Complexity of Indexing: Efficient and Learnable Large Database Indexing*

Michael Werman and Daphna Weinshall
Institute of Computer Science, The Hebrew University of Jerusalem
91904, Jerusalem, Israel
{werman,daphna}@cs.huji.ac.il

Abstract

Object recognition starts from a set of image measurements (including locations of points, lines, surfaces, color, and shading), which provides access into a database where representations of objects are stored. We describe a complexity theory of indexing, *a meta-analysis which identifies the best set of measurements (up to algebraic transformations) such that: (1) the representation of objects are linear subspaces and thus easy to* **learn***; (2) direct indexing is* **efficient** *since the linear subspaces are of minimal rank. The index complexity is determined via a simple process, equivalent to computing the rank of a matrix. We readily rederive the index complexity of the few previously analyzed cases. We then compute the best index for new and more interesting cases: 6 points in one perspective image, 6 directions in one para-perspective image, and 2 perspective images of 7 points. With color we get the following result: 4 color sensors are sufficient for color constancy at a point, and the sensor-output index is irreducible; the most efficient representation of a color is a plane in* 3D *space. For future applications with any vision problem where the relations between shape and image measurements can be written down, we give an automatic process to construct the most efficient database that can be directly obtained by learning from examples.*

1 Introduction

Any recognition system starts with a finite list of n measurements computed on the image (the list may include the grey-levels of the pixels). The vector of n measurements defines the indexing space, R^n. Each image corresponds to a point in R^n. Since objects appear in many different forms in different images (due to changes in viewing position, a change of illumination, etc), each object is mapped to a collection of points. This collection, the set of all points corresponding to a feasible image of the object, is the object's *picture manifold* in the *indexing space* .

The particular form of an object's *picture manifold* is very important for object recognition because:

Direct indexing: in the simplest approach each object's *picture manifold* is stored in the database. Now given a set of measurements computed from an image, which is a point in $\mathcal{R}^n$, we use the point as an index into a table, pointing to a cell where the identity of the observed object is written. An example of a recognition approach which uses this method is geometric hashing [6].

Search: If we have parametric representations of the picture manifolds, we can efficiently store those representations. Given a set of measurements computed from an image, we search the database to find the picture manifold which covers the measurements point in $\mathcal{R}^n$. An example of a recognition approach which uses this method is alignment [4].

The first approach is the most time-efficient since no search is required. However, clearly the form of the *picture manifold* is critical here: if it is too complex, it may not be feasible to represent this picture manifold in memory. On the other extreme, the second approach is space-efficient since only a parametric representation of the picture manifold is stored. However, recognition requires search, and may therefore take too long to be useful.

For an optimal system we probably need to consider a hybrid of the 2 approaches to object recognition. In order to do that, we need to better understand the form of objects' *picture manifold.* We need to understand what a *picture manifold* of classes of objects look like. Finally and most importantly, we need to be able to manipulate the measurements defining the indexing space, optimizing for simplicity and efficiency of the picture manifold at the same time. That is, we seek a transformation of the measurements such that:

Simplicity- the picture manifold is a linear subspace, a manifold which is both easy to represent and easy to learn from examples.

Efficiency- the picture manifold is of a low rank.

Indexing spaces where objects' picture manifolds are both simple and low-dimensional can be used in recognition systems which are based on either one of the approaches described above. Defining and computing good indexing spaces is the subject of the present paper.

1.1 Previous work:

The concepts we study here are closely related to the concept of invariants, which was a subject of much interest in computer vision and pattern recognition. To analyze the picture manifold, we need to start with a relation between the vector of measurements and the object. This relation should not depend on variables of the imaging process, such as the camera orientation (viewing position) or the color of the illumination source; thus we

*This research was sponsored by ARPA through the U.S. Office of Naval Research under grant N00014-93-1-1202, R&T Project Code 4424341—01.

need an invariant relation between image measurement and objects.

Invariant relations were divided in the literature into 2 types:

Model-free invariant: there exist image measurements which identify the object uniquely, so that their value is completely determined by the object regardless of the details of the imaging process. This is what is usually called an invariant in the literature. Such invariant relations do not exist for $2D$ images of general $3D$ objects [1, 8]; for special classes of objects, such as planar or symmetrical objects, invariant relations may be found [8, 11].

Model-based invariant: a relation which includes mixed terms, representing image measurements or model parameters. Model-based invariant relations exist for many interesting vision problems [14].

If a model-free invariant is used to construct the database (the indexing space), then the picture manifold of an object is a point. For example, geometric hashing [6] uses the class of planar objects composed of features, where model-free invariants exist and therefore the picture manifolds are points. If on the other hand a model-based invariant is used, the picture manifold may be a complex hypersurface in the indexing space. For example, Murase & Nayar [9] use a very general class of objects and illumination conditions, and therefore they have to represent picture manifolds which take on rather complex shapes.

Clearly, for object recognition using either direct indexing or search, it is better to use an indexing space where the picture manifolds are as simple as possible (for example, hyperplanes rather than quadrics), and have as low a rank as possible.

Jacobs [2, 5] defined the concept of indexing space, and studied the complexity of picture manifolds. In comparison: (1) Jacobs was not interested in the question of how to transform the indexing space to obtain a simpler picture manifold, which is the central question of our study. (2) Jacobs' rank computation method was specific to the problem at hand, objects composed of points and varying viewing position, whereas we seek an *automatic* method which is completely general and can in principle be applied to all picture manifolds. (3) Jacobs considered only picture manifolds obtained by change of viewpoint, and calculated the rank of the picture manifold for objects composed of sets of points only; he did not always compute the actual parametric representation of the picture manifolds, as we do here.

Wallack & Canny [13] assumed that the complexity of the picture manifold is given, and developed a database structure to implement a multi-dimensional index efficiently. Tomasi [12] used the phrase *picture locus* for what we call here picture manifold, while analyzing a special $2D$ case of perceived directions of features under perspective projection.

Many recognition systems that are based on direct indexing use databases where objects are represented by their manifold of images, i.e., all their images are stored. Typically there is a training phase where images are collected, and the manifold of images is approximated using a neural network [3], an RBF interpolation network [10], or a parametric representation [9]. In search-based recognition systems (e.g., [4]) the database contains not the manifolds themselves but their parametric representations, yet automatically obtaining those representations is as important as ever.

Up to now there was no study addressing the manipulation of the indexing space in such a way that manifolds of images are easier to learn; achieving this goal, obtaining linear manifolds that are easy to learn from examples while maintaining low rank (efficiency), is a major contribution of our present study.

1.2 Description of our approach:

We start from a general vision problem, with one or more cameras, where the image measurements include points, lines, or surfaces, with or without shading, color and texture. For each such problem we find an optimal set of transformed measurements, to define an indexing space where the picture manifolds are linear (hyperplanes) and of the lowest possible rank (among all linear picture manifolds). The lowest rank of the picture manifold is called the **index complexity** of the vision problem.

At the end we get the best linear relation between objects and image measurements. Objects are then represented by low rank hyperplanes[1]. We advocate the use of linear representations of objects even at a modest cost of space efficiency for the following reasons:

- Hyperplanes can be learned (or interpolated) robustly and efficiently from a small number of examples; this is NOT the case for general manifolds, where the minimal number of measurements needed is unknown in advance and potentially very large.

  Moreover, parametric representations of linear subspaces can be stored and used by search-based recognition approaches (like alignment); on the other hand, parametric representations of general manifolds may not even be computable (cf. [9]).

- Using noisy measurements, it is straightforward to find the distance of a measurement point from a hyperplane, which is the problem that an object recognition algorithm needs to solve. It is very hard to find the distance to a general manifold, and even the distance to a quadric surface cannot always be computed reliably (not to mention quickly).

Our approach allows us to do one more thing: find class constraints, on the permitted set of objects, such that the index efficiency is increased (and the rank of the picture manifold is decreased). With sufficient class constraints we can reduce the rank of the picture manifold to 0, achieving the most efficient indexing possible via model-free invariants. This was the goal of many studies [8, 11]; these studies gave examples of class constraints but did not offer a general way to generate them, like we do here.

In summary, our theory unifies the 2 main approaches to object recognition under the same theoretical roof. We give an automatic way of finding the optimal object representation for the benefit of both direct indexing and

[1] When non-linear picture manifolds are permitted, there may exist one whose rank is lower than the most efficient linear picture manifold. However, in most of the examples worked out below we obtained the minimal rank known while considering only linear picture manifolds; only in the example of 4 points under weak perspective projection was the rank compromised by 1 for the sake of simplicity and learnability.

search approaches by optimizing both the learnability and the indexing efficiency of the representation. In addition, we give an automatic way to generate all class constraints that would increase indexing efficiency.

2 Complexity of indexing

This section describes our indexing complexity theory, and is organized as follows: Section 2.1 defines the context, what a vision problem is. Section 2.2 reviews how a set of invariant relations can be obtained for a given vision problem. In Section 2.3 the concept of index complexity is formally defined and motivated, and its computation is discussed in Section 2.4. Vision problems with multiple invariant relations are discussed in Section 2.5.

2.1 A vision problem:

A vision problem, describing a set of images, is defined by 3 types of variables:

1. $\mathbf{T}$ denotes the set of parameters describing the unknown **imaging variables**, including camera transformation and color of illumination. For example, 2 images obtained under perspective projection require a 22-dimensional vector $\mathbf{T}$, 11 parameters for the camera transformation in each image.
2. $\mathbf{S}$ denotes the set of parameters describing the **object shape**, where shape denotes both geometrical and physical features (including color and shading). For example, 6 lines in 3-space require a 24-dimensional vector $\mathbf{S}$, 4 parameters for each line. A color requires 3 parameters, defining its RGB color code.
3. $\mathbf{u}$ denotes the set of **image measurements**. For example, 6 points in 1 perspective image require a 12-dimensional vector $\mathbf{u}$, 2 parameters for each point.

Each image provides a number of relations between these variables, of the form $F(\mathbf{T}, \mathbf{S}, \mathbf{u}) = 0$; for example, an image of 1 point provides 2 such relations, one for the x coordinate and one for the y. The vision problem is fully described by the set of all equations, relating these variables over the given images:

$$\mathcal{V} = \{F_j(\mathbf{T}, \mathbf{S}, \mathbf{u}) = 0\}_{j=1}^{L}$$

2.2 Database indexing:

Each vision problem $\mathcal{V}$ produces an ideal $\mathcal{I}$, which is the set of all the possible relations between the image measurables and the model shape. $\mathcal{I}$ is generated by a set of relations, obtained from the relations in $\mathcal{V}$ by eliminating the imaging variables $\mathbf{T}$. Thus:

$$\mathcal{I} = \{f_j(\mathbf{S}, \mathbf{u}) = 0\}_{j=1}^{l}$$

$l \leq L$. The ideal $\mathcal{I}$ includes precisely all true relationships between the model parameters $\mathbf{S}$ and the image measurements $\mathbf{u}$. The polynomials in $\mathcal{I}$ do not include the imaging variables $\mathbf{T}$. We will often not distinguish between the ideal and its set of generators. The ideal is the smallest set including the generators closed under addition and multiplication by polynomials in $\mathbf{S}$ and $\mathbf{u}$.

[15] describes a general elimination method to obtain the set of model-image relations from the vision problem using Gröbner bases (which is equivalent to Gauss elimination in linear systems). The space of model-image relations $\mathcal{I}$ is an ideal (equivalent to a linear subspace in linear systems). The dimension of the ideal, l, in some sense, describes the number of independent indices that can be used to point to the representation of the object in the database, given the context of the particular vision problem.

2.3 Complexity of indexing:

For simplicity, we assume that the set of possible model-image relations $\mathcal{I}$ of our vision problem $\mathcal{V}$ includes a single relation $I : f(\mathbf{S}, \mathbf{u}) = 0$. The number of relations is often chosen to be one, as this requires the smallest number of constraints in the vision problem so that the imaging variables can be eliminated, also taking a larger set equations can result in a combinatorial blowup. We define the complexity of I, a measure of the space and time needed to implement the database index obtained from it. As explained in the introduction, the complexity of an index reflects the size of the smallest hyperplane that needs to be stored in the database, so that the object can be retrieved by the index.

To begin with, we rewrite the invariant relation I in the following compact way, explicitly separating image variables $\mathbf{u}$ from shape variables $\mathbf{S}$:

$$f(\mathbf{S}, \mathbf{u}) = \sum_{k=1}^{r+1} g_k(\mathbf{S}) * h_k(\mathbf{u}) = 0 \tag{1}$$

where g_k and h_k are polynomial functions of the shape $\mathbf{S}$ and the image $\mathbf{u}$ respectively. We call (1) the canonical representation of $f(\mathbf{S}, \mathbf{u})$.

Definition 1: *[Index] The index, which includes all the image measurables in the invariant relation (1), is the r-dimensional vector:*

$$[\frac{h_1(\mathbf{u})}{h_{r+1}(\mathbf{u})}, \ldots, \frac{h_r(\mathbf{u})}{h_{r+1}(\mathbf{u})}] \quad \in \quad \mathcal{R}^r \tag{2}$$

The indexing space is therefore $\mathcal{R}^r$. The manifold of all the possible images of object $\mathbf{S}$ is the hyperplane of rank $(r-1)$ defined in (1).

In the simplest case where $r = 1$, we have:

$$(\mathbf{S}, \mathbf{u}) = h_1(\mathbf{u})g_1(\mathbf{S}) + h_2(\mathbf{u})g_2(\mathbf{S})$$

$$\Rightarrow \frac{g_2(\mathbf{S})}{g_1(\mathbf{S})} + \frac{h_1(\mathbf{u})}{h_2(\mathbf{u})} = 0 \Rightarrow \frac{h_1(\mathbf{u})}{h_2(\mathbf{u})} = -\frac{g_2(\mathbf{S})}{g_1(\mathbf{S})}$$

This means that the object $\mathbf{S}$ has a model-free invariant: the database is a 1-dimensional table, where each object is represented by a point (cell) in the table, and the cell's value is $-g_2(\mathbf{S})/g_1(\mathbf{S})$. The image provides a direct access to the table via $h_1(\mathbf{u})/h_2(\mathbf{u})$.

More generally, the index given in (2) is an element of $\mathcal{R}^r$. If $r = 1$ it is a model-free invariant; if $r > 1$ it is a model-based invariant as defined in [14]. The database now is an r-dimensional table, and a pointer to object $\mathbf{S}$ is stored in all the cells of the table that satisfy (1). In order to achieve efficient recognition, we would like to find a representation of the form (1) with the smallest r. Clearly $r = 1$ is the best case, since an object is represented most efficiently by a single point (rather than a manifold).

Definition 2: *[Index Complexity] The index complexity $\mathcal{C}$ of a relation $f(\mathbf{S}, \mathbf{u}) = 0$ is the rank of the smallest* *hyperplane that the relation defines. Thus if r is the smallest number of terms such that:*

$$f(\mathbf{S},\mathbf{u}) = \sum_{k=1}^{r+1} g_k(\mathbf{S}) * h_k(\mathbf{u}) = 0 \tag{3}$$

then $\mathcal{C} = r - 1$.
When the index ideal $\mathcal{I}$ of a vision problem has a single relation in it, $\mathcal{C}$ is also the index complexity of the vision problem.

The theorem in the following section gives a simple way to compute $\mathcal{C}$.

2.4 Computing index complexity:

Writing the polynomial $f(\mathbf{S},\mathbf{u})$ as a sum of multiplications, let

$$f(\mathbf{S},\mathbf{u}) = \sum_{i=1}^{n}\sum_{j=1}^{m} m_{ij} S_i u_j \tag{4}$$

where S_i and u_j are distinct products of element of $\mathbf{S}$ and $\mathbf{u}$ respectively.

Definition 3: *The* **complexity-matrix** *of a relation $f(\mathbf{S},\mathbf{u}) = 0$, denoted M, is the $n \times m$ matrix whose elements are m_{ij} from (4).*

Theorem 1: *The index complexity $\mathcal{C}$ of a relation whose minimal canonical representation has $r+1$ terms*

$$f(\mathbf{S},\mathbf{u}) = \sum_{k=1}^{r+1} g_k(\mathbf{S}) * h_k(\mathbf{u})$$

is equal to the rank of the complexity-matrix M minus 2.

Proof: The theorem follows from the fact that elementary operations on the rows and columns of a matrix are algebraic operations and that the rank of a matrix is the minimal number of outer products of vectors that sum to the matrix.

Algorithm to compute the complexity $\mathcal{C}$ and the corresponding index:

Complexity: compute SVD decomposition of $M = U\Sigma V^T$; the rank of M is equal to the number of non-0 elements in the diagonal matrix Σ, and $\mathcal{C} = \mathrm{rank}(M) - 2$.

Most efficient index: let

- U'_k denote the inner product of the k-th column of matrix U with the vector of monomials S_i from (4).
- V'_k denote the inner product of the k-th column of matrix V with with the vector of monomials u_j from (4).
- σ_k denote the k-th diagonal element of Σ.

$g_k(\mathbf{S})$, the k-th element of the parametric representation of the linear manifold of images, is $\sqrt{\sigma_k}U'_k$. $h_k(\mathbf{u})$, the k-th element of the index into this manifold, is $\sqrt{\sigma_k}V'_k$.

Although the rank is unique, the decomposition to index and parameterization is not; other decompositions can all be derived using the same SVD decomposition: The complexity matrix of any minimal-rank decomposition (of type (3)) can be written as $M = (U_1 H)(H^{-1}V_1^T)$, where $U_1 = U\sqrt{\Sigma}$ a $n \times (\mathcal{C}+2)$ matrix, $V_1 = \sqrt{\Sigma}V$ a $(\mathcal{C}+2) \times m$ matrix, and H any regular $(\mathcal{C}+2) \times (\mathcal{C}+2)$ matrix. Thus there are essentially $(\mathcal{C}+2)^2 - 1$ (-1 from homogeneity) independent complexity matrices.

2.5 Handling multiple indices

Up to now we considered the case where the ideal of indices $\mathcal{I}$, obtained via elimination from the vision problem $\mathcal{V}$, has a single index function in it. That is, the ideal is generated by a single polynomial. If the rank of the ideal is $l > 1$, l independent low-complexity indices can be computed from the image. The larger l is, the more tables are required for the vision problem. We don't have an efficient way to compute the relation in the ideal with minimal complexity other than a fairly computer intensive search; two such examples are described below.

3 Examples

In this section we work out specific examples, from rederivation of known results in Section 3.1, to the new results described in Section 3.2.

3.1 Reformulation of known results

$\mathcal{C} < 0$: symmetry

Bilateral symmetry is a property that can be identified in an image regardless of the shape of the object. All the lines connecting pairs of symmetrical points are parallel in the world, and are therefore projected to either parallel lines (under weak perspective or affine projection) or a pencil of lines (under perspective projection). These are properties that can be tested in the image directly, regardless of the particular symmetrical shape of the object.

Indeed, writing down the set of equations describing the perspective projection of 6 points, and adding the symmetry constraints on the coordinates of the 3 pairs of symmetrical points, we derive a single index $f(\mathbf{S},\mathbf{u})$. The rank of the complexity matrix of this index is 1. The most efficient relation does not differentiate symmetrical models from each other, and its index complexity is -1.

$\mathcal{C} = 0$: planarity

Images of planar objects are obtained by a projection from $\mathcal{R}^2$ to $\mathcal{R}^2$, and therefore model-free invariants exists in images of such objects. In particular, 5 points under perspective projection have one model-free invariant, the cross-ratio of the points. Therefore given a vision problem describing one image of 5 coplanar points, the index complexity would be 2.

Indeed, writing down the set of equations describing the perspective projection of 5 points, and adding the coplanarity constraint, we derive a single index $f(\mathbf{S},\mathbf{u})$. The rank of the complexity matrix of the index is 2. Thus the index complexity is 0, namely, the rank of the most efficient picture manifold is 0 (it is a point rather than a surface). The most efficient index is found to be the cross-ratio, which is known to be the model-free invariant of the problem.

3.2 New results

$\mathcal{C} = 3$: 6 points and a perspective camera

We took a vision problem of 6 points projected perspectively to a single image. We wrote down the set of equations describing the perspective projection of 6 points, deriving a single index $f(\mathbf{S}, \mathbf{u})$. We denote by x_i and y_i the image coordinates of the i-th point, $1 \leq i \leq 6$ (these are the elements of vector $\mathbf{u}$). The shape of the object is written in a projective invariant way: the first five points are chosen as the standard basis of the projective space $\mathcal{P}^3$, and the coordinates of the 6th point in this basis are denotes (p, q, r, s) (these are the elements of the vector $\mathbf{S}$).

The rank of the complexity matrix of the index is 5. In other words, the rank of the smallest picture manifold is 3. The minimal rank form of the relation $f(\mathbf{S}, \mathbf{u}) = 0$, or the description of the picture manifold as a hyperplane in $\mathcal{R}^4$, is the following:

$$Ind(\mathbf{u}) \cdot Model(p, q, r, s) = 1$$

where the parameterization of the hyperplane defining the picture manifold of the object is:

$$Model(\mathbf{S}) = \left[\frac{(pq - sr)}{(qs - rs)}, -\frac{(pr - sr)}{(qs - rs)}, \frac{(qr - rs)}{(qs - rs)}, \frac{(ps - rs)}{(qs - rs)}\right]$$

and the most efficient index $Ind(\mathbf{u}) \in \mathcal{R}^4$ is

$$\begin{aligned} Ind(\mathbf{u}) \quad = \quad & \Phi(1,3,5,2,4,6)^{-1}[\Phi(1,2,6,3,4,5), \\ & \Phi(1,3,6,2,4,5),\ \Phi(2,3,6,1,4,5),\ \Phi(2,3,5,1,4,6)] \end{aligned}$$

where

$$\Phi(i_1, i_2, i_3, i_4, i_5, i_6) = \begin{vmatrix} x_{i_1} & x_{i_2} & x_{i_3} \\ y_{i_1} & y_{i_2} & y_{i_3} \\ 1 & 1 & 1 \end{vmatrix} \begin{vmatrix} x_{i_4} & x_{i_5} & x_{i_6} \\ y_{i_4} & y_{i_5} & y_{i_6} \\ 1 & 1 & 1 \end{vmatrix}$$

It is interesting to compare this result with the literature. Jacobs [5] showed recently that for 6 points in a single perspective image, the lowest rank picture manifold is a non-linear manifold of rank 3 in $\mathcal{R}^4$. We obtained the same rank 3, while using a different 4-dimensional measurement vector (a different indexing space). In this transformed space, the picture manifold is of the same low rank of 3 (provenly the lowest rank possible), and yet it is linear with all the benefits that come from linearity. In addition, our result was obtained automatically and relatively effortlessly, as we are using general and not problem-specific tools.

$\mathcal{C} = 3$: 6 lines and a para-perspective camera

We took a vision problem of 6 lines projected para-perspectively (using an affine camera model) to a single image. We wrote down the equations relating the 6 unknown camera parameters, the 24 unknown parameters of the lines, and the 6 known projected directions in the image. Using elimination we obtained a single index $f(\mathbf{S}, \mathbf{u})$. The rank of the complexity matrix of the index is 5, thus the index complexity is 3.

For convenience of notation, we define the function $\Theta(a, b, c, d, e, f)$ as:

$$\Theta(a, b, c, d, e, f) = \begin{vmatrix} a & b & b \\ d & d & c \\ e & f & e \end{vmatrix} - \begin{vmatrix} a & a & b \\ c & d & c \\ e & f & f \end{vmatrix}$$

Using the analysis described above we computed the most efficient index (a 4-dimensional vector), which is the following:

$$\begin{aligned} I \quad = & \ \Theta(sl_1, sl_5, sl_4, sl_2, sl_6, sl_3)^{-1} \\ & [\Theta(sl_1, sl_2, sl_3, sl_4, sl_5, sl_6),\ \Theta(sl_1, sl_2, sl_3, sl_5, sl_4, sl_6), \\ & \ \Theta(sl_1, sl_3, sl_5, sl_4, sl_6, sl_2),\ \Theta(sl_1, sl_3, sl_4, sl_2, sl_6, sl_5)] \end{aligned}$$

where sl_i is the projected slope of the i-th line.

$\mathcal{C} = 2$: 2 perspective images of 7 points

We took a vision problem of 7 points projected perspectively to two images. We wrote down the equations relating the 22 unknown camera parameters, the 6 unknown parameters of the points (5 points define a projective basis, leaving only 6 unknowns), and the 14 known image coordinate. Using elimination we obtained a ideal of indices of rank larger than 1 (more than 1 invariant relation holds for this system). Continuing to search through this ideal, we found the minimal complexity linear manifold with a rank of 2.

$\mathcal{C} = 2$: The color at a point

Using the linear combination color model [7], the color of an object or a light source can be described as a linear combination of a set of basis color functions of wavelength λ. It has been argued that 3 basis functions can approximate well the color of natural objects and light sources. We use a model where the color spaces of objects and light sources are approximated by the span of 3 basis functions each. Our camera model has 4 different color sensors. Thus we have 4 measurements at each point (the output of the 4 sensors), and 6 unknowns - the 3 coefficients of the object color space and the 3 coefficients of the illuminant color space. With 4 equations we can eliminate the 3 illumination parameters, leaving us with a relationship between the measured colors (the sensors' output) and the reflectance properties (the color coefficients) of the object at a given point.

The index complexity of this relationship is shown to be 2, meaning that all the possible measured colors of a point (regardless of illumination color) sit in a plane in a 3-dimensional color space. The analysis also gives that there is no smaller subspace of the colors, which is insensitive to the illuminant color.

4 Adding class constraints

Once it has been shown that model-free invariants do not exist for unconstrained objects [1, 8], attention had turned to characterizing the constraints (or classes of objects) which would lead to model-free invariants [8, 11], or in our language, lead to index complexity 0. The present analysis allows us to do so directly, as part of a more general question: what class constraints on objects reduce the index complexity (and thus make their recognition more efficient)? We determine sufficient and necessary conditions on class constraints to reduce index complexity, in particular to reduce it to 0 (implying the existence of model-free invariants).

We start from a relation in the indices set

$$f(\mathbf{S}, \mathbf{u}) = \sum_{k=1}^{\mathcal{C}+2} g_k(\mathbf{S}) * h_k(\mathbf{u}) = 0$$

where $g_k(\mathbf{S})$ and $h_k(\mathbf{u})$ are polynomial functions of the shape and image measurements respectively. Every class constraint of the form: $\lambda = 0$ where λ divides some

$\sum \alpha_i g_k$ reduces the dimension of the best index by at least 1. Thus:

Theorem 2: *[class constraints:] To reduce the index complexity of a vision problem from $\mathcal{C}$ to $p < \mathcal{C}$, the class constraints should provide at most $(\mathcal{C} - p)$ independent constraints of the form λ_i, where each λ_i divides some $\sum \alpha_i g_k$ modulo the λ_j, $\quad j < i$.*

Clearly there is a trade off between index complexity, which is higher for more general (and less constrained) classes, and the density of the database, which is smaller for more general classes (as there are fewer types of such general objects).

Example: given 6 points and a perspective camera, the index complexity is 3 (see Section 3.2).

From the minimal-complexity invariant relation developed in Section 3.2, and the theorem above, it immediately follows that if any of the parameters of the 6th point, p, q, r, s, equals 0, the complexity goes down from 3 to 1; if any 2 parameters of the 6th point are equal, the complexity also goes down from 3 to 1. Thus *if 4 of the 6 points are coplanar, the index complexity of the system is 1.*

The case where 2 pairs of the 4 parameters p, q, r, s are equal has complexity 0, namely, there is a model-free invariant. The geometry of this case is as follows: one point lies on the line of intersection of 2 of the planes, each spanned by triplets of the remaining 3 points.

5 Discussion:

We described a theory of the complexity of database indexing, in the recognition of $3D$ objects from $2D$ images. Objects correspond to manifolds in the database, where a point on the manifold represents a possible image of the object. We first transform the manifolds to linear subspaces, and then compute the rank of the smallest linear subspace possible. This rank is the complexity of the index.

We discussed in the introduction the advantages of linear representations: they are easy to learn from examples, they can be stored in parametric forms, and noisy points can be reliably assigned to adjacent hyperplanes. By preferring linear representations, however, we risk a polynomial explosion in the rank of the database, which is a high price to pay. However, in all the examples we worked out it turned out that the best linear representation was no more complex in its rank than the best non-linear representation, with one exception where the rank increased by 1. This seems like a reasonable price to pay, since the advantages of linear representation far outweigh the disadvantages.

Our formalism is completely automatic. It gives us automatic tools to construct the most efficient database that can be directly obtained by learning from examples, regardless of whether we are using a direct indexing approach (like geometric hashing) or a search approach (like alignment). Moreover, since we optimize for the 2 approaches, a hybrid object recognition algorithm can be used with the final database:

Finally, we provide an automatic tool for finding class constraints which make the indexing problem easier. Finding class constraints that lead to model-free invariants (i.e., index complexity 0, cf. [8, 11]) is a special case of this general problem. Our approach is the first general method to automatically obtain such class constraints.

References

[1] J.B. Burns, R. Weiss, and E. Riseman. View variation of point-set and line segment features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(1):51–68, 1993.

[2] 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.

[3] S. Edelman and D. Weinshall. A self-organizing multiple-view representation of 3D objects. *Biological Cybernetics*, 64:209–219, 1991.

[4] D. P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5:195–212, 1990.

[5] D. W. Jacobs. Matching 3-D models to 2-D images. *International Journal of Computer Vision*, 1995. in press.

[6] Y. Lamdan and H. Wolfson. Geometric hashing: a general and efficient recognition scheme. In *Proceedings of the 2nd International Conference on Computer Vision*, pages 238–251, Tarpon Springs, FL, 1988. IEEE, Washington, DC.

[7] L. T. Maloney and B. Wandell. A computational model of color constancy. *Journal of the Optical Society of America*, 1:29–33, 1986.

[8] Y. Moses and S. Ullman. Limitations of non model-based recognition schemes. In G. Sandini, editor, *Proc. 2nd European Conf. on Computer Vision, Lecture Notes in Computer Science*, volume 588, pages 820–828. Springer Verlag, 1992.

[9] H. Murase and S. K. Nayar. Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, 1995. in press.

[10] T. Poggio and S. Edelman. A network that learns to recognize three-dimensional objects. *Nature*, 343:263–266, 1990.

[11] C.A. Rothwell, D.A. Forsyth, A. Zisserman, and J.L. Mundy. Extracting projective structure from single perspective views of 3d point sets. In *Proceedings of the 4th International Conference on Computer Vision*, pages 573–582, Berlin, Germany, 1993. IEEE, Washington, DC.

[12] C. Tomasi. Picture and trails: a new framework for the computation of shape and motion from perspective image sequences. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 913–918, June 1994.

[13] A. S. Wallack and J. F. Canny. Efficient indexing techniques for model based sensing. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 259–266, June 1994.

[14] D. Weinshall. Model-based invariants for 3D vision. *International Journal of Computer Vision*, 10(1):27–42, 1993.

[15] M. Werman and A. Shashua. Elimination: An approach to the study of 3D-from-2D. In *Proceedings of the International Conference on Computer Vision*, June 1995.