Indexing with Unknown Illumination and Pose

Ira Kemelmacher and Ronen Basri*
Dept. of Computer Science and Applied Math.
The Weizmann Institute of Science
Rehovot, 76100 Israel
{ira.kemelmacher, ronen.basri}@weizmann.ac.il

Abstract

The task of identifying 3D objects in 2D images is difficult due to variation in objects' appearance with changes in pose and lighting. The task is further complicated by the presence of occlusion and clutter. Shape indexing is a method for rapid association between features identified in an image and their corresponding 3D features stored in a database. Previous indexing methods ignored variations due to lighting, restricting the approach to polyhedral objects. In this paper, we further develop these methods to handle variations in both pose and lighting. We focus on rigid objects undergoing a scaled-orthographic projection and use spherical harmonics to represent lighting. The resulting integrated algorithm can recognize 3D objects from a single input image; furthermore, it recovers the pose and lighting of each familiar object in the given image. The algorithm has been tested on a database of real objects, demonstrating its performance on cluttered scenes under a variety of poses and illumination conditions.

1. Introduction

Shape indexing is a method of associating features found in an image to features of 3D models stored in a database. A key factor in indexing is the distinction between the offline preprocessing stage and the online recognition stage. In indexing, features are extracted from the objects and the image. Each set of model features is preprocessed by constructing entries into an *indexing table*. Then, at recognition time, collections of image features are used to compute indices to access the table. Using these indices, corresponding sets of model features are identified. The preprocessing of model sets is performed offline, making the online recognition stage as fast as possible (faster than a sequential scan of the database, as in *alignment* [6, 11]). A further advantage of indexing is its high inherent parallelism both in the preprocessing stage and in the recognition stage. Also, entries in the indexing table are constructed using small collections of features, so that each set accounts for relatively local scene information. This allows the method to overcome occlusion and clutter.

The most efficient way to perform indexing is by using functions that are invariant to transformations relating different views of an object [7, 13] (see [15]). In this case every collection of both model and image features gives rise to exactly one entry in the indexing table. However, due to loss of depth information as a result of projection, invariant functions cannot be used generically to identify 3D feature configurations in 2D images [4, 5, 14]. Nevertheless, indexing is still possible in this case by constructing indexing functions which map either model or image features to collections of entries (usually a line or a curve) in the indexing table [12, 22]. Other approaches, based on probabilistic inference and k-d tree search were proposed [2, 4, 17, 19].

Most existing indexing methods use point and line features, restricting their applicability to polyhedral shapes or to objects painted with prominent surface markings. The vast information contained in the intensities of objects is largely ignored by these methods. This not only severely restricts the type of shapes such methods can handle, but, as our experiments demonstrate, decreases their efficiency and reduces their performance. Only a few studies attempt to target smooth shapes (e.g., [20]), but this study too uses only the silhouette boundaries of objects. Below we present an attempt to increase the applicability of indexing to a much wider class of common objects. Our method handles variations in both pose and lighting and can handle both polyhedral and smooth shapes. It still relies on point features, but it only needs to locate very few of those, and it makes use of intensities to filter out incorrect matches.

There exists a sizable body of work that addresses the problem of recognition under unknown pose and lighting,

*Research was supported in part by the Israel Science Foundation grant number 266/02 and by the European Commission Project IST-2002-506766 Aim Shape. The vision group at the Weizmann Inst. is supported in part by the Moross Laboratory for Vision Research and Robotics.

Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
1063-6919/05 $20.00 © 2005 **IEEE**

which uses intensity information, particularly in the context of face recognition. These studies use manifold representations [16], statistical considerations [21], morphable models [3], or explicit lighting models (light fields [9], illumination cone [8], and representations based on spherical harmonics [1]). These methods scan a database of models sequentially. When these methods are applied to faces, pose can be compensated simultaneously for all models, exploiting the common location of face features. But when these methods are applied to large databases of general objects the search for the appropriate pose and lighting parameters may be prohibitive.

The indexing scheme we present in this paper identifies general objects in real, cluttered scenes efficiently and accurately under a wide range of poses and illumination conditions. Our scheme is based on a combination of reflectance and geometric properties of objects. In particular, we incorporate the spherical harmonics representation of lighting [1, 18] in an indexing scheme based on the 3D to 2D matching algorithm of [12]. Our algorithm has been tested successfully on a database of real 3D objects, recognizing the objects in a variety of scenes. We present these experiments and compare our results to results obtained using a pure geometric approach.

2. Image formation

For our method we need to model the image formation process as a function of pose and lighting. For pose, we assume that images of an object are formed by applying an arbitrary rigid transformation to the object and projecting it using the weak-perspective projection. In particular, an object can be rotated, translated and scaled.

For light, we use an analytically derived representation of the images produced by a convex Lambertian object illuminated by distant light sources [1, 18]. According to these derivations, the set of images of a convex Lambertian object obtained under arbitrary lighting conditions can be approximated accurately by a low dimensional linear subspace (4 or 9 dimensional). These results were accomplished by expressing lighting in terms of spherical harmonics and representing the operation of reflection as the analog of a convolution of the lighting with the clamped-cosine function, which is called the *Lambertian kernel*. Specifically, any image of an object can be described as: $\mathbf{I}^T = \mathbf{l}^T\mathbf{S}$ where $\mathbf{I}$ is an n-dimensional column vector containing the intensity of each pixel, $\mathbf{l}$ is an r-dimensional vector (r = 4 or 9, depending on the approximation order) describing the low order components of lighting, and $\mathbf{S}$ is $r \times n$ matrix whose rows each describe an image the object produces when lighting consists of a single spherical harmonic function. The superscript T represents the transpose operation. For a first order approximation to lighting, $\mathbf{S}$ is $4 \times n$ with each column containing the albedo of a pixel, and the three components of the surface normal at the pixel scaled by the albedo, i.e., $\rho, \rho n_x, \rho n_y, \rho n_z$. If we take a second order approximation to lighting, $\mathbf{S}$ is $9 \times n$, where the first four components of each column are the same as for the first order and the other five components are: $\rho(3n_z^2 - 1)$, $\rho n_x n_z$, $\rho n_y n_z$, $\rho(n_x^2 - n_y^2)$, $\rho n_x n_y$. (Normalization factors due to the spherical harmonics are omitted from $\mathbf{S}$. As a consequence $\mathbf{l}$ contains the low order harmonic coefficients of lighting scaled by these normalization factors.)

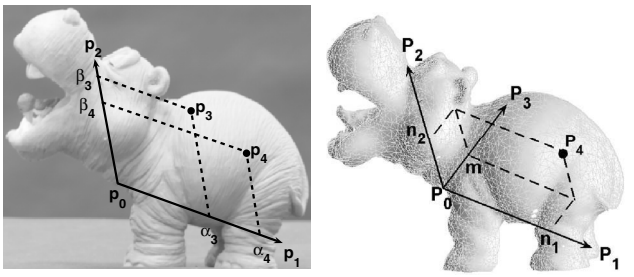

Figure 1. The affine coordinates of 2D points $\mathbf{p}_3$ and $\mathbf{p}_4$ (left) and 3D point $\mathbf{P}_4$ (right). Note that the basis vectors in an affine frame need not be orthonormal.

3. Indexing with pose

In this section we describe how we build an indexing space to match sets of image features to corresponding sets of objects features. We will use an indexing table based on the work of Jacobs [12], which assumes an affine projection model. Jacob's scheme is described in Section 3.1. We introduce certain modifications to this scheme in Section 3.2. Finally, we remove nonrigid configurations using a rigidity test due to Weinshall [22] (Section 3.3). Throughout this section we assume that feature points on both an image and a model are available to us by the application of some feature detector.

3.1. Indexing with affine projection model

For the construction of the indexing table we allow a more general model of projection - the affine projection model. In this model 2D image points are produced by applying an arbitrary 2×3 rank 2 linear transformation $\mathbf{A}$ to 3D object points followed by a translation $\mathbf{t}$ (2-dimensional vector). Explicitly, denote a collection of model points by $\mathbf{P}_0, \mathbf{P}_1, ..., \mathbf{P}_n$ and their corresponding image points by $\mathbf{p}_0, \mathbf{p}_1, ..., \mathbf{p}_n$, then $\mathbf{p_i} = \mathbf{AP_i} + \mathbf{t}$. The affine projection model significantly simplifies the indexing spaces (relative to a rigid transformation), but also adds additional degrees of freedom to the model, which may result in false positive

Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
1063-6919/05 $20.00 © 2005 **IEEE**

matches. We show later on in this section how to remove these false positives.

In an affine projection model any four model points can produce any four image points. So for indexing, model and image sets must each consist of at least five points. As the model lies in 3D, we use the first four (non-coplanar) points to define an affine basis, and describe the location of the fifth point using affine coordinates derived with respect to this basis. That is, we remove the translation by fixing the first point as an origin of coordinate frame: $\mathbf{Q}_i = \mathbf{P}_i - \mathbf{P}_0$ and then find the affine coordinates of the fifth point by

$$\mathbf{Q}_M = \begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2 & \mathbf{Q}_3\end{bmatrix}^{-1}\mathbf{Q} = \begin{bmatrix}0 & 1 & 0 & 0 & n_1\\ 0 & 0 & 1 & 0 & n_2\\ 0 & 0 & 0 & 1 & m\end{bmatrix}, \quad (1)$$

where the 3×5 matrix $\mathbf{Q} = [\mathbf{Q}_0, \mathbf{Q}_1, ..., \mathbf{Q}_4]$. We apply a similar procedure to the image set. We use the first three points to define a basis, and describe the location of the forth and fifth points using their affine coordinates with respect to this basis. Explicitly, we let $\mathbf{q}_i = \mathbf{p}_i - \mathbf{p}_0$ and the 2×5 matrix $\mathbf{q} = [\mathbf{q}_0, \mathbf{q}_1, ..., \mathbf{q}_4]$. So that

$$\mathbf{q}_I = \begin{bmatrix}\mathbf{q}_1 & \mathbf{q}_2\end{bmatrix}^{-1}\mathbf{q} = \begin{bmatrix}0 & 1 & 0 & \alpha_3 & \alpha_4\\ 0 & 0 & 1 & \beta_3 & \beta_4\end{bmatrix}. \quad (2)$$

Figure 1 illustrates the affine frames constructions. Due to the special structure of $\mathbf{Q}_M$ and $\mathbf{q}_I$ (which include the identity as sub-matrices) they are related by a 2×3 matrix:

$$\mathbf{q}_I = \begin{bmatrix}1 & 0 & a\\ 0 & 1 & b\end{bmatrix}\mathbf{Q}_M, \quad (3)$$

for some scalars a and b, and therefore

$$\begin{bmatrix}\alpha_3 & \alpha_4\\ \beta_3 & \beta_4\end{bmatrix} = \begin{bmatrix}1 & 0 & a\\ 0 & 1 & b\end{bmatrix}\begin{bmatrix}0 & n_1\\ 0 & n_2\\ 1 & m\end{bmatrix}. \quad (4)$$

Elimination of a and b results in two lines

$$\begin{cases}\alpha_4 = \alpha_3 m + n_1\\ \beta_4 = \beta_3 m + n_2.\end{cases} \quad (5)$$

These two line equations are independent and describe all image parameters that five model points can produce. Image parameters form a 4-dimensional $(\alpha_3, \alpha_4, \beta_3, \beta_4)$ affine space, which can be divided into two orthogonal subspaces, an α space, and a β space. An image with five ordered points is mapped into a point in these spaces. So matching an image set to a model set is reduced to matching a pair of points in two 2-dimensional spaces (representing the image set), to a pair of parallel lines in these spaces (representing the model set). Notice that in the special case that all the model points are coplanar, the affine coordinates of the projected model points are invariant to pose, and each model point beyond the first three is represented by a point in affine space. So in this case matching an image set to a model set is reduced to matching a point to a point.

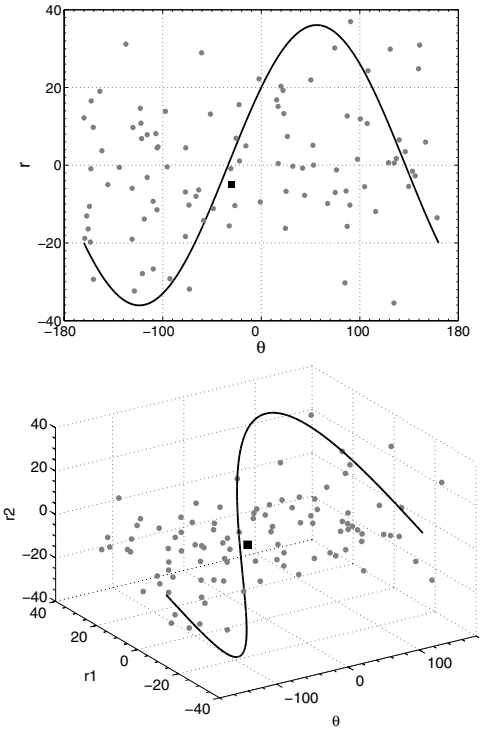

Figure 2. 2D (top) and 3D (bottom) indexing tables. An image set is represented by the sinusoid and its corresponding model set by the black square. The gray dots represent other model sets stored in the indexing tables.

3.2. Modifications

In our implementation we chose, unlike in [12], to modify the indexing table so as to represent a model set by a point and an image set by a curve. Such a representation is space efficient since we no longer need to place a pointer to a model in every cell intersected by its line. In particular, this allows us to represent the spaces by a linked list or sparse matrices. The disadvantage of this representation is that during the online recognition stage every set of image features will require accessing the table in many cells to extract all the possible models that can produce this set. However, the critical factor for the complexity of the method is the number of model sets retrieved in this process, and this number is the same whether we choose to represent a model set in the indexing table by a pair of points or by a pair of lines.

A line can be represented by its shortest (signed) distance from the origin (denoted by r) and its orientation (denoted by θ). From this we can obtain equations for the lines representing a model set, equivalent to (5),

$$\begin{cases}r_1 = \alpha_3 \cos\theta + \alpha_4 \sin\theta\\ r_2 = \beta_3 \cos\theta + \beta_4 \sin\theta.\end{cases} \quad (6)$$

By combining these two line definitions we can derive an expression for r_1, r_2 and θ in terms of n_1, n_2 and m:

$$r_i = \frac{n_i}{\sqrt{m^2+1}} \qquad \theta = \arctan(-\frac{1}{m}) \tag{7}$$

(where $i = 1, 2$). Consequently, a model set is represented by a pair of points in (θ, r_1) and (θ, r_2) index spaces, and an image set is represented by a pair of sinusoids in these spaces. Hence, matching between them is reduced to matching between points and sinusoids.

There are now several ways to implement indexing. In preprocessing, for each model set of each object we compute θ, r_1, and r_2. We then place a pointer to the set in the (θ, r_1) cell in the α table and the (θ, r_2) cell in the β table. For recognition, given an image set, we access the two tables tracing a sinusoid path (see the 2D table in Figure 2) to retrieve all the relevant model sets (possibly allowing for small errors). Subsequently, we then intersect the two lists retrieved to obtain all the model sets that are consistent with our image set.

As our experiments indicate (Section 7) the number of model sets retrieved by the α and β tables alone can be very large, and so tracing the two sinusoid curves to retrieve these sets can be prohibitively time consuming. Fortunately, intersecting the two lists dramatically reduces the number of candidate sets. A simple modification allows us to trace only one of the two tables. For each model set in the (θ, r_1) table we can store the value of r_2, and so we can immediately eliminate model sets that are incompatible in (θ, r_2) already while we trace the α table. A more significant saving is obtained if we choose to represent both indexing tables in a single, 3D table (θ, r_1, r_2). In this case (see the 3D space in Figure 2) every model set produces a single point in this space, whose coordinates are given by (7). An image corresponds to a 1D curve in this space of the form given by (6). Tracing this curve will allow us to retrieve *only* those model sets that are compatible with both the α and the β space simultaneously. This will result in a significant reduction of complexity, as is demonstrated in Section 7. We further disregard sets with $r_1 \approx 0$ and $r_2 \approx 0$ since these are usually non-discriminative.

It is important to note that for each set of model points we store in the indexing table all the orderings of these points. Hence, for recognition we will be able to retrieve the corresponding model sets using any one of these orderings of the corresponding image set. This will further reduce the runtime complexity by a factor of $5! = 120$.

3.3. Removal of nonrigid configurations

Each set of image feature points is associated now with a list of matching sets of model points. To eliminate the false positive matches that are due to the affine projection model assumption we use the *inverse Gramian* test proposed by Weinshall [22]. We calculate the inverse Gramian matrix $\mathbf{B}$ of three basis points of a model set $\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3$ (after removing the translation by fixing the origin):

$$\mathbf{B} = \begin{bmatrix} \mathbf{Q}_1^T\mathbf{Q}_1 & \mathbf{Q}_1^T\mathbf{Q}_2 & \mathbf{Q}_1^T\mathbf{Q}_3 \\ \mathbf{Q}_1^T\mathbf{Q}_2 & \mathbf{Q}_2^T\mathbf{Q}_2 & \mathbf{Q}_2^T\mathbf{Q}_3 \\ \mathbf{Q}_1^T\mathbf{Q}_3 & \mathbf{Q}_2^T\mathbf{Q}_3 & \mathbf{Q}_3^T\mathbf{Q}_3 \end{bmatrix}^{-1}. \tag{8}$$

The elements of $\mathbf{B}^{-1}$ contain all the 3D information on the geometry of the four basis points, their angles and lengths, so that $\mathbf{B}$ is invariant to rotations of the coordinate system but not to general linear transformations. Weinshall [22] showed that model and image sets that satisfy the relation

$$\frac{|\mathbf{x}^T\mathbf{B}\mathbf{y}| + |\mathbf{x}^T\mathbf{B}\mathbf{x} - \mathbf{y}^T\mathbf{B}\mathbf{y}|}{|\mathbf{x}|\|\mathbf{B}\||\mathbf{y}|} = 0 \tag{9}$$

($\mathbf{x}$ and $\mathbf{y}$ are 3-dimensional vectors denoting the x and y coordinates of the three corresponding basis points in the image) are consistent geometrically. To overcome noise this expression is allowed to deviate slightly from zero by applying a small threshold. The expression is normalized by the denominator so that its value is insensitive to uniform scaling of the object.

4. Consistency with lighting

The indexing scheme described until now provides us, for every collection of image features, candidate sets of model features whose location is consistent with the image collection. We next wish to test which of these candidate correspondences could produce the intensity patterns observed in the image. This will allow us to remove many false correspondences and identify the object accurately.

We use the linear relation described in Section 2. Given a collection of model points and their corresponding image points we construct for the model the matrix $\mathbf{S}$, which contains the unscaled harmonic images of these points. Denote the intensities of the image points by a vector $\mathbf{I}$, then $\mathbf{I}^T = \mathbf{l}^T\mathbf{S}$. To test consistency with lighting we measure the distance between the vector $\mathbf{I}$ of image intensities and the space spanned by the harmonic images (the rows of $\mathbf{S}$) normalized by the squared norm of $\mathbf{I}$

$$\frac{\|\mathbf{I} - \mathbf{S}\mathbf{S}^+\mathbf{I}\|^2}{\|\mathbf{I}\|^2 + \varepsilon}, \tag{10}$$

where $\mathbf{S}^+$ denotes the pseudo inverse of $\mathbf{S}$, $\mathbf{S}^+ = (\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T$. The small scalar ε is added in the denominator to account for zero intensities. This measure is small for true matches, and generally high for false ones.

Note that since the harmonic images $\mathbf{S}$ form an r-dimensional linear subspace ($r = 4, 9$ depends on the approximation order), any r model points can produce any r

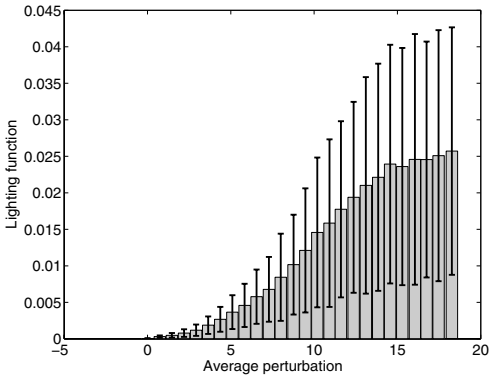

Figure 3. Illumination indexing measure (10) as a function of error in point location. The plot shows mean values over 1000 runs for an average error in point location in all directions. Error bars denote standard deviations.

image intensities. So the function requires sets of at least $r + 1$ points to be of any use.

To test the robustness of the measure to small perturbations in the location of points we have run an experiment in which we tested the measure under various displacements. This is important because of inaccuracies of the feature detection process. A typical behavior of the measure is shown in Figure 3. We plot the behavior of the indexing function up to almost 18 pixels average displacement (the length of the object in this experiment was 450 pixels). We observe that the minimal value is obtained at zero displacement, and the measure slowly increases as the displacement grows. An average perturbation of more than 5 pixels can be considered as an incorrect selection of points. Indeed, by testing the same model set with intensities of points acquired with large perturbations we get much larger values. This means that our illumination test, with high probability, is able to distinguish between true and false matches of image and model sets even with small errors in feature location.

A straightforward approach to incorporating illumination variations is to test using (10) whether the intensities of the feature points extracted from the image can be produced by the normals and albedos of the corresponding feature points on the model. Testing the intensities of feature points, however, is problematic since feature points often lie on the boundaries of the object or near corners, and so the surface orientation near the feature may change rapidly. In addition, some feature points may lie in dark spots on the object which are created by deep concavities or dark markings on the object. Such locations may appear dark under a wide variety of illumination conditions, and so these points will not carry much information regarding the lighting.

To overcome this problem we associate with every set of model feature points a list of several dozens additional points that are selected at random within smooth portions of the surface. (We select these points roughly inside the convex hull produced by the five basis points.) We then store their affine coordinates with respect to the model basis points, together with their harmonic basis. During the recognition stage, when candidate correspondence between model and image feature sets is found, we access these affine coordinates in the image and test if the intensities found in these locations satisfy the lighting consistency test (10). To avoid degeneracies we store in the indexing table only sets of such 'smooth' points for which the condition number of their matrix $\mathbf{S}$ of harmonic images, is below a certain threshold. If the harmonic images are linearly dependent, their condition number tends to infinity and the test is usually non-discriminative. This procedure can be performed at the preprocessing stage since it depends only on the harmonic images which are functions of the albedos and normals of the model points.

Sets of points that pass the illumination test are then used to vote for their respective model. Finally all models receive scores. The scores reflect the fraction of image sets for which a model appears as minimum. To account for accidental minima we exclude those image sets for which their minimum exceeds a certain threshold. Once a model is selected by this voting procedure its corresponding subsets are used to determine its pose and lighting. We recover these parameters using a *Robust Estimation* technique. In particular, we solve for pose and lighting for each image set, obtaining a parameter vector. Then we iterate by removing in each iteration a quarter of the vectors whose parameters are most distant from their median. Finally, we are left with the transformation that accounts for the majority of the image sets. We use this transformation to render the model under the same conditions as in the query image. In case the query image contains several objects we divide the image into a collection of overlapping squares and check which model obtained the largest voting score in each square. The shapes that receive the largest number of votes are then rendered and verified against the query image.

5. Summary of the algorithm

Preprocessing: For each model object:

1. Extract feature points.
2. For each ordered set of five model features:
   - compute the θ, r_1, r_2 coordinates (§3.2).
   - calculate the normalized inverse Gramian matrix $\mathbf{B}/\|\mathbf{B}\|$ (§3.3).
   - select points on smooth sections on the surface of the object and compute their affine coordinates and harmonic images $\mathbf{S}$ (§4).
3. Use θ, r_1, r_2 coordinates to place pointers to the corresponding cells in indexing spaces.

Recognition: Given an image:

1. Extract feature points from the image.
2. For some sets of five image features:
   - compute the affine coordinates $\alpha_4, \alpha_5, \beta_4, \beta_5$ (§3.1).
   - calculate the curve $(r_1(\theta), r_2(\theta))$ and retrieve candidate model sets from cells intersected by this curve (§3.2).
   - apply the inverse Gramian test (§3.3).
   - locate the 'smooth' points in the image (using the affine coordinates calculated in preprocessing).
   - use the illumination consistency test to obtain a score for each match (§4).
3. Vote for the best model candidates.

Postprocessing

1. Recover the pose and light for the model that received the largest number of votes over all image sets.
2. Render the output image.
3. Compare this image to the query image.

6. Experimental results

To test our method we have constructed a database of eight real objects. Each model includes a point cloud and surface normals acquired using a 3D laser scanner. The shapes are presented in Figure 4. To collect model feature points we photographed the objects, located feature points on the images and back-projected them to the models. To estimate the albedo, we averaged several images of the same object taken at the same pose and with different lightings. Feature points were collected automatically using the Harris feature detector [10].

Figure 5 shows the results obtained with our algorithm on four input images (each column represents a different image). For each result we present, from top to bottom, the original input image, the same image with image points (both feature and 'smooth' points) painted in colors to indicate the models to which they vote, the output image with the winning objects rendered under the recovered pose and lighting and a difference between the input and rendered images. The images were photographed under various natural poses and lighting conditions and with clutter and occlusion. Specifically, the first three images were pictured outdoors with the first two pictures taken at night with the objects illuminated by street light and the third pictured in daylight. The last image was pictured inside an office with florescent lighting.

Large concentrations of points that vote for the correct model are seen in regions where an object appears. The rendering shows that roughly the correct pose and lighting were recovered. The rendered intensities (we used 9 harmonics in these examples, although 4 harmonics essentially resulted in similar performance) matched the actual intensities to 80-88%. These accuracies are somewhat inferior to the accuracies derived theoretically [1, 18] due to some specularities and cast shadows.

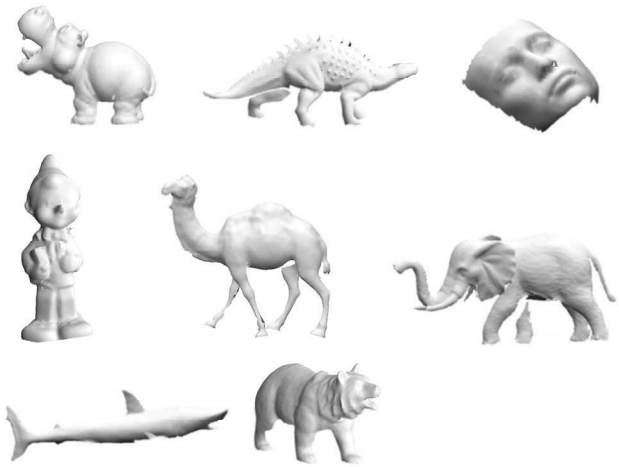

Figure 4. Database of 3D Models

To demonstrate the importance of the illumination consistency test, we compare our method to a geometric approach that uses only the location of feature points in Table 1. We show results obtained with three methods: "Affine", "Gramian", and "Light" (our method). For the first method we calculate the fraction of image sets, for each model, that received the maximum number of matchings. In this approach the model that received the maximum number of matchings is the first to be considered by the verification procedure. In the second method, the voting is performed according to the values of the inverse Gramian function (9). In particular, each image set votes for the model candidate that received the minimal measure value. The third method is our method where the lighting information is used. The second and third methods were tested with different thresholds (the thresholds applied to remove accidental matches). We can see that the illumination test applied in our method greatly improves the results of indexing compared to the geometric methods.

7. Complexity

We next turn to calculating the complexity of our algorithm. Denote by n the number of image feature points, by m the number of object feature points and by M the number of objects in the database. Also, note that each image or model set consists of five feature points and let c denote the

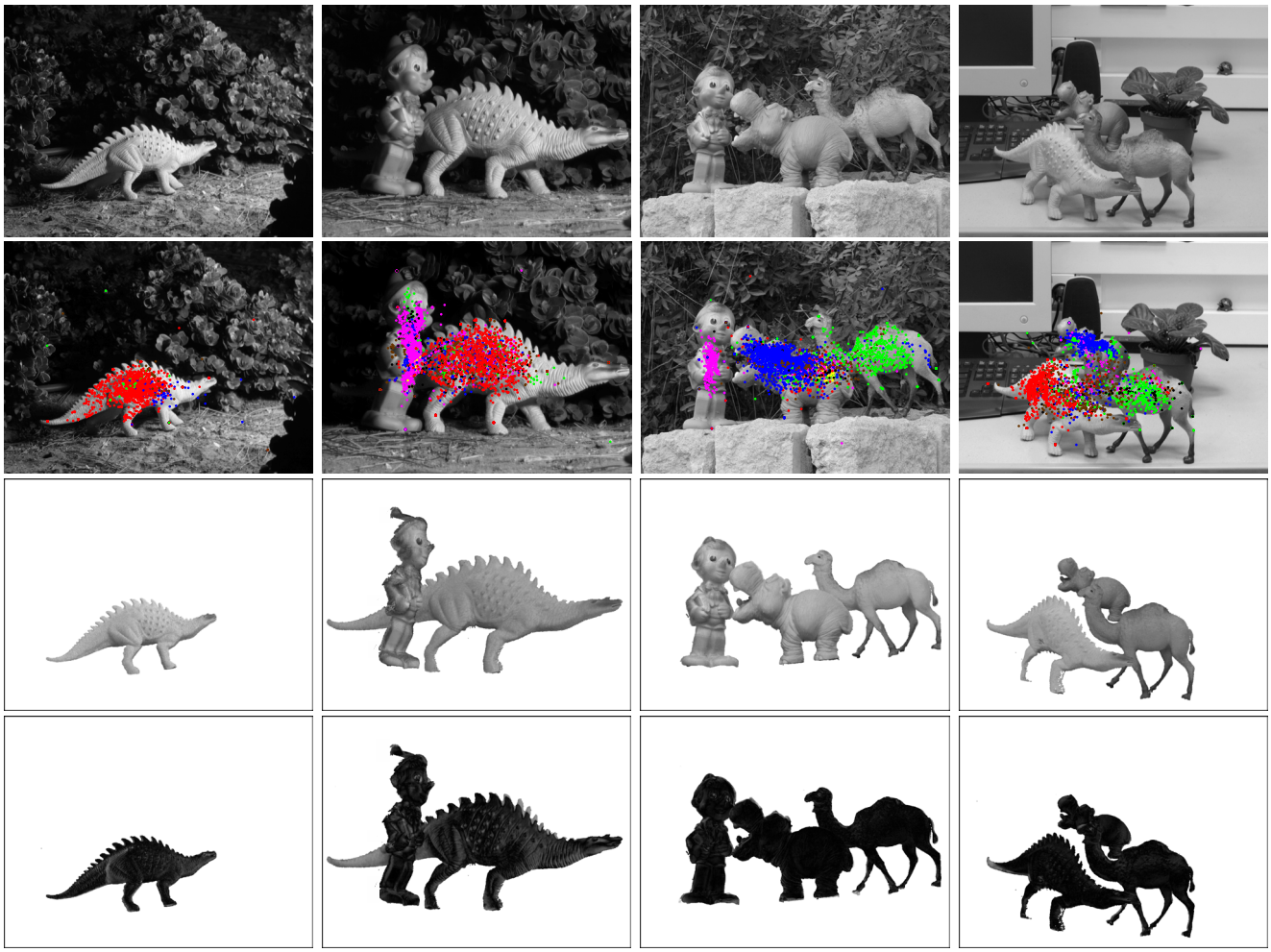

Figure 5. Example of four different query images and the output of our algorithm (each column is a different query). From top to bottom: The query image. The point sets that passed through all the tests colored to indicate the model to which they vote. These sets are used to recover the pose and lighting parameters. Rendered images using the recovered pose and lighting, and the difference between the input and rendered images. (Color index: dinosaur - red, hippo - blue, camel - green, pinokio - magenta, bear - yellow, elephant - black, shark - brown, face - cyan.)

average number of matched queries per image set.

At preprocessing, we represent all possible orderings of each model set in the indexing table. Thus the number of sets to be processed at the preprocessing stage is $M\binom{m}{5}5!$. At runtime, we consider some of the quintuples of feature points in the image. Suppose that the number of image sets is $\binom{n}{5}$ - all possible sets with no permutations. In addition, we have to take into account the number of model sets extracted by the indexing procedure for each image set. Hence, the total number of queries, at runtime, are $\binom{n}{5}c$.

In Table 2 we present the average number of sets matched per image set during each stage of the algorithm, and the fraction of correct image sets remained. This experiment was performed on the leftmost query image in Figure 5, where $M = 8$, $m = 17$ and $n = 35$. The total number of sets stored in the indexing table was $M\binom{m}{5}5!$ (almost $6,000,000$). Using the α-table almost $50,000$ model sets were retrieved for every image set. The β-table resulted in a similar number of sets. Intersecting the α and β candidates reduced the number of matches considerably, justifying the use of a 3D indexing table (Section 3.2). The subsequent inverse Gramian and lighting tests further reduce the number of potential matches. Only a few of the correct candidates were eliminated in this process (right column).

Our implementation performed the offline preprocessing stage preparing the database of eight objects in just 35 seconds. For recognition, the critical factor is the number of model sets retrieved from the indexing table. Our implementation performed the recognition step in 4 seconds for $n = 22$ feature points and 80 seconds for $n = 40$. In our

	Affine	Gramian		Light	
Thresh	Naive	0.04	0.00002	0.1	0.02
No. Sets	323,317	18,578	98	2,143	64
Dinosaur	0.13	0.06	0.31	0.25	0.91
Hippo	0.14	0.11	0.21	0.1	0.06
Camel	0.05	0.12	0.29	0.17	0.02
Pinokio	0.06	0.08	0.05	0.08	0
Bear	0.01	0.35	0.03	0.16	0
Elephant	0.42	0.1	0.07	0.09	0
Shark	0.1	0.08	0.02	0.1	0.03
Face	0.09	0.1	0.02	0.05	0

Table 1. Results of the algorithm for the first input image in Figure 5 (which contains the dinosaur) and comparison to two geometric schemes. Indexing that uses lighting information produces favorable results.

	Total	True
Total stored	5,940,480	0.98
α-space	48,815	0.97
$\alpha \cap \beta$	1916	0.94
Gramian	331	0.94
Lighting	18	0.86

Table 2. Average number of model sets matched per image set in each stage of the algorithm, together with the fraction of correct matches that remain after each discrimination (right column). The β-space produces a similar number of matches as the α-space.

experiments we automatically diluted the number of feature points extracted with the Harris detector to 40 using criteria of distance and saliency. No segmentation or grouping algorithms were used, although these could reduce the number of image sets need to be considered. The postprocessing took about 5sec. These times were obtained with a combined Matlab and Mex file implementation (using a Pentium 4, 2.8 GHz).

8. Conclusion

We have presented an indexing algorithm for identifying 3D objects in single 2D images under unknown pose and illumination and in the presence of occlusion and clutter. We have demonstrated the importance of intensity cues to complement the geometric cues in eliminating false matchings between model and image sets of points. Our experiments demonstrate that the method can work with general, real objects and is not restricted to polyhedral shapes as previous methods are. It can also handle a wide variety of lighting conditions despite the Lambertian and convexity assumptions. We have presented experiments in which we successfully detected the presence of objects in complex scenes under a large variety of poses and lightings. Future work includes incorporating color information, testing the method on a database of hundred models, and parallel implementation of the model.

References

[1] R. Basri and D.W. Jacobs. Lambertian reflectance and linear subspaces. *PAMI*, 25(2):218–233, 2003.

[2] J.S. Beis and D.G. Lowe. Indexing without invariants in 3d object recognition. *PAMI*, 21(10):1000–1015, 1999.

[3] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *PAMI*, 25(9):1063–1074, 2003.

[4] J. Burns, R. Weiss, and E. Riseman. View variation of point-set and line-segment features. *PAMI*, 15(1):51–68, 1993.

[5] D. Clemens and D. Jacobs. Space and time bounds on indexing 3d models from 2d images. *PAMI*, 13(10):1007–1018, 1991.

[6] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[7] D. Forsyth, J.L. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell. Invariant descriptors for 3d object recognition and pose. *PAMI*, 13(10):971–991, 1991.

[8] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: generative models for recognition under variable pose and illumination. *PAMI*, 23(6):643–660, 2001.

[9] R. Gross, I. Matthews, and S. Baker. Eigen light-fields and face recognition across pose. *Int. Conf. on Automatic Face and Gesture Recognition*, pages 1–7, 2002.

[10] C. Harris and M. Stephens. A combined corner and edge detector. *4th Alvey Vision Conf.*, pages 147–151, 1988.

[11] D.P. Huttenlocher and S. Ullman. Object recognition using alignment. *ICCV*, pages 102–111, 1987.

[12] D.W. Jacobs. Matching 3d models to 2d images. *IJCV*, 21(1/2):123–153, 1997.

[13] Y. Lamdan and H.J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. *ICCV*, pages 238–249, 1988.

[14] Y. Moses and S. Ullman. Limitations of non model-based recognition schemes. *ECCV*, pages 820–828, 1992.

[15] J.L. Mundy and A. Zisserman, editors. *Geometric Invariance in Computer Vision*. MIT Press, 1992.

[16] H. Murase and S. Nayar. Visual learning and recognition of 3d objects from appearance. *IJCV*, 14(1):5–25, 1995.

[17] C.F. Olson. Probabilistic indexing for object recognition. *PAMI*, 17(5):518–522, 1995.

[18] R. Ramamoorthi and P. Hanrahan. On the relationship between radiance and irradiance: Determining the illumination from images of a convex lambertian object. *JOSA*, 18(10):2448–2459, 2001.

[19] I. Shimshoni and J. Ponce. 3d probabilistic object recognition. *IJCV*, 36(1):51–70, 2000.

[20] B. Vijayakumar, D.J. Kriegman, and J. Ponce. Invariant-based recognition of complex curved 3d objects from image contours. *CVIU*, 72:287–303, 1998.

[21] P. Viola and W.A. Wells. Alignment by maximization of mutual information. *IJCV*, 24:137 – 154, 1997.

[22] D. Weinshall. Model based invariants for 3d vision. *IJCV*, 10(1):27–42, 1993.