Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems Raleigh, NC July 7-10, 1992

Object Recognition using Prediction and Probabilistic Matching

Octavia I. Camps,
email: camps@whale.ece.psu.edu
Dept. of Electrical and Computer Engineering
The Pennsylvania State University
University Park, PA 16802

Linda G. Shapiro,
email: shapiro@.cs.washington.edu
Dept. of Computer Science, FR-35
University of Washington
Seattle, WA 98195

Robert M. Haralick
email: haralick@ee.washington.edu
Electrical Engineering Department, FT-10
University of Washington
Seattle, WA 98195

Abstract

PREMIO is a CAD-based object recognition and localization system that uses CAD models of 3D objects and knowledge of lighting and sensors to predict the detectability of features in various views of the object. The predictions that PREMIO produces are powerful new tools in recognizing and determining the pose of a 3D object. In order to take advantage of these tools, we have developed a new matching algorithm: an iterative-deepening-A^* search that explicitly takes advantage of the predictions to guide the search and reduce the search space. The purpose of this paper is to describe the matching algorithm and illustrative results.

I Introduction

Most feature-based matching schemes assume that all the features that are potentially visible in a view of an object will appear with equal probability. The resultant matching algorithms have to allow for "errors" without really understanding what the errors mean. **PREMIO** [2] is an object recognition/localization system that attempts to model some of the physical processes that can cause these "errors". It uses CAD models of 3D objects and knowledge of lighting and sensors to predict the detectability of features in various views of the object. From these predictions, PREMIO calculates probabilities for each feature of being detected as a whole, being missed entirely, or breaking into pieces and conditional probabilities of the detection of one feature given the detection or nondetection of other features. The predictions that PREMIO produces are powerful new tools in recognizing and determining the pose of a 3D object. In order to take advantage of these tools, we have developed a new matching algorithm: an iterative-deepening-A^* search that explicitly takes advantage of the probabilities to guide the search and prune the tree. The matching algorithm represents a large theoretical effort that is actually independent of the PREMIO system. The algorithm has been implemented as a C program and tested on data specifically generated to fit the abstract paradigm for the probabilistic search. The purpose of this paper is to describe the theory, the algorithm, and illustrative results.

II Relation to Previous Work

The matching algorithm described in this paper can be thought of in two ways, as a relational matching algorithm and as a heuristic search. The theory behind heuristic search is well known [9]. Grimson [3] showed that the number of nodes expanded during a depth first search of an interpretation tree in the presence of spurious data is exponential, due to the combinatorics of the problem. Relational matching has been expressed in several different formalisms. Early papers concentrated on graph or subgraph isomorphisms [15]. This led to many algorithms for discrete relaxation and the introduction of probabilistic relaxation [11]. The exact matching problem was generalized to the consistent labeling problem [6] and to the inexact matching problem [13]. This was extended further to the problem of determining the relational distance between two structural descriptions [14, 12]. Some recent related work includes structural stereopsis using information theory [1]. The present algorithm differs from all of these in its attempt to provide a solid theoretical probabilistic framework for the matching problem that can be used to reduce the search space.

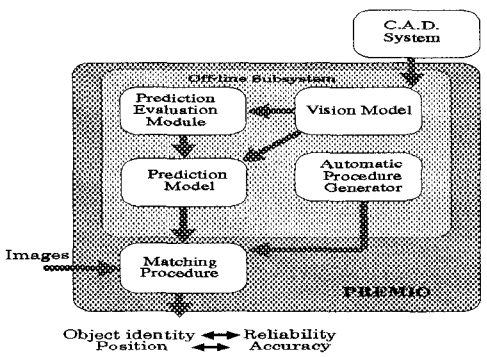

Figure 1: PREMIO: A Model-Based Vision System

III PREMIO: A Model–Based Vision System

PREMIO (**PRE**diction in **M**atching **I**mages to **O**bjects) is a model–based object recognition/localization system. PREMIO uses CAD models of 3D objects and knowledge of surface reflectance properties, light sources, sensors characteristics, and the performance of feature detectors to build a model called the *vision model*. The system is illustrated in Fig. 1. PREMIO's vision model is a more complete model of the world than the ones presented in the literature. It not only describes the object, light sources and camera geometries, but it also models their interactions.

The feature predictor of the system uses the vision model to predict and evaluate the features that can be expected to be detected in an image of an object, taken from a given viewpoint and under a given light source and sensor configuration. The output of the prediction module is organized as the prediction model. The automatic procedure generator takes as its input the prediction model and generates

0-7803-0737-2/92$03.00 1992©IEEE

the matching procedure to be used for matching the image features against the object models.

IV Definitions and Notation

Models and images are represented by their features, the relationships among them, and the measurements associated with them. As in the consistent labeling formalism [6], we will call the image features *units* and the model features *labels*. The matching algorithm must determine the correspondences between the units and the labels. Formally, a *model* M is a quadruple $M = (L, R, f_L, g_R)$ where L is the set of model features or labels, R is a set of relational tuples of labels, f_L is the attribute-value mapping that associates a value with each attribute of a label of L, and g_R is the strength mapping that associates a strength with each relational tuple of R. Similarly, an *image* I is a quadruple $I = (U, S, f_U, g_S)$ where U is the set of image features or units, S is a set of relational tuples of units, f_U is the attribute-value mapping associated with U, and g_S is the strength mapping associated with S.

An image is an observation of a particular model. Not all the labels in L participate in the observation, only a subset of labels $L^o \subseteq L$ is actually observed. Furthermore, only the relational tuples of labels representing relationships among labels in L^o can be observed, and only a subset of them, $R^o \subseteq R$, are actually observed. The set U consists of the unrecognized units. Some of the units observed in U come from labels in L^o; others are unrelated and can be thought of as clutter objects. The set S is a set of observed relational tuples of units in U.

The relational matching problem is to find an unknown one-to-one correspondence $h: L \rightarrow U$ between a subset of L and a subset of U, associating some labels of L with some units of U. The mapping h is called the *observation mapping*. Notice that the matching process consists not only of finding the model M, but also of finding the correspondence h, which is the explanation of why the model M is the most likely model.

V Matching by Tree Search

The matching process can be thought of as a state space search through the space of all possible interpretations Σ. The state space Σ is called the *matching space* and it is defined as follows.

Def. V.1 The *matching space*, Σ, is the state space of all possible interpretations, in which each state σ is defined by an observation mapping h_σ with degree of match $k_\sigma = \#\mathrm{Dom}(h_\sigma)$.

The search through the state space Σ can be achieved by doing an ordered search on a interpretation tree $\mathcal{T}$ such as the one shown in Fig. 2. Each node in $\mathcal{T}$ represents a unit and each of its branches represents an assignment of the unit to a label. A search state σ in Σ is represented by a path $\mathcal{P}$ in the tree $\mathcal{T}$. In the rest of the paper, the terms "path" and "partial mapping" will be used interchangeably.

The main difficulty in solving the matching problem by a tree search is the high combinatorics involved in the problem [3]. The number of possible interpretations in the tree grows exponentially with the number of labels and units. The number of interpretations could be reduced by stopping the search before having a complete mapping. The problem of course, is to determine when to stop.

A usual approach towards solving this problem is as follows: a path in the interpretation tree $\mathcal{T}$, $\mathcal{P}$, defines an observation mapping $m_\mathcal{P}$ with an associated cost $C_\mathcal{P} = C(m_\mathcal{P}, M, I)$ that measures the correctness of the mapping;

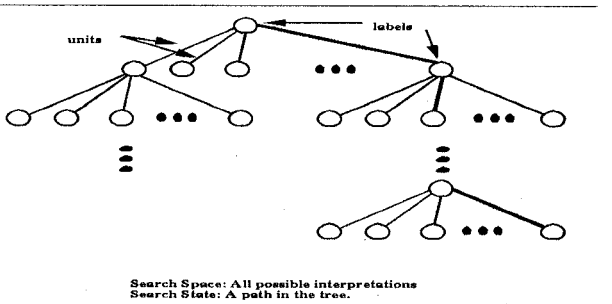

Figure 2: Search tree $\mathcal{T}$.

then, the matching process consists of finding the path $\mathcal{P}^*$ such that its associated observation mapping $m_{\mathcal{P}^*}$ has the least cost.

In this way, the problem of selecting the correct interpretation has been relegated to the problem of defining an adequate cost function such that the interpretation having the least cost is indeed the correct one.

In the following sections we will introduce a theoretical probabilistic framework for the matching problem. The proposed framework allows us to define the cost of a mapping in a rigorous way, with a strong physical meaning. Furthermore, we will show how to find lower bounds of the defined cost, so that it can be used in guiding an heuristic search.

VI Solving the Relational Matching Problem

The relational matching problem is a special case of the pattern complex recognition problem [5]. In the pattern complex recognition paradigm, the relational matching problem can be stated as follows:

> Given a model $M = (L, R, f_L, g_R)$ and an image $I = (U, S, f_U, g_S)$, find the observation mapping (h, H) such that the *a posteriori* probability $P(M, h|I)$ is maximum.

That is, we want to maximize the probability of the model being M and the observation mapping being h, given that the image I is observed. Hence, solving the relational matching problem requires a search procedure that can identify the model M and the mapping h such that the probability $P(M, h|I)$ is maximized. In order to define such a procedure, this probability must be further broken down and related to a cost function to be used in the search.

A *Probability of an Observation Mapping*

An observation mapping h consists of a domain contained in the set of labels L, and of the correspondences of these labels to some units belonging to the set of units U. Let $H \subseteq L$ be the domain of the mapping h. In the following, whenever we want to make explicit the need to consider both the correspondence h and its domain H, we will denote them as the pair (h, H). By the definition of conditional probability,

$$P\Big(M, (h, H)\,\Big|\, I\Big) = P\Big(M, (h, H), I\Big)/P(I)\,. \qquad (1)$$

Since M, and (h, H) do not appear in the denominator, maximizing the conditional probability $P(M, (h, H)|I)$ is equivalent to maximizing $P(M, (h, H), I)$. Assuming that measurements and relationships are conditionally independent given M, U, f_U, and (h, H), and further, that the

relationships are independent of the feature measurements, we have after some algebra manipulation:

$$P\Big(M,(h,H),I\Big) = P_M \cdot P_U \cdot P_S \cdot P_{f_U} \cdot P_{g_S} \ , \qquad (2)$$

where $P_M = P(M)$, $P_U = P(U,(h,H)|M)$, $P_{f_U} = P(f_U|U,M,(h,H))$, $P_S = P(S|U,M,(h,H))$, and $P_{g_S} = P(g_S|U,f_U,M,(h,H))$.

Equation (2) breaks down the joint probability of observing the model M through the mapping h as the image I into five terms. The first term is the prior probability of the model M. The other four terms are such that each one of them can be directly related to one of the four elements that describe the model and the image.

B Relational Matching Cost

In this context it is natural to define the *relational matching cost* of the observation mapping h as the *information* content in the probabilistic event that h is the observation mapping between the model M and the image I:

Def. VI.1 Let $h: L \rightarrow U$, with $\mathrm{Dom}(h) = H$, be an observation mapping. The *relational matching cost* of h, is defined by, $C(M,(h,H),I) = -\ln P(M,(h,H),I)$.

Taking the logarithm on both sides of equation (2), and changing the sign, we have that maximizing $P(M,(h,H),I)$ is equivalent to minimizing:

$$C\Big(M,(h,H),I\Big) = C_M + C_U + C_S + C_{f_U} + C_{g_S} \ , (3)$$

where $C_M = -\ln P_M$, $C_U = -\ln P_U$, $C_{f_U} = -\ln P_{f_U}$, $C_S = -\ln P_S$, and $C_{g_S} = -\ln P_{g_S}$.

Equation (3) shows that the cost C depends on the model, the *label–unit* assignments, the relational structures, and their measurements.

C A Probabilistic Model

To compute the relational matching cost defined in the previous section we need the corresponding probabilities. In this section we present a model to compute these probabilities based on their physical meaning.

Model Cost

The probability $P_M = P(M)$ is the *prior* probability for the model M to be observed, and it is available from the prediction system. The cost C_M is the cost associated with the model being considered, and it penalizes the selection of models whose prior probability of occurring is low.

Label–Unit Assignment Cost

The probability $P_U = P(U,(h,H)|M)$ evaluates the likelihood of the number of labels in H being matched through the mapping h to a subset of the observed units U. Since for the model M, the set L designates the set of possible labels, it is natural for P_U to depend on the difference between the size of the set L and the size of the domain of h, as well as on the difference between the size of the set U and the size of the range of h. This probability should be high for observation maps that assign corresponding labels and units, and lower for those mappings that either miss assignments or assign labels to spurious units. Therefore, it is reasonable to model[1]

[1] This assumption, based on the central-limit theorem, has been verified experimentally.

$$P_U = \frac{1}{\sqrt{2\pi}\sigma_f} e^{-\frac{1}{2}\left(\frac{(\#L+\#U-2\#H)-\mu_f}{\sigma_f}\right)^2} . \qquad (4)$$

Relational Structural Cost

The probability $P_S = P(S|U,M,(h,H))$ evaluates how well the relationships among the labels are preserved by the mapping h. We will take this probability to be dependent on the number of relational tuples that are not preserved by the mapping.

Let Tr be the set of all possible types of relational tuples of labels. We define the *composition* of a relational tuple of labels of order N, $r \in (Tr \times H^N) \subseteq R$, with the mapping h, as a relational tuple of units of the same type as r such that each unit of its feature vector corresponds to a label in r through the mapping h:

Def. VI.2 Given the one-to-one mapping $h: H \rightarrow U$, and the relational tuple of labels $r \in (Tr \times H^N) \subseteq R$, the *composition* of r with h is denoted as $h \circ r$, and is defined as the relational tuple of units given by, $h \circ r = (t,(u_1,u_2,\ldots,u_N))$, where t is the type of tuple r, N is the number of labels participating in tuple r, and $u_i = h(\mathrm{F}_R(r,i))$ for $0 < i \leq N$, where $\mathrm{F}_R(r,i)$ denotes the i^{th} element of the feature vector of r.

The *composition* of the set of relational tuples of labels R with the mapping h is defined as the set of the relational tuples of units resulting from composing each of the relational tuples $r \in (Tr \times H^N) \subseteq R$ with h.

Def. VI.3 Given the set of relational tuples of labels R, and the one-to-one mapping $h: H \rightarrow U$, the *composition* of R with h is denoted as $h \circ R$, and is defined as the set of relational tuples of units given by $h \circ R = \{s = h \circ r \mid r \in (Tr \times H^N) \subseteq R\}$.

The compositions of a relational tuple of units s and of the set of relational tuples of units S with the inverse mapping h^{-1} are denoted by $h^{-1} \circ s$ and $h^{-1} \circ S$, respectively and are defined in a similar way.

We will model the probability P_S to penalize the number of relational tuples not preserved in the match, as well as those relational tuples matched to spurious relational tuples. Thus, it is natural to use the concepts introduced above to model:

$$P_S = \frac{1}{\sqrt{2\pi}\sigma_r} e^{-\frac{1}{2}\left(\frac{\#(R-S\circ h^{-1})+\#(S-R\circ h)-\mu_r}{\sigma_r}\right)^2} . \qquad (5)$$

Therefore, the relational structural cost $C_S = -\ln P_S$ is the part of the cost that accounts for the differences between the set of observed relationships S and the set of relationships of the model R.

Metric Costs

Since the probability $P_{f_U} = P(f_U|U,M,(h,H))$ and the probability $P_{g_S} = P(g_S|U,f_U,M,(h,H))$ are both probabilities of mappings that associate values to elements of a set, their treatment is similar.

The probability P_{f_U} can be expressed as the probability $P(f_U \circ h|U,M,H)$, where $f_U \circ h$ is the composition of f_U with h defined by $(f_U \circ h)(l) = f_U(h(l))$, $l \in H$.

Since f_L is the attribute-value mapping associated with L,

$$P_{f_U} = \frac{1}{\sqrt{2\pi}\sigma_{f_U}} e^{-\frac{1}{2}\left(\frac{\rho(f_U \circ h, f_{L|_H}) - \mu_{f_U}}{\sigma_{f_U}}\right)^2}, \quad (6)$$

where ρ is a suitable metric function, and $f_{L|_H}$ represents the attribute-value mapping f_L restricted to the labels in the domain H. Hence, the farther the measurements of the units are from the measurements of the corresponding labels, the larger the cost term C_{f_U}.

Reasoning in an analogous way, the probability P_{g_S} can be modeled as

$$P_{g_S} = \frac{1}{\sqrt{2\pi}\sigma_{g_S}} e^{-\frac{1}{2}\left(\frac{\rho(h \circ g_S, g_R) - \mu_{g_S}}{\sigma_{g_S}}\right)^2}, \quad (7)$$

where ρ is a suitable metric function.

Having modeled the probabilities involved, we can now compute the relational matching cost. Substituting equations (4) to (7) in equation (3), we have:

$$C(h) = A + \quad (8)$$
$$\frac{1}{2}\|E_f(h) - \mu_f\|^2_{\sigma_f} + \frac{1}{2}\|E_r(h) - \mu_r\|^2_{\sigma_r} +$$
$$\frac{1}{2}\|E_{f_U}(h) - \mu_{f_U}\|^2_{\sigma_{f_U}} + \frac{1}{2}\|E_{g_S}(h) - \mu_{g_S}\|^2_{\sigma_{g_S}},$$

where $E_f(h) = \#L + \#U - 2\#H$, is the *feature error*, $E_r(h) = \#(R - S \circ h^{-1}) + \#(S - R \circ h)$, is the *relational error*, $E_{f_U}(h) = \rho(f_U \circ h, f_{L|_H})$, is the *feature metric error*, $E_{g_S}(h) = \rho(h \circ g_S, g_R)$, is the *relational metric error*, $\|x - \mu\|^2_\sigma = (x - \mu)^2/\sigma^2$, is the squared *Mahalanobis distance* from x to μ, and $A = -\ln P(M) + 4\ln\sqrt{2\pi} + \ln(\sigma_f \sigma_{f_U} \sigma_r \sigma_{g_S})$, is a constant for a given model M.

VII Iterative-Deepening-A^* Matching

A match can be found by using the relational matching cost defined in section B, and the well known *branch–and–bound* tree search technique. In the standard branch and bound approach during search there are many incomplete paths contending for further consideration. The one with the least cost is extended one level, creating as many new incomplete paths as there are branches. This procedure is repeated until the tree is exhausted.

The branch and bound search can be improved greatly if the path to be extended is selected such that an estimate of the total cost using that sub-path is minimal. This search technique is usually known as A^*. An important and well known result is that if the estimate of the total cost is always less than the actual cost, the path found by A^* is optimal. The drawback of this algorithm is the same as that of breadth-first search, namely its memory requirement. The algorithm must maintain a list of all contending paths. In each cycle, the number of contending paths is increased by $b - 1$, where b is the branching factor of the node being extended. Thus, the space complexity of A^* is $O(b^d)$ where d is the solution depth level.

Korf [9] presented a new search algorithm, called iterative-deepening-A^* (IDA^*) that gets around the memory problem of A^* without sacrificing optimality or time complexity. The algorithm consists of a sequence of depth-first searches. IDA^* starts with an initial threshold value equal to the estimated total cost for the root of the tree. In each iteration, the algorithm is a pure depth-first search, cutting off any branch that has an estimated total cost larger than the current threshold value. If a solution is expanded, the algorithm is finished. Otherwise, a new threshold value is set to the minimum estimated cost that exceeded the previous threshold, and another depth-first search is started from scratch.

As in the case of A^*, if the estimated total cost is an underestimate of the real total cost, IDA^* finds the optimal solution. The advantage of IDA^* over A^* is that since each iteration of the algorithm is a depth-first search, the memory complexity is $O(d)$, instead of exponential. The number of nodes opened by IDA^* is asymptotically the number of nodes opened by A^*, provided that the tree grows exponentially. In practice, IDA^* runs faster than A^*, since its overhead per node is less than the overhead for A^*.

A *Relational Matching Cost Underestimate*

The IDA^* algorithm requires an underestimate of the relational matching cost of an observation mapping that is an *extension* of the current partial mapping. In this section we formally define the extension of a partial mapping and use this concept to find a lower bound of the relational matching cost given in equation (8).

Def. VII.1 Given two one-to-one mappings h and m, such that $\mathrm{Dom}(m) \subseteq \mathrm{Dom}(h)$, and $m(l) = h(l)$ for all $l \in \mathrm{Dom}(m)$, we say that the function h is an *extension* of the function m, and that the function m is a *restriction* of the function h. The *order* of the extension h with respect to m is the difference between the cardinalities of the sets $\mathrm{Dom}(h)$ and $\mathrm{Dom}(m)$.

Let $m\colon L \to U$ be a partial mapping assigning some labels to some units, and let m_j be an extension of order j of m. Using equation (8), the relational matching cost of m_j is bounded by:

$$C(m_j) \geq A + \quad (9)$$
$$\frac{1}{2}\|E_f(m_j) - \mu_f\|^2_{\sigma_f} + \frac{1}{2}\|E_r(m_j) - \mu_r\|^2_{\sigma_r}.$$

The term $\|E_f(m_j) - \mu_f\|^2_{\sigma_f}$ can be exactly computed by using the definition of feature error. The feature error for the extended mapping m_j, $E_f(m_j)$, is given by:

$$E_f(m_j) = \#L + \#U - 2\#\mathrm{Dom}(m_j) = E_f(m) - 2j\ .$$

Hence,

$$\|E_f(m_j) - \mu_f\|^2_{\sigma_f} = \|E_f(m) - 2j - \mu_f\|^2_{\sigma_f}. \quad (10)$$

To find a lower bound of the term $\|E_r(m_j) - \mu_r\|^2_{\sigma_r}$, we start by noticing that a partial mapping partitions the sets of relational tuples into disjoint subsets:

Def. VII.2 The *set of used relational tuples of labels*, $R^u(m)$, is the subset of relational tuples of labels in R such that all the labels in their feature vectors have been associated a correspondent unit in U through the mapping m.

Def. VII.3 The *set of i–partially free relational tuples of labels*, $R^p_i(m)$, is the subset of relational tuples of labels in S such that all but $i \geq 0$ of the labels in their feature vectors have been associated a correspondent unit in U through the mapping m.

The set of used relational tuples of units, $S^u(m)$, and the set of i–partially free relational tuples of units, $S^p_i(m)$, are defined in a similar way. Fig. 3 shows the sets S and R, and the partition induced on them by the partial match m.

The relational error for the mapping m_j, $E_r(m_j)$, is given by:

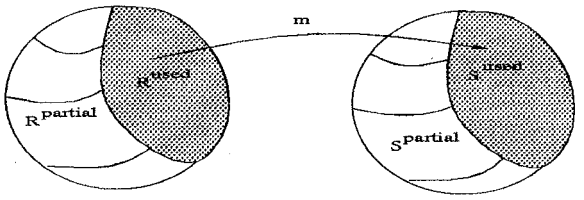

Figure 3: Partition of the sets of relational tuples of features induced by a partial match.

$$E_r(m_j) = \#(R - S \circ m_j^{-1}) + \#(S - R \circ m_j). \quad (11)$$

As the order of extension j increases, some partially free relational tuples become used and the cardinal of the sets $(R - S \circ m_j^{-1})$ and $(S - R \circ m_j)$ can not increase. Hence,

$$E_r(m) \geq E_r(m_j). \quad (12)$$

A partially free relational tuple of labels $r \in R_i^p(m)$ and a partially free relational tuple of units $s \in S_i^p(m)$ are *compatibles* if they agree on the features that have been already matched. Let $c_i(m)$ be the minimum between the number of relational tuples of labels i–partially free that have compatibles, and the number of relational tuples of units i–partially free that have compatibles through the mapping m. That is, $c_i(m)$ is the maximum number of partially free relational tuples that could be correctly matched by extending the mapping m with i correspondences. Then, the relational error of the extended mapping m_j, $E_r(m_j)$, has the lower bound:

$$E_r(m_j) \geq E_r(m) - 2\sum_{i \geq j} c_i(m) = E_r^{\min}(m, j). \quad (13)$$

Thus, using equations (12) and (13) we have,

$$\|E_r(m_j) - \mu_r\|_{\sigma_r}^2 \geq \|E_r^{\text{bound}}(m, j) - \mu_r\|_{\sigma_r}^2 \quad (14)$$

where

$$E_r^{\text{bound}}(m, j) = \begin{cases} E_r(m) \text{ if } E_r(m) < \mu_r, \\ \mu_r \text{ if } E_r^{\min}(m, j) < \mu_r < E_r(m), \\ E_r^{\min}(m, j) \text{ otherwise.} \end{cases} \quad (15)$$

Substituting equations (10) and (14) in equation (9), we have:

$$C(m_j) \geq A + \frac{1}{2}\|E_f(m) - 2j - \mu_f\|_{\sigma_f}^2 + \frac{1}{2}\|E_r^{\text{bound}}(m, j) - \mu_r\|_{\sigma_r}^2. \quad (16)$$

Inequality (16) provides an easy to compute underestimate of the total cost of a partial match. This lower bound of the total cost can be used to quickly guide an IDA^* algorithm to the correct mapping reducing dramatically the number of nodes to be opened during the search. The complete matching algorithm using IDA^* is given in Fig. 4.

VIII Use of Prediction in Matching

In the previous section we proposed a probabilistic model to solve the relational matching problem. In this section we discuss how to use the output produced by PREMIO's prediction module to estimate the proposed model parameters.

Step 0: Initialization.
Set Threshold $Th = \text{EstCost}(root)$.
Set ϵ to the desired matching cost.
If $Th > \epsilon$
  Begin
    The model and the image do not match.
    Go to step 4.
  End if.
Until the solution is found or the tree is exhausted, **do** :
Begin
  Step 1: Start Depth First.
  Form a stack $Q_{\mathcal{P}}$ of partial matches
  Let $\mathcal{P}_0$ be the initial partial match.
  Set $MinPruned :=$ highest value.
  Step 2: Iterate over current paths.
  Until $Q_{\mathcal{P}}$ is empty, **do**
  Begin
    $\mathcal{P} := \text{FIRST}(Q_{\mathcal{P}})$
    $m :=$ partial mapping associated with $\mathcal{P}$
    $C_m :=$ relational cost of m
    Step 2.1: Test if $\mathcal{P}$ can be extended.
    If the path $\mathcal{P}$ can be extended,
    Begin
      Step 2.1.1: Select a label to extend the path.
      Look for two partially free compatible tuples.
      Step 2.1.2 Extend the path
      For each $u \in U^r$, **do**
      Begin
        $h_1 :=$ path m extended with the pair (l, u).
        $\mathcal{P}' :=$ path associated with the mapping h_1.
        Step 2.1.2.1 Compare to the upper bound.
        If $\text{EstCost}(h_1) \leq \epsilon$
        Begin
          Step 2.1.2.1.1 Check if done.
          If $\text{Cost}(h_1) \leq \epsilon$
          Begin
            $\mathcal{P}'$ is a satisfactory match.
            Exit .
          End if.
          Step 2.1.2.1.2 Add the path to the stack.
          If $\text{EstCost}(h_1) \leq Th$
          Begin
            $\text{FIRST}(Q_{\mathcal{P}}) := \mathcal{P}'$.
          Else
            $MinPruned := \min\{MinPruned, \text{EstCost}(h_1)\}$
          End if.
        End if.
      End for.
    End if.
  End until.
  Step 3: Update Threshold
  $Th = MinPruned$
End until.
Step 4: End of Algorithm
Announce failure.

Figure 4: Matching Algorithm

A *Probabilistic Prediction Models*

Given an object, a set of sensor and light configurations corresponding to a view aspect[2] of the object, PREMIO summarizes all the predictions obtained by the prediction module into a probabilistic model called the *probabilistic prediction model* (PPM). A PPM consists of a model $M = (L, R, f_L, g_R)$ and a set of statistics $\Theta = \{P(M), P_U, P_S, P_{f_U}, P_{g_S}\}$.

The model M

PREMIO's prediction module predicts which 2D features should be detectable for a given configuration of light and sensor and a given image processing sequence. Each of the detectable 2D features correspond to their originating 3D feature, and have associated attribute values.

Given a set of n predictions for a set of n sensor and lighting configurations, we approximate the detectability of a 2D feature by the frequency rate of its appearance.

[2] A view aspect is defined as a set of views with similar properties. In this paper, an aspect corresponds to a set of views that have the same visible faces.

Two 2D features appearing in two different images are considered to be the same feature if they have a common 3D originating feature. The set of labels L is formed by those 2D features that have high enough probability of being detected (above threshold t_f), as a whole or in pieces for the given set of sensors and light sources. Furthermore, each feature in L has associated attributes which are given by the mean and the standard deviation of the attribute values of the feature for the n predictions.

Similarly, PREMIO's prediction module predicts which relationships among features would be detected and their strength, for a given configuration of light and camera and a given image processing sequence. Given the n predictions we approximate the probability of a relation among a set of features to holding by the frequency rate of its appearance. The set of relational tuples R is formed for those relations among features in L such that they have high enough probability of holding (above threshold t_R). As with feature attributes, the relationship strength values of the tuples in R are represented by the mean and standard deviation of the relational tuples for the n predictions.

The model $M = (L, R, f_U, g_S)$ obtained in this way, is a *probabilistic model* of the object for the given set of configurations of sensors and lights. Note that neither all the features in L, nor all the relational tuples in R need to be present in a single prediction. Neither do all the features of a particular prediction need to be in L. The model M combines a group of predictions into a single model, which is a sort of "average" model. The differences between the model M and the individual predictions that were used to build the model are summarized in the statistics Θ.

The statistics Θ

Once the model M is obtained, the individual predictions can be used to generate samples for the four error distributions P_U, P_S, P_{f_U}, and P_{g_S}. Let $I_1, \ldots, I_n$ be a set of n predictions. Each prediction can be represented by the four tuple $I_i = (U_i, S_i, f_{U_i}, g_{S_i})$ where U_i is the set of units, S_i is the set of relational tuples of units, f_{U_i} is the attribute mapping for the units in U_i, and g_{S_i} is the relationship strength mapping for the relational set S_i. By construction we know the true observation mapping for each of the predictions. Let $h_i: H_i \rightarrow U_i$, with $H_i \subseteq L$ be the true observation mapping for the prediction i. Then, we can compute the quantities $\#L + \#U_i - 2\#H_i$, $\#(R - S_i \circ h_i^{-1}) + \#(S_i - R \circ h_i)$, $\rho(f_{U_i} \circ h_i, f_{L|H_i})$, and $\rho(g_{S_i} \circ h_i, g_{R|H_i})$ for $i = 1, \ldots, n$.

Now, the problem of finding the statistics Θ reduces to the well-know problem of estimating the parameters of normal distributions given sets of n samples. A detailed treatment of this topic can be found in statistics textbooks [10].

IX Experimental Protocol

Controlled experiments are an important component of computer vision, for the controlled experiment demonstrates that the algorithm designed by the computer vision researcher recognizes, locates, and/or measures what it was designed to do [4]. In this section we describe the experimental protocol, based on the one presented in [8], that we designed to evaluate PREMIO's matching algorithm.

A Probabilistic Prediction Model Generation

In the previous section we introduced the concept of the probabilistic prediction model (PPM). In our experiments, we use a PPM that summarizes the predictions obtained by the prediction module for a set of views in the same *aspect* of the object. Next, we describe how to generate the two components of a PPM, the model M and the statistics Θ.

Model Generation

To generate a model M we need to do the following steps:

1. Select a region $\mathcal{V}$ of the viewing space. The region $\mathcal{V}$ is a spherical sector between two spheres. It is specified by the range of the longitude $(\Phi_{v\min}, \Phi_{v\max})$ and latitude $(\theta_{v\min}, \theta_{v\max})$ angles and the radius of the viewing sphere $(R_{v\min}, R_{v\max})$.
2. Select a region $\mathcal{I}$ of the illumination space. This region is specified in an analogous way as the viewing space region, by the range of the longitude $(\Phi_{i\min}, \Phi_{i\max})$ and latitude $(\theta_{i\min}, \theta_{i\max})$ angles and the radius of the illumination sphere $(R_{i\min}, R_{i\max})$.
3. Sample the viewing space and illumination space regions. Given the desired number of samples N_v, and N_i, the viewing and illumination space regions previously defined are uniformly sampled. Let $\mathcal{V}_s$ be the set of the sampled viewing positions, and $\mathcal{I}_s$ the set of the sampled lighting positions.
4. Generate the predictions. For each pair $(v, i) \in \mathcal{V}_s \times \mathcal{I}_s$, use the prediction module to predict the subset of detectable labels L_{vi}, its associated attribute mapping $f_{L_{vi}}$, the subset of detectable relational tuples R_{vi}, and its associated relationship strength mapping $g_{R_{vi}}$. The prediction module also generates the corresponding set of units U_{vi}, the associated attribute mapping $f_{U_{vi}}$, the set of relational tuples S_{vi}, and the associated strength mapping $g_{S_{vi}}$.
5. Obtain detectability frequencies. We will approximate the probability of a label/relationship being detected, given that the view and light are in the considered regions $\mathcal{V}$ and $\mathcal{I}$, by the observed frequency rate of their detectability in the generated predictions.
   These approximations are based on the fact that the predictions were made from uniformly sampled camera and light positions as well on the CTL (provided, that N_v and N_i are large enough).
6. Select desired detectability. Select the desired minimum label detectability t_f and the minimum relational detectability t_R.
7. Combine the predictions. The $N_v \times N_i$ predictions are combined into a single model $M = (L, R, f_L, g_R)$ such that the labels in L have a detectability greater than t_f and the relational tuples in R have a detectability greater than t_R.

Statistics Generation

The statistics Θ are generated as follows:

1. Obtain the observation mappings. For each generated prediction, find the true observation mapping between the predicted image and the obtained model M, $h_{vi}: H_{vi} \rightarrow U_{vi}$, with $H_{vi} \subseteq L$, and $(v, i) \in \mathcal{V}_s \times \mathcal{I}_s$. These observation mappings only include correspondences with units that were originated from labels with detectability greater than or equal to t_f.
2. Obtain matching errors. The previous step produced $N_v \times N_i$ true observation mappings, h_{vi}. For each prediction compute the quantities: $\#L + \#U_{vi} - 2\#H_{vi}$, $\#(R - S_{vi} \circ h_{vi}^{-1}) + \#(S_{vi} - R \circ h_{vi})$, $\rho(f_{U_{vi}}, f_{L|H_{vi}})$, and $\rho(g_{S_{vi}}, g_{R|H_{vi}})$.

3. Parameter estimation. Use the matching errors generated in the previous step to estimate the parameters of the four Gaussian distributions P_U, P_S, P_{f_U}, and P_{g_S}: μ_f,σ_f,μ_R,σ_R, μ_{f_U},σ_{f_U},μ_{g_S}, and σ_{g_S} by using the sample means and variances.

B *Image Generation*

The matching algorithm has been tested on simulated images and some real images. Next, we describe how to generate a set of simulated images to be matched against the previously generated PPM.

1. Sample the viewing space and illumination space regions. Given the desired number of samples N'_v, and N'_i, the viewing and illumination space regions $\mathcal{V}$ and $\mathcal{I}$ are uniformly sampled again. Let $\mathcal{V}'_s$ be the new set of the sampled viewing positions, and $\mathcal{I}'_s$ the new set of the sampled lighting positions.
2. Generate the images. For each pair $(v,i) \in \mathcal{V}'_s \times \mathcal{I}'_s$, use the prediction module to predict the subset of detectable labels L_{vi}, its associated attribute mapping $f_{L_{vi}}$, the subset of detectable relational tuples R_{vi}, and its associated relationship strength mapping $g_{R_{vi}}$. The prediction module also generates the corresponding set of units U_{vi}, the associated attribute mapping $f_{U_{vi}}$, the set of relational tuples S_{vi}, and the associated strength mapping $g_{S_{vi}}$.

C *Matching*

To test the performance of the matching algorithm, we matched the randomly-generated images against the PPM, varying the number of correspondences sought, and we compared the obtained camera position against the known "true" camera position:

1. The number n of correspondences to find using the matching algorithm was chosen.
2. For each of the generated images:
   (a) The matching algorithm was applied to search for an observation mapping with n correspondences between the PPM and the image.
   (b) If such observation mapping was found, the n correspondences found were used to compute the camera position.
   (c) The distance between the camera position and the true camera position, d was computed. This distance is referred to as the *position error* and is a measure of the strength of the mapping found. The smaller the error is, the greater the strength of the mapping found.

D *Performance Evaluation*

If an observation mapping is not found, the experiment is referred to as a *misdetection error* (ME). The probability of a misdetection error, given the number of correspondences sought n, is defined as:

$$P(\text{ ME } | n) = \frac{\#\text{ Mappings not Found}}{\#\text{Images}} .$$

If an observation mapping is found, in order to decide whether or not the system has found the correct observation mapping, an accuracy criterion C must be applied to the position error. If the position error of an image is larger than the accuracy criterion C, the observation mapping found is declared incorrect, and the experiment is referred to as a *pose error* (PE). That is, C is the maximum position error allowed. In order to study the performance of the system, the accuracy criterion C is varied through a set of values. The probability of a pose error, given that a mapping m_n with n correspondences was found, and given the accuracy criterion C is defined as:

$$P(\text{ PE} | m_n, C) = \frac{\#\text{ Incorrect Mappings}}{\#\text{Mappings Found}} .$$

The performance of the matching algorithm is characterized by both, the probability of a misdetection error and the probability of a pose error. Hence, the performance of the algorithm changes as the number of correspondences sought and the accuracy criterion are varied. The results of the experiments described in the presented protocol can be summarized by plotting the *receiver operating curves* (ROC) of the algorithm. The ROC are obtained by plotting the probability of misdetection against the probability of pose error, parametrical on the number of correspondences n and the accuracy criterion C.

X Experiments

In our experiments we used a CCD camera with focal length 4.8 mm. and a resolution of 1.25901 mm./pixel $\times$ 1.18758 mm./pixel. The light is a point source of unpolarized light, of intensity 1, located at a fixed position. The set of features L are segments. The feature attribute mapping f_U associates to each segment four attributes: its midpoint coordinates, its length, and its orientation. The set of relational tuples of segments R is formed by three different types of relationships: *junctions* of two segments, junctions of three segments, and *triples* of segments. A *junction* of two/three segments is an ordered set of two/three lines which meet at a junction. The segments are ordered such that the angles between the segments are less than 180 degrees when the lines are traced clockwise. A *triple* of segments [7] is defined as an ordered set of three lines, two pairs of which meet at a junction. The angles at the two junctions must both be less than 180 degrees when the lines are traced clockwise, so the triple has a well defined "inside". For this set of experiments, the relationship strength mapping g_R was not used. Conceptually, this amounts to having a constant relationship strength mapping.

Fig. 5 shows a perfect line drawing of *Cube3Cut*, one of the objects modeled in Premio. In what follows we describe the results of a series of experiments with a PPM model of Cube3Cut combining over a hundred of predictions. These predictions were generated with Cube3Cut located at the origin, the light fixed at $(-3.0cm., -2.0cm., 60.0cm.)$, and the camera moving on a sphere of radius $R = 35.3857cm.$, with longitude $20° \leq \Phi_v \leq 70°$ and latitude $20° \leq \theta_v \leq 70°$. The minimum feature detectability was set to $t_f = 0.0$, and the minimum relational detectability was set to $t_R = 0.15$. Fig. 6 shows some of the predictions used to build the PPM. Fig. 7 shows a line drawing of the corresponding model M. The segments are shown with their mean attributes, labeled in descending order of detectability. The parameters for the error distributions for this model are: $\mu_f = 9.6875$, $\sigma_f = 2.5042$, $\mu_R = 10.4107$, $\sigma_R = 3.3867$ (junctions of two segments), $\mu_R = 7.9375$, $\sigma_R = 1.5024$ (junctions of three segments), and $\mu_R = 17.9107$, $\sigma_R = 5.6226$ (triples), and $\mu_{f_U} = 32.5366$, $\sigma_{f_U} = 8.9240$.

Fig. 8 shows the operating curves obtained when the matching algorithm was tested on a set of over sixty artificial images. The number of correspondences between

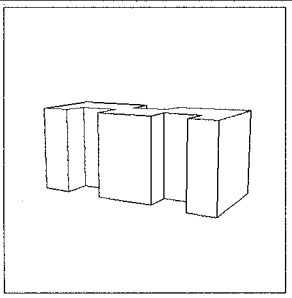

Figure 5: Cube3Cut: an object modeled in Premio. The figure shows a perfect line drawing of the object Cube3Cut.

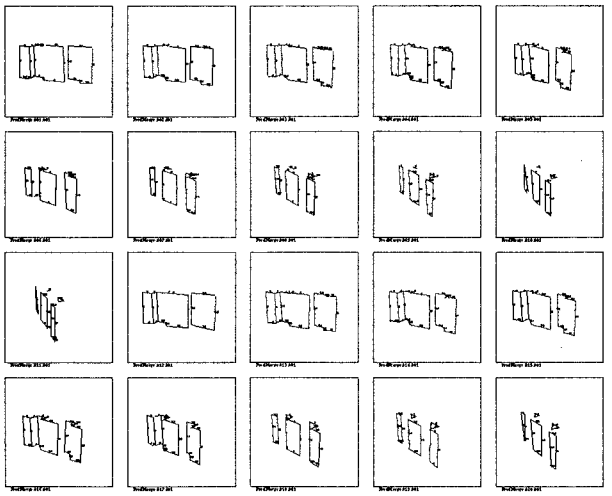

Figure 6: Cube3Cut: predicted images. The figure shows a few predicted segmented images of Cube3Cut, when the object is at the origin, the light is at $(-3.0cm., -2.0cm., 60.0cm.)$, and the camera moves on a sphere with radius $R = 35.3857cm.$, with $20° \leq \Phi_v \leq 70°$, and $20° \leq \theta_v \leq 70°$.

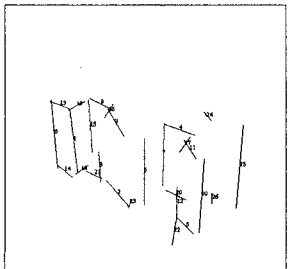

Figure 7: Probabilistic prediction model: model M component. The figure shows a line drawing using the mean values of the attributes of the segments in the model M of a PPM of Cube3Cut. The shown PPM combines over a hundred images of Cube3Cut when the object is at the origin, the light is at $(-3.0cm., -2.0cm., 60.0cm.)$, and the camera moves on a sphere with radius $R = 35.3857cm.$, $20° \leq \Phi_v \leq 70°$, and $20° \leq \theta_v \leq 70°$.

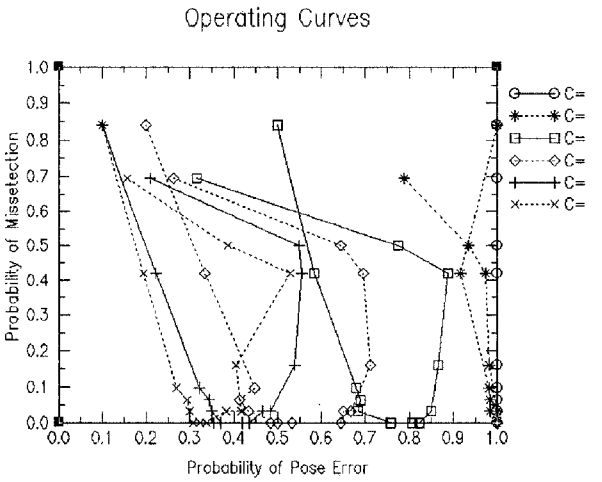

Figure 8: Matching Operating Curves for Cube3Cut. The plots show the misdetection probability versus the probability of pose error, given that a camera position was found, parametrical on the number of correspondences sought, for different accuracy criterion C value.

segments sought was varied between 3 and 25. The correspondences found between segments were used to determine correspondences between points in the images and points on the object, which in turn were used to determine the camera position. Let n be the number of correspondences sought between segments. For $n = 3$, the probability of misdetection is high. If too few segments are matched, few point correspondences are found, resulting in a high misdetection rate. For $n = 4$, the number of point correspondences found increases, and hence the probability of misdetection decreases. However, the number of point correspondences found remains low, and hence the pose error is large. For $n = 5$ to $n = 8$, the number of point correspondences increases and hence both probabilities decrease. For $n = 9$ to $n = 11$, the rate of misdetection is zero. The more correspondences found, the more accurate the computed camera position is, and the smaller the probability of pose error is. For $n > 11$, there are images for which the matching algorithm can not find n correspondences and hence the misdetection rate increases. However, for those images that a set of correspondences is found, the accuracy of the computed camera increases, since more point correspondences are available, and therefore the probability of pose error decreases. In general, we would like the system to have both low probability of misdetection, *and* probability of pose error. Having this in mind, the operating curves can be used to select the number of correspondences sought during the matching. For example, for an accuracy of $C = 5.0\%$, and a number of correspondences $n = 13$, the probability of misdetection is equal to 0.3 and the probability of pose error is equal to 0.3226.

Fig. 9(a) shows a real image of Cube3Cut. Fig. 9(b) shows a perfect line drawing of Cube3Cut for the camera position used in (a). Fig. 9(c) shows a segmented image of (a) and Fig. 9(d) shows the predicted segmentation for the camera and light configuration used in (a). Fig. 9(e) shows the line drawing of Cube3Cut for the camera position obtained from matching the line segments in (c) against the PPM showed in Fig. 7, overimposed the line drawing showed in Fig. 9b.

The IDA^* matching algorithm presented in this paper

takes 2 iterations to find nine correspondences (all of them correct) between the image given in Fig. 9(c) and the prediction model given in Fig. 7. During the search, only 42 nodes were opened and 32 of them were pruned (76%).

XI Conclusion

In this paper we have posed the relational matching problem as a special case of the pattern complex recognition problem. This probabilistic approach allows us to make explicit statements about how an image is formed from a model, and hence to find theoretical underestimates of the matching cost to direct and reduce the search. Furthermore, we have described how the predictions generated by PREMIO's prediction module can be used to estimate the probabilistic model parameters. Finally, we have laid out a rigurous methodology to characterize the performance of the proposed matching algorithm and we have presented experimental results using artificial and real images.

References

[1] K. L. Boyer and A. C. Kak. Structural stereopsis for 3-d vision. *IEEE Transactions on Systems, Man and Cybernetics*, PAMI-10(2):144–166, March 1988.

[2] O. I. Camps, L. G. Shapiro, and R. M. Haralick. PREMIO: The Use of Prediction in a CAD-Model-Based Vision System. Technical Report EE-ISL-89-01, Department of Electrical Engineering, University of Washington, 1989.

[3] W. E. L. Grimson. The combinatorics of object recognition in cluttered environments using constrained search. In *Proc. of the International Conference on Computer Vision*, pages 218–227, 1988.

[4] R. Haralick. Performance assessment of near perfect machines. *Journal of Machine Vision and Applications*, 1988.

[5] R. Haralick. The pattern complex. In R. Mohr, T. Pavlidis, and A. Sanfeliu, editors, *Structural Pattern Analysis*, pages 57–66. World Scientific Public. Co, 1989.

[6] R. Haralick and L. G. Shapiro. The consistent labeling problem: part i. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-1(2):173–184, April 1979.

[7] J. Henikoff and L. Shapiro. Interesting patterns for model-based matching. In *ICCV*, 1990.

[8] T. Kanungo, M. Jaisimha, R. Haralick, and J. Palmer. An experimental methodology for performance characterization of a line detection algorithm. In *SPIE Conference on Optics, Illumination and Image Sensing for Machine Vision V*, pages 104–112, November 1990.

[9] R. E. Korf. Search: A survey of recent results. In H. E. Shrobe and T. A. A. for Artificial Intelligence, editors, *Exploring Artificial Intelligence*, chapter 6, pages 197–237. Morgan Kaufmann Publishers, Inc., 1988.

[10] L. Ott. *An Introduction to Statistical Methods and Data Analysis*. Duxbury Press, Boston, Ma., second edition, 1984.

[11] A. Rosenfeld, R. A. Hummel, and S. W. Zucker. Scene labeling by relaxation operations. *IEEE Trans. Syst. Man Cybern.*, SMC-06, June 1976.

[12] A. Sanfeliu and K. S. Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, MAn and Cybernetics*, SMC-13(13):353–362, May 1983.

[13] L. Shapiro and R. Haralick. Structural descriptions and inexact matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-3(5):504–519, September 1981.

[14] L. Shapiro and R. Haralick. A metric for comparing relational descriptions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-7, 1985.

[15] J. R. Ullman. An algorithm for subgraph homomorphisms. *J. Assoc. Comput. Mach.*, 23:31–42, January 1976.

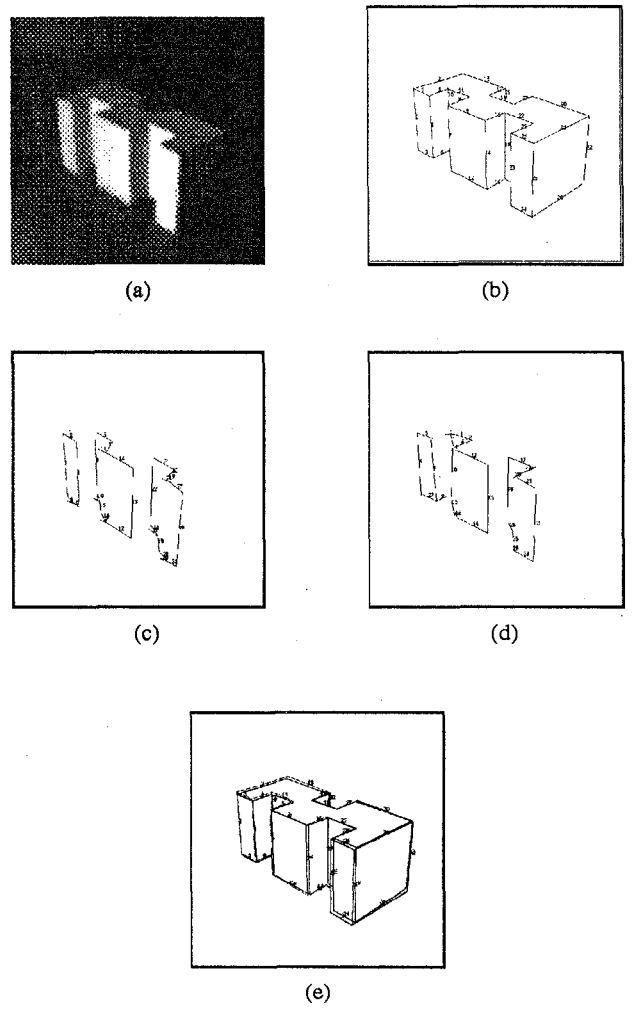

Figure 9: A real image of Cube3Cut: (a) A real image of Cube3Cut. (b) A perfect line drawing of Cube3Cut for the camera position used in (a). (c) A segmented image of (a). (d) The predicted segmentation for the camera position and lighting conditions used in (a). (e) A perfect line drawing of Cube3Cut for the camera position obtained from matching the line segments showed in (c) against the corresponding PPM, overimposed the line drawing shown in (b).