the constraints of the image. Constraint satisfaction is achieved by minimizing the energy function of (3). To avoid exponential search, we must give up on finding the optimal solution. Fortunately, there is an energy minimization procedure called Highest Confidence First (HCF) [25] which is efficient in practice and, though not optimal, finds good (useful) local minima.

After HCF estimation is completed, if $\omega_{R_i,M_j} = ON$, then the match (R_i, M_j) is considered for verification; matches with $\omega_{R_i,M_j} = OFF$ are discarded. To verify a hypothesis, we need enough matches to get a reasonable estimate of the object's pose. For this domain, three, and sometimes two (using center of gravity as an additional constraint), matches between planar regions and planar model faces are sufficient to roughly estimate the pose of the hypothesized object. Thus, we need to extract sets of two or three matches from the set of active matches to form object hypotheses. The constituent matches of each object hypothesis will be consistent if the matches form a 2- or 3-clique in the neighborhood system N^+.

The verification phase must determine which of these hypotheses describe objects that are in the scene. Traditionally, hypotheses are ordered by saliency or size of the image features [4], [7]. These heuristics have proven useful; however, to truly minimize verifications, we want to first verify the most likely hypotheses which are not necessarily the most salient or largest.

We order the hypothesis cliques by their contribution to the posterior energy function (see (3)). By ordering in terms of least energy contribution first, we are checking the most likely hypotheses first. Thus, the ordering of hypotheses for verification is "optimal" with respect to our prior knowledge and observed evidence.

C. Hypothesis-Generation Algorithm Complexity

Each step of our hypothesis-generation algorithm is examined to determine its overall complexity. The number of variables in the MRF is at most mn, where m is the number of model faces and n is the number of image regions. In HCF

search, each MRF variable is modified a constant number of times on average (in our experience and Chou and Brown's [25], this number is close to one). Each modification of a variable requires a heap update, costing $O(\log mn)$ operations, for each of its neighbor variables. Each variable has at most mn neighbor variables. Thus, if each variable is modified a constant number of times, HCF search will take $O((mn)^2 \log mn)$ operations. After HCF search, 3- and 2-cliques in N^+ are extracted from the MRF. The number of 3-cliques in N^+ is $O((mn)^3)$. Sorting the list of cliques is $O((mn)^3 \log mn)$. Thus, the overall complexity of our algorithm is $O((mn)^3 \log mn)$. In Section VI we demonstrate that the actual performance of our algorithm is much better than the worst case complexity.

V. Robust Pose Refinement and Verification

Given a hypothesized set of matches, we must determine where the object is (*localization*) and whether it is really present in the image or not (*verification*). To succeed, verification depends on an accurate pose estimate. If the estimate is poor, the hypothesis must be rejected. Unfortunately, even slight location inaccuracies[3] can cause rejection of a hypothesis. In many cases, a small refinement of the pose may be the difference between throwing away a valid hypothesis and recognizing the object.

Several factors exacerbate the localization problem. We may not have enough constraints from our matches to accurately determine the object's location. Inaccuracies in the region data due to noise and partial occlusion will lead to errors in location estimates, as will inaccuracies in the CAD models. Using only a small number of primitive matches, we are able to get rough pose estimates.[4] The solution adopted here and by others [6], [8], [10], [14] is to use this location estimate as a starting point for a local search for the true pose. In this section, we describe our approach for robust pose refinement, 3D template matching (3DTM), and our verification metric.

A. 3D Template Matching

Typically, verification is performed by rendering an image of the object and computing some measure of match between the rendering and the image [4], [7]. With range images, a range map of the object would be rendered and compared with the range image data. We can efficiently approximate this comparison using a 3D template consisting of a set of points sampled from the surface of the model. Given a set of points on the objects surface, it is much more efficient to transform the object points into world coordinates and compare them with range data than to render surfaces using ray-tracing techniques. This template can be used for pose refinement as well as verification. Our constraint on the templates is that visible points on the model surface match range data points in the image. The template of a rigid model is parameterized by six rigid-body transformation parameters. An energy function is specified over the parameters which relates how closely the model matches the image data. Then, we find the best model parameters by minimizing the energy function.

To build the model representation required by 3DTM, the model is sampled over each surface by projecting a uniformly spaced grid of points onto the surface. The sampling is uniform in the sense that no two points are within the spacing width of each other. To compute the visibility of the point, each point is annotated by the list of faces to which it belongs. The viewing sphere is divided into 80 cells. The list of faces visible within each cell forms an aspect.[5] Given the current viewing direction, the appropriate aspect can be determined and, hence, the visibility of individual points. The list of aspects of the model are included in the model's 3DTM description. Thus, the model's 3DTM description consists simply of the list of model points and the aspect information.

B. Robust Estimation for Pose Refinement

Typically, algorithms for pose estimation and refinement compute the model parameters which minimize the squared error of model point to image point matches. Assuming a distribution $P(z) \propto e^{-\rho(z)}$ of independent errors, z, over the model points with respect to the corresponding image points, we can find the MAP estimate of the model parameters q by minimizing the energy function

$$E(q) = \sum_i \rho(z_i(q)) \tag{6}$$

where z_i is the error of the i^{th} point in the model. Thus, least-squared error (i.e., $\rho(z) = z^2$) is equivalent to MAP estimation of model parameters assuming a Gaussian distribution of errors. To minimize E, we can use the gradient-descent update rule:

$$\Delta q \propto -\frac{\partial E}{\partial q} = \sum_{i=1}^{n} \psi(z_i) \frac{\partial z_i}{\partial q} \tag{7}$$

where

$$\psi(z_i) = \frac{d\rho(z)}{dz}.$$

When viewing objects, it is often true that parts of the object surfaces are occluded. Occlusion can be due to self-occlusion, other objects in the scene, or sensor shadows (visible portions of the scene which don't receive light from the light-striper in the light-stripe range finders). Occluded points on an object will produce errors much larger than the visible points of the object because they violate the assumption that the data point actually corresponds to an object point. Data points with errors that are larger than predicted by the prior model are referred to as outliers. Least-squares estimates are very sensitive to outliers since the effect of each error is proportional to its magnitude. Thus, the points which have the most effect on a least-squares solution would be the outliers. If

[3] In our experiments, alignment errors of a few degrees rotation and a few millimeters translation are sufficient to cause a hypothesis to be rejected (see the third example in Section VI.B.).

[4] The inaccuracy of pose estimates from a small set of correspondences is well documented [6], [8], [10], [14], [26].

[5] In this work, an "aspect" of a model denotes a set of viewing directions and associated visible faces.

outliers are likely, a least-squared-error estimation procedure is not desirable.

Instead, we would like an estimation procedure which eliminates or reduces the effect of the true outliers. This is achieved by computing a MAP estimate using an error distribution, $P(z)$, where large errors are (relatively) more likely than in a normal distribution. This is the approach taken by Haralick et al. [20]. In this work, we assume a *Lorentzian* error distribution [27]

$$\rho\left(\frac{z}{\sigma}\right) = -\log P\left(\frac{z}{\sigma}\right) = \log\left(1+\frac{1}{2}\left(\frac{z}{\sigma}\right)^2\right). \tag{8}$$

The Lorentzian is similar in shape to the Gaussian distribution, except that large errors are assumed to occur with a higher relative probability than in the Gaussian error model. Fig. 4 compares the (unnormalized) Gaussian distribution with the Lorentzian distribution. The important graph is Fig. 4(b) which shows the weighting, $w(z)=\frac{\psi(z)}{z}$, relative to magnitude of the error vectors under the Gaussian and Lorentzian distributions. The Lorentzian gives much lower weight to the true outliers thus improving parameter estimation. There are other effective functions ρ(), such as the Tukey Biweight or Huber function [20], which can be used; however, the Lorentzian gave us the best results (see Section VI.B.).

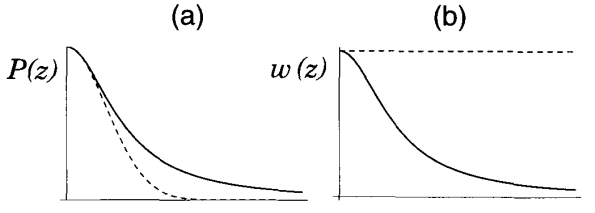


Fig. 4. Comparison of the (a) probability distributions and (b) weight functions of a Gaussian (dashed lines) and Lorentzian (solid lines).

We define $q = (q_t, q_\theta)^T$ to be the the vector of model parameters where $q_t = (q_x, q_y, q_z)^T$ is the vector of translation parameters and $q_\theta = (q_s, q_w, q_v, q_w)^T$ is the vector of rotation parameters using the quaternion representation [5]. We can find the MAP estimate of q, where $P(q) = \prod_i P$

$$\left(\frac{z_i(q)}{\sigma}\right),$$

by minimizing the energy function

$$E(q)=\sum_{i\in V(q)} \rho\left(\frac{z_i(q)}{\sigma}\right) \tag{9}$$

where $V(q)$ is the set of visible model points for the model parameters q, $z_i(q)$ is the error of the i^{th} model point, and σ is the normalizing factor for the Lorentzian function (σ is set to the approximate standard deviation of our sensor's error, here 2 millimeters). The aspect information (see Section V.A.) in the 3DTM model defines an efficient approximation of $V(q)$.

We define the error, $z_i(q)$, to be the distance between the model point and the data point nearest the model point

$$z_i(q)=\min_{\vec{a}\in D}\left\|\vec{x}_i(q)-\vec{a}\right\| \tag{10}$$

where D is the set of three-dimensional data points in the image, and $\vec{x}_i(q)$ is the world coordinate of the i^{th} model point transformed using the model parameters q. The calculation of the nearest data point $\vec{a}$ is optimized by using a k-dimensional nearest-neighbor search [28]. This search requires a k-d tree to be built once per image which takes $O(|D|\log|D|)$ time. Using the k-d tree, finding the nearest neighbor takes expected time proportional to $\log|D|$.

With the definition of the energy E (9), we are able to apply any of a number of minimization procedures. Haralick et al. [20] use weighted least squares to solve for the pose. In our experience, weighted least squares is only appropriate when the correspondences are fixed as in [20]. When correspondences are dynamic (as in our approach where correspondence is a function of q), the energy function being minimized can change drastically between large changes in q. When using WLS, the weights and correspondences are held constant during each minimization iteration. Thus, the function being minimized is different from the objective function, and the computed step may take the pose into a different minima.

To ensure that we are minimizing the desired energy function, we use a line search in the gradient direction (7), recomputing the correspondences at each function evaluation. The search alternates between the translation parameters and the rotation parameters. This is necessary because the differences in sensitivity of the parameters create a scale problem. Simply scaling the gradient results in the energy being minimized in terms of the most sensitive parameters only. Knowing the required accuracy of the pose estimates, we can specify a minimum step size in terms of each parameter (we use 1 millimeter in translation and 1° in rotation). The search ends when a minimum-size step in the gradient direction no longer improves the estimate. In order to eliminate discontinuities in E caused by aspect changes, the current aspect is computed before each line search and is used to determine point visibility throughout the line search; thus, the energy function is kept smooth throughout the line search.

A pseudo-code description of our complete 3DTM pose refinement algorithm is presented below:

```
INPUT: image points D, 3DTM model, initial pose q
OUTPUT: final pose q
ALGORITHM:
create k-d tree from D
REPEAT
   compute the set of visible model points: V(q)
   E0 = E(q) ;; current energy
   dq = -dE/dqt ;; translation gradient
   Δq = (Δqt, Δqθ) = (dq/|dq| · 1mm, 0) ;; step size
```

WHILE $(E(q + \Delta q) < E(q))q = q + \Delta q$

$\vec{dq} = -\dfrac{\partial E}{\partial q\theta}$;; rotational gradient

$\Delta q = (\Delta q_t, \Delta q_\theta) = \left(0, \dfrac{\vec{dq}}{|\vec{dq}|} \cdot 1°\right)$;; step size

WHILE $(E(q + \Delta q) < E(q))q = q + \Delta q$

UNTIL $E(q) = E_0$;; until no improvement is made.

In our experience, our method has a reasonable convergence time and produces accurate results.

C. Verification

The hypothesis-generation phase produces a sorted list of object hypotheses. Each hypothesis is first localized then verified. The verification procedure must determine which of these hypotheses describe objects in the scene.

We use a verification metric similar to a matching metric proposed by Breuel [29] who uses first- and second-order statistics on the matches between image and object features. Each model point is considered *matched* if it is within a distance $2\sigma = 4$ millimeters of an image point. The first-order statistic, α, is the percentage of visible model points that *match* an image point

$$\alpha = \frac{|M(q)|}{|V(q)|}$$

where $V(q)$ is the set of model points visible in pose q, and $M(q)$ is the set of *matched* points from $V(q)$. The second-order statistic, β, is the percentage of neighbor model points that match an image point

$$\beta = \frac{\Sigma_i |M(q) \cap N_i(q)|}{\Sigma_i |N_i(q)|}$$

where $N_i(q)$ is the set of point i's neighbor points which are visible in pose q. β measures the local consistency of the matching points of the model and penalizes random scattered matches compared to locally coherent matches.

To further reduce the chances of false positives, we compute a negative evidence metric γ

$$\gamma = \frac{|Occ(q)|}{|V(q)|}$$

where $Occ(q)$ is the set of model points which occlude range data points in the image. To eliminate cases where the hypothesis occludes range data, we must ensure that γ is below a certain level. For the results described here, the hypothesis was accepted if $\alpha \geq 0.5$, $\beta \geq 0.4$, and $\gamma \leq 0.175$.

The verification procedure takes the ordered list of hypotheses and performs the following set of steps on each element of the list. The pose of the hypothesized object is estimated using the available matches. The pose estimate is refined using 3DTM, and the verification metrics are computed. If the hypothesis is accepted, all remaining hypotheses which have regions in common with the recognized object are removed from the list.

VI. Experiments and Results

We describe experiments that evaluate our sensor model, localization method, and the complete recognition system. Experiments were conducted using the model database of eight polyhedral objects shown in Fig. 5.

Fig. 5. Eight objects used for experiments (from left to right, top to bottom): stapler, holed cube, castle, stick, note dispenser, pencil box, Rolodex, and tape dispenser.

A. Constraint Accuracy using Sensor Models

Since we are forced to make many approximations when modeling our range finder, it is important to determine whether the constraints generated by our sensor model are accurate enough to be useful. To measure the accuracy of our sensor model, we would like to compare the constraints generated by sensor modeling with constraints generated from real images of the object.

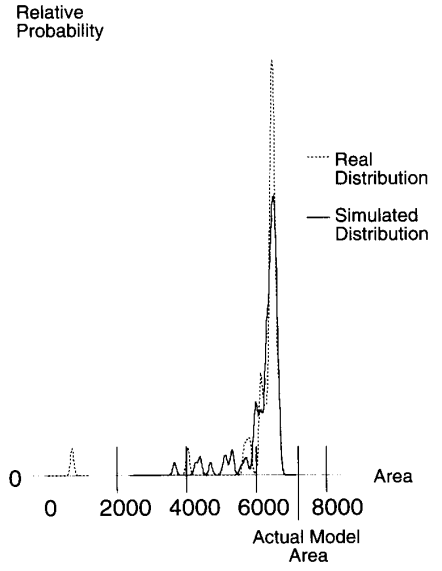


Fig. 6. A comparison of distributions computed from real images and our sensor model.

In a time-consuming process, we manually acquired real samples of a surface to do the comparison. The object (pencil box) was placed at the center of a rotary table in view of our range finder. The table was rotated 3° between each image. Forty images of the object were taken and planar regions were extracted from each image. We selected one surface of the object for comparison. For each image, we manually labeled all planar regions belonging to the selected model surface. The distribution of the area of these regions was computed. In Fig. 6, the distribution acquired from real images is compared

with the corresponding distribution computed by sensor modeling. As this figure shows, the distribution generated by our sensor model is quite similar to the distribution of the real data. The real distribution is slightly narrower and thus has a higher peak. The simulated distribution appears to be a good approximation. The accuracy of the constraints may not be this good for every model surface; however, we believe that the constraints generated by sensor modeling are nonetheless useful for recognition with real images and an improvement over constraints used in previous work.

B. Localization Results

Our recognition strategy is based on the hypothesize-and-verify paradigm. To be successful, the verification decision must be reliable. This can only be achieved with accurate location estimates for the hypothesized objects. Thus, our pose refinement algorithm must return reliable estimates, given slightly inaccurate initial estimates.

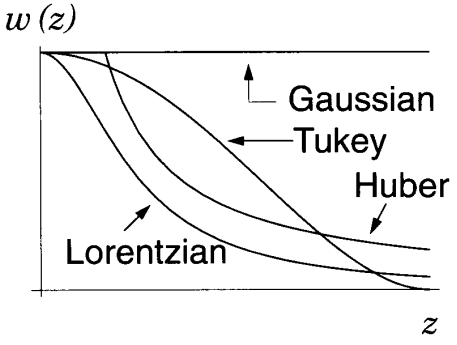


Fig. 7. Comparison of weight functions.

TABLE I.
RESULTS OF LOCALIZATION PERFORMANCE COMPARISON

ρ( )	1: Rolodex		2: Tape Dispenser		3: Stapler	
	%	Cost	%	Cost	%	Cost
Lorentzian	97	71	99	78	85	68
Tukey	97	74	99	86	70	77
Huber	60	68	96	79	45	67
Gaussian	0	150	0	107	0	68

We tested the convergence ability of our algorithm for a variety of objects using real images containing significant occlusion and clutter. For each test, we randomly perturbed the starting location of the object within 2 centimeters and 20° of the known location. For each experiment, 400 tests were executed. To compare the relative performance of the various functions ρ( )in our 3DTM algorithm, we repeated each experiment using the Tukey Biweight [20], Huber function [20], Gaussian, and Lorentzian. The Tukey and Huber functions were implemented as specified in [20] using their recommended constants with respect to σ = 2 millimeters. The corresponding weighting functions $w(z)$ for each $\rho(z)$ are shown in Fig. 7. Experiments were performed on three different images. Table I lists two numbers for each experiment. The first is the percentage of times that the solution converged to an acceptable pose estimate (i.e., the pose satisfies the verification thresholds of Section V.C.). The second number listed is the average number of times the energy function of (9) is computed during 3DTM localization. This number measures the average cost of refining the pose.

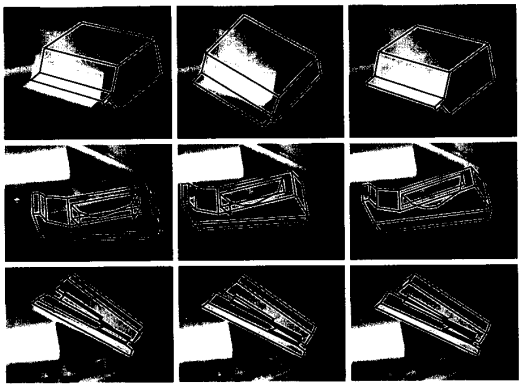


Fig. 8. Examples from the localization experiments: initial model location (left), final model location using the Gaussian distribution (middle), and final model location using Lorentzian (right).

Example runs from each of the three experiments are shown in Fig. 8. These examples compare the resulting location estimates of our method using Gaussian distributions and Lorentzian distributions. As discussed in Section V.B. and as you can see in the first two examples in Fig. 8, the least-squares solutions are noticeably occlusion sensitive while the results using the Lorentzian distribution are relatively insensitive to occlusion. The results shown in the third example (stapler) are difficult to visually distinguish; however, the Gaussian estimate contains a small translation and rotation error that is sufficient to cause the verification algorithm to reject it. Even when initialized to the correct location, Gaussian estimation will diverge from the correct location because of a few outliers.

The results of the comparison show us that the Lorentzian and Tukey Biweight functions have quite good performance. The Huber function performs reasonably well, and the Gaussian failed to produce a useful result. The Lorentzian does consistently better than the Tukey in terms of number of energy function evaluations by about 10%. The results also reinforce the fact that local minima are a problem. However, this is not a significant problem considering that our initial location errors from correspondences will usually be much smaller than the 2 centimeter and 20° initial offsets used in the tests.

C. Recognition Experiments

To evaluate the performance of our object-recognition system, we are interested in the number of verifications performed, since verifications are the most expensive part of our

algorithm. In this section, we present the results of the experimental evaluation of our algorithm, first using synthetic images then using real images.

C.1. Synthetic Images

To determine the peak performance of our hypothesis-generation algorithm, we performed some tests using synthetic range images. We generated synthetic images—using our sensor model—of our objects. Images were generated of each object at 20 distinct viewpoints—giving a total of 160 images. Each image was then processed by our recognition algorithm. The performance was very encouraging. The correct object was recognized in 148 of the images for a recognition rate of 92.5%. The average number of verifications performed was 2.04, while the average number of hypotheses selected for verification was 404. The correct object hypothesis was almost always ranked first or second in the sorted list of object hypotheses. These results indicate that the hypothesis generation algorithm is doing a very good job of selecting hypotheses as well as ordering the hypotheses for verification.

Some failures were due to singular views where only one surface of the object was visible; the current system is not able to handle this situation (the feature set would have to be extended). One false positive occurred when a castle was mistaken for a holed cube. These two objects are very similar geometrically, and the negative evidence metric γ was not sufficient enough to disregard the hypothesis. When the correct hypothesis was not ranked first, the problem was most often the result of object symmetry. The binary relational constraints used by our system are unable to distinguish mirror image views of a symmetric object.

The results on synthetic images give us an indication of the potential performance of our hypothesis-generation techniques. The sensor model used to compile the prior constraints is a perfect model of the simulated imaging process. This leads us to believe that as the accuracy of our constraints improve, the performance of our hypothesis-generation algorithm on real images may approach the performance achieved on synthetic images.

C.2. Real Images

We performed a similar experiment using real images of our model database objects. Each object was placed on a rotary table and 10 range images were taken, for a total of 80 images. Each image was processed by the recognition algorithm. Out of 80 images, 62 of the objects were correctly identified for a recognition percentage of 77.5%. The average number of verifications performed was 29, while the average number of hypotheses selected for verification was 247. The drop in performance between synthetic and real experiments is to be expected; as shown in Section VI.A., the difference between the real and synthetic constraints may be extremely small, but these differences will result in some hypotheses being pruned and the ordering of hypotheses to be wrong. However, the ordering is quite reasonable; as our results show, only about 10 percent of the generated hypotheses are tested on average.

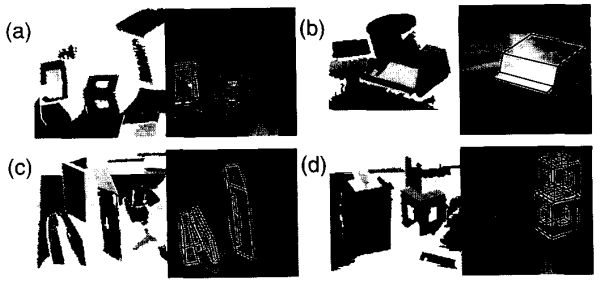
Our algorithm has been tested on many images containing multiple objects, unknown objects, and partial occlusion. We present four examples in Fig. 9 to show successes as well as failures and limitations of our system. The left image of each pair is an image of the extracted planar regions, and the right image is the intensity image of the scene with the wire-frame model of recognized objects overlaid. The examples shown in Fig. 9 depict several important points. The first image (Fig. 9(a)) shows correct recognition of the tape dispenser despite poor segmentation. The image in Fig. 9(b) demonstrates both the successes and limitations of our system with respect to recognizing partially occluded objects. The partially occluded Rolodex is found because unoccluded parts of the object were observable and consistent with the constraints. The partially occluded stapler is not recognized because not enough surfaces are unoccluded, thus causing the correct matches to violate the first-order constraints. A similar problem exists in the third image with respect to the partially occluded pencil box. In the fourth image (Fig. 9(d)), the partially occluded holed cube is correctly recognized, while the castle above it is incorrectly identified as a holed cube. Because the two objects are shaped similarly the verification metrics do not distinguish the two well. This error is also due to inaccuracies in the constraints generated by sensor modeling which rank the holed cube as a better hypothesis than the castle. Another error in this image is that the pencil box was not recognized. This error is the result of the need for thresholds α, β, and γ in our verification decision. The pencil box was actually correctly located in the image, but the β and γ metrics were beyond acceptable limits. It is also important to notice that none of the unknown objects in the images were incorrectly identified as objects from our model database.

The table in Fig. 9 lists important statistics from the execution of the algorithm. The first number in the table is the number of planar regions found in the image. The second number is the number of possible correspondences. The third number is the number of correspondences that satisfy the first-order attribute constraints $\left(P\left(\vec{f}_{R_i} \middle| M_j\right) > 0\right)$. Fourth is the number of 2- and 3-cliques of hypotheses that are active after probabilistic constraint satisfaction. The final number lists the number of verifications actually performed by the algorithm—this is the dominant factor in the recognition time of the algorithm. The first four entries in the table correspond to the four example images. The last two entries in the Fig. 9 table list the average of the hypothesis-generation statistics for the synthetic and real image test sets. From the tests on real images, the number of required verifications was of the order 100 compared to the cost of the basic alignment algorithm which would be of the order 10^6.

For a more realistic comparison of our algorithm's performance, we implemented a simple constrained Alignment/ Interpretation Tree search algorithm [8] (searching the tree to a depth of three). From tests on several images, the IT search would test approximately half of all generated hypotheses on average. For the synthetic tests, using IT search would result in testing 200 hypotheses on average as compared to two by our

algorithm. For the real image tests, the comparison would be about 120 to 30. These figures reflect the performance of the IT search using the constraints generated by sensor modeling. Using constraints from the CAD models alone (i.e., without a sensor model), the number of tests performed increased by a factor of more than 10—making the algorithm impractical to run. This shows that there is a great speedup from using accurate, tight constraints.

Our recognition algorithm sometimes failed to recognize a visible object (false negative). Some failures were due to singular viewpoints (i.e., insufficient visible surfaces for localization). Some objects were correctly hypothesized and located, but due to significant occlusion, their verification metrics were too low to reliably accept them. Other false negatives were due to constraint inaccuracies causing correct hypotheses to be eliminated. For example, lighting, sensor shadows, and changes in depth of object were not modeled. These omissions have some detrimental effects on constraint accuracy which result in recognition errors.



	Regions	Possible Matches	Thresh. Matches	Sel. Cliq.	Verif. Cliq.
a	21	3507	291	29	13
b	16	2672	234	91	81
c	35	5845	421	138	89
d	25	4175	318	171	53
Synth	5.4	902	123	408	2.02
Real	9.9	1653	222	247	29

Fig. 9. Example recognition results: region image (left), wire-frame overlay of recognized models (right).

Our algorithm is capable of recognizing partially occluded objects assuming that a sufficient number of object features are detected. Due to the relative simplicity of our current objects, this assumption is often violated when the objects are partially occluded. One solution is to use features that are more dense than simple surfaces (e.g., edges).

Robust localization had a significant impact on the reliability of the verification decision. Without robust localization, it was clear that the thresholds on α, β, and γ (see Section V.C.) for accepting hypotheses (to avoid false negatives) would need to be so relaxed that false positives would become very common. Thus, we feel pose refinement is necessary before testing the hypothesis.

In this work, we used an OGIS light-striping range finder which provided us with 240×256 images of (x, y, z) coordinates at approximately 0.1 millimeters resolution. Our prototype recognition program was implemented in Common Lisp on a Sun 4 workstation. The approximate execution time was two minutes for the feature extraction, two to five minutes for building the MRF, and two to five seconds to perform HCF and order the hypotheses for verification. The prototype of 3DTM takes approximately one to two minutes to perform localization (much of which is image input and output). We have not concentrated on making the implementation efficient. Instead, we have opted for a fast development environment in which to test our ideas.

VII. Conclusions

We have presented techniques for object recognition that exploit prior knowledge to make recognition efficient and accurate. The techniques are built upon a probabilistic framework that provides principled approaches to the problems encountered in model-based vision. We have implemented and tested these ideas in a system that recognizes polyhedral objects in range images. We offer a few conclusions from our experience in building this system.

Sensor models are necessary for efficient model-based vision. If the system's constraints only account for model geometry, unnecessary search will inevitably be performed. Image feature formation depends on the sensor (quantization, digitization, and noise), the scene environment (lighting and background), the feature-extraction algorithm (biases and thresholds), as well as the object (geometry and photometric properties)—a comprehensive model should account for these effects. Though ray-tracing is expensive, we feel that it is the most realistic approach available to capture all of the interactions of the various processes. A significant benefit is that we do not need analytic models (as in [10], [16]) of the performance of feature extraction—it can be considered to be a black box.

Statistical representations of constraints generated from many sample views automatically solves the problem of feature and threshold selection for recognition. Thresholds are difficult to manually choose and inevitably lead to incorrect or missed matches. Our statistical representation of the mapping between object and image features provides a general framework for determining the valid range of the observed feature's attributes. The distribution of an attribute is represented explicitly, and, thus, the need to specify some approximating distribution or thresholds over allowable values of the invariant is avoided. Statistical constraints also implicitly represent the relative utility of object features for recognition; features that are not stably detected over a wide range of viewpoints will have low prior probabilities and, hence, little effect in the matching process. Thus, the statistics automatically select the most useful features for recognition.

There are benefits to be gained from both interpretation-tree-like correspondence search and constraint satisfaction/relaxation searches. Interpretation trees offer robustness while constraint satisfaction offers efficiency. We use a constraint satisfaction style search to quickly eliminate/prune

many correspondences from consideration; following that, a more deliberate interpretation tree style search is used to test the remaining hypotheses. Our approach should not be confused with the idea of relaxation labeling used by Bhanu [11] or the MAP model matching technique of Wells [14]. Our constraint satisfaction search does not search for *the* most likely matching; it searches for sets of matches that are likely enough to consider verifying. The difference is important since it is possible for incorrect hypotheses to have higher probabilities than correct hypotheses due to random configurations of features.

Most-likely-first search is a principled way of optimizing correspondence searches for efficient recognition. The execution time of correspondence searches is often dependent on how fortunate the system is to start the search with correct correspondences. Improvements have been obtained by using heuristics to select initial matches, thus removing the need for arbitrary selection. Others rely on grouping operations to further constrain the matching problem—effectively reducing the complexity of the problem. Our approach prunes unlikely hypotheses while considering all constraints on the hypotheses simultaneously. The filtered hypotheses are ordered by their relative likelihoods for verification. By doing so, the number of unnecessary verifications is reduced significantly—resulting in a significant efficiency gain as demonstrated by our recognition results.

Pose refinement is a requirement for reliable verification. If the location estimate based on a minimal set of correspondences is used, we have seen that verification will be very unreliable, and the recognition algorithm will be practically useless because of the high false positive rate. To be reliable, the decision procedure requires a more accurate estimate of the object's pose than is attainable from a minimal set of matches.

For pose refinement, we have observed that least-squares techniques fail when a significant portion of object features are occluded or undetected. Our solution utilizes an estimation technique which is relatively insensitive to outliers. We find that the key to a solution is to use an error distribution, as in [20], that has a higher relative probability for large errors compared to a Gaussian error distribution. Using our technique, the pose estimate converges to the true object position with great reliability in spite of partial occlusion or missing features.

In summary, we have presented new techniques for object recognition that exploit prior knowledge to make the search process efficient and the results accurate. Sensor modeling is used to generate accurate statistical constraints for object recognition that reflect the effect of the sensing process on the appearance of the object. Our hypothesis-generation algorithm uses probabilistic constraint satisfaction to apply the statistical constraints to select likely hypotheses and orders the selected hypotheses by most likely first. To ensure reliable verification, we developed a pose refinement algorithm that is robust to partial occlusion and noise. When combined into a complete system, these techniques make progress towards improving recognition efficiency and accuracy. In experiments on synthetic images generated by our sensor model, our approach has demonstrated nearly optimal performance in terms of selecting and ordering the hypotheses for verification. In experiments on real images, our approach has demonstrated the ability to greatly limit the number of verifications. The techniques presented here represent a promising new paradigm for probabilistic object recognition based on accurate statistical constraints.

We are currently working on improving and extending these techniques in the following areas: improving the accuracy of our constraints by improving our sensor models and using real images; applying our techniques to other sensors; and increasing verification reliability. The techniques presented here can be applied to recognize curved objects. Statistics over the appearance of segmented curved surfaces can be computed just as in the planar surface case and the hypothesis-generation algorithm would remain the same. Our pose refinement algorithm has already been applied to localization of curved objects.

ACKNOWLEDGMENTS

This paper is an enhanced version of work presented at the Second CAD-Based Vision Workshop [1]. This research was sponsored in part by the U.S. Advanced Research Projects Agency under the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U. S. Air Force, Wright-Patterson AFB, OH 45433-6543 under Contract F33615-90-C-1465, ARPA Order No. 7597; in part by the Advanced Research Projects Agency under the Department of the Army, Army Research Office under grant number DAAH04-94-G-0006; and in part by the Department of the Navy, Office of Naval Research under grant number N00014-93-1-1220. The first author was supported by a National Science Foundation Graduate Fellowship. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Advanced Research Projects Agency, the U.S. Government, or the National Science Foundation.

The authors would like to thank Martial Hebert for the use of his segmentation code, Harry Shum for the use of his rotary table, Kathryn Porsche, Sing Bing Kang, Kevin Lynch, Fredric Solomon, and the anonymous reviewers for helping to improve this paper.

REFERENCES

[1] M. D. Wheeler and K. Ikeuchi, "Sensor modeling, probabilistic hypothesis generation, and robust localization for object recognition," *Second CAD-Based Vision Workshop,* pp. 46-53, 1994.

[2] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int'l J. Computer Vision*, vol. 2, no.1, pp. 322-331, 1987.

[3] W.E.L. Grimson and T. Lozano-Perez, "Localizing overlapping parts by searching the interpretative tree," *IEEE Trans. Patt. Anal. Machine Intell.,* vol. 9, no. 4, pp. 469-482, 1987.

[4] R.C. Bolles and P. Huraud, "3DPO: A three-dimensional part orientation system," *Intl. J. Robotics Research,* vol. 5, no. 3, pp. 3-26, 1986.

[5] O. Faugeras and M. Hebert, "The representation, recognition, and locating of 3D objects," *Intl. J. Robot. Research,* vol. 5, no. 3, pp. 27-52, 1986.

[6] D.G. Lowe, *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, 1985.
[7] P.J. Flynn and A.K. Jain, "Bonsai: 3D object recognition using constrained search," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 13, no. 10, pp. 1066-1075, 1991.
[8] D.P. Huttenlocher and S. Ullman, "Recognizing solid objects by alignment with an image," *Int'l J. Computer Vision*, vol. 5, no. 2, pp. 195-212, 1990.
[9] K. Ikeuchi, "Generating an interpretation tree from a CAD model for 3-d object recognition in bin-picking tasks," *Int'l J. Computer Vision*, vol. 1, no. 2, pp. 145-165, 1987.
[10] O.I. Camps, R.M. Haralick, and L.G. Shapiro, "PREMIO: An overview," in *IEEE Workshop on Directions in Automated CAD-Based Vision*, pp. 11-21, 1991.
[11] B. Bhanu, "Representation and shape matching of 3D objects," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 6, no. 3, pp. 340-351, 1984.
[12] R.M. Bolle, A. Califano, and R. Kjeldsen, "A complete and extendable approach to visual recognition," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 14, no. 5, 1992.
[13] P. Cooper, *Parallel Object Recognition from Structure (The Tinkertoy Project)*. PhD thesis, Dept. of Computer Science, Univ. of Rochester, 1989.
[14] W.M. Wells, *Statistical Object Recognition*, PhD thesis, Massachusetts Inst. of Technology, 1992.
[15] J. Ben-Arie, "The probabilistic peaking effect of viewed angles and distances with application to 3D object recognition," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 12, no. 8, pp. 760-774, 1990.
[16] K. Ikeuchi and T. Kanade, "Automatic generation of object recognition programs," *Proc. IEEE Special Issue Computer Vision*, vol 76, pp. 1016-1035, 1988.
[17] D. Lowe, "Fitting parameterized three-dimensional models to images," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 13, no. 5, pp. 441-450, 1991.
[18] D. Kriegman and J. Ponce, "On recognizing and positioning curved 3D objects from image contours," in *DARPA IUS Workshop*, pp. 461-470, 1989.
[19] R.M. Bolle and D.B. Cooper, "On optimally combining pieces of information, with application to estimating 3D complex-object position from range data," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 8, no. 5, 1986.
[20] R.M. Haralick, H. Joo, C.-N. Lee, X. Zhuang, V.G. Vaidya, and M.B. Kim, "Pose estimating from corresponding point data," *IEEE Trans. Syst. Man Cybern.*, vol. 19, no. 6, pp. 1426-1446, 1989.
[21] P.J. Besl and N.D. McKay, "A method for registration of 3D shapes," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 14, no. 2, pp. 239-256, 1992.
[22] K. Ikeuchi and J.-C. Robert, "Modeling sensor detectability with {VANTAGE} geometric/sensor modeler," *IEEE Trans. Robotics Automat.*, vol. 7, no. 6, pp. 771-784, 1991.
[23] Y. Fujiwara, S. Nayar, and K. Ikeuchi, "Appearance simulator for computer vision research," Tech. Report CMU-RI-TR-91-16, Carnegie Mellon Univ., 1991.
[24] J.B. Burns and E.M. Riseman, "Matching complex images to multiple 3D objects using view description networks," in *Proc. CVPR*, pp. 328-334, 1992.
[25] P.B. Chou and C.M. Brown, "The theory and practice of Bayesian image labeling," *Int'l J. Computer Vision*, vol. 4, pp. 185-210, 1990.
[26] W.E.L. Grimson, D.P. Huttenlocher, and D.W. Jacobs, "Affine matching with bounded sensor error: A study of geometric hashing and alignment," Tech. Report 1250, Massachusetts Inst. of Technology, 1991.
[27] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge Univ. Press, 1991.
[28] J. Friedman, J. Bentley, and R. Finkel, "An algorithm for finding best matches in logarithmic expected time." *ACM Trans. on Math. Software*, vol. 3, no. 3, pp. 209-226, 1977.
[29] T.M. Breuel, "Fast recognition using adaptive subdivisions of transformation space," in *Proc. CVPR*, pp. 445-451, 1992.

Mark D. Wheeler received the BSE degree (summa cum laude) in computer engineering from Tulane University in 1989 and the MS degree in computer science from Carnegie Mellon University in 1993, where he is currently a PhD candidate in the School of Computer Science. His research interests include computer vision, object recognition, machine learning, neural networks, and robotics.

Katsushi Ikeuchi received his B Eng degree in Mechanical Engineering from Kyoto University, Kyoto, Japan, in 1973 and a PhD degree in Information Engineering from the University of Tokyo, Tokyo, Japan, in 1978. He is a Principal Research Scientist in the Computer Science Department and the Robotics Institute, School of Computer Science, Carnegie Mellon University. His research interests include image understanding and the use of specular reflections to recover surface orientations.