

Recognizing Articulated Objects Using a Region-Based Invariant Transform

Isaac Weiss and Manjit Ray

Abstract—In this paper, we present a new method for representing and recognizing objects, based on invariants of the object's regions. We apply the method to articulated objects in low-resolution, noisy range images. Articulated objects such as a back-hoe can have many degrees of freedom, in addition to the unknown variables of viewpoint. Recognizing such an object in an image can involve a search in a high-dimensional space that involves all these unknown variables. Here, we use invariance to reduce this search space to a manageable size. The low resolution of our range images makes it hard to use common features such as edges to find invariants. We have thus developed a new "featureless" method that does not depend on feature detection. Instead of local features, we deal with whole regions of the object. We define a "transform" that converts the image into an invariant representation on a grid, based on invariant descriptors of entire regions centered around the grid points. We use these region-based invariants for indexing and recognition. While the focus here is on articulation, the method can be easily applied to other problems such as the occlusion of fixed objects.

Index Terms—Object recognition, invariance, range images, transform.

1 INTRODUCTION

In this paper, we address the problem of recognizing articulated objects from single range images. In addition to the usual challenges of such a task, such as an unknown viewpoint, the objects are also at unknown articulation angles and the images are of quite low resolution because the sensor is far from the objects. On the other hand, we know the absolute coordinates x, y, z of each pixel. Our goal here is to find both the identity of the observed object and the articulation angles.

The approach is necessarily model-based. Like any object recognition method, it requires a method of representation for both the models and the visible objects. Broadly speaking, most representations can be classified into two main categories: local and global. Local methods rely on local features such as edges, normals, and curvatures. They require reliable extraction of such features, which can be a problem particularly in low-resolution images such as ours. Global methods, on the other hand, rely on properties of the whole object such as moments or approximating polynomials. These are usually sensitive to occlusion.

Our approach lies in-between the local and the global representations and tries to capture the advantages of both. It can be called region-based as it is based on regions of the objects. These regions are smaller than the whole object, so that if a region of the object is occluded others can be used to identify it. They are larger than local neighborhoods so the shape descriptors we define on them are more robust to noise than local features. Our shape descriptors are region-based invariants, derived by the canonical frame method, enabling us to achieve invariance to changes in viewpoint as well as to deal with articulation and occlusion.

The size of the regions can be controlled. It can range from the whole object, yielding a global method, to very small regions yielding a local method. The degree of invariance is also controlled in this way from only global pose invariance at one extreme to invariance at every neighborhood at the other. A complete

- The authors are with the Center for Automation Research, University of Maryland, College Park, MD 20742.
E-mail: {weiss, manjit}@cfar.umd.edu.

Manuscript received 27 Mar. 2003; revised 17 Jan. 2005; accepted 23 Feb. 2005; published online 11 Aug. 2005.

Recommended for acceptance by C. Taylor.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0011-0303.

representation involves using several region sizes or scales of description.

2 HIGHLIGHTS OF OUR APPROACH

Our region-based approach is based on the following main ideas:

1. We transform *regions* of the object into a representation on a grid. The regions are bigger than local neighborhoods but smaller than the whole object. Briefly, we proceed as follows: We first define a grid of points on the visible object. This grid is generated in the image plane and projected onto the 3D surface. Around each grid point, we define a sphere of a given radius, and look at the region of the object enclosed within this sphere. This enclosed part is the region associated with the grid point. We then calculate invariant shape descriptors (a small set of numbers) that characterize this region of the object and assign them to its grid point. This is our invariant region-based transform. Because the descriptors were calculated on whole regions they are less sensitive to errors than strictly local quantities. Because the sphere is smaller than the whole object and is defined at each grid point, this representation is less sensitive to occlusion than global methods. We can be missing a region of the object and still obtain enough descriptors to recognize the object.
2. We take into account the scale space properties of the shape. A shape can have different descriptors at different scales of representation. This is controlled in our method by setting the radii of the spheres defined above. A larger sphere radius represents the shape at a larger, coarser scale. We use several preset radii which sample a whole range of scales. In the extreme case, one sphere includes the whole object, yielding a global method. This can be a useful transform of nonarticulated objects.
3. Our representation is Euclidean invariant. Since we deal with 3D range images, there is no problem of projection into 2D images, but the object can still undergo the Euclidean motions of translations and rotations. Previous methods used simple invariants such as distances and angles. Our shape descriptors are invariant quantities describing the regions of the object that are enclosed within the spheres. Furthermore, the invariants are used here as a complete representation or a transform of the object.
4. Our approach is able to deal with articulated objects by reducing the number of degrees of freedom that we have to deal with. A complicated object such as a back-hoe can have some 10 DOFs which makes the search space for the correct articulation angles prohibitively large. However, the smaller regions contain at most two of the moving parts and many contain only one or none, so they are much easier to deal with. The invariants are in effect used to eliminate many of the relative poses between parts of the articulated object, in addition to eliminating the global pose.
5. We use the invariant transform as a means of indexing the object, eliminating the search for point correspondences between models and images. Both the spatial (invariant) and articulation descriptors of each model are indexed within a (discrete) hypersurface that makes recognition easy.

2.1 Finding Region-Based Invariants

At the core of the transform is a method of finding invariant descriptors of the region enclosed in the sphere. There are obviously many ways to find invariants, but we have chosen one that is best-suited to our low-resolution images. Since we cannot extract local features reliably, we find invariants that describe the enclosed region as a whole. At the same time, these descriptors are not too sensitive to the sampling parameters such as the grid spacing or the sphere radius.

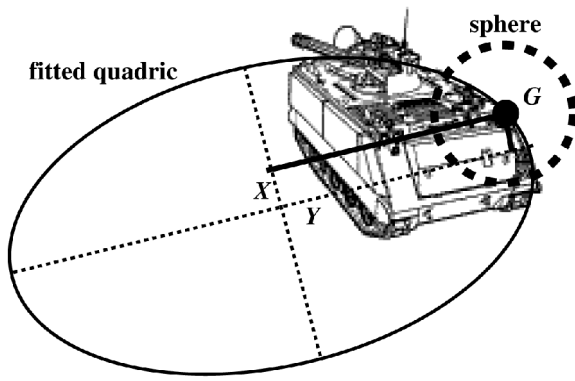


Fig. 1. Ellipsoid fits a region of the object inside the sphere. Ellipsoid axes define invariant coordinates X, Y, Z for grid point G at the sphere center.

The invariants are calculated as follows: Given the region enclosed in the sphere, we fit to it a quadric surface (Fig. 1). The quadric does not need to fit the shape closely but it has to fit it *invariantly*. That is, when looking from a different viewpoint, we want to obtain the same quadric fitted to the surface (but viewed from the new viewpoint). Achieving this in discrete images, gave rise to several problems which needed to be overcome. Once we obtain the fitted quadric, we can calculate invariants. We do that by moving to a *canonical coordinate frame*, namely, a frame that depends only on the shape itself and not on any external entities and is therefore invariant. Such a frame is provided by the three major axes of the quadric surface (e.g., the axes of an ellipsoid). Since this frame is invariant, any quantity expressed in it is also an invariant. One example of such invariants is the coordinates of the grid point at the sphere center, which is generally different from the quadric's center (i.e., the new system's origin). Another example is the centroid of the region enclosed in the sphere. These invariants are assigned to the grid point. Repeating this process for each grid point, we obtain an invariant transform of the whole object.

2.2 Region Size Considerations

The scale, or regions' size, is controlled by the spheres radius. A smaller radius results in smaller and more numerous regions. Smaller regions are more independent of each other and, thus, more invariant, i.e., more independent Euclidean coordinate frames (or poses) are eliminated. This helps with articulation, but if the regions are too small we can lose some of the geometric relations between regions and thus some discriminative power. At the other extreme, one sphere encloses the whole object, so each grid point has the same quadric fitted to the whole object. (The invariants are still different at each grid point). In this case, we eliminate only the global pose of the object so we fully preserve its geometric structure, but we have problems with articulation and occlusion. To avoid the pitfalls of both extremes, we use medium radii, yielding many overlapping medium size regions, as well as a multiscale approach involving representations with several radii. The larger radii capture the large-scale structure of the object while the smaller scales are more invariant.

2.3 Summary of the Algorithm

We outline here the steps of our algorithm. Further details are given in subsequent sections.

Offline processing of models:

1. Choose a set of sphere radii r_i .
2. Sample each given model on a uniform grid. Around each grid point, draw a sphere of radius r_1 .
3. Find the invariants of the region enclosed in each sphere as described above. This yields an invariant representation of the object, expressed on the grid. This is our invariant region-based transform of the object at a scale r_1 . Repeat

for the other scales. For articulated objects, repeat for a discretized set of articulation parameters.

4. Index all results in a database consisting of a hyper-space spanning both spatial (invariant) and articulation coordinates.

Online recognition of images:

1. Sample the unknown object's image on a grid and calculate the invariants, at different scales r_i , as was done for the models. The articulation is obviously fixed and no sampling of it is possible or needed.
2. Match the invariants of each grid point against the invariants indexed in the database. Each such match yields candidates for models with appropriate articulation parameters.
3. Accumulate the resulting candidate models and articulations in an accumulator array, a form of a contingency table. There is one accumulator for each candidate model.
4. Find the highest peak in the accumulators, i.e., where the largest number of "votes" lie. This yields the correct model with the correct articulation angles.

2.4 Comparison with Previous Work

One of the main issues in 3D object recognition is how to represent the object. Representations for 3D objects can be generally classified into:

1. *Local surface representations*: In this class of representations, surface properties such as surface normals and Gaussian curvatures are estimated and used in the description of the surface. These approaches depend on a reliable extraction of local features such as edges and derivatives which are quite sensitive to errors in low resolutions. A Euclidean invariant representation based on local curvatures was used in [1]. These curvatures require a good accuracy in extracting local features such as derivatives, unlike the present work which is "featureless." In [13], an invariant "splash" representation based on local Gaussian mapping is used. In [14], a surface representation that is invariant under projective transformations is presented.
2. *Volumetric, or global representations*: These representations use volumes rather than surface descriptors. They include generalized cylinders [2], superquadrics, and spherical harmonics. The volumetric and similar approaches are *global*, i.e., they depend on large-scale characteristics of the shape. These are less vulnerable to errors but they have to deal with the object as a whole and are thus sensitive to occlusion.
3. *Region-based representations*: A relatively new approach lies in-between the above approaches. Our approach here can be called "region-based." It is based on object regions that are bigger than local neighborhoods but smaller than the whole object. Other intermediate shape representations are based on using histograms of point coordinates ("spin images") as descriptors of neighborhoods around mesh points, e.g., [4], [7], [12]. These methods require at each neighborhood:
 - a. an accurate surface normal and
 - b. enough points to build a reliable histogram.

Our approach is applied here to low resolution images in which neither of these requirements is satisfied. Furthermore, our approach uses the invariant descriptors in an indexing scheme to avoid the search for point correspondences between the models and the image, while the methods cited above rely partly on random search.

Invariants to viewpoint changes, without articulation, have been used before, e.g., in [14], [18], [3], [9], [5], [8], [10], [17], [15], [16], [11]. Most of these deal with affine and projective invariants of the projection from 3D to 2D. In the present work, we have 3D range

images so the problem of projection does not appear. Viewpoint transformations here involve only translations and rotations. Thus, we can use Euclidean invariants which are more robust than the projective or affine invariants needed for 2D projected images.

Euclidean invariants such as distances and angles have been used before (e.g., [19]). Our use of Euclidean invariants differs from other methods as follows: 1) We use Euclidean invariants of whole regions, based on quadric fitting. 2) We use the invariants to obtain a complete transform of the object for efficient indexing, avoiding the search for correspondences between models and images. 3) We use the invariants to eliminate the relative poses between articulating parts in addition to eliminating the global pose of the object.

3 FITTING THE QUADRATIC

The quadric surface is the lowest order 3D surface from which properties invariant to Euclidean transformations can be extracted. Even though a higher-order surface would yield more invariant quantities, fitting it to our low-resolution data would result in parameter values that are less reliable. At the same time, with the exception of the degenerate case when the surface is a plane, enough information can be extracted from the parameters of such a surface to obtain a 3D canonical coordinate frame.

3.1 The Fitting Metric

Here, we define a metric for the distance between the quadric and the region. The metric has to be Euclidean invariant but does not need to represent a close fit.

A surface can be represented as the zero set of a function, namely, $f(\mathbf{x}) = 0$, with \mathbf{x} being a vector in an n -dimensional space, $\mathbf{x} = (x_1, \dots, x_n)^t$. To find the distance of some point \mathbf{x}_1 from the surface, we can use a first order Taylor approximation of f around this point to get:

$$f(\mathbf{x}) \approx f(\mathbf{x}_1) + \nabla f(\mathbf{x}_1) \cdot (\mathbf{x} - \mathbf{x}_1) = 0.$$

It is easy to show that a point \mathbf{x}_0 on this surface has approximately the minimal distance from \mathbf{x}_1 when the difference vector $(\mathbf{x}_0 - \mathbf{x}_1)$ has the same direction as the gradient vector $\nabla f(\mathbf{x}_1)$, up to a sign. In this case, the above equation reduces to:

$$f(\mathbf{x}_1) \pm |\nabla f(\mathbf{x}_1)| |(\mathbf{x}_0 - \mathbf{x}_1)| = 0.$$

The distance $|\mathbf{x}_1 - \mathbf{x}_0|$ can now be derived from the above as:

$$d^2 = |\mathbf{x}_1 - \mathbf{x}_0|^2 = (\mathbf{x}_1 - \mathbf{x}_0)^2 = \left(\frac{f(\mathbf{x}_1)}{\nabla f(\mathbf{x}_1)} \right)^2, \quad (1)$$

where the square of a vector is defined as the square norm, e.g., $(\nabla f(\mathbf{x}_1))^2 = (\nabla f(\mathbf{x}_1)) \cdot (\nabla f(\mathbf{x}_1))^t$. The numerator in d is known as the "algebraic distance" which is not invariant. Normalizing it by the gradient as above makes d invariant and also more similar to the usual geometric distance.

Given a set of m points \mathbf{x}_i , the average (squared) distance from this set to the surface is

$$\bar{d}^2 = \frac{1}{m} \sum_{i=1}^m \left(\frac{f(\mathbf{x}_i)}{\nabla f(\mathbf{x}_i)} \right)^2.$$

Trying to minimize this distance measure, we will obtain a nonlinear problem. To avoid that, we can approximate this measure to obtain our fitting metric:

$$\bar{d}^2 = \frac{1}{m} \frac{\sum_i (f(\mathbf{x}_i))^2}{\sum_i (\nabla f(\mathbf{x}_i))^2}. \quad (2)$$

As mentioned before, the approximations are of no consequence, because we do not try to make the quadric fit closely to the data. All we require is that the fit be invariant. It is easy to prove the Euclidean invariance of this metric.

3.2 Finding the Surface

We want to choose our surface f such that it minimizes the metric \bar{d} . A general surface $f(\alpha, \mathbf{x})$ depends on a set of parameters α over which we want to minimize the metric. A way to do this is to minimize the numerator in (2) with the constraint of a fixed denominator, namely, minimize $\sum_i (f(\alpha, \mathbf{x}_i))^2$ subject to the constraint $\sum_i (\nabla f(\alpha, \mathbf{x}_i))^2 = \text{const}$. We can handle the problem by the method of Lagrange multipliers, namely, solve the set of equations

$$\frac{\partial}{\partial \alpha} \sum_i (f(\alpha, \mathbf{x}_i))^2 + \lambda \frac{\partial}{\partial \alpha} \sum_i (\nabla f(\alpha, \mathbf{x}_i))^2 = 0$$

with λ being an additional unknown variable.

We now specialize to a surface f that can be written as a polynomial

$$f = \alpha F \equiv \sum_{k=1}^n \alpha_k F_k$$

with the parameter vector $\alpha = (\alpha_1, \dots, \alpha_n)$ multiplying a vector of monomial functions $F = (F_1, \dots, F_n)^t$. This includes lines, planes, conics, quadrics, etc. Our quadric surface can be written as

$$f = ax^2 + by^2 + cz^2 + 2fyz + 2gzx + 2hxy + 2px + 2qy + 2rz + d = 0 \quad (3)$$

so, in this case, the functions f_k are of the form x^2, xy , etc. The Lagrange equations above can now be written as

$$\frac{\partial}{\partial \alpha} \sum_i (\alpha F(\mathbf{x}_i))^2 + \lambda \frac{\partial}{\partial \alpha} \sum_i (\alpha \nabla F(\mathbf{x}_i))^2 = 0.$$

Taking the derivatives above, and considering the vectorial nature of α, F , it is easy to obtain

$$\alpha \sum_i F(\mathbf{x}_i) F(\mathbf{x}_i)^t + \lambda \alpha \sum_i \nabla F(\mathbf{x}_i) \nabla F(\mathbf{x}_i)^t = 0.$$

Defining the two matrices

$$\bar{F} = \sum_i F(\mathbf{x}_i) F(\mathbf{x}_i)^t, \quad \bar{G} = \sum_i \nabla F(\mathbf{x}_i) \nabla F(\mathbf{x}_i)^t, \quad (4)$$

we can now write the Lagrange equations as

$$(\bar{F} + \lambda \bar{G}) \alpha^t = 0.$$

This is known as the generalized eigenvalue problem, namely, find eigenvalues λ and eigenvectors α that satisfy the above equation. With \bar{F}, \bar{G} being symmetric nonnegative matrices, we solve the problem using the well-known QZ factorization algorithm [6]. From (2) for \bar{d} , it is obvious that we need the smallest λ .

3.3 Resolution and Discretization Invariance

We have encountered a problem in which the above metric depends on the discretization, or the sampling of the surface, which is not viewpoint invariant. This is because the visible object contains many surface patches which are posed at different angles with respect to the sensor. Each patch is thus sampled at a different sampling rate per unit area of the real surface, although the sampling rate is the same per unit area as projected on the sensor. A more oblique patch will have fewer pixels per unit of real area. Thus, the number of pixels per unit area differs for different surface patches. When we change the viewpoint, the patches pose changes and, thus, the sampling rate changes in a way which is different for each patch. Thus, the relative contribution of each patch to the sums in the metric \bar{d} (2) changes and, so, the value of this metric changes with the viewpoint.

The solution is to replace the summations in the matrices of (4) by integrations over the surface. This is done using a numerical

integration technique based on Bode's rule. This eliminates the effect of the discrete sampling. A surface patch will now contribute the same amount to the metric regardless of the resolution or the viewpoint.

4 RECOGNITION OF ARTICULATED OBJECTS

4.1 Articulation invariants

We face a significant problem when we want to derive invariant descriptors for the articulation degrees of freedom. Invariant descriptors generally apply to particular transformations; for instance, if an object is viewed from different viewpoints, we can derive invariants of the viewpoint transformations. Specific forms of viewpoint invariants involve well-defined groups of transformations, for example, Euclidean transformations in the case of range images, and can be applied to any object we want to recognize, i.e., the transformation group is independent of the object. Thus, these viewpoint invariants can be applied generically to all objects.

However, in the case of articulated objects, this is no longer the case. Each model has its own set of articulation degrees of freedom which translates into its own specific transformation group. It is difficult to determine the degrees of freedom of an unknown object from its range image and, hence, its transformation group remains unknown. Therefore, while it is theoretically possible to derive articulation-invariant functions for each particular object, it is difficult to determine generic invariants that can be applied to all objects.

Although we cannot use articulation invariants as such, we can reduce the number of DOFs using our region-based transform. In a relatively small region, only one or two parts of the object will be present. Thus, most regions will contain at most two DOFs, and many will have no articulation DOFs at all. Thus, in effect, the invariants eliminate some of the relative poses between regions of the object, in addition to eliminating the unknown viewpoint.

4.2 Hypersurface Indexing

In this section, we describe the indexing scheme used for the objects with their articulation DOFs.

Denoting our set of three invariants at a grid point by \mathbf{x} , we note that they are functions of the articulation parameters \mathbf{u} , i.e., $\mathbf{x}(\mathbf{u})$. Indexing the articulations amounts to inverting this relation, i.e., given the invariants \mathbf{x} , we want to look up the articulation parameters $\mathbf{u}(\mathbf{x})$. This inversion is easily done when the relationship is expressed as a implicit hypersurface $f(\mathbf{x}, \mathbf{u}) = 0$. This hypersurface is defined in a hyperspace with invariant (i.e., spatial) coordinates \mathbf{x} and articulation parameters \mathbf{u} . To construct this hypersurface, we can vary the articulation parameters, and for each value \mathbf{u} we perform the invariant transform to determine the invariant coordinates \mathbf{x} . This is done offline for each model. All models are represented in a similar way in the hyperspace. We have a separate hyperspace for each of three sphere radii r_i .

The hypersurface is now a form of indexing in which, given an invariant image point \mathbf{x}_1 , we can look up models with corresponding articulations $\mathbf{u}(\mathbf{x}_1)$. This can be done by intersecting the hypersurfaces representing all the models with the hyperplane $\mathbf{x} = \mathbf{x}_1$ and recording the parameters \mathbf{u} at the intersections for each model. We repeat this for all image points \mathbf{x}_i and collect the results in a contingency table. For most points \mathbf{x}_i , there will be relatively few corresponding models, because most models do not span the full extent of the hyperspace even with articulation. Thus, we will not have many candidate models to choose from. An object is recognized when the corresponding model gets the most votes in the contingency table.

In discretizing this method, the hypersurface is represented as a sparse multidimensional array, with three indices representing the spatial invariants and the remaining indices representing the articulation parameters. The array has the value of 1 where the surface passes and 0 elsewhere. The contingency table is another array, one for each model, which we call the *accumulator*. This array

has indices corresponding to the articulation parameters \mathbf{u} (only), and its values correspond to the number of "votes" we obtain for these particular \mathbf{u} s. Building the accumulator is a simple matter of looking up the \mathbf{u} indices of the nonzero elements of the sparse arrays, for given \mathbf{x} indices. This represents the intersections mentioned above. We used 1D and 2D accumulators. For non-articulated objects, the accumulator is a scalar counter containing the votes for a particular model.

One advantage of this indexing method lies in the smoothness of the hypersurface, resulting from the fact that neighboring articulation values \mathbf{u} produce neighboring invariants \mathbf{x} . This makes the method quite insensitive to the details of the discretization. Obviously, the articulation parameters can only be indexed as a discrete sample of values, and the invariant transform can only be performed on some predefined grid points which may be different for the object and for the model. The smoothness of the hypersurface means that even if the correct articulation parameters are not indexed we will still obtain values that are close to them. It also means that we have considerable latitude in choosing the grid spacing and position.

5 EXPERIMENTS

We have conducted experiments to test our recognition scheme with synthetic range data. The simulated sensor was a single-look range finder with each pixel representing an angle-angle region and having two values: range and intensity. The goals of the experiments were to match the correct model to each given test image, as well as to find the correct articulation angles. Thus, the experiments test a relatively small number of models at a wide range of articulations rather than a large database of fixed models. We have also tested a range of viewpoints and noise levels.

Following the general algorithm of Section 2.3, we first build the hypersurface using the invariant transforms of our reference (model) range images. These are obtained from 3D models whose articulation angles were varied through their full range. This is done offline on a dense grid. We have a separate hypersurface for each model and each of the three sphere radii. The test images, to be recognized online, consist of images of the 3D models taken at an unknown viewpoint and articulation angles. We calculate their invariant transforms on a coarse grid.

Matching consists of building an accumulator as described earlier and then finding its peak. A sharp peak should be obtained for the correct model at the correct articulation parameters. To build the accumulator for a candidate model, the invariant coordinates of each grid point of the test image are used as lookup indices to find the corresponding articulation indices in the sparse array representing the model hypersurface. This is done with the help of a smoothing algorithm to overcome the discretization effects and to remove outliers.

The experiments led us to make certain choices concerning the canonical system:

1. The quadric's axes can determine a canonical frame only up to the signs of the axes, e.g., an axis can point left or right and still belong to the same quadric. We need another method to set the axes' signs. We have set the signs so that the positive octant of the canonical system will contain the biggest part of the region in the sphere. This is an invariant choice.
2. In fitting the quadric surface, it turned out that one axis is quite often very long so the error in finding the quadric's center can be quite large. Thus, we do not use the quadric's center as the origin of the system as we did at first, but instead we use the centroid of the object's region contained in the sphere. The axes can still be used as canonical directions. Since the centroid depends only on the shape itself, this choice of a canonical system is invariant too.

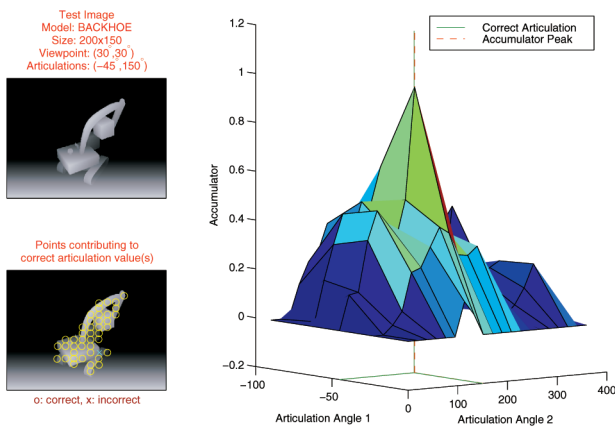


Fig. 2. Accumulator for correct match with two DOFs. Object BACKHOE, view 1, radius R_1 .

5.1 Summary of Calculating the Invariants

With the above choices, the invariants at each grid point are calculated as follows:

- Define a sphere around the grid point. Fit a quadric surface to the object's region inside the sphere, using the metric (2) with the integration scheme discussed in Section 3.3.
- Define a canonical coordinate system such that: 1) its axes directions, up to signs, coincide with the quadric's axes directions, 2) its axes signs are set so that the system's positive octant contains the largest part of the object's region, and 3) its origin coincides with the region's centroid.
- Calculate the coordinates of the grid point in this canonical system. These are our invariants at this grid point.

5.2 Results

We show the accumulators in a few examples, while most of the results are summarized in performance graphs. A correct match is recorded when the accumulator shows a peak at the correct articulation angle.

Figs. 2 and 3 show examples of accumulators obtained for correct matches, for one and two DOFs. We see that we have

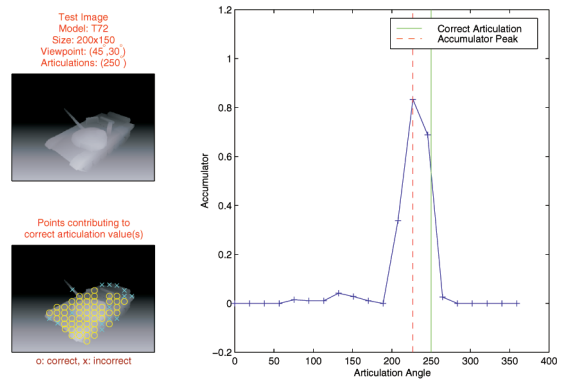


Fig. 3. Accumulator for correct match with one DOF. Object T72, view 2, radius R_1 .

obtained very sharp peaks at the correct articulation angles of the back-hoe and the T72. The grid points that contributed to the correct articulation values and those that did not are also indicated. Incorrect matches did not produce significant peaks.

The above examples were obtained with noiseless images. To test the performance under noisy conditions, we have added a Gaussian noise with a varying noise parameter σ . Fig. 4 summarizes a series of experiments, varying both the noise parameter and the viewpoint. Each figure shows four separate curves for four different viewpoints of the same object as indicated. For each curve, we depict the relative peak p , (namely, the height of the peak above random peaks relative to the noiseless case), as a function of the noise parameter σ . We show results for the T72. The results for the other models are quite similar.

Fig. 4a shows correct matches. We can see that we obtain strong correct peaks for viewpoint changes of up to 45 degrees. This is because, as the viewpoint changes, most of the object's regions still remain in full view and only those in the margins change. Of course, performance degrades somewhat with larger viewpoint differences and higher noise levels as our graphs show.

Fig. 4b shows experiments with (worst case) incorrect matches. We can see that the relative peak values are much lower than for the

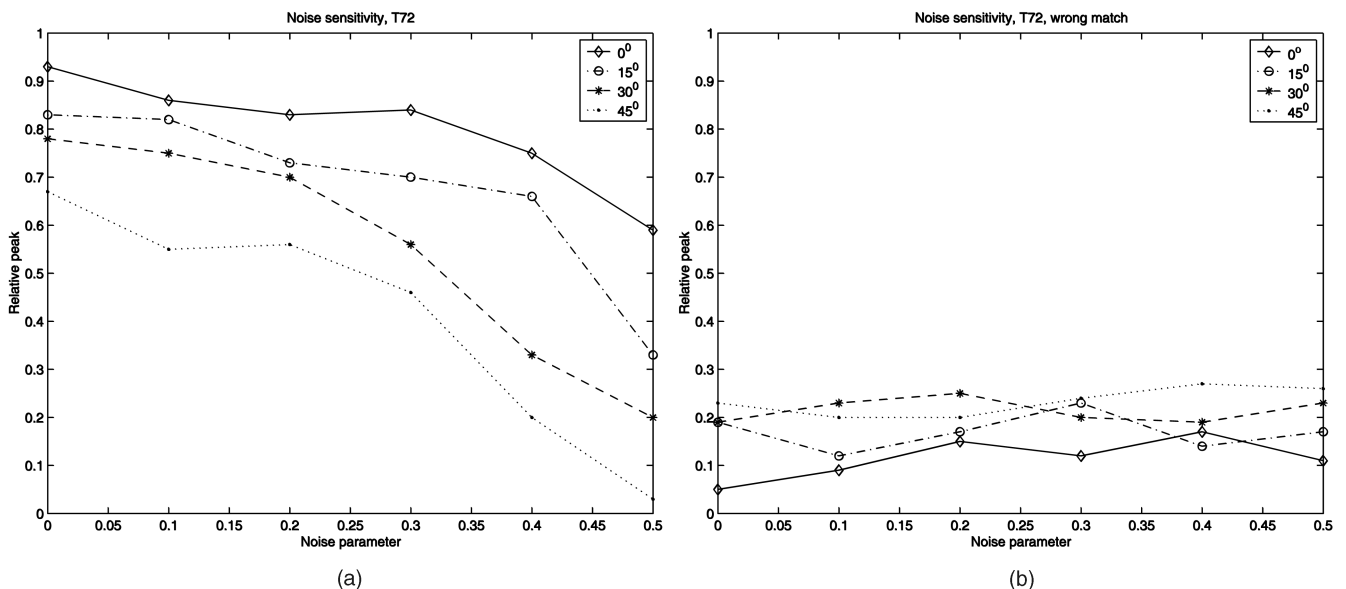


Fig. 4. Object T72, various viewpoints, peak height versus noise for (a) correct matches and for (b) incorrect matches.

correct matches, except for high noise with highly different view-points. Thus, except under these rather adverse conditions, the method can easily distinguish between correct and incorrect matches.

We have also tested several other factors that can affect the performance of the system such as smoothing methods, sphere sizes, and grid resolutions. Smoothing the hypersurface is necessary because we use different grid densities for the models and for the test images. We have also tried an iterative "cooperative" algorithm for further smoothing and removing outliers. After a few iterations, this algorithm provides a significant although not a decisive improvement over noniterative smoothing. The spheres need to capture meaningful regions of the object to obtain a reliable fit. We have found that we obtain useful peaks for sphere sizes ranging from a 1/3 of the object size and up. As for the grid resolution, we have found that about 10-15 grid points (not image points) along each dimension of the visible object were normally sufficient for our purposes. For the models (namely, for the hypersurfaces), we used at least 20 points in each dimension.

6 CONCLUSIONS

We have presented a new object representation which is region-based and Euclidean invariant. It lies in between the traditional global and local representations, with its position in this spectrum being controllable by the radius of the spheres enclosing the regions. A smaller radius results in smaller and more numerous regions. Smaller regions are more independent and yield more invariance, while larger regions preserve more of the object's larger scale structure. We use a multiscale approach that captures the advantages of different radii. Using the invariants for indexing and eliminating search, we have successfully implemented an object recognition method that works efficiently under adverse conditions of articulation, low resolution, and noise. It will be worthwhile to test the method in other applications such as fixed objects with significant occlusion.

ACKNOWLEDGMENTS

This research was supported by the US Defense Advanced Research Projects Agency under ARPA Order No. E655, the Air Force Wright Lab under Grant F33615-97-1-1015, and the US Office of Naval Research under Grant N00014-95-1-0521. The authors thank the reviewers for their careful reading and many valuable comments on the paper.

REFERENCES

- [1] P.J. Besl and R.C. Jain, "Segmentation through Variable-Order Surface Fitting," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, pp. 167-192, 1988.
- [2] T.O. Binford, "Visual Perception by Computer," *Proc. IEEE Conf. Systems and Control*, 1971.
- [3] A. Bruckstein, E. Rivlin, and I. Weiss, "Scale Space Invariants for Recognition," *Machine Vision and Applications*, vol. 15, pp. 335-344, 1997.
- [4] A.E. Johnson and M. Hebert, "Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433-449, May 1999.
- [5] D. Keren, R. Rivlin, I. Shimshoni, and I. Weiss, "Recognizing 3D Objects Using Tactile Sensing and Curve Invariants," *J. Math. Imaging and Vision*, vol. 12, no. 1, pp. 5-23, Feb. 2000.
- [6] C.B. Moler and G.W. Stewart, "An Algorithm for Generalized Matrix Eigenvalue Problems," *SIAM J. Numerical Analysis*, vol. 10, 1973.
- [7] G. Mori, S. Belongie, and J. Malik, "Shape Contexts Enable Efficient Retrieval of Similar Shapes," *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 723-730, 2001.
- [8] *Geometric Invariance in Machine Vision*. J.L. Mundy and A. Zisserman, eds. Cambridge, Mass.: MIT Press, 1992.

- [9] D. Renaudie, D. Kriegman, and J. Ponce, "Duals, Invariants, and the Recognition of Smooth Objects from Their Occluding Contour," *Proc. European Conf. Computer Vision*, 2000.
- [10] E. Rivlin and I. Weiss, "Local Invariants for Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 3, pp. 226-238, Mar. 1995.
- [11] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3D Object Recognition and Modeling Using Affine-Invariant Patches and Multi-View Spatial constraints," *Proc. Int'l Conf. Computer Vision*, Oct. 2003.
- [12] S. Ruiz-Correa, L.G. Shapiro, and M. Meilva, "A New Paradigm for Recognizing 3D Object Shapes from Range Data," *Proc. Int'l Conf. Computer Vision*, Oct. 2003.
- [13] F. Stein and G. Medioni, "Structural Indexing: Efficient 3D Object Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 125-145, Feb. 1992.
- [14] I. Weiss, "Projective Invariants of Shape," *Proc. Computer Vision and Image Processing*, pp. 291-297, 1988.
- [15] I. Weiss, "Noise Resistant Invariants of Curves," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 943-948, Sept. 1993.
- [16] I. Weiss, "Geometric Invariants and Object Recognition," *Int'l J. Computer Vision*, vol. 10, pp. 207-231, 1993.
- [17] I. Weiss, "Model-Based Recognition of 3D Curves from One View," *J. Math. Imaging and Vision*, vol. 10, pp. 1-10, 1999.
- [18] I. Weiss and M. Ray, "Model-Based Recognition of 3D Objects from Single Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 116-128, Feb. 2001.
- [19] M.D. Wheeler and K. Ikeuchi, "Sensor Modeling, Probabilistic Hypothesis Generation, and Robust Localization for Object Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 3, pp. 252-265, Mar. 1995.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.