Presented at the 1994 International Conference on Computer Vision in Seattle, Washington

Efficient Indexing Techniques for Model Based Sensing

Aaron S. Wallack * John F. Canny †
Computer Science Division, 571 Evans Hall
University of California, Berkeley, CA 94720, USA

Abstract

Indexing is a model-based recognition technique, in which unknown objects are identified using lookup tables. Indexing coordinates are extracted from sensed features, and the indexing coordinates specify a table entry containing the object's identity. Usually, only a small fraction of the possible indexing coordinates correspond to modeled objects, and hash tables are often used to save space. In this paper, we present a new indexing data structure called a tree grid *which has two advantages over hash tables: (i) The tree grid preserves spatial ordering, so that nearby indexing entries can be retrieved efficiently (ii) The tree grid compacts the storage size of the table by a factor of as much as two orders of magnitude. k coordinates index an ordering of the interpretations, and 1 coordinate determines the consistent interpretations for objects with k degrees of freedom. We also show that for almost all model sets, $2k+1$ indexing coordinates are sufficient to discriminate between two generic models, implying that $2k+1$ indexing coordinates specify a unique interpretation. We have implemented an indexing algorithm for recognizing 3D objects from pairs of image rays using the tree grid technique, and the results are reported.*

1 Introduction

State-of-the-art machine vision technology can *interpret* scenes, i.e. recognize objects; the next step is real-time model-based recognition. Recognition involves interpreting the sensed features as model features. This could require checking an exponential number of hypothetical interpretations, without heuristics for pruning the search. One approach to pruning the set of possible interpretations is *indexing*, in which the observed features directly specify a subset of consistent interpretations.

*Supported by Fannie and John Hertz Fellowship and NSF IRI-9114446

†Supported in part by David and Lucile Packard Fellowship and National Science Foundation Presidential Young Investigator Award (# IRI-8958577).

Definition 1 *Hypothesis H* is a matching between a group of model features and a group of sensed features.

Indexing techniques are becoming prevalent in the field of machine vision for solving the recognition problem and the correspondence problem [1, 8, 11, 9, 3, 5]. Indexing involves distilling/extracting an overconstraining indexing coordinate from an observation (a group of sensed features) and then interpreting the observation via a precomputed lookup table. Utilizing geometric invariants (descriptors which are constant irregardless of the configuration) benefits indexing since each hypothesis corresponds to a single table entry [8, 11, 5]. Unfortunately, it has been shown that there are no *invariants* for an arbitrary number of three-dimensional points for orthographic [3] or perspective vision [2]. Indexing tables include predicted observations for all configurations.

Overconstraint is crucial to indexing because each observation not only specifies a configuration for each hypothesis, but the observation also rates the quality of the match between the observation and each hypothesis and configuration as well. Only a small fraction of the indexing coordinates are consistent with each hypothesis. One advantage of indexing is using a single lookup to compare an observation with all of the hypotheses. This is achieved by merging the lookup tables for various hypotheses into a composite lookup table. The two advantages of indexing are simplified online computations and performing only a single lookup operation, which enable real-time performance.

In this paper, we present an efficient indexing technique, the *tree grid*, which trades off online speed for space compression and spatial faithfulness (order preservation). Lookup time increases by a factor of 10, but table size shrinks by a factor of 100, allowing more tables, of finer resolution, to share fast main memory. Spatial faithfulness enables the user to change the resolution of the search online, to determine the most consistent hypothesis without performing an exponential search. We present experimental results of this technique for a three-dimensional object recogni-

tion algorithm from pairs of edge-detected rays.

1.1 Previous Work

Clemens and Jacobs proved that under orthographic projection, indexing coordinates from three-dimensional point groups must be represented by at least a two-dimensional surface in a single index space [3], requiring an inordinate number of table entries. Jacobs developed an indexing algorithm for the more general class of affine transformations, which separates the two-dimensional surface in indexing space into two one-dimensional surfaces (lines) in two smaller spaces [7]. This separation reduces the size of the tables and also simplifies table construction. The drawback of this approach is that for affine transformations, having eight degrees of freedom,; overconstraint requires five point groups.

This work also stems from work in the field of coding theory. A set of similar lists can be stored in $O(e)$ space, where e is the total number of edits involved in synthesizing the lists, which can be much less than the sum of the lengths of the lists. Cole [4] presented an algorithm which required examining all of the lists simultaneously. Sarnak and Tarjan [10] described an incremental algorithm which precluded memoization. In the future, we plan implementing a variant of Sarnak and Tarjan's algorithm in order to allow memoization, and we expect this to reduce the storage size by a factor of $O(\log(m))$, where m is the list length.

1.2 Framework

Definition 2 *Configuration* $\vec{x}$ describes the observable state. The configuration corresponds to the object's pose with respect to the camera. *Configuration space* (C) describes the set of all configurations ($\vec{x} \in$ C). For example, under unscaled three-dimensional orthographic projection, configuration space can be defined by coordinate axes of x, y, θ, ϕ, ψ; z is not a coordinate axis because it is unobservable. The observation technique may normalize out observable degrees of freedom; for example, the rotation in the image plane is normalized by rotating a group of model points so that the vector from the first point to the second point is horizontal. k refers to the number of normlized degrees of freedom ($\mathrm{C} = \mathbb{R}^k$).

Definition 3 *Indexing coordinate* y_i refers to a real valued scalar data value distilled from a group of sensed features; $\hat{y}$ refers to an n-tuple, a set of n simultaneously observed indexing coordinates. *Indexing space* ($\mathbb{I}$) describes the set of all possibly observed n-tuples ($\mathbb{I} = \mathbb{R}^n$, $\hat{y} \in \mathbb{I}$). q image points provide $2q$ indexing coordinates.

Definition 4 *Hypothesis function* F_H for a given hypothesis H, provides a map from configuration space into indexing space: $\hat{F_H} : \mathrm{C} \rightarrow \mathbb{I}$. The hypothesis set is the range of this mapping, the set of all indexing coordinates consistent with a particular hypothesis H.

The purpose of indexing is to map *from* the indexing coordinates $\hat{y}$ *to* the configuration $\vec{x}$. Indexing tables $T_H(\vec{y})$ list the configurations $\{\vec{x}\}$ consistent with an n-tuple of indexing coordinates (equation (1)).

$$T_H(\hat{y}) = \{\vec{x} | \hat{F_H}(\vec{x}) = \hat{y}\} \tag{1}$$

Since each table T_H only covers a small fraction of the indexing space $\mathbb{I}$, many such tables can be merged into a composite table ($T_{\bigcup_{H_i}}$ in equation (2)); thereby, one can usually determine both the hypothesis and the configuration consistent with given indexing coordinates in a single lookup operation. Each n-tuple of indexing coordinates $\hat{y}$ usually implies a unique hypothesis $H_{\hat{y}}$ and configuration $x_{\hat{y}}$. Essentially, indexing techniques are used to solve the problems of the form: given an indexing coordinates $\hat{y}$, enumerate all consistent hypotheses H_i and configurations $\{\vec{x_i}\}$ which satisfy $\hat{y} = \hat{F_{H_i}}(\vec{x_i})$ (equation (2)).

$$T_{\bigcup_{H_i}}(\hat{y}) = \{< H, \vec{x} > | \vec{F_H}(\vec{x}) = \hat{y}\} \tag{2}$$

Indexing techniques are based on the assumption that the predicted indexing coordinates and the sensed indexing coordinates will vary so slightly that both index the same table entry. When the hypotheses only cover a small fraction of the indexing coordinates, indexing tables are usually implemented as hash tables. In order to index table entries, the indexing coordinates must first be discretized into integral indices. For completeness and correctness, indexing tables must account for any hypothesis and configuration indexing coordinates which discretize to a table entry's indices.

The multiple hypotheses scenario is not uncommon: one example is the task of recognizing modeled objects in an image. The first step involves identifying salient features, such as edge-based features, from the sensor data. The majority of the computation is spent in the second step: interpreting the sensed features groups as groups of modeled features. Relative to a group of sensed features, each group of model features constitutes a separate hypothesis.

For a system with k degrees of freedom, k data values only constrain the configuration, $k+1$ data values overconstrain the configuration and thereby substantially narrow the set of consistent hypotheses. Furthermore, we rigorously show that $2k+1$ generic data

values are necessary and sufficient in order to specify a unique interpretation.

Current indexing implementations have three drawbacks: the completeness requirement, the size of the composite tables, and the non-spatially faithful nature of hash tables. Constructing complete tables is a difficult task. The size of the composite tables is also important because a large composite table may thrash, obviating the theoretical constant-time performance. Of lesser importance is spatial faithfulness, which simplifies the task of determining the *closest* hypothesis; this is normally accomplished by searching all of the nearby indexing coordinates, a procedure which, for hash tables, takes time exponential in n.

1.3 Algorithmic Overview

In this paper, we present a table indexing technique, the tree grid, which provides compaction and spatial faithfulness. The tree grid achieves these results by ordering the hypotheses, and performing binary searches on these orderings. k indexing coordinates are used to index an ordering of the hypotheses, and 1 indexing coordinate is used to search through that ordering. Storage space is lessened because the hypothesis orderings are similar lists, and the storage space for similar lists depends upon the number of edits, not the total number of elements. This approach exploits the coherence of the hypothesis orderings; we want to reduce the task of indexing to performing a binary search on the hypothesis orderings. In this paper, we present an ordering on the hypotheses.

The tree grid technique relies on three major ideas: concentration, segmentation, and discretization. We concentrate on $k+1$ data values of the indexing n-tuple. These $k+1$ data values are segmented into k independent coordinates and 1 dependent coordinate. The k independent coordinates are discretized to a grid point, indexing an ordering of the hypotheses. Finally, we determine the hypotheses consistent with the indexing coordinates by performing a binary search using the dependent coordinate, and then validating each of those hypotheses using the $n-(k+1)$ indexing coordinates.

1.4 Overview

In section two, we rigorously prove that $2k+1$ data values are necessary and sufficient to generically specify a unique hypothesis. In section three, we develop a theoretical framework for the tree grid indexing technique. In section four, we detail the implementation of the tree grid indexing technique. In section five, we present experimental results of the storage space compaction of the tree grid technique. Finally, we conclude by highlighting the contributions of this work.

2 Sparse Observation Theorem

In this section, we present a rigorous proof of the sparse observation theorem.

Theorem 1 *$2k+1$ observed values are necessary and sufficient to distinguish generically between two objects with k generic normalized degrees of freedom.*

Proof There is a lower bound on the number of highly precise data values necessary to differentiate two dissimilar generic objects, and this bound can be shown by a dimension counting argument. Consider an object with k normalized degrees of freedom and n observed data values. Each configuration $\vec{x} \in \mathbb{R}^k$ maps to a set of n-tuples of observed data values, and this set has dimension k, and codimension $n-k$ in data value space. The codimension of the intersection of two generic sets is equal to the sum of the codimensions of those stratifications. For $n > 2k$, the codimension of intersection $> 2k$ which implies that the dimension of the intersection is < 0.

Let P be a parametrization of the space of models. Let $M : P \times C \rightarrow \mathbb{I}$ be the *meta-hypothesis function*. So $\hat{y} = M(p, \vec{x})$ is the sensor tuple arising from model p in configuration $\vec{x}$. We can then give an alternative definition for the hypothesis set for a model p as $M(p, C) \subset \mathbb{I}$. Note that we seem to be ignoring the correspondence problem between model and image features. For the purposes of this proof, which relates to dimension, we are including the feature assignment as part of the model. Since there are finitely many possible correspondences, it does not affect our result whether we do this or not.

Now let $H_1 = M(p_1, C)$ be the hypothesis set for hypothesis p_1, and $H_2 = M(p_2, C)$ be the hypothesis set for hypothesis p_2. Under a certain condition on the model map, and for generic, i.e. almost all choices of p_1 and p_2, H_1 and H_2 will be disjoint. If H_1 and H_2 are disjoint, there is no possibility of confusing p_1 and p_2. There are several conditions on M that imply this. The first and strongest is:

Condition 1:
A map $M : P \times C \rightarrow \mathbb{I}$ is *regular* if its differential (jacobian) is everywhere surjective. This condition is stronger than necessary, but is easy to state, and probably easier to verify in most situations. In essence it says that by perturbing both pose and model parameters, we can move a sensor value $\hat{y} \in \mathbb{I}$ a small distance in any direction.

Condition 2:
A map $M : P \times C \rightarrow \mathbb{I}$ is *transversal* to a stratified set H_1 if M is transversal to all the strata in H_1.

For definitions of transversality, we refer the reader to [6]. Condition 1 implies condition 2 for any H_1, so it is strictly stronger, and we take it as the premise for our theorem:

Theorem 2 *Suppose condition 2 above holds for a hypothesis function (mapping) M, and that the dimension of the indexing space is at least $2k+1$. Then for almost all choices of $p_2 \in P$, $H_2 = M(p_2, C)$ is disjoint from H_1.*

Proof: A basic property of transversality is that if a parametrized class of mappings (M in this case, parametrized by P) is transversal to a manifold (the manifolds in H_1), then almost every map in the class is transversal to H_1. See e.g. the remark before Corollary 4.7 of [6].

Now M is transversal to H_1 by condition 2. Therefore $M(p_2, \cdot) : C \rightarrow \mathbb{I}$ is transversal to H_1 for almost every p_2. The transversality condition implies that the codimension of $M(p_2, \cdot)^{-1}(H_1)$ equals the codimension of H_1 in $\mathbb{I}$. Since H_1 has dimension k and O has dimension $2k+1$, the codimension of H_1 is $k+1$. The codimension of $M(p_2, \cdot)^{-1}(H_1)$ is also $k+1$, but it inhabits configuration space C of dimension k, which means it must be empty. $M(p_2, \cdot)^{-1}(H_1)$ empty says that there is no point in both H_1 and H_2. □

3 Theoretical Framework

In the tree grid indexing technique, consistent hypotheses are found using only $k+1$ of the n indexing coordinates. The projection $\Pi : \mathbb{R}^n \rightarrow \mathbb{R}^k$ extracts k indexing coordinates, and the projection $^{k+1}\Pi : \mathbb{R}^n \rightarrow \mathbb{R}^{k+1}$ which extracts $k+1$ coordinates.

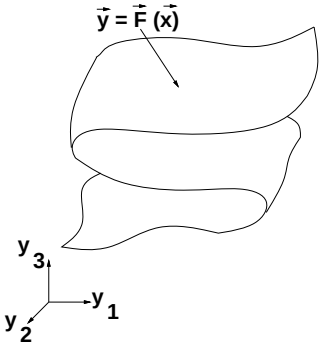

Figure 1: An example of a two-dimensional hypothesis set parameterized as $\hat{y} = \hat{F}(\vec{x})$. In this case, C is three-dimensional (y_1, y_2, y_3), and I is two-dimensional (x_1, x_2).

Definition 5 *Independent coordinates* correspond to the first k indexing coordinates ($\Pi(\hat{y})$). k independent coordinates provide sufficient constraint so that only a finite number of configurations are consistent with $\Pi(\hat{y})$. In figure 1, y_1 and y_2 were chosen as the independent coordinates.

Definition 6 *Dependent coordinate* is $k+1^{st}$ indexing coordinate (y_3 in Figure 1). Generically, since only a finite number of configurations are consistent with the independent coordinates, only a finite number of dependent coordinates are consistent with the independent coordinates.

3.1 Separating Sets Into Monotone Sheets

In order to realize a unique mapping from independent to dependent coordinates, we want to impose monotonicity. This is achieved by separating the hypothesis sets into monotone hypothesis sheets. On these hypothesis sheets, the dependent coordinate is a single valued function of the independent coordinates.

Definition 7 *Hypothesis sheet* $\mathcal{S}_H$ is a continuous subset of the hypothesis set such no two points in the sheet share the same independent coordinates (Figure 2).

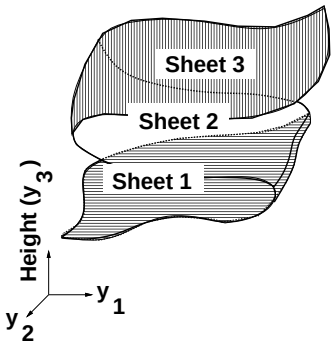

Figure 2: The hypothesis set is separated into sheets such that no two points on a sheet share the same independent coordinates.

3.1.1 Base Grid

Indexing into tables requires integral indices; therefore, the indexing coordinates need to be discretized.

Definition 8 *Base grid*, $\mathcal{B}$ refers to a k-dimensional grid in $\mathbb{R}^k$. The term *grid point* ($\lfloor \Pi(\hat{y}) \rfloor \in \mathcal{B}$) refers to the discretized independent coordinates $(\lfloor \frac{y_1}{\delta_1} \rfloor, \lfloor \frac{y_2}{\delta_2} \rfloor, \ldots, \lfloor \frac{y_k}{\delta_k} \rfloor)$. The term M refers to the maximum number of hypothesis sheets over any point on the base grid.

Definition 9 *Height function*, $(h(\lfloor \Pi(\hat{y}) \rfloor))$ formulates the dependent coordinate y_{k+1} as a function of the independent coordinates $\lfloor \Pi(\hat{y}) \rfloor$.

Definition 10 *Ordered hypothesis list* $(A_{\lfloor \Pi(\hat{y}) \rfloor})$ describes an ordering of the hypothesis sheets at a particular grid point $\lfloor \Pi(\hat{y}) \rfloor$ according to their dependent coordinates (refer Figures 3 and 4).

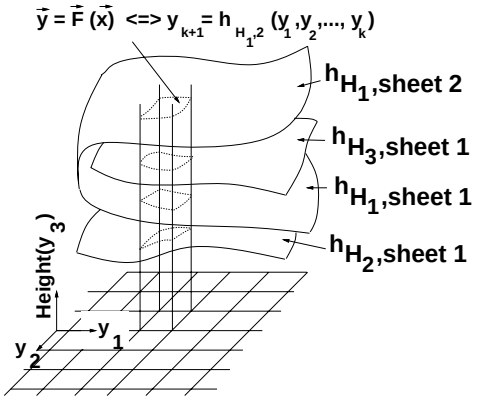

Figure 3: The hypothesis sheets above a particular base grid point $\lfloor \Pi(\hat{y}) \rfloor$ are ordered according to their dependent coordinates.

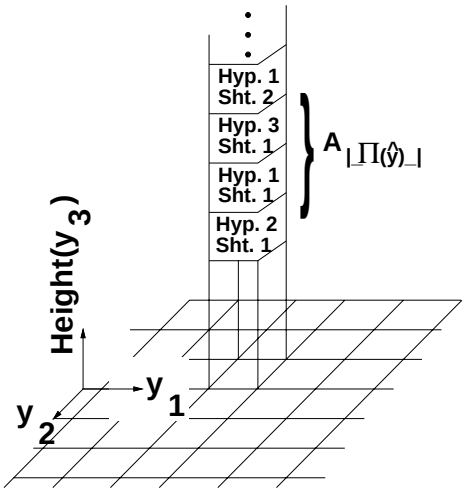

Figure 4: The ordered hypothesis list $A_{\lfloor \Pi(\hat{y}) \rfloor}$ of the hypothesis sheets above grid point $\lfloor \Pi(\hat{y}) \rfloor$.

4 Indexing Table Implementation

In this section, we describe a table indexing implementation technique, the tree grid, which reduces the storage size of the table and preserves order (spatial faithfulness). In the first steps, only $\overset{k+1}{\Pi}(\hat{y})$ is used. For each hypothesis sheet $< H_i, j >$ (j an sheet index of hypothesis H_i), $\Pi(\hat{y})$, the independent coordinates specify a unique configuration (refer equation (3), implying that there exists a function $\hat{F_H}^{-1} : \Pi(\hat{y}) \rightarrow \vec{x}$ (refer equation 4). Therefore, there also exists a height function $h_{H_i,j}$ which is the composition of $\hat{F_H}^{-1}$ and f_{k+1}, so that the 1 dependent coordinate y_{k+1} validates the hypothesis sheet and that configuration. The main idea is that we define ordered lists $(A_{\lfloor \Pi(\hat{y}) \rfloor})$ of the hypothesis sheets above all of the grid points $\mathcal{B}$. At runtime, the consistent hypotheses are determined by performing binary search on $A_{\lfloor \Pi(\hat{y}) \rfloor}$ with y_{k+1}.

$$y_1, y_2, \ldots, y_k \rightarrow (x_1, x_2, \ldots x_k)_j \quad (3)$$

$$\exists j \; s.t. \; y_{k+1} = f_{k+1}((x_1, x_2, \ldots x_k)_j) \quad (4)$$

$$y_{k+1} = h_{H_i,j}(y_1, y_2, \ldots, y_k) \quad (5)$$

4.1 Sketch of Algorithm

Given a n-tuple of indexing coordinates $\hat{y}$, we discretize k values ($\lfloor \Pi(\hat{y}) \rfloor$) to index an ordered hypothesis list ($A_{\lfloor \Pi(\hat{y}) \rfloor}$), and then search for y_{k+1} in that ordered list. This produces a subset of hypotheses consistent with $\overset{k+1}{\Pi}(\hat{y})$ which are then validated with respect to the entire observation $\hat{y}$.

4.2 Ordered Hypotheses Lists

This algorithm is based upon the ordered hypothesis lists $A_{\lfloor \Pi(\hat{y}) \rfloor}$ in order to exploit the coherence (slowly varying nature) of those hypothesis orderings. This strategy saves space because storing similar lists takes up less space than storing the lists individually (equation (6)). This strategy also provides spatial faithfulness.

$$\sum_{b \in \mathcal{B}} \min_{\nu \in neighbors(b)} |A_b - A_\nu| \ll \sum_{b \in \mathcal{B}} |A_b| \quad (6)$$

The only drawback in the tree grid approach is that it involves performing a binary search on the hypothesis sheets at runtime; the search time can be bounded ($O(\log(M))$) by using balanced trees (see Figure 4.2). Executing the search involves recomputing the heights $h_{H_i,j}(\lfloor \Pi(\hat{y}) \rfloor)$ of $\log(M)$ hypothesis sheets.

4.3 Constructing Space Efficient Ordering Trees

Definition 11 *Ordering trees* $(\mathcal{T}_p)$ describe the ordering A_p above a grid point p.

Ordering trees are generated in the following manner in order to share subtrees: $\mathcal{T}_{p_{child}}$, for hypothesis ordering $A_{p_{child}}$, is generated by performing a sequence of tree operations (change, insertion, deletion) on a *parent* tree ($\mathcal{T}_{p_{parent}}$) with a similar hypothesis ordering $A_{p_{parent}}$; each edit corresponds to one tree

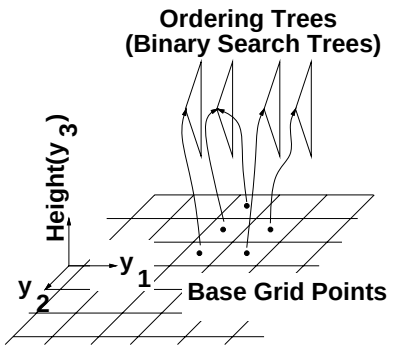

Figure 5: Height ordering trees above the base grid

operation. Notice that each tree operation only affects one path through the ordering tree, allowing both trees to share the other subtrees (see Figure 6).

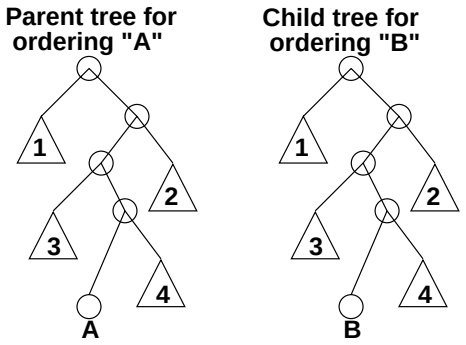

Figure 6: An ordering tree $\mathcal{T}_B$, for ordering B, shares many subtrees with $\mathcal{T}_A$, for ordering A, because A and B differ by only a single element. Describing $\mathcal{T}_B$ using $\mathcal{T}_A$'s subtrees involves a sequence of edit operations, where each edit operation requires $O(\log(M))$ nodes.

Ordering trees were implemented using persistent tree structures so that tree operations on $\mathcal{T}_{p_{parent}}$ left $\mathcal{T}_{p_{parent}}$ intact. Persistent AVL trees were used rather than a more specialized data structure [10] even though each AVL edit generates $O(\log(M))$ new nodes (compared to $O(1)$ new nodes/edit). The total number of nodes to store all of the ordered lists is approximately $O(e_\Sigma \log(M))$, where e_Σ is the total number of edits between the hypothesis orderings.

4.4 Indexing Algorithm

The algorithm determines the hypothesis sheet closest to a projected indexing coordinates $\overset{k+1}{\Pi}(\hat{y})$ in $O(\log(M))$ time.

1. Given $\hat{y}$, lookup the ordering tree $\mathcal{T}_{\lfloor \Pi(\hat{y}) \rfloor}$.
2. Determine the hypothesis sheet(s) consistent with the projected indexing coordinates $\overset{k+1}{\Pi}(\hat{y})$ by performing a binary search on the ordering tree $\mathcal{T}_{\lfloor \Pi(\hat{y}) \rfloor}$ using the dependent coordinate, y_{k+1}.
3. Validate each hypothesis using $\hat{y}$.

4.5 Generating the Ordering Trees $\mathcal{T}$

1. Compute the minimum edit distances between hypothesis orderings $A_b, A_{b'}$ of neighboring grid points $b, b' \in \mathcal{B}$.
2. Construct a graph G in which grid points are nodes and the edges between neighboring grid points are weighted by the minimum edit distances.
3. Compute the minimum spanning tree M_G of G.
4. Select an initial grid point p_{init} and construct an ordering tree $\mathcal{T}_{p_{init}}$ for hypothesis ordering $A_{p_{init}}$.
5. Perform depth first search on M_G starting with p_{init}. When expanding node p_{parent}, for all previously unexplored neighbors p_{child} construct the ordering tree $\mathcal{T}_{p_{child}}$ by performing a series of edit operations on $\mathcal{T}_{p_{parent}}$ determined by the minimum edit sequence between $A_{p_{child}}$ and $A_{p_{parent}}$.

4.6 Analysis

4.6.1 Tree Grid Construction Running Time

The majority of time constructing the trees is spent in step 2, computing all of the minimum edit distances between hypothesis orderings of neighboring grid points, and step 5, modifying trees using the minimum edit sequences between hypothesis orderings of neighboring grid points. Computing the minimum edit distance between two lists of length M takes Me time, where e is the number of edits. Computing all of the minimum edit distances for $k|\mathcal{B}|$ pairs of hypothesis orderings (we need to check k neighbors for each grid point) takes $k|\mathcal{B}| \times M\overline{e}$ time, where $\overline{e}$ refers to the average number of edits between hypothesis orderings ($|\mathcal{B}|\overline{e} = e_\Sigma$). Recomputing the minimum edit sequences for tree operations takes $|\mathcal{B}| \times M\overline{e}$ time. The algorithm's total time complexity is $O(|\mathcal{B}|kM\overline{e})$. The total number of nodes to describe the tree grid is $e_\Sigma \log(M)$ using AVL trees, or e_Σ using more specialized data structures. e_Σ is bounded by the number of different hypothesis orderings.

4.6.2 Bounds on the Number of Orderings

Next we derive an upper bound on the number of different height orderings for M hypothesis sheets above

the grid points. We invoke a theorem which says that M algebraic surfaces of constant degree in $\mathbb{R}^k$ produce an arrangement with $O(M^k)$ cells. This analysis depends upon the assumption that the intersection of each pair of hypothesis-sheets is a surface of constant degree in $\mathbb{R}^k$.

Orderings change in only three ways: removing a hypothesis sheet from the current ordering, inserting a hypothesis sheet into the current ordering, and interchanging the order of two hypothesis sheets. We assume that the majority of orderings result from interchanging the order of two hypothesis sheets.

Since we are only concerned with orderings above the base grid points, we can restrict our view to base lines including rows of grid points ($y_1 = r_1\delta_1, y_2 = r_2\delta_2 \ , \ldots | \ r_1, r_2, \ldots \in \mathbb{I}$) Consider the height functions of the models above each of these base lines as one-dimensional curves in the two-dimensional cross-section. Areas of constant ordering correspond to cells in the arrangement of these M height function cross-section curves. The theorem says that the number of cells in the arrangements, and thereby the number of edits, for M such curves is $O(|\mathcal{B}|M^2)$.

4.7 Efficient Performance Heuristics

4.7.1 Separating Hypotheses Sheets

Dividing the hypotheses sheets into l subsets shrinks the composite table because there are fewer edits, and each edit involves fewer nodes. If a single set contains all of the hypothesis sheets, then there are $O(M^2)$ possible different hypothesis orderings, but if the hypothesis sheets are separated into l subsets, then there are only $O(l(\frac{M}{l})^2)$ different hypothesis orderings. For a single set, each AVL edit involves $O(\log(M))$ nodes; for l subsets each AVL edit involves $O(\log(M) - \log(l))$ nodes. Dividing the hypotheses sheets into subsets increases the lookup time by a factor of l, since each subset must be individually checked.

4.7.2 Memoization

A naive implementation of tree operations generates subtrees for every new subsequence, *even if these subsequences already have been transcribed into trees.* The motivation for memoizing the subsequences is that memoization can only shrink the table size, and memoization takes very little time and space. Checking the balanced subtrees of a sequence with n elements takes linear time. Memoizing the hypothesis ordering subsequences results in table compaction by a factor of between three and ten.

5 Experiments and Results

In this section, we describe a model-based three-dimensional recognition technique from two-dimensional image data assuming unscaled orthonormal projection. Observations were generated from pairs of rays in the manner shown in Figure 7, an extension of the vertex pairs described by Thompson and Mundy [12]; image ray pairs are symmetric to groups of four image points. The rays parameterization implicitly normalizes out three of the five degrees of freedom of unscaled orthographic projection: absolute x and y position information is normalized because vertex v_1 is translated to the origin, absolute orientation is normalized because we assume that the first ray of v_1 is transformed to be parallel to the x-axis. In this manner, there are two normalized degrees of freedom ($k = 2$): θ_x, rotation around the x-axis, and θ_y, rotation around the y-axis. The scale factor could be normalized away by scaling v_1 to length 1.

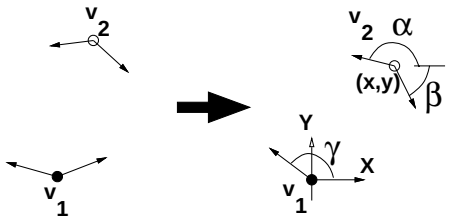

Figure 7: After normalizing the rays by rotating and translating vertex v_1 to the origin and one of its rays is aligned with the x-axis, the image ray parameterization is: x, y position of v_2, and α, β, γ, the orientations of the other rays relative to the x-axis.

For a given hypothesis (m_1, m_2), the four configurations $\{(\theta_x, \theta_y)_i\}$ can be extracted as follows:

1. Determine the two orientations $\theta_{x_1}, \theta_{x_2}$ which rotate the vertex m_2 to the correct y position $(v_2.y)$. Rotate m_2 by $\theta_{x_1}, \theta_{x_2}$ to predict m_2' and m_2''.

2. For m_2' and m_2'', determine the two orientations $\theta_{y_1}, \theta_{y_2}$ which rotate m_2' or m_2'' to the correct x position $(v_2.x)$.

The dependent coordinate can be any of the orientations of the other rays. Notice that five values $(2k+1)$ are observed. This should sufficiently differentiate the hypotheses and ensure a unique interpretation. The index table implementation was tested by using real model groups (*, the stapler in [3]) and random model groups, and then constructing the table for the respective hypothesis sheets. The size of the compacted tables, and the compaction ratio as a function of the

M	$\sqrt{\lvert\frac{1}{\mathcal{B}}\rvert}$ Grid Spacing	# Entries	# Nodes	Compaction
25*	$\frac{1}{100}$	1959460	16983	115.38
25*	$\frac{1}{150}$	4399560	18417	238.89
50	$\frac{1}{100}$	3520520	26808	131.32
50	$\frac{1}{150}$	7902660	28189	345.72
50	$\frac{1}{100}$	2852068	18639	153.02
75	$\frac{1}{100}$	4772700	82156	58.09
75	$\frac{1}{150}$	10631948	105477	100.79
100	$\frac{1}{100}$	6416292	200033	31.28
100	$\frac{1}{100}$	6564076	220021	38.97

Table 1: Compaction ratios for lookup tables of real (*) and randomly generated hypotheses. The hypothesis sheets are separated into eight subsets.

number of hypothesis, and resolution (grid spacing) is given in Table 1. This recognition technique was experimentally validated by interpreting groups of four image points as groups of stapler model vertices from data given in [3].

5.1 Discussion

Notice that increasing the resolution increases the compaction ratio, and increasing the number of hypotheses decreases the compaction ratio. The case of increasing resolution can be explained in the following way: if increasing the resolution does not significantly increase the total number of edits, e_Σ because most of the hypothesis orderings already appear at the coarser resolution, while increasing the resolution significantly increases the number of table entries, the compaction ratio increases. The decreasing compaction for larger hypothesis sets can be explained in the following way: the number of orderings increases as the square of the number of hypotheses; if there are more orderings, each grid point has a higher probability of being associated with a different ordering, requiring more edits. Since the table storage size depends upon the number of edits, the compaction ratio decreases with more hypotheses. Increasing the number of subsets counteracts increasing the number of hypotheses because separating the hypothesis sheets into more subsets improves compaction.

6 Conclusion

In this paper, we presented a indexing table technique, the tree grid, which combined space compaction with spatial faithfulness. This technique exploited the fact that $k+1$ observation values prune the set of consistent hypotheses; we also presented a proof on the minimum number $(2k+1)$ of generic observation values necessary to specify uniquely a generic hypothesis. The tree grid approach enables the user to trade-off storage space and lookup time via a binary search on the hypotheses. In some cases, this approach compacted the total storage size of the table by a hundred-fold. We presented experimental data for three-dimensional object recognition from two image rays.

Acknowledgements

The authors wish to acknowledge: Clark Olson, Ruth Rosenholtz, and Dieter Koller for helpful comments.

References

[1] Nicholas Ayache and Oliver D. Faugeras. Hyper: A new approach for the recognition and positioning of two-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):44–54, 1986.

[2] J. Brian Burns, Richard S. Weiss, and Edward M. Riseman. View variation of point-set and line-segment features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):51–68, 1993.

[3] D. T. Clemens and D. W. Jacobs. Space and time bounds on indexing 3-d models from 2-d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1007–1017, 1991.

[4] Richard Cole. Searching and storing similar lists. *Journal of Algorithms*, 7:202–220, 1986.

[5] D. Forsyth, L. Mundy, A. Zisserman, A. Heller, and C. Rothwell. Invariant descriptors fo 3-d object recognition and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:971–991, Oct 1991.

[6] M. Golubitsky and V. Guilleman. *Stable Mappings and Their Singularities*. Number 14 in Grad. Texts in Math. Springer Verlag, New York, 1973.

[7] D. W. Jacobs. Space efficient 3d model indexing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 439–444, 1992.

[8] Alan Kalvin, Edith Schonberg, Jacob T. Schwartz, and Micha Sharir. Two-dimensional model-based boundary matching using footprints. *International Journal of Robotics Research*, 5(4):38–55, 1986.

[9] Yehezkel Lamdan, Jacob T. Schwartz, and Haim J. Wolfson. Object recognition by affine invariant matching. *International Conference of Computer Vision*, 1988.

[10] Neil Sarnak and Robert E. Tarjan. Planar point location using persistent search trees. *Communications of the ACM*, 29(7):669–679, July 1986.

[11] Jacob T. Schwartz and Micha Sharir. Identification of partially obscured objects in three dimensions by matching noisy characteristic curves. *International Journal of Robotics Research*, 6(2):29–44, 1987.

[12] D. W. Thompson and J. L. Mundy. Three-dimensional model matching from an unconstrained viewpoint. *IEEE International Conference on Robotics and Automation*, 1:208–220, 1987.