Chapter 3

Three-Dimensional Laser Radar Recognition Approaches

Gregory Arnold[1], Timothy J. Klausutis[2], and Kirk Sturtz[3]

[1] Air Force Research Lab, AFRL/SNAT, Bldg. 620, 2241 Avionics Circle, Dayton, Ohio 45433 Gregory.Arnold@wpafb.af.mil
[2] Air Force Research Lab, AFRL/MNGI, Bldg. 13, 101 West Eglin Blvd., Eglin AFB, FL 32542 Timothy.Klausutis@eglin.af.mil
[3] Veridian Incorporated, 5200 Springfield Pike, Suite 200, Dayton, Ohio 45431 ksturtz@mbvlab.wpafb.af.mil

Summary. Three-dimensional laser radars measure the geometric shape of objects. The shape of an object is a geometric quality that is more intuitively understood than intensity-based sensors, and consequently laser radars are easier to interpret. While the shape contains more salient (and less variable) information, the computational difficulties are similar to those of other common sensor systems. A discussion of common approaches to 3D object recognition, and the technical issues (called operating conditions), are presented. A novel method that provides a straightforward approach to handling articulating object components and multiscale decomposition of complex objects is also presented. Invariants (or more precisely covariants) are a key element of this method. The presented approach is appealing since detection and segmentation processes need not be done beforehand, the object recognition system is robust to articulation and obscuration, and it is conducive to incorporating shape metrics.

3.1 Introduction

Lasers provide many advantages for the object recognition problem, especially when compared to passive electro-optical (video) sensors. For robust object recognition, it is desirable for the sensor to provide measurements of the object that are stable under many viewing and environmental conditions. Furthermore, these sensor measurements, or signatures, should be easily exploitable and provide enough richness to allow object-to-object separability. Specifically, 3D imaging laser radar (ladar) greatly simplifies the object recognition problem by accurately measuring the geometric shape of an object in 3D (preserving scale). In contrast many other sensing techniques inherently suffer a loss of information by projecting 3D objects onto 2D or 1D images. Another advantage of ladar shape measurements for object recognition is that the object signature is far less variable than other sensing modalities (e.g., the shape

of the object does not vary due to lighting, diurnal affects, thermal loading, range, etc.). However, there is still difficulty since the ladar effectively samples an object's surface slightly differently each time. Another complicating factor is that ladar does not sample a scene on a uniform sampling lattice. Furthermore, a general complication for recognition is the lack of a general theory of discrimination (i.e., how to tell objects apart). This chapter will present the fundamentals of generic ladar systems, and detail a method for handling the differences in images due to articulation and viewpoint changes.

Generically, ladars can be thought of as an orthographic projection of the world onto the sensor (see Figure 3.1). Many 3D imaging ladars provide a range and intensity measurement at every point in the sampling lattice. Direct detection and coherent detection are two common ladar detection techniques. A complete treatise of ladar detection techniques is beyond the scope of this chapter [1]. The intensity value is a measurement of the amount of energy reflected from the appropriate region of the sampling lattice. This measurement is directly related to the monostatic bidirectional reflectance distribution function (mBRDF) of the material illuminated by the laser pulse. The intensity image is effectively a narrow-band, actively illuminated 2D image. This chapter focuses on the range measurement. The inherent data coordinate system for 3D ladar is {angle, angle, range}=$\{\theta, \phi, \rho\}$, where θ is the depression angle and ϕ is the azimuth angle from which the transmitted laser energy propagates from the sensor for each point in the sampling lattice. This is a polar coordinate system that can be transformed into a rectilinear $\{x, y, z\}$ coordinate system. Many different types of ladars exist, but for simplicity a flash ladar constructed with a focal plane array (FPA) of detectors will be assumed as the standard in this chapter. The term flash implies that the whole range and intensity image is measured at one time by spotlight illuminating the entire scene with one laser pulse. Alternately, a scanning 3D ladar images a scene by scanning one or several Laser beams over the entire sampling lattice. Although scanning ladars will be briefly described, the assumption is that appropriate motion compensation for platform motion has been done such that the resulting range image from the scanning ladar is equivalent to a (3-D) flash ladar. This assumes that the scanning mechanism operates in a linear fashion.

Operating conditions [2] have been discussed in many papers since their inauguration during the DARPA Moving and Stationary Target Acquisition and Recognition (MSTAR) program. Essentially the operating conditions are an attempt to describe everything that can affect the sensed image. They include sensor, target, and environmental parameters. Understanding all the standard and extended operating conditions is a first step to the development of an object recognition system. For the purposes of this chapter, they are categorized as (i) conditions whose effect on the image can be modeled (i.e., by a group action), (ii) conditions that obscure the image of the object (but are not easily modeled), or (iii) conditions that do not affect the part of the image corresponding to the object (i.e., changes in background). The goal is to

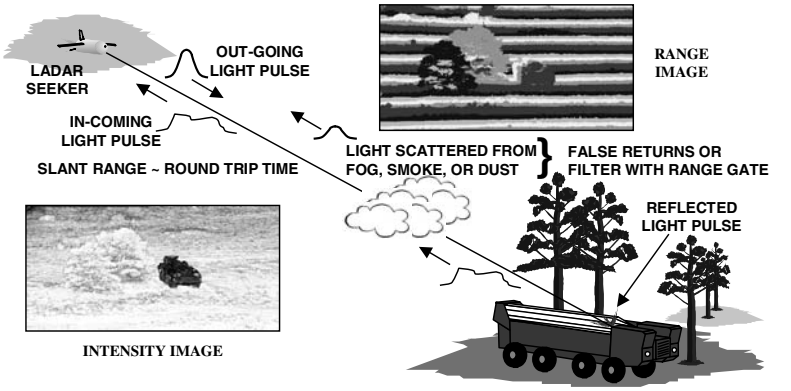


Figure 3.1. General ladar scene. This conceptual diagram illustrates the typical scenario and primary drivers of the sensed imagery. Most ladars return both a range and intensity at each pixel.

model and mitigate sufficient geometric effects to create a robust recognition system. The approach is model–based and object-centric.

A 3D ladar object recognition system can be thought of as a "simple" data fusion technique. It is simpler than more general fusion techniques because (typically) only one sensor and one image is involved. Even for this simple case the algorithm must efficiently fuse the information collected from each image pixel (or voxel, short for volumetric picture element, since it contains a third dimension). Enough pixels must be considered simultaneously to remove the unknown nuisance parameters of the model. For example, a single range return (without any other knowledge) cannot contain any information about the shape of the object. For the purposes of this chapter, "shape" is defined as what is left after translation and rotation have been removed. At least four points are needed (for 3D) to extract the local shape. Noise and small changes in the sampling grid can seriously affect the measured shape if the four points are in close proximity. Close proximity is preferred to mitigate obscuration and articulation, but dispersed locations are preferred to mitigate noise and achieve separability. Thus, the algorithm must make intelligent and efficient use of multiscale information to balance these issues.

The goal of 3D ladar object recognition systems is to classify images based on how similar or dissimilar the shape from the image is as compared to the shape of each object in a database. A metric is required to quantitatively compare and sort objects based on shape. The triangle inequality property of a metric enables efficient sifting through a large database by removing whole classes of objects that are different enough to be immediately removed from consideration. A modified form of the Procrustes metric is used for this object recognition system. This metric is constructed by considering the quotient space that is invariant to translation and rotation.

While 2D and 1D ladar systems exist and have many useful applications, for the purposes of this chapter ladar will imply 3D imaging ladar. A 3D model will be referred to as an *object*, and an {angle, angle, range} projection of that object (with a ladar at an arbitrary orientation) as an *image* or *range image*.

3.2 Ladar Sensors

3.2.1 Hardware

ladar sensors are active devices that avoid numerous issues inherent in passive systems and in stereo 3D reconstruction approaches. The two primary types of active range sensors are direct detection (also known as time-of-flight) and coherent detection.

Most direct-detection sensors periodically emit a short intermittent pulse and the distance is calculated by measuring the round-trip time (given the speed of light). This eliminates one type of shadow region that occurs with standard stereo 3D reconstruction approaches when a portion of the scene is not visible to both sensors [3]. However, shadow regions will still occur due to self-occlusion and occlusion of the background by objects in the foreground. This is one of the major differences in 3D image formation using ladar and the sensors used in the medical community. ladar is a reflective sensor while many 3D medical imaging sensors are partially transmissive.

Coherent detection techniques measure distance with a continuous (or multipulse) laser beam by measuring the phase shift of the reflected signal with respect to the original signal. Continuous-beam lasers require more power and are less covert. Furthermore, coherent detection techniques require more complex hardware than direct detection techniques.

Many ladar sensors have a single emitter and detector that are scanned across the scene at a constant angular resolution (see Figure 3.2). Although this is (currently) less expensive than the flash arrays that follow, there are numerous disadvantages. The primary disadvantage is that it takes longer to image a scene by scanning. A secondary problem is the nonlinearities induced by the scanning motion and by scene and object motion during the scanning process.

3.2.2 Projection

The appropriate choice of the projection model for ladar is muddled since the projection model will be different depending upon which coordinate system is utilized! In the native {angle, angle, range} coordinate system the projection model is a pinhole camera (full perspective model). Consider two parallel line segments in a plane orthogonal to the line of sight of the ladar. As the lines are moved farther away from the sensor, the angle subtended with respect

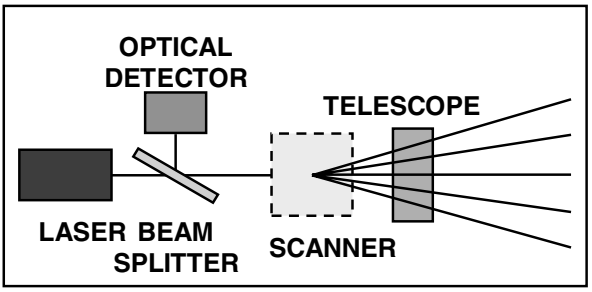


Figure 3.2. Typical ladar sensor hardware. A simplified diagram of the most common scanning, direct–detect system.

to the sensor decreases as $\frac{1}{\text{range}}$ (in each dimension). Therefore the angular extent of the lines within the image decreases as a function of their range.

The farther an object is from the sensor, the fewer pixels that will be on the object's surface (commonly called pixels-on-target). Lines that are parallel in 3D (but not parallel to the FPA) converge to a vanishing point in the $\{\text{angle}, \text{angle}, \text{range}\}$ image. This is known as the "train tracks phenomenon." Standard video cameras are also perspective projection. A fundamental difference is that video cameras cannot recover the absolute size of the object. In contrast, since a ladar measures the range, the size of the object can be calculated (up to the pixelation error).

An orthographic projection model is appropriate for a rectangular coordinate system. It is not possible to perform this coordinate transformation directly with video cameras. The addition of range information enables the conversion for ladar from polar to rectangular coordinates. This is not precisely correct since the rectangular voxels should grow larger as the depth increases analogous to the change in size of the polar voxels. Alternately, the x–y precision decreases for voxels that are farther from the sensor. For those familiar with radar systems, which are also orthographic, this is analogous to the fact that the signal-to-noise ratio decreases as the object moves farther away from the sensor.

3.2.3 Angle–Angle

A flash ladar's FPA is analogous to the CCD arrays used in digital cameras and video cameras where each array element contains a tiny receiver. Unlike CCD cameras, ladar is an active sensor and a single laser is used to spotlight illuminate the entire scene with each laser pulse. The laser is co-located with the receiver to form a monostatic system. Therefore, each element in the focal plane array can be viewed as containing both a receiver and a transmitter.

The FPA structure is inherently different from scanning systems as illustrated in Figure 3.3. Lenses can be used to transform to either array format, but FPAs are easier to build with a constant array size. A constant angular

size has a modest advantage during the polar-to-rectangular transformation that is commonly done in software. Neither system will obtain a uniform sampling of an observed surface in general. In Figure 3.3 a constant array size would obtain a uniform sampling of a flat surface that is parallel to the FPA, but as the surface rotates out of plane it will be sampled non uniformly. The only surface that would be sampled uniformly in general is a spherical surface with its origin placed at the focal point of a constant angular size ladar system. In summary, most ladar systems are natively {angle, angle, range} (i.e., polar coordinates). Algorithms that are conducive to polar coordinate systems will have an inherent computational and noise advantage, because it is not possible to transform from the (discretized) polar coordinate system to the (discretized) rectangular coordinate system without some interpolation scheme.

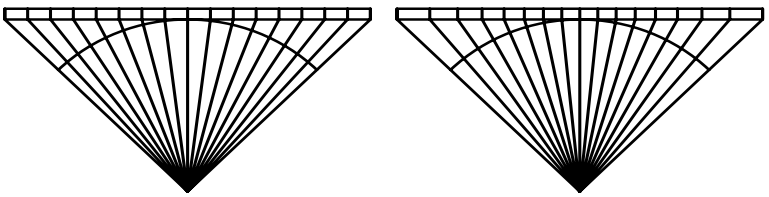


Figure 3.3. ladar focal plane arrays. A notional one-dimensional focal plane array is portrayed with a constant array size (**a**), and with a constant angular size (**b**). The choice of coordinate system will affect sensor design and the corresponding algorithm development. For FPAs, uniformly sized arrays, (**a**), are easier to construct and can achieve a constant angular size using a lens. For scanning ladar systems, it is easier to use constant angular step scan mirrors creating a constant angular array (**b**). Caution is still necessary since neither a constant angular or constant array size imply a uniform sampling of the object surface.

3.2.4 Range

Each pixel of the ladar measures the energy returned as a function of time. This is called the "range profile," and a notional example of a range profile for one pixel is illustrated in Figure 3.4. The majority of systems return the location and intensity of the nth-peak, where n is typically the first or last peak. This is commonly called "first bounce" or "last bounce." Some newer systems return the entire range profile or the first m detected pulse returns. This has advantages for reasoning about obscuration and validating edges and surfaces more complex than flat plates.

The returned pulse is the convolution of the transmitted signal with the object surface. Thus, the determination of the actual range is accomplished

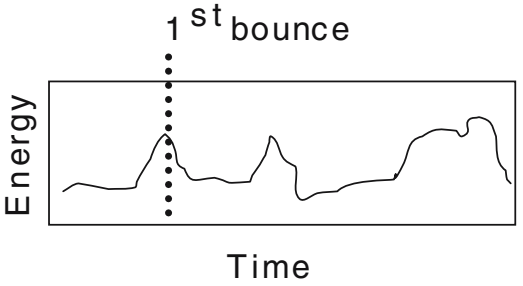


Figure 3.4. ladar range profile for one pixel. The range profile is the returned signal as a function of time (or distance). When multiple objects are within the field of view, more than one peak will occur. Current ladar systems typically return the first or last bounce peak of the range profile. Newer systems return multiple peaks or the entire profile.

by deconvolving the transmitted signal from the returned signal. This is generally an ill-posed problem. The common solution is to assume that the object surface is flat and therefore a simple peak detector is used to determine the estimated range. Hardware designers employ several common pulse detection strategies and these directly affect the location of the sensed return pulse. Furthermore, the selection of pulse detection technique will change the noise statistics. When multiple surfaces are within the same resolution cell, multiple peaks occur within the returned signal such as are illustrated in Figure 3.4. The nth-peak approach is a trade-off between accuracy and speed.

3.2.5 Transmission

Atmospheric effects such as scintillation (atmospheric transmission) and semi-transparent obscuration (clouds and smoke) limit the useful range of ladar systems. Platform vibration mitigation can also be difficult to achieve for high-resolution systems. Currently, ranges beyond 5 km are typically relegated to 1D or 2D systems unless advanced techniques, like pulse-doublets, is used. Experimental high-resolution 3D systems have greater operational ranges.

The reflectance properties of the object can severely affect the returned signal, or cause a complete dropout (i.e., where no return is recorded for the given pixel). Dropouts can occur if the reflectance is too high (shiny metal reflects the emitted energy away from the detector) or too low (the material attenuates the reflected energy to a peak value below the threshold of the detector).

Range gating is a technique that is commonly applied to mitigate the affects of obscuration and semitransparent obscurants. The concept is to only examine the part of the signal that could have come from something near the estimated object location. In other words, if some portion of the energy is returned too quickly, then it is probably the result of reflecting off nearby

clouds or smoke. Multiple reflections can cause a portion of the energy to return late. The finer the ability to accurately range gate, the better the signal-to-noise ratio will be. The associated peak detector or more advanced algorithms are more robust by applying this simple geometric constraint to the data.

The wavelength of the ladar is an interesting question from the point of view of computer vision beyond the visible spectrum. Lasers in the visible wavelengths are obviously very common; however, most sensor platforms would prefer to be covert. Atmospheric transmission and lack of generic tunable band-gap materials also limit the choice of wavelength. A common wavelength today is 1.06μm. However, this wavelength is not eye-safe and therefore the potential applications are limited. Longer wavelengths are being investigated not only for eye safety, but also for better weather and aerosol penetration.

3.2.6 Synopsis

Many other topics with regard to ladar system design could be addressed here. For example, polarization, noise statistics, and hardware limitations are all very important considerations for the design and exploitation of ladar data. However, this section is intended to be a basic initiation into key features of ladar hardware and phenomenology.

The primary purpose of this chapter is to discuss object recognition capabilities given a ladar sensor. ladar sensor engineers often want to know how to optimize their sensor design for a particular application. However, a discrimination theory does not currently exist and therefore sensor optimization for object recognition is often based on standard pattern recognition techniques applied to very limited data sets. Approaches that are beginning to address this shortcoming are discussed after the presentation of existing algorithm approaches.

3.3 Is 3D Ladar Object Recognition a Solved Problem?

ladar object recognition is not a solved problem, primarily due to computational complexity issues. It has been argued that computer vision is generally an ill-posed problem that will never be solved. From an information-theoretic point of view, 3D sensors contain more information than 2D sensors because 2D sensors suffer a loss of information from the projection of $\mathbb{R}^3 \mapsto \mathbb{R}^2$. Still, the computational problems are no easier for 3D than they are for 2D or 1D.

3.3.1 Technical Challenges

The computation complexity issues are primarily due to the operating conditions briefly mentioned in Section 3.1 and further illustrated in Table 3.3

on p. 92. Ross [2] is an excellent paper on defining operating conditions. The following list represents the subset of the operating conditions that are key to the 3D ladar object recognition problem:

- Translation
- Rotation
- 2.5D projection
- Surface resampling
- Number of pixels-on-target
- Point correspondence (labeling, registration)
- Obscuration
- Articulation
- Fidelity
- Unknown objects

An algorithm that can efficiently and effectively handle all of these problems would be a major advancement in the fields of object recognition and computer vision. Note that "noise" is not explicitly listed. This is intentional since it is the belief of the authors that noise is not the fundamental limitation of current object recognition approaches.

Added to all these difficulties, the development process is seldom conducive to solving these hard problems. Typically, most of the money is spent on the hardware and data collections. The exploitation efforts are not begun until the final stages of the development. This leads to an additional set of issues:

- There is never enough data.
- The sensor parameters are undefined or incorrect.
- The ground truth will be incomplete or incorrect.
- The operational "requirements" will be way beyond the state-of-the-art.
- The money will be limited.
- Time will be limited.
- Expectations will be too high.
- The capabilities will be oversold.
- The sensor models will not be understood.
- Mother Nature is enigmatic.

Each of the technical challenges will now be addressed in more detail.

Translation

The first challenge refers to the fact that the coordinate system is sensor-centered, not object-centered. Therefore, the $\{x, y, z\}$ or $\{\theta, \phi, \rho\}$ pixel locations for a particular point on the object can change arbitrarily. In other words, the algorithm should recognize the object whether it appears at the top of the image or the bottom of the image or anywhere in between. Translation is the easiest challenge to solve, but caution is necessary. The standard approach is to move the centroid of a region or volume of interest to the origin.

If the object is consistently segmented from the background every time then this simple method will suffice. However, approaches that are more complicated are needed if the algorithm is to successfully match partially obscured or articulated objects.

Rotation

The above method of removing translation can be interpreted as putting the data into a standard position. A proof exists [4] that the equivalent approach cannot be done to handle rotation. In general, however, the eigenvectors of the inner product of the (rectangular) coordinates are invariant to rotation. The primary difficulty is with degenerate objects or completely symmetric objects such as a sphere. The difficulty manifests itself as the need for heuristic algorithms to choose the sign and ordering of the eigenvectors. Mathematically, quotienting out 3D rotation from the ladar data does not result in a smooth manifold.

2.5-D Projection

The term 2.5D is used to convey the fact that ladar sensors cannot see through an object [1]. As noted in Section 3.2.1, ladar is a reflective sensor, not a transmissive sensor like most medical 3D sensors. A "true" 3D sensor would return a computer-aided design (CAD) model of the object (i.e., front, back, and all the information in between). A ladar only returns the range to the first dispersive surface. Thus, there is a requirement to match an "image" to a model in the database. Ideally, the ladar data can be compared directly to the appropriate portion of the CAD models that are used to describe the objects of interest.

Surface Resampling

The sampling grid's relative location on the surface is slightly different each time the surface of the object is sampled. Sometimes these differences are negligible, but a shift of half a pixel can be significant, especially for low resolutions or near, sharp edges. Typically, algorithm developers consider each point in a point cloud as the distance to the object at that particular grid point. This assumption would be correct if the instantaneous field of view (IFOV) of the detectors in the FPA were infinitesimally small. However, this is never the case and the detectors see a portion of the target larger than an infinitesimally small point. Therefore, the range measurement returned is the average distance of the surface(s) within the pixel's extent (depending upon the peak extraction algorithm). This simplification is harder to make when working with the complete range profile. The implication is that approaches (like graph matching) based on vertex locations will fail without compensating for this affect. Point correspondence algorithms must carefully handle this affect too.

Number of Pixels-on-Target

The absolute range from the sensor to the surface of the object is measured and therefore size (scale) is a known quantity. However, the farther the object is from the sensor, the fewer the pixels per unit surface area. For example, for a fixed angular resolution ladar, halving the range to an object quadruples the number pixels on the object. It is possible to calculate the area of the pixel at the object's surface. However, at the initial stages of the object recognition system it is often more computationally efficient to normalize out the number of pixels-on-target. This is a trade-off between discrimination capability and computational cost.

Point Correspondence

Even ignoring the issues about resampling the surface, registration is a computationally complex problem—potentially factorial in the number of points to be corresponded. There are some nice algorithms available from the operational research and pattern recognition literature (called the bipartite matching or Hungarian algorithm)[5, 6, 7]. A nice summary and approach is provided in [8] and a powerful new approach is presented in [9]. However, even these have complexity that is polynomial in the number of points to be imaged.

An additional consideration is that "point correspondence" typically implies that there are equal numbers of points to register. The problem becomes even more complex when the goal is to match as many points as possible from two different point clouds. The above references have various approaches to solve this problem.

An elegant solution to this problem is required before any algorithm is computationally feasible. Two (previously mentioned) complications are that corresponding points will never be precisely the same due to noise, and the points may not physically represent precisely the same region from the scene or CAD model, i.e., the image grid shifted. The first complication necessitates the use of a metric, and the second complication implies that the ultimate goal is to register the measured point cloud directly to a CAD model. In other words, there may *not* be an advantage to converting the CAD model to a point cloud for the purposes of matching.

Obscuration

Can an object still be recognized if it is not completely visible? Of course one can never see *all* of an object simultaneously, and algorithms must work even if everything that is expected to be visible in an image is not. Some parts of the object are more salient than others, so a true understanding of an algorithm's capability to handle obscuration can only be made with an understanding of the saliency of the visible surfaces. Jones [10] shows relatively good results against obscuration and articulation in 2D. Hetzel [11] presents an approach for 3D data that has an additional advantage as segmentation is not required.

Articulation

Articulation and obscuration each force a tradeoff between local and global features for object recognition algorithms. Articulation refers to the fact that objects have specific ways of changing. For example, cars have doors that can open, and people can move their arms, legs, and head. It can also be something that may or may not appear on an object, such as a spoiler on a sports car. The ability to recognize an object is typically limited to one instantiation without the ability to recognize each of these variations as allowable changes of the same object.

Fidelity

A physical and geometric model of the world, the sensor, and everything in between may be conceivable, but it is computationally unachievable. Each step in the processing chain, from the ladar probing the world to the output of the probability of a matching database model, makes simplifications to achieve computability and potentially compromising fidelity. Ideally these simplifications would be achieved while preserving the fundamental discrimination capability or be constructed in such a way as to provide incremental steps back to achieving the ideal discrimination capability. The goal is to be able to recognize all the objects (and corresponding poses) in the image with a finite number of computations.

In general, increased fidelity requires greater computation. The link between complexity and database indexing (search) is very strong. The indexing step is an n-class recognition problem, whereas the final validation is a one-class hypothesis verification. Most of the computations are consumed by the search (throwing out the models that do not match). The fundamental question is how to minimize the required fidelity and still guarantee that potential matches cannot be incorrectly pruned.

Unknown Objects

This problem has been addressed the least in current literature. The question is how to realize that something is not represented in the database. Ultimately, it boils down to drawing a threshold in a one-class recognition problem, but that threshold should make sense geometrically and with respect to the appropriate noise model. All that can typically be done with current systems is to decide which database object the image looks most like. Thus, the final threshold is drawn based on experiments with limited data sets. As mentioned in the previous technical challenge, the indexing step must be dependent upon the objects in the database, but the final step of verifying the hypothesized identity should be independent of the other objects in the database. Only a controlled environment, where unknown objects cannot occur, would allow the algorithm to bypass the final verification step.

3.3.2 Shape Representations

Object representation is a critical first issue in the construction of an object recognition system. From an information-theoretic point of view, the algorithm should use the raw data directly from the sensor since each transformation potentially introduces additional noise into the system or loses relevant information.

A rough hierarchy can be imposed upon the different representations that have been proposed by various authors. The following list of shape representations is ordered in an ascending level of abstractness:

1. Point cloud.
2. Features (points, edges, corners, normals).
3. Triangular mesh.
4. NURBS (biquadratic surfaces).
5. Superquadrics/generalized cylinders (geons).

Note that the number of parameters necessary to describe the shape of an object decreases as the level of abstractness increases. Consequently, object matching based on the more abstract representation is less complicated. A useful survey of techniques for data representation can be found in [12, 13, 14]. Each representation is briefly discussed subsequently.

Point Cloud

A point cloud generally denotes the raw data that is available directly from the sensor.

Features

Features are a first-level abstraction of point clouds. Feature points could be as simple as pruning the point cloud to only include points at "interesting" places, such as along edges or corners. The features could also be an augmentation of the point cloud with a local surface normal.

Triangular Mesh

A triangular mesh is the most common form of mesh that is used. Triangular meshes can be interpreted as a linear approximation (and interpolation) of the point cloud data. Ideally, the choice of vertices for the triangles is made to minimize the difference between the measured point cloud and the corresponding point cloud that would be generated from the mesh. A mesh is typically constrained to be continuous; however this is generally not sufficient to enforce a unique mesh representation. Additional constraints are often applied in order to achieve a unique representation, such as attempting to make the all of the triangles nearly equilateral. However, this is not possible in general, so corresponding recognition algorithms must be able to handle different representations for the same object.

NURBS

Nonuniform rational B-splines (NURBS) [15] are a generalization of biquadratic surfaces. These representations typically model a 2D surface embedded in 3D [16]. Patches of NURBS can be thought of as the generalization of meshes. The patches form a complete covering of the object's (measured) surface. Then, the surface within the bounds of each of these patches is represented by a NURBS surface. An immediate complication is defining a robust and unique decomposition of data into model patches, similar to meshes. Thus, the recognition algorithm cannot rely on the same object being modeled the same way for every instantiation of the object.

Superquadrics/Generalized Cylinders

Superquadrics [17] and generalized cylinders [18] (or geons) model a volume of data. They define specific mathematical forms to model the data. For example, generalize cylinders estimate a cross-section and then sweep the cross-section along the measured data. At each step along the swept out path the centroid of the cross-section is located, and the size of the cross-section is estimated. Objects are constructed by assembling multiple generalized cylinders together. This is clearly an ill–posed problem for obscured or unseen portions of the object. Generally, a symmetry assumption is made, or efforts are made to demarcate what portions of the surface were constructed from measured data.

Synopsis

All of these higher-level object representations have great promise; however, none of them have lived up to that promise to date. While a successful higher-level representation would greatly simplify the object recognition process, in practice these representations have simply traded simplicity in one portion of the overall system for added complexity in another with zero or negative gain.

The "negative gain" comes from the fact that these representations are exactly that — representations. Object recognition is based on discrimination. While one can argue that there is a functional dependence between representation and discrimination, it is easy to generate examples such that any given representation is the worst possible choice to differentiate the exemplars. However, this is not meant to encourage the other common extreme, which is to collect *enough* data to distinguish the classes (i.e., data-driven approaches). The goal is to find the right representation to optimize the discrimination capability with respect to the storage or speed requirements. This can only be discovered by modeling the entire object recognition system and using the scientific method such that the data is simply an experiment to validate or refute a hypothesized system model.

3.3.3 Shape Recognition/Indexing Approaches

Two obvious approaches to recognition are image-based and model-based. Jain [19] provides an excellent overview of statistical pattern analysis techniques. The most common image-based approach is template matching, and correlation is the most common similarity measure used in template matching. Some common approaches are briefly described below. Every approach has three common problems:

1. How to detect and segment the object from the background.
2. How to build and search a large database (avoiding local minima).
3. How to interpret results when the match is not perfect.

The following list and detailed descriptions represents many common approaches to shape recognition that have been developed and tested. References [12] and [13] are good survey papers.

1. Image-based matching (matched filter, template match).
2. Model-based matching.
3. Geometric hashing.
4. Hough transform.
5. Evidence accrual.
6. Learning approaches (neural networks, genetic algorithms, etc.).
7. Tree search (graph matching).
8. Principal components (eigenspace, eigenface, appearance-based).
9. Invariance (Fourier, moments, spherical harmonics, spin images).

Table 3.1. A summary of some typical characteristics of various approaches to shape recognition. A more detailed description of each method follows.

Approach	Model	Empirical	Voting	Alignment
Image-based		√	√	√
Model-based	√		√	√
Hashing	√	√	√	
Hough		√	√	
Evidence accrual		√	√	
Learning		√		√
Tree	√	√		√
Principal Components	√	√		√
Invariance	√	√	√	√

Image-Based Matching

Image-based or pixel-level matching generally implies the most common matched-filter-type comparison. This could be a comparison between two different measurements, but more generally is a comparison between a synthesized signature and the measured image using a "hypothesize and verify" type approach.

Intuitively, a matched filter is the average of the squared error between each measured pixel and the corresponding template pixel, or a nearest-neighbor type algorithm in a very high-dimensional space. A plethora of variations on this approach is available for purposes such as robustness to outliers and improving relative separability. "Image-based" refers to the fact that the algorithm is trained on measured or synthetic image data.

Image-based approaches suffer from both the complexity of the correspondence problem (finding the template in the image) as well as the complexity of building the template database. Moreover, it has been shown that "every consistent recognition scheme for recognizing 3D objects must in general be model based" [20]. Although this was referring to 2D imagery, the pretext still stands and thus pixel-level matching is best suited as a final validation step.

Pixel-level validation promises the maximum discrimination capability that is achievable (again from an information theoretic point of view). Several additional considerations are implied by pixel-level validation that are not commonly lumped into standard matched filtering techniques. First, a template is generated from a modeling and simulation capability since precise information is required which cannot be created apriori by data collections (due to cost and complexity). Second, the projection of the model into the image is used to determine the best estimated segmentation. This information should then be used to (a) perform a pixel-level validation for the visible parts of the object, (b) ignore the portions of the object that are obscured, and (c) provide a consistency check that the background looks like background (as opposed to unmodeled portions of the object).

A background consistency check is crucial to avoid the "box-in-a-box" problem. This problem appears when trying to recognize different sized objects. For example, a small box looks exactly like a larger box, except (from the sensor point of view) there is additional information that would be ignored without the consistency check. In other words, it would be easy to "recognize" the smaller box within an image of the larger box (however, it would be hard to recognize the larger box given an image of the smaller box). Alternately, is half a car still a car? It is possibly still a car if the other half of the car is obscured, but not likely if the measurements indicate the other half of the car is missing (as evidenced by measurements coming from behind where the other half of the car should be). Even if the algorithm was intelligent enough to realize two known objects are easily confused for each other, a background consistency check is still required so that unknown objects will successfully be rejected.

Model-Based Matching

The fundamental premise of model-based approaches is that the desired object and few other things will obey the constraints defined by the model. The system's ultimate performance will be bounded by how well the models can predict the real world and all of its subtleties. As with other methods, inverse

methods do not currently exist for efficiently matching measured data back to these models. Therefore, current versions of model-based matching are very similar to template matching. The basic approach is "hypothesize and verify," where the geometry and physics models are used to synthesize an image corresponding to the hypothesis and then a verification technique is applied to compare the measured and synthesized data. Most of these approaches can be classified as

1. voting-based (geometric hashing, pose clustering, or generalized Hough transforms): voting in a parameter space for potential matches, or
2. alignment-based: searching for additional model-to-image matches based on the transformation computed from a small number of hypothesized correspondences.

The selectivity and an error analysis based on these approaches has been reported [21, 22, 23]. These techniques are discussed in more detail subsequently.

The different voting-based techniques primarily differ in their choice of a transformation space in which to tabulate votes to elect potential matches [24, 10]. These techniques are computationally efficient, accurate, and robust theoretically. The difficulty is in applying these techniques with a noise model. Most of the demonstrated systems assume that the robustness of the system will handle the noise. However, Grimson [23] demonstrates that the affect of the noise is dependent on location. This violates a fundamental assumption of the geometric hashing technique that the noise is independent of where the selected basis was located. Grimson suggests a modified voting technique, but the computational expense and loss in selectivity is clear in comparison to the alignment-based approach, especially as the noise increases.

Huttenlocher and Ullman [25] presents an alignment-based approach. Jacob and Alter [21] demonstrates that an alignment-based approach has better error characteristics than voting–based techniques, primarily due to the error approximations being much more accurate in the alignment-based approach (the transformation into the voting space generally make the error very difficult to estimate). Grimson [23] demonstrates, based on the selectivity of a matching set, that a large number of matches are necessary to reduce the probability of false match toward zero. Although Jacobs [21] developed a linear approach for matching points, this still does not handle the combinatorics associated with determining the initial minimal match between the image and object.

To conclude, the model-based approach has significant advantages such as limited dependence on measured data (so it is extendible to unsampled data regimes), and orders of magnitude reduction in online storage requirements. The typical disadvantage to model–based approaches is the online cost of synthesizing images corresponding to the hypothesis, especially when complex interactions occur between objects and backgrounds.

Geometric Hashing

Geometric hashing is essentially an index tabulating an exhaustive enumeration of feature values. The goal is to build an index offline so that the occurrence of a feature in the image is linked to the occurrence of those features for a particular model or models in the database. For example, given a set of features with integer labels between one and five, a hash table would record which models have which features. In addition, it may record the relative frequency of occurrence of the objects for each features value (using some apriori information about the relative frequency of occurrence of the models).

Given the following occurrence data:

- Model 1: Features 2,4
- Model 2: Features 1,5
- Model 3: Features 5,5,4,2

the hash table in Table 3.2 would be constructed.

Table 3.2. An example hash table. The table contains the number of times a feature is present for each model. The occurrence of a feature in the image can be directly indexed into potential matches from the database.

Features	Model 1	Model 2	Model 3
1	0	1	0
2	1	0	1
3	0	0	0
4	1	0	1
5	0	1	2

Now, given an image containing a feature of type 1, it is obvious that Model 2 is the only possible match in the database. Furthermore, if a feature of type 3 is found, no models match. If feature type 5 is found, Model 3 is more likely than Model 2.

Additional information, such as the look angle from which the feature was visible, may also be stored in the hash table. This is useful for further refining the hypothesis to include pose, especially when there is significant overlap between the features and models.

This is a simplistic model of geometric hashing. The principal difficulties are: (a) enumerating and binning all the images so that they can be recorded in the hash table, (b)the separability of the hash table once extensive enumeration has been accomplished, and (c) noise forces the discrete features to be treated probabilistically. This makes the matching much more complex, and once again highlights the need for metrics. References [26] and [27] implement hash tables to achieve efficient indexing.

Hough Transform

The Hough transform converts images into the parameter space of lines. This technique is presented in most pattern recognition and computer vision books [28]. The generalized Hough transform extends this technique to parameter spaces for other object descriptors (generally geometric curves and surfaces). The concept is that an edge extraction technique has been applied to the image, and now the goal is to combine the evidence to identify lines despite noise and missing segments.

Lines in a 2D image could be parameterized by an angle and the location of an intercept along an axis. However, this parameterization does not uniquely identify lines that are parallel to the axis of intercept. Many different parameterizations are possible, but for demonstration a simple angle–angle representation will be used. A notable benefit of this particular representation is that both parameters have the same domain and range. Thus, the bin size and number of bins should be the same for each parameter. The relative proportion of bins is not as obvious if, for example, slope and intercept are used as the parameters. The general goals in choosing the parameterization include minimizing the number of parameters, nondegeneracy of the parameters, computational complexity, boundedness of the parameters, and uniformity of the parameter distributions. Figure 3.5 illustrates a binning scheme that might be used for the angle–angle parameterization. Notice that the middle bins are effectively more forgiving (coarser quantization) than the bins near the edges. Also, some bins cannot possibly have any support from the image. Thus, an improvement upon this scheme would make the bins a uniform size in the image.

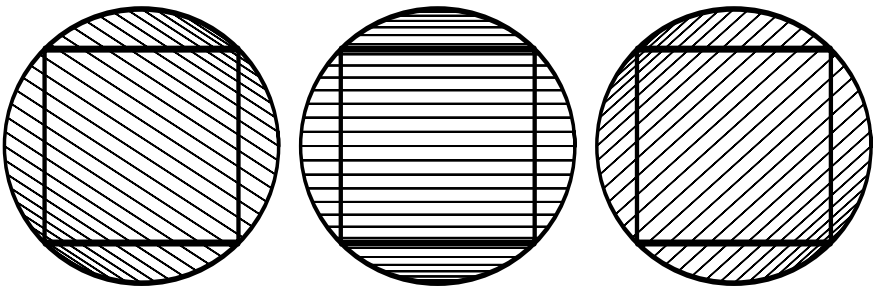


Figure 3.5. Hough transform binning. Here are three examples of the 30×30 different bins corresponding to sampling the circle every 6 degrees. The square represents the image and the circle is shown to demonstrate the construction of the bins. The bins correspond to approximately equivalent lines in the image. Any pixels falling within a bin gives support to that particular line.

Lines in a 2D image intersect a circle circumscribing the image in two locations (the circle must be around the outside of the image in order to

avoid lines tangent to the circle). These intersections uniquely define the line, and provide a minimal global parameterization of lines in the image. The circle is discretized into 30×30 bins, and a two-dimensional Hough space is created, each axis corresponding to the bin locations along the circle. Each feature in the image votes for all the bins that contain that appropriate point in the image. The relation between the image locations and Hough bins can be precomputed offline so that the online computation is minimal. Bins containing multiple features will always have more votes than bins containing few or no features. Thus, the bins with the most votes are the most likely to contain a line. The presence of a line still needs to be verified.

Evidence Accrual

Evidence accrual is most commonly another name for Bayesian statistical inference methods [29, 26], although Dempster–Schafer is also common. The approaches and applications are well beyond the scope of this chapter. In general, they are similar to Hough transforms in that ultimately a voting space is established (appropriately normalized to produce a valid probability). Both positive and negative evidence can be accrued. "Negative evidence" refers to evidence that contradicts the hypothesis. Evidence accrual can be used both for determining what models are most likely given an image, and for deciding whether to accept or reject a particular hypothesis given an image. Dempster–Schafer's advantage in this respect is that it has a framework to incorporate the *unknown* class and it can generate confidence levels with so-called belief functions [30].

Learning Approaches

Neural networks are another area that are well beyond the scope of this chapter [31]. Many different types of networks (generally biologically inspired) have been designed, all with the basic idea of taking the available input data as well as truth information and producing the desired output data. A learning (training) process is used along with feedback (when available) to generate the desired output for a given input. The networks are generally constructed by multiple levels of massively interconnected computational nodes and thresholds. Neural networks are an excellent tool for a quick assessment of the computability of a desired process. However, the inability to guarantee robust decisions and handle unknown objects limits their applicability to object recognition. The reliance of the training process on available data is its fundamental weakness, both because a model is required [20] and because it is typically difficult to predetermine how the decision boundaries generalize to new data. In particular, it is necessary to avoid overtraining, a condition in which the network memorizes the training data and recalls it perfectly but does not generate the true class boundaries.

Genetic algorithms are another biologically inspired approach to object recognition. Genetic algorithms are essentially an optimization technique akin to simulated annealing and gradient descent. The "training" is an iterative process of creation and destruction of potential solutions. Creation involves taking existing solutions and randomly mutating or hybridizing to create new solutions. All the solutions are evaluated based on a measure of success, and those that fall below some threshold are destroyed. This very useful optimization technique is appropriate when other methods of modeling and simplifying the problem are not possible.

Tree Search

Tree search algorithms are easily as numerous as pixel-level validation variations. The most common approach is to assume the data can be efficiently and correctly segmented into chunks based on local similarity constraints. Then a tree structure is used to describe the relative geometric relation between the chunks. Labels are added to the nodes of the tree that correspond to information extracted from the individual chunks. Finally, trees may be replaced by linked graphs so that missing information will not be detrimental to the matching process [16]. The primary difficulty is how to handle the nonuniqueness of the various partitionings of the image without making the search computationally intractable. "Decision trees" are an automated approach to building trees [32].

Principal Components

Principal components, also known as eigenspaces, Hotelling, and Karhunen–Loeve transforms, minimize the mean-squared error in the reconstructed data as elements are dropped from a linear basis. The concept is developed in most pattern recognition and computer vision books [32]. Variations on the basic concept have been suggested that are invariant to translation, rotation, and other useful group actions. It is both an advantage and a disadvantage that the optimal (linear) transformation is data dependent. This enables the data to be transformed such that the principal components are linearly independent. However, the ability to determine and remove any linear dependence between the coefficients representing the objects is limited to the available data. Furthermore, any functional dependence between the components cannot be addressed. Finally, the standard technique is not appropriate for object recognition, since it is optimized to minimize representational error, not discrimination. Section 3.4.2 will detail how Procrustes analysis successfully uses the eigenspace to determine the distance between objects. Reference [33] is the seminal paper on eigenfaces, and [34] develops appearance-based techniques.

Invariance

Some constraints are required so that not everything can be matched to everything. Data-driven approaches infer the constraints from the available data, whereas invariance-based approaches use a model (group action) of how the objects transform. The invariance approach is to equivalence sets of images that differ only by some (predefined) group actions. The concept is best expressed in the statistical shape analysis literature, shape is what is left after translation, rotation, and scale are removed [35]. Therefore, if the goal is to measure changes in shape, then invariance is a natural tool.

Invariance to a particular group action can be achieved many different ways. This includes both variations in the actual invariant function, but also in methods of calculating the function. In particular, invariants can be explicit or implicit. Implicit invariants are functions that are independent of the group parameter. For example, $x_1 - x_2$ is invariant to translation along the x-axis. Explicit functions achieve invariance by fixing the group parameter. This is also known as "standard position." The best example is moving the centroid of the data to the origin. No matter how the data is translated, by moving the centroid back to the origin, the compensated data is explicitly invariant to translation.

It only takes a quick example to demonstrate why invariant approaches are important. Consider a standard hypothesis-and-verify technique. Even if the operating condition parameters can be safely quantized as in Table 3.3, the total number of hypothesis is beyond exhaustive computational capabilities.

Table 3.3. Enumeration of quantized operating conditions. The total number of hypothesis is beyond exhaustive computational capabilities.

Parameter	Quantized Bins
Object type	20
Object aspect	72
Depression angle	5
Articulation (1 DoF)	36
Configuration (4 binary)	16
Obscuration	400
Correspondence	20
Netting	5
Total Hypotheses	$165,888,000,000 = 1.6 \times 10^{11}$

The most common problem with invariant-based techniques is a lack of understanding of the affect of the transformations into an invariant space. It is trivially true that a constant function is invariant to any group action, so it is only illustrative to point out that many objects are mapped to the same equivalence class by this function. An invariant basis can be derived such that it only equivalences objects that are the same up to the desired

transformation; however, the affect of noise can be substantially different for each invariant basis. Therefore the choice of invariant basis must be carefully considered. Furthermore, it is not trivial to determine the distribution of the objects in invariant space given a distribution of objects in parameter space.

Invariants provide the most promising theoretical approach, but current approaches have not achieved their potential for *solving* the object recognition problem. However, progress is being made using several different techniques. Johnson's thesis [36] is the seminal paper on spin images and their application to ladar object recognition. Funkhouser et al. [37] present spherical harmonics. Lo and Don [38] present an excellent summary of invariants formed using moments, and [39] makes significant advances.

3.4 Shape Metrics

3.4.1 Background

The motivation for metrics is illustrated by some common questions, "What is the potential efficacy of this sensor to my application?" and "How close is my algorithm to achieving the maximum achievable separability?" However, no theory of discrimination exists and therefore it has not been possible to answer these questions in a general context. Object recognition is fundamentally driven by the ability to differentiate objects. Alternately, a method is needed to measure the difference between objects.

The metrics of interest for object recognition are invariant to the modeled group action (e.g., rotation and translation). This is intuitively obvious as one would not expect the distance (measurement of shape difference) between two objects to change when those objects have been rotated.

Shape metrics provide a basis for answering these questions. However, metrics for object recognition have been notoriously difficult to find and very little work has been done in developing methods for constructing them. An exception comes from the field called statistical shape analysis or morphometrics [4, 35, 40]. Specifically, these books summarize and extend the research (primarily for biological applications) that although founded in work developed throughout the century, really accelerated with papers by Kendall [41] and Bookstein [42].

The goal in this field is to develop a shape metric based on "landmark" features. From the object recognition point of view, these are simply pixel locations of extracted features. Since shape is of fundamental interest, two objects are considered similar independent of translation, rotation, and scale. In other words, two objects are considered equivalent if they can be brought into correspondence by translating, rotating, and scaling. This is called the similarity group.

The metric developed for statistical shape analysis is commonly called the Procrustes, Procrustean, or (in one specific case) the Fubini–Study metric.

The proof that it is a metric is developed in Section 3.4.2. The Procrustes metric is a quotient metric. Intuitively, it is easy to conceptualize considering the space of objects modulo the similarity group. Quotienting the group action out of objects can be very difficult both analytically and numerically. Despite these difficulties, this does provide a constructive approach to developing metrics invariant to other group actions.

Although an object recognition system must eventually make a hard decision (i.e., make a binary decision as to whether an image is consistent with a particular hypothesized object), the concept is that this decision will occur in the final stages of the algorithm when a pixel-level validation and background consistency is performed. This chapter is applicable to both this final validation stage and the weeding or indexing problem where the goal is to order the most likely candidates first. The indexer should not predetermine how many candidates the algorithm will examine before making a final decision. The metric will naturally rank order the models based on their similarity to the image. Therefore the system will validate the best matches first (based on the extracted data or features) and continue until either the validation is highly confident in a match, it rejects all the potential matches, or a time constraint is exceeded.

A preliminary study on the fundamental separability of objects (sets of landmarks) based on this metric can be found in [43]. The need for appropriate metrics is a major theme of this chapter. There have been numerous metrics, measures, and distances proposed throughout the history of object recognition, computer vision, and pattern recognition. Alternative 3D pseudometrics are presented in [3] and [36]. More general metrics can be found in [44] which provides a nice summary of different metrics. Csiszár argues for a unique choice of metrics in [45]. Finally, [46] questions whether human perception satisfies any of the axioms of a metric, and [47] contains an excellent summary of what is known about biological vision systems.

3.4.2 Procrustean Metrics

In this section the partial Procrustes metric is derived and compared relative to other choices of a metric. The three metrics commonly used in statistical shape analysis are the *full* Procrustes, the *partial* Procrustes, and the Procrustes metric. Each metric corresponds to a different model and the appropriate selection of a given metric, in the context of the object recognition problem, depends upon the sensor being employed.

Let G be the similarity group — the group generated by the rotation, translation, and scale (dilation)— and consider the componentwise group action $\mathbb{R}^m \times \cdots \times \mathbb{R}^m \times G \to \mathbb{R}^m \times \cdots \times \mathbb{R}^m$. The associated quotient space, Σ_m^k, is called the shape-space of order (m, k)

$$\Sigma_m^k \equiv \mathbb{R}^m \times \cdots \times \mathbb{R}^m / G,$$

where k corresponds to both the number of features and the number of copies of $\mathbb{R}^m$. Note that $m = 2$ represents a two-dimensional "image" space, $m = 3$ represents three-dimensional "volume" space, and higher values of m represent higher-dimensional feature spaces. The procedure to calculate the (representative) equivalence classes of this quotient space as well as a metric on this space follows.

The first step in determining the shape of an object, $\mathbf{X}$, is to quotient out translation by moving the centroid along each axis to a standard position (the origin)

$$\mathbf{X} \to \mathbf{X} - \bar{\mathbf{X}},$$

where $\bar{\mathbf{X}}$ represents the centroid of $\mathbf{X}$. This can be rewritten as a matrix product

$$\mathbf{X} \to \underbrace{-\frac{1}{k}\begin{bmatrix} 1-k & 1 & 1 & \dots & 1 \\ 1 & 1-k & 1 & \dots & 1 \\ 1 & 1 & 1-k & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1-k \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ \vdots & \vdots & \vdots \\ x_k & y_k & z_k \end{bmatrix}}_{\mathbf{X}}$$

where $\mathbf{C}$ is the $k \times k$ centering matrix (the matrix that takes the centroid to the origin), which is symmetric and idempotent (so $\mathbf{C} = \mathbf{C}^T = \mathbf{C}\mathbf{C}$).

The centering matrix has rank $k-1$ and taking the Cholesky decomposition of $\mathbf{C}$ results in the first row consisting entirely of zeros. Therefore, the Cholesky decomposition is a method for removing a linearly dependent row from $\mathbf{C}$. In particular, the Helmert submatrix, $\mathbf{H}$, is defined by

$$\mathbf{H}^T\mathbf{H} = \mathbf{C},$$

where $\mathbf{C}$ is the centering matrix presented above, and $\mathbf{H}$ is the Cholesky decomposition into a lower triangular matrix followed by the removal of the row of zeros. Thus $\mathbf{H}$ removes translation and reduces the dimension of the object (analogous to dropping one point). This is exactly the number of parameters that were removed from the group action (one parameter for the translation of each axis). Therefore, the Helmertized object, $\hat{\mathbf{X}} = \mathbf{H}\mathbf{X}$, is of dimension $(k-1) \times m$ and is a representative of the original object $\mathbf{X}$ in the quotient space for translation. Note that $\mathbf{H}$ can be computed directly [35].

The scale is then removed by the scaling operation $\mathbf{x} \mapsto \mathbf{x}/\|\mathbf{x}\|$, where $\|\cdot\|$ is the l_2-norm as this norm is invariant to rotation. The resulting space, $\mathbb{R}^m \times \cdots \times \mathbb{R}^m$ modulo translation and scale, is called the preshape space and the equivalence class of $\mathbf{X}$ is $\mathbf{W} = \mathbf{H}\mathbf{X}/\|\mathbf{H}\mathbf{X}\|$. Elements of the preshape space are denoted by $\mathbf{W}$ and will also be of dimension $(k-1) \times m$.

A continuous global quotient space does *not* exist for rotation [4]. However, the distance between the equivalence classes of $\mathbf{W}$ modulo $\mathrm{SO}(m)$ can be defined and does exist,

$$d_P(\mathbf{X}_1, \mathbf{X}_2) = \inf_{\mathbf{R}_1, \mathbf{R}_2 \in \mathrm{SO}(m)} \|\mathbf{W}_1\mathbf{R}_1 - \mathbf{W}_2\mathbf{R}_2\|, \tag{3.1}$$

where the matrix norm $\|\mathbf{W}\| = \sqrt{\mathrm{Tr}(\mathbf{W}^T\mathbf{W})}$ is once again the l_2-norm, and $\mathrm{SO}(m)$ is the set of $m \times m$ rotation matrices. This function can be simplified by defining $\mathbf{R} = \mathbf{R}_1\mathbf{R}_2^{-1}$ and using $\mathrm{Tr}(\mathbf{BAB}^{-1}) = \mathrm{Tr}(\mathbf{A})$ to obtain

$$\begin{aligned} d_P(\mathbf{X}_1, \mathbf{X}_2) &= \inf_{\mathbf{R}_1, \mathbf{R}_2 \in \mathrm{SO}(m)} \|\mathbf{W}_1\mathbf{R}_1 - \mathbf{W}_2\mathbf{R}_2\| \\ &= \inf_{\mathbf{R}_1, \mathbf{R}_2 \in \mathrm{SO}(m)} \left\|(\mathbf{W}_1\mathbf{R}_1\mathbf{R}_2^{-1} - \mathbf{W}_2)\mathbf{R}_2\right\| \\ &= \inf_{\mathbf{R}, \mathbf{R}_2 \in \mathrm{SO}(m)} \|(\mathbf{W}_1\mathbf{R} - \mathbf{W}_2)\mathbf{R}_2\| \\ &= \inf_{\mathbf{R} \in \mathrm{SO}(m)} \|(\mathbf{W}_1\mathbf{R} - \mathbf{W}_2)\|. \end{aligned}$$

The properties of a metric can be found in any standard math reference book [48]. The properties of the Procrustes metric are ultimately induced by the properties of the norm. In particular, the triangle inequality property of the metric follows from the triangle inequality for the norm

$$\begin{aligned} d_P(\mathbf{X}_1, \mathbf{X}_3) &= \inf_{\mathbf{R} \in \mathrm{SO}(m)} \|\mathbf{W}_1\mathbf{R} - \mathbf{W}_3\| \\ &= \inf_{\mathbf{R}, \mathbf{R}_2 \in \mathrm{SO}(m)} \|\mathbf{W}_1\mathbf{R} - \mathbf{W}_2\mathbf{R}_2 + \mathbf{W}_2\mathbf{R}_2 - \mathbf{W}_3\| \\ &\leq \inf_{\mathbf{R}, \mathbf{R}_2 \in \mathrm{SO}(m)} (\|\mathbf{W}_1\mathbf{R} - \mathbf{W}_2\mathbf{R}_2\| + \|\mathbf{W}_2\mathbf{R}_2 - \mathbf{W}_3\|) \end{aligned}$$

because by defining $\overline{\mathbf{R}} = \mathbf{R}\mathbf{R}_2^{-1}$

$$\begin{aligned} &\inf_{\mathbf{R}, \mathbf{R}_2 \in \mathrm{SO}(m)} (\|\mathbf{W}_1\mathbf{R} - \mathbf{W}_2\mathbf{R}_2\| + \|\mathbf{W}_2\mathbf{R}_2 - \mathbf{W}_3\|) \\ &\quad = \inf_{\overline{\mathbf{R}} \in \mathrm{SO}(m)} \left\|\mathbf{W}_1\overline{\mathbf{R}} - \mathbf{W}_2\right\| + \inf_{\mathbf{R}_2 \in \mathrm{SO}(m)} \|\mathbf{W}_2\mathbf{R}_2 - \mathbf{W}_3\| \\ &\quad = d_P(\mathbf{X}_1, \mathbf{X}_2) + d_P(\mathbf{X}_2, \mathbf{X}_3). \end{aligned}$$

Thus this function defines a metric on the shape space. Expanded out in terms of the l_2-norm, the distance is

$$\begin{aligned} d_P(\mathbf{X}_1, \mathbf{X}_2)^2 &= \inf_{\mathbf{R} \in \mathrm{SO}(m)} \|\mathbf{W}_1\mathbf{R} - \mathbf{W}_2\|^2 \\ &= \inf_{\mathbf{R} \in \mathrm{SO}(m)} \mathrm{Tr}((\mathbf{W}_1\mathbf{R} - \mathbf{W}_2)^T(\mathbf{W}_1\mathbf{R} - \mathbf{W}_2)) \\ &= \inf_{\mathbf{R} \in \mathrm{SO}(m)} \mathrm{Tr}(\mathbf{W}_1^T\mathbf{W}_1 + \mathbf{W}_2^T\mathbf{W}_2 - 2\mathbf{R}^T\mathbf{W}_1^T\mathbf{W}_2) \\ &= \mathrm{Tr}(\|\mathbf{W}_1\|^2 + \|\mathbf{W}_2\|^2 - 2 \sup_{\mathbf{R} \in \mathrm{SO}(m)} \mathbf{R}^T\mathbf{W}_1^T\mathbf{W}_2) \\ &= 2(1 - \sup_{\mathbf{R} \in \mathrm{SO}(m)} \mathrm{Tr}(\mathbf{R}^T\mathbf{W}_1^T\mathbf{W}_2)). \end{aligned} \tag{3.2}$$

Globally Optimal Solution

Finding the argument of the supremum in (3.2) is equivalent to finding the argument of the infimum in (3.1). An analytic globally optimal solution to this optimization problem follows using singular value decomposition

$$\mathbf{W}_1^T\mathbf{W}_2 = \mathbf{U}^T\mathbf{\Lambda}\mathbf{V},$$

where $\mathbf{U}$ is orthonormal, $\mathbf{V} \in \mathrm{SO}(m)$, and $\mathbf{\Lambda}$ is zero with a diagonal of singular values. The singular values are positive except the smallest, λ_m, is the negative if and only if $\det(\mathbf{W}_1^T\mathbf{W}_2) < 0$ [35]. Note that the singular values, $\{\lambda_i\}_{i=1}^m$, are the square roots of the eigenvalues of the matrix $\mathbf{W}_1^T\mathbf{W}_2\mathbf{W}_2^T\mathbf{W}_1$ and has a maximum $\sum_{i=1}^m \lambda_i = 1$ (which corresponds to $\mathbf{W}_1$ and $\mathbf{W}_2$ matching).

Now the metric can be written as

$$d_P(\mathbf{X}_1, \mathbf{X}_2)^2 = 2(1 - \sup_{\mathbf{R}\in\mathrm{SO}(m)} \mathrm{Tr}(\mathbf{R}^T\mathbf{U}^T\mathbf{\Lambda}\mathbf{V})).$$

Since $\mathrm{Tr}(\mathbf{R}^T\mathbf{U}^T\mathbf{\Lambda}\mathbf{V}) \leq \mathrm{Tr}(\mathbf{\Lambda})$ with equality occurring when $\mathbf{R} = \mathbf{V}\mathbf{U}^T$, the supremum occurs at $\mathbf{R} = \mathbf{V}\mathbf{U}^T$. Thus

$$d_P(\mathbf{X}_1, \mathbf{X}_2) = \sqrt{2\left(1 - \sum_{i=1}^m \lambda_i\right)}. \tag{3.3}$$

This metric is called the partial Procrustes metric in the statistical shape literature [35]. The full Procrustes metric between two objects $\mathbf{X}_1$ and $\mathbf{X}_2$ can be written with respect to their corresponding preshapes $\mathbf{W}_1$ and $\mathbf{W}_2$ as

$$d_F(\mathbf{X}_1, \mathbf{X}_2) = \inf_{\mathbf{\Gamma}\in\mathrm{SO}(m),\beta\in\mathbb{R}} \|\mathbf{W}_2 - \beta\mathbf{W}_1\mathbf{\Gamma}\|.$$

Expressing this metric in terms of the eigenvalues of $\mathbf{W}_1^T\mathbf{W}_2\mathbf{W}_2^T\mathbf{W}_1$, as done above for the partial Procrustes metric, gives

$$d_F(\mathbf{X}_1, \mathbf{X}_2) = \sqrt{1 - \left(\sum_{i=1}^m \lambda_i\right)^2}. \tag{3.4}$$

The Procrustes metric is defined as the closest great circle distance between $\mathbf{W}_1$ and $\mathbf{W}_2$ on the preshape sphere. Kendall [41] shows that the preshape space is indeed a sphere. In fact, the Procrustes distance on $\sum_2^k$ is equivalent to the Fubini-Study metric on $\mathbf{CP}^{k-2}(4)$. From trigonometry it follows

$$d(\mathbf{X}_1, \mathbf{X}_2) = \arccos\left(\sum_{i=1}^m \lambda_i\right). \tag{3.5}$$

Finally, another distance measure is the full generalized Procrustes metric defined by

$$d_g(\mathbf{X}_1, \mathbf{X}_2) = \inf_{\substack{\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_2 \in \mathrm{SO}(m) \\ \beta_1, \beta_2 \in \mathbb{R} \\ \boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2 \in \mathbb{R}^m}} \|(\beta_1 \mathbf{X}_1 \boldsymbol{\Gamma}_1 + \mathbf{1}_k \boldsymbol{\gamma}_1) - (\beta_2 \mathbf{X}_2 \boldsymbol{\Gamma}_2 + \mathbf{1}_k \boldsymbol{\gamma}_2)\|$$
$$= d_F(\mathbf{X}_1, \mathbf{X}_2).$$

where an additional constraint such as $\beta_1 \beta_2 = 1$ is necessary to prevent degenerate solutions. The full generalized form is typically used in statistical shape analysis for estimating the mean shape of a set of objects. This development demonstrates it for comparing just two objects. Therefore, the only difference between the full generalized form and the full ordinary form (as it is called in [35]) is the inclusion of the translation in the optimization. For two objects, the full generalized form is equivalent to the full ordinary Procrustes metric (by using the fact that the objects can be centered, i.e., have their centroids moved to the origin, without loss of generality). This follows since the cross terms in the norm due to the translation will disappear because $\mathbf{C}\mathbf{1}_k = \mathbf{0}$ for any centering matrix $\mathbf{C}$ such as the Helmert matrix ($\mathbf{C} = \mathbf{H}^T\mathbf{H}$).

The three metrics can be interpreted as a path in shape space corresponding to the chord length (partial Procrustes), arc distance (standard Procrustes), and orthogonal length (full Procrustes). It should be noted that these three metrics are order preserving relative to each other, i.e.,

$$\begin{aligned} d_F(\mathbf{X}_1, \mathbf{X}_2) &< d_F(\mathbf{X}_1, \mathbf{X}_3) \\ \Updownarrow & \\ d_P(\mathbf{X}_1, \mathbf{X}_2) &< d_P(\mathbf{X}_1, \mathbf{X}_3) \\ \Updownarrow & \\ d(\mathbf{X}_1, \mathbf{X}_2) &< d(\mathbf{X}_1, \mathbf{X}_3). \end{aligned}$$

Therefore, any of these metrics can be used if all that matters is the relative ordering of the objects. Dryden and Mardia [35] argues that the full Procrustes distance is the natural choice as it optimizes over the full set of similarity parameters and because it "appears exponentiated in the density for many simple probability distributions for shape" (p.44).

Point Correspondence

The existing literature on shape metrics presumes the point correspondence between the objects is known or is handled elsewhere. This is also known as the labeling problem. Conceptually, this is the internal labeling problem (matching k image features to k model features) as opposed to the external labeling problem (choosing which subset of the image features and which subset of the model features to compare). An exhaustive computation of the external problem requires $\binom{P}{k}$ computations (where P is the number of image features) for each of the $\binom{K}{k}$ feature sets (where K is the number of model features) of each

model in the database. The internal problem requires $k!$ computations (exhaustive). See [49] for more information. Algorithms for point correspondence that are polynomial in complexity were referenced in Section 3.3.1, however the ultimate goal is to develop an analytic solution to this problem so that theoretical analysis of the label-invariant metric is possible.

An implicit assumption has been made that the same subset of points can be found on the image and the object to compare. Zhang [50] presents a technique for consistently extracting the same number of points using Legendre polynomials. The internal labeling problem for Procrustes is explored in [51].

3.5 Current Approach

The subsequently described algorithm handles most of the technical challenges previously discussed for a successful 3D object recognition system. The conclusions and future research sections will discuss desired improvements.

The method is based on looking at "spheres" of data extracted from the image and comparing these to "spheres" of data extracted from the models. This method specifically avoids feature detection for improved robustness, and this approach specifically handles articulated objects. Recognizing such an object can involve a search in a high-dimensional space that involves all the articulating degrees of freedom, in addition to the usual unknown viewpoint. This algorithm uses invariants to reduce the search space to a manageable size.

The input is a range image (in rectangular coordinates) of an unknown object in an unknown articulation position, such as a backhoe with each link in an arbitrary position. The desired output of the system is the identity of the object, and the articulation and viewpoint parameters. The object is identified using a database of known models. Since an object, like the backhoe, could have ten degrees of freedom (DoF) it is infeasible to store all images of each articulated object. Subsampling and storing 10 images of each DoF for just this one object would require 10^{10} images, which is prohibitive. Consequently an efficient procedure for representing and searching a database of objects is described.

Unique features of the method described in this chapter include an integral approach to feature detection and recognition that is more efficient than considering them separately and improves robustness to noise. The approach includes scanning the image, so it avoids the combinatorial explosion of hypothesizing feature correspondences [52, 53, 54]. This approach simultaneously uses the data from all the points in a neighborhood (thus the robustness to noise and obscuration).

3.5.1 The Sensor Model

The sensor model characterizes the transformation group acting on the object, and the projection from $\mathbb{R}^3$ to $\mathbb{R}^3$. The model presented here is written for

four points on an object undergoing the same rigid motion. This is a good starting point, with the articulation being incorporated subsequently.

The transformation group is the rigid transformation (rotation and translation). The relation between the measured feature location, $\{u, v, r\}$, and the model feature location, $\{x, y, z\}$ (expressed in rectangular coordinates) is

$$\begin{bmatrix} u_1 & u_2 & u_3 & u_4 \\ v_1 & v_2 & v_3 & v_4 \\ r_1 & r_2 & r_3 & r_4 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} & & & a \\ & \mathbf{R} & & b \\ & & & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

such that $\mathbf{R} \in \mathrm{SO}(3)$ (rotations), and $\{a, b, c\}$ is a vector denoting a rigid translation of $\mathbb{R}^3$. The trailing 1's are included to permit writing the translation as a matrix product. As written, both the front and back of the object would appear in the image. The fact that a ladar cannot see the through an object is why the projection is referred to as 2.5D in Section 3.3.1.

3.5.2 Invariants and the Object–Image Relations

Object–image relations express a geometric relation (constraint) between a 3D object and its image. This is a general approach and the particular invariants and number of points required for an invariant depends upon the transformation group associated with the sensor model.

Although constructing invariants is difficult, once an invariant has been found it is typically simple to compute. Object–image (O-I) relations are an application of invariance theory as applied to the world viewed by a sensor (ladar in this case). O-I relations provide a formal way of asking, "What are all possible images of this object?" and "What are all possible objects that could produce this image?" Clearly, this is a very powerful formalism and it is well suited to object recognition.

Recent research has yielded the fundamental geometric relation between "objects" and "images" (for RADAR, SAR, UHRR, EO, IR, and other sensors) [55, 56, 57, 58, 59, 52]. Although the object–image relations are very simple for ladar, they are useful for demonstrating the benefits of using covariants.

Fundamentally the O-I relations can be viewed as the result of elimination of the unknown parameters in the model describing the projection of the 3D world onto the sensor. Thus, the ladar O-I relation can be derived by eliminating the group parameters associated with object motion, and the parameters associated with the ladar look direction (for this case, the group actions are the same). The result is an equation relating the object to its ladar range image written in terms of their associated invariants. More specifically, the equation relates 3D rigid invariants (inter-scatterer 3D Euclidean distances, determinants, or inner products) to rigid invariants measured from the ladar image.

3.5.3 The ladar Object–Image Relation

The above model of a ladar is essentially an orthographic projection from 3D to 3D. An invariant of 3D objects (undergoing rigid transformations) requires a minimum of four points (in general position), $P_i = \{X_i, Y_i, Z_i, 1\}$. Let the point P_i correspond to the $i-$th column of a 4×4 matrix. By translating and rotating appropriately, one can always transform the points into the standard position,

$$\begin{bmatrix} 0 & I_1 & I_2 & I_3 \\ 0 & 0 & I_4 & I_5 \\ 0 & 0 & 0 & I_6 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

where $\{I_1, I_2, I_3, I_4, I_5, I_6\}$ are the object model invariants. They are invariants under 3D rigid transformations. Linear algebra shows a unique transformation (up to sign) exists to make this change of basis. It is not obvious without further explanation, but the invariants are functions of the Euclidean distance, determinants, and inner products.

The range image can also be transformed to the standard position,

$$\begin{bmatrix} 0 & i_1 & i_2 & i_3 \\ 0 & 0 & i_4 & i_5 \\ 0 & 0 & 0 & i_6 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

where $\{i_1, i_2, i_3, i_4, i_5, i_6\}$ are the image invariants. They are also invariant under 3D rigid transformations.

The fundamental object–image relation can now be determined by solving the model projection equations with respect to the unknown rotation and translation. The resulting object–image relations are

$$i_j = I_j, \qquad \forall j \in \{1, 2, 3, 4, 5, 6\} \tag{3.6}$$

An algorithm that uses these object–image relations does not need to worry about any rigid transformations. In other words, use of the object–image relations guarantees that all the continuous group actions that were modeled have been factored out of the resulting equalities. What remains are the discrete permutations, i.e., the correspondence and ordering of sets of features between the object and image. However the formulation, as presented above, is not feasible for two reasons:

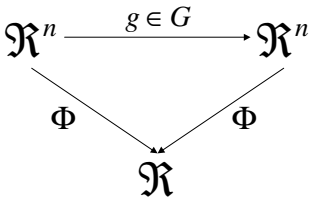
1. *Noise issues: As presented, these equations treat the initial points as "special." A formulation is desired that treats all points equivalently, as this should inject some robustness to noise.*
2. Complexity issues: Arnold [49] demonstrated that the complexity of a brute force search of the discrete permutations is not computationally feasible in general.

Therefore a modified formulation is desired that provides an invariant relation between the object and image, and also handles the noise and complexity issues.

3.5.4 Covariants

Covariants provide an alternative approach to determining O-I relations by considering equivariant functions. This section will formalize the approach, including how projection is naturally handled in the technique.

Let G be a Lie group (a continuous transformation group) acting on an n-dimensional manifold M, $\phi : G \times M \to M : (g, p) \mapsto gp$. An important example of such an action is $G = \mathrm{GL}_3(\mathbb{R})$, the set of invertible matrices, acting on the function space $M = \prod_{i=1}^{3} \mathbb{R}_i^3$. An element of this function space is simply an ordered triple,$\{p_i\}_{i=1}^{3}$, where each "point" p_i is an ordered 3-tuple, $\{x_i, y_i, z_i\}^T$. Here the action is a componentwise matrix multiplication—hence the action is linear. A G-invariant function is a function Φ satisfying $\Phi(gp) = \Phi(p)\ \forall p \in M\ \forall g \in G$. The following figure characterizes an invariant function Φ under an action $\phi : G \times \mathbb{R}^n \mapsto \mathbb{R}^n$



where the horizontal arrow denoted by g is the induced map $\phi_g : \mathbb{R}^n \to \mathbb{R}^n : p \mapsto \phi(g, p)$.

With respect to how this linear action is used in applications, view p as the object variables (i.e., 3D coordinates), and $q = \rho(gp)$ as the image variables (i.e., 2D coordinates), where ρ is a projection from 3-space to 2-space, e.g.,

$$\rho = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & c \end{bmatrix}$$

and where c is a fixed parameter. The variable q is embedding into 3-space—thus enabling standard invariant theoretic techniques. Embedding $\boldsymbol{\rho} \hookrightarrow \overline{\boldsymbol{\rho}}$, where

$$\overline{\boldsymbol{\rho}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & c \\ 0 & 0 & 1 \end{bmatrix}$$

and noting that this gives an automorphism of $G = \mathrm{GL}_3(\mathbb{R})$ allows one to write $\overline{q} = (\overline{\rho} g)p = \overline{g}p$. Using the property characterizing an invariant function, $\Phi(\overline{g}p) = \Phi(p)\ \forall p \in M \quad \forall \overline{g} \in G$, it follows $\Phi(\overline{q}) = \Phi(p)$. This equation provides the relationship between the "object" variables p and the "image" variables q. Thus the O-I relation is $f(p, q) = \Phi(\overline{q}) - \Phi(p) = 0$. Computationally, this idea can be implemented by (1) finding the invariants of the given

group action, and (2) "equating invariants" $\Phi(\overline{q}) = \Phi(p)$, and (3) eliminating any artificial components associated with the embedding (in the example, an artificial component to $\overline{q}$ was introduced, namely the third component). Note that the invariants found in step one have already eliminated the group parameters from the second step. It should be noted that there are a set of fundamental invariants $\Phi_i \quad i = 1, \ldots, k$ for each group action. Hence in practice there is a set of O-I relations $\Phi_i(\overline{q}) - \Phi_i(p) = 0$ for $i = 1, \ldots, k$. The technique of Lie group analysis makes determination of the invariants relatively simple. This directly avoids the conceptually simple but often computationally difficult task of eliminating the group parameters and camera parameters directly.

The latter approach naturally lends itself to consideration of covariant functions. A covariant involves two actions. The figure below characterizes the definition of a covariant function Φ under the two actions $\phi : G \times \mathbb{R}^n \to \mathbb{R}^n$ and $\psi : G \times \mathbb{R} \to \mathbb{R}$

$$\begin{array}{ccc} \mathfrak{R}^n & \xrightarrow{\phi \in G} & \mathfrak{R}^n \\ \downarrow \Phi & & \downarrow \Phi \\ \mathfrak{R} & \xrightarrow{\psi} & \mathfrak{R} \end{array}$$

where the two horizontal arrows correspond to the two induced maps $\phi_g : \mathbb{R}^n \to \mathbb{R}^n : p \mapsto \phi(g, p)$ and $\psi_g : \mathbb{R} \to \mathbb{R} : q \mapsto \psi(g, q)$. Only nontrivial choices of Φ, ϕ, and ψ are interesting. Choosing ψ as the identity is equivalent to absolute invariants. As with invariants, the goal is to find a basis from which all other covariants can be written. Vector-valued covariants, $\Phi : \mathbb{R}^n \mapsto \mathbb{R}^m$, are a fairly simple extension that will be used below. The desire is to find a covariant with a small m.

The defining property of a covariant function Φ is $\Phi(gp) = g\Phi(p)\ \forall p \in M\ \forall g \in G$. The covariants are constructed into a set of O-I relations $\Phi_i(gp) = g\Phi_i(p)$ for $i = 1, \ldots, k$. Similar to the case with invariant O-I relations, any artificial components associated with the embedding must be eliminated. Note that the group parameters have not been eliminated. The benefit of using covariants is that it allows one to explicitly solve (estimate) for the group transformation to move back to the standard position. This advantage follows from the discussion on alignment versus voting techniques in Section 3.3.3. Similarly, noise analysis is easier to perform in the original parameter space.

Covariants arise with the rigid group, $E_n = SO_n \rtimes \mathbb{R}^n$, acting componentwise on the product space $\prod_{i=1}^m \mathbb{R}^n$. Consider the case $n = 2$. The two group actions are

$$\begin{aligned} \phi : (SO_2 \rtimes R^2) \times \prod_{i=1}^{m} \mathbb{R}_i^2 &\to \mathbb{R}^2 \\ : ((\mathbf{A}(\theta), b), \{p^i\}_{i=1}^m) &\mapsto \{\mathbf{A}(\theta)p^i + b\}_i^m \end{aligned}$$

and

$$\begin{aligned}\mu : (SO_2 \rtimes R^2) \times \mathbb{R}_i^2 &\to \mathbb{R}^2 \\ : ((\mathbf{A}(\theta), b), q) &\mapsto \mathbf{A}(\theta)q + b,\end{aligned}$$

where

$$\mathbf{A} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix},$$

$$\mathbf{b} = \begin{bmatrix} t_x \\ t_y \end{bmatrix},$$

and

$$\mathbf{p}^i = \begin{bmatrix} x^i \\ y^i \end{bmatrix}.$$

For brevity, denote $\mathbf{p} = \{\mathbf{p}^i\}_{i=1}^m$.

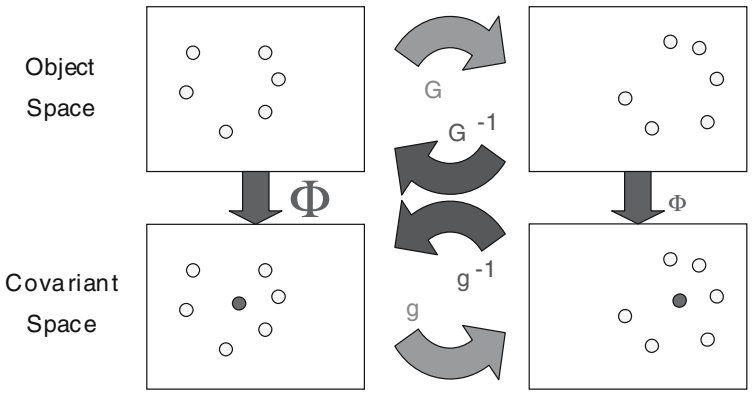


Figure 3.6. Transformations in the object space induce transformations in the covariant space. The top row, from left to right, represents points in object space being translated and rotated by G. The bottom row shows the covariant (the dark point in the middle; the object points are shown for reference) representation, specifically the centroid for this example. From left to right, the centroid is transformed by g, the action induced by G, the object transformation. The inverse g^{-1} is readily calculated, and by applying G^{-1} to the object it can be returned to its canonical coordinate system.

This system gives the covariant function "centroid"

$$\Phi = \frac{1}{m} \begin{Bmatrix} \sum_{i=1}^m x^i \\ \sum_{i=1}^m y^i \end{Bmatrix}.$$

An example is shown in Fig. 3.6. This result easily generalizes to 3D.

Similarly it can be shown that the eigenvectors are covariant under rotation in 3D. These two covariants form the basis of the object recognition algorithm to be described subsequently.

Advantages of Covariants

The advantage of using a covariant-based method is that for a large number of points the transformations can still be computed efficiently. Whereas invariants require searching $\binom{p}{r}$ combinations of features, where p is the number of image points and r is the number of features required to compute the invariants, the covariant-based technique scans the image (a neighborhood is defined by a sphere) thereby avoiding the combinatorial problem. Therefore, the covariant-based approach is computationally more efficient and more robust to noise. Also, as previously mentioned, the covariant-based approach is conducive to alignment versus voting techniques. Finally, noise analysis is easier to perform in the original parameter space.

Covariants in the Algorithm

Covariants are simply an (improved) approach to achieving invariance. Specifically, calculating covariants is an intermediate step toward transforming the selected data into a canonical coordinate system. To summarize, covariants appear in two places in the algorithm that follows:

1. *Translation: A covariant of translation is the centroid.*
2. Rotation: The eigenvectors are covariant with respect to rotation.

3.5.5 Articulation Invariants?

Invariance can be used to reduce the articulation problem. When considering invariants, the imperative question is "what transformations should the function be invariant with respect to?" For instance, when the same object is viewed from different viewpoints, invariance with respect to the viewpoint transformation is useful. In ladar range images, the viewpoint invariants are the rigid invariants (in a rectangular coordinate system). This is a well-defined group of transformations and it applies to any (static) object in the ladar range images, i.e., the transformation group is independent of the object. Thus, the viewpoint invariants can be applied generically to all objects.

When it comes to articulation this is no longer the case. Each object has different articulation degrees of freedom, i.e., a different transformation group. An object's DoF (and therefore its transformation group) cannot be determined from a single image. Therefore, while it is mathematically possible to find articulation invariants for each individual object, generic articulation invariants that apply to all objects do not exist. Lacking articulation invariants, the goal is to turn as many of the articulation DoF into generic viewpoint DoF as possible. The remaining DoF are parameterized and used to define a manifold with respect to the canonical coordinate system.

3.5.6 Dividing the Object

To simplify the articulation problem, viewpoint invariants are applied to parts of the object (subobjects). To avoid explicit segmentation, these smaller parts are not necessarily the "functional" object parts. They are arbitrary sections of the object as partitioned by a sphere of a certain center and radius. For example, one sphere may contain part of the body of a backhoe, and another may contain a joint of a backhoe's arm. The sphere that contains a rigid body part has viewpoint DoF but no articulation. The sphere that contains a joint has both viewpoint and articulation DoF. However this joint has only one articulation parameter, namely the angle between the two segments of the arm. All other DoF of these arm segments have been turned into viewpoint DoF. In other words, using this approach, the articulation of each arm segment is independent of the backhoe's body. The joint within each sphere is viewed as a separate object that is seen from an unknown viewpoint. Consequently dividing the backhoe into subobjects reduces the number of articulation DoF from ten to usually at most two. Viewpoint invariance methods can now be used for each subobject such as the joint, obtaining invariants that depend only on the angle of the joint. These invariants are a smooth function of the angle.

A major advantage of the object division is the use of so-called "global" invariants of each subobject. A global invariant is a function that depends on the entire subobject rather than on isolated features such as points or lines, i.e., a global invariant is a function of all the voxels within the sphere. This achieves two purposes: (a) it avoids the problem of feature extraction, with the high sensitivity associated with feature-based methods; (b) it avoids calculating global invariants of the whole object, which would be sensitive to occlusion and missing parts.

3.5.7 Method Summary

1. Divide each modeled object into subobjects.
2. Find global viewpoint (rigid) invariants for each subobject as functions of its articulation parameters.
3. Use these invariant manifolds for storage, matching, and identification.

A brief description of these steps follows in the next section.

3.6 Implementation

3.6.1 Object Division

Each modeled object is divided into parts that are contained within spheres. Spheres are utilized since they are preserved under rotation. An important

question is how to choose the spheres' centers and radii. Ideally all possible centers and radii would be used, but of course this is infeasible. Consequently, a finite set of radii starting from the biggest, containing the whole object, to the smallest, in which the data looks planar, are selected. This results in the description of the object on different scales. For each radius a set of centers is chosen. Sufficient numbers of centers are necessary to describe an object uniquely at a given scale. Although the object is 3D, its visible range image is described as a 2D array of ranges. Currently, a rectangular grid is draped over this array to describe where the sphere centers will be placed. Nominally, the grid spacing is some integer subsampling of the array at some fraction of the radius. The sphere centers are placed at the grid coordinates, $\{\theta, \phi, \rho\}$. Obviously the division is dependent on the image since each image will have a different grid and it is different from the grid on the image stored in the database. However, the invariant functions calculated on each sphere vary smoothly from one center to another. Consequently, it is easy to interpolate between grid points when matching is performed.

3.6.2 Finding Invariants

Once the object has been partitioned into (generally overlapping) spheres, the invariants of these subobjects are calculated. Note that invariants of a 3D object as a whole cannot be used since only partial views of the object are visible to the range sensor. Thus, at a minimum, a few different views of each object, such as front, back, and sides are required. The views are chosen such that any other view has the same invariants as one of these views. Thus, invariants are calculated for representative views, at various scales, and stored in a database. These are used as reference "models."

There are several ways to calculate such invariants. Ideally, the ones chosen are the least sensitive to changes in the boundaries of the subobjects, resulting from changing the radius or center of the sphere. Currently, the covariants explained in Section 3.5.4 are used to transform the sphere into a canonical, or standard, coordinate system. Specifically, the centroid of the subobject is the canonical origin, and the eigenvectors form the canonical axes. The new origin and axes are independent of the viewpoint; therefore the new coordinate system is viewpoint invariant. See Figure 3.7 for a simplistic example.

Transforming the subobject into this system, the grid point (sphere center) now has new coordinates that are invariant since they are given in the invariant coordinate system. Hence for each subobject (or sphere) three invariants are extracted, namely the 3D coordinates of the sphere's center in the invariant coordinate system. By using invariants to describe every grid point, the full description of the object is invariant. This description does not depend on point features and is insensitive to occlusion. A noteworthy complication is the affect of image discretization. This approach assumes planar patches connect the data points in order to facilitate the necessary integration and normalization to marginalize the affects of discretization [3].

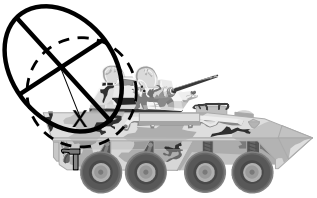


Figure 3.7. A 2D analogy to the ellipsoid fit. The dashed circle determines the data to be fitted by the ellipse. The axes of the ellipse (eigenvectors) define a canonical coordinate system. Thus any point X within the sphere is invariant in these coordinates.

The current approach and results shown here are based on this simple choice of invariants. This choice was made to provide maximum robustness to low numbers of pixels-on-target. As previously mentioned, all of the points are invariant once they have been expressed in the canonical coordinate system. The results are excellent; however, further discrimination and robustness would be achievable by using more information about the data. Obvious alternative approaches include using the spin images, spherical harmonics, or moments of the "sphere of data." Ultimately, the chosen method should include a shape metric, such as the Procrustes metric presented in Section 3.4.2.

3.6.3 Indexing

The invariants found above are functions of the articulation parameters. Denoting the invariants of each grid point by a vector $\mathbf{x}$ and the articulation parameters by a vector $\mathbf{u}$, the invariants are $\mathbf{x}(\mathbf{u})$. Indexing amounts to inverting these functions, i.e., given the invariant coordinates $\mathbf{x}$ in the image, find the articulation parameters $\mathbf{u}(\mathbf{x})$. To do that, the above relations are represented as a surface in a hyperspace, namely $f(\mathbf{x}, \mathbf{u}) = 0$. To build this surface, the articulation parameters are varied, and for each vector $\mathbf{u}$ all the corresponding vectors $\mathbf{x}$ are found. In this hyperspace, the voxels lying on the surface f are marked by 1 and all other voxels are left as 0. This is a digital representation of the hypersurface. This is done off-line for every model in the database. The functions for all models are thus represented in the hyperspace. Thus indexing has been obtained such that given the invariant coordinates $\mathbf{x}$, the corresponding models can be found with articulations $\mathbf{u}$. This can be done by intersecting all the hypersurfaces with the hyperplane $\mathbf{x} =$ constant. For most points $\mathbf{x}_i$ there will be relatively few corresponding models, since most models do not go through all points in the hyperspace even with articulation. Thus the indexing space is rather sparse.

3.6.4 Matching

Given an image, the algorithm should match it to the closest model in the database. An initial scale is picked, which fixes the spheres' radii. Then the

invariant coordinates $\mathbf{x}_i$ are computed at each grid point as described above. Note that the grid used for the matching step can be much more sparse than the grid used to build the database. The next step is to find a model in the database with an articulation $\mathbf{u}$ that has the same invariant spatial coordinates $\mathbf{x}_i$. The algorithm starts with the invariant coordinates $\mathbf{x}_1$ of one point of the given range image.

For the given point $\mathbf{x}_1$, the surfaces $\mathbf{u}(\mathbf{x}_1)$ for all models in the neighborhood of this point are extracted. The whole hypersurface need not be extracted, only the portion in the neighborhood of $\mathbf{x}_1$ is necessary. Several techniques make it possible to reconstruct the surface in the neighborhood of $\mathbf{x}_1$ although the original surface was constructed using a slightly different grid [3]. Next, the intersection of all the model surfaces with the hyperplane $\mathbf{x}_1 = \text{constant}$ is computed. This provides a list of all models with all articulation parameters $\mathbf{u}_j$ that match the given point $\mathbf{x}_1$. The process is repeated for another image point $\mathbf{x}_2$. This new point will have a different set of matching models. Continuing with other points $\mathbf{x}_i$, all the models (and articulations) are collected in a voting table. Each additional point $\mathbf{x}_i$ will contribute votes to certain models. The models with the most votes will be the best candidates for possible further verification.

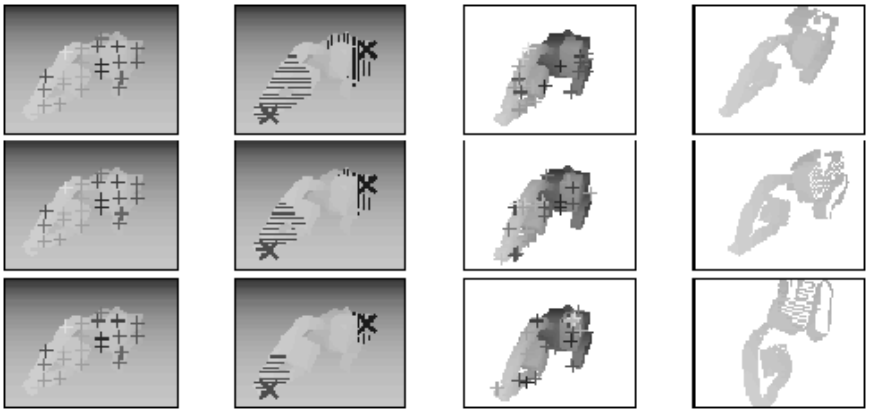


Figure 3.8. Matching of a backhoe. In each row matching is done using a progressively smaller sphere radius. Column (**a**) shows crosses at the spheres' centers. Column (**b**) shows representative subobjects (hatched areas). Column (**c**) shows crosses on a matching model. The invariant coordinates match in the neighborhood of each cross. Column (**d**) is a projection of the model to match the images' pose.

3.7 Conclusions

The current algorithm, as presented, efficiently handles nine of the ten primary technical challenges. These include translation, rotation, surface projection, surface resampling, varying numbers of pixels-on-target, point correspondence, obscuration, articulation, and fidelity. The final technical challenge, unknown objects, is partially achieved.

The extensive verification step would reject unknown objects, but to do this reliably requires a metric. Ideally, the verification will contain more information than the location of the grid point in the canonical coordinate system. Substantially different surfaces could produce the same grid point, so shape metrics are required to remove this possibility and further refine the indexing procedure.

The major contribution of this work is an approach that simultaneously addresses segmentation, recognition, and articulation in an efficient manner. The efficiencies are achieved by decomposing the image into subobjects, applying invariants, using a multiresolution decomposition, and hypothesis voting. The final algorithm explicitly avoids segmenting objects from the background, reduces the articulation parameters to a small number that are found within small subobjects, and encapsulates the benefits of a scanning-based approach (i.e., it is not combinatoric). This approach has demonstrated robustness in experiments.

3.8 Future Research

Future research includes developing and testing an alignment-based technique for comparison with the current voting technique. Future goals include developing an improved data comparison technique that has the desirable properties of metrics. However, this will in turn require readdressing the questions of point correspondence, surface resampling, and varying numbers of pixels-on-target. Ultimately, the goal is to minimize the overall computational cost for the same performance. The tradeoff is computational complexity versus saliency of the invariant features.

Quantifying and qualifying the affects of noise on the system have not been studied extensively, and this is necessary in order to understand the robustness of the system. Initial results have been promising. At the lowest level, this would include estimating the confidence region of the range and angular measurements from the ladar (this is dependent upon the range and the relative surface orientation). Finally, a large-scale assessment of the algorithm on both background and modeled objects remains to be completed.

3.9 Acknowledgements

This work was supported in part by the U.S. Air Force Office of Scientific Research under laboratory tasks 93SN03COR and 00MN01COR. The authors also acknowledge and appreciate the input of Dr. Isaac Weiss and Dr. Manjit Ray for their description of the algorithm and Figure 3.8. Dr. Peter Stiller has been instrumental in forming the viewpoint expressed in this chapter.

References

[1] Jelalian, A.V.: Laser Radar Systems. Artech House (1992)
[2] Ross, T.D., Bradley, J.J., Hudson, L.J., O'Conner, M.P.: SAR ATR — so what's the problem? — an MSTAR perspective. In Zelnio, E., ed.: Proceedings SPIE International Conf. Algorithms for Synthetic Aperture Radar Imagery VI, Bellingham, WA, SPIE—The International Society for Optical Engineering (1999)
[3] Ray, M.: Model-based 3D Object Recognition using Invariants. PhD thesis, University of Maryland, College Park (2000)
[4] Small, C.G.: The Statistical Theory of Shape. Springer-Verlag New York (1996)
[5] Papadimitriou, C.H., Stieglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Prentice-Hall (1982)
[6] Ettinger, G.J., Klanderman, G.A., Wells, W.M., Grimson, W.E.L.: A probabilistic optimization approach to SAR feature matching. In Zelnio, E., Douglass, R., eds.: Proceedings SPIE International Conf. Volume 2757 of Algorithms for Synthetic Aperture Radar Imagery III., Bellingham, WA, SPIE—The International Society for Optical Engineering (1996) 318–329
[7] Hung-Chih, C., Moses, R.L., Potter, L.C.: Model-based classification of radar images. IEEE Transactions on Information Theory **46** (2000) 1842–1854
[8] van Wamelen, P.B., Li, Z., Iyengar, S.S.: A fast expected time algorithm for the 2D point pattern matching problem. see http://www.math.lsu.edu/~wamelen/publications.html (2002)
[9] Maciel, J., Costeira, J.P.: A global solution to sparse correspondence problems. IEEE Transactions on PAMI **25** (2003) 187–199
[10] Jones III, G., Bhanu, B.: Recognition of articulated and occluded objects. IEEE Transactions on PAMI **21** (1999) 603–613
[11] Hetzel, G., Leibe, B., Levi, P., Schiele, B.: 3D object recognition from range images using local feature histograms. In: Proc. IEEE CVPR. Volume 2., Seattle, WA, IEEE Computer Society (2001) 394–399
[12] Campbell, R.J., Flynn, P.J.: A survey of free-form object representation and recognition techniques. Computer Vision and Image Understanding **81** (2001) 166–210

[13] Arman, F., Aggarwal, J.K.: Model-based object recognition in dense-range images: A review. ACM Computing Surveys **25** (1993) 5–43
[14] Bhanu, B., Ho, C.C.: CAD-based 3D object representation for robot vision. Computer **20** (1987) 19–35
[15] Piegl, L.A., Tiller, W.: The Nurbs Book (Monograph in Visual Communications). 2nd edn. Springer-Verlag (1997)
[16] Fan, T.J., Medioni, G., Nevatia, R.: Recognizing 3D objects using surface descriptions. IEEE Transactions on PAMI **11** (1989) 1140–1157
[17] Solina, F., Bajcsy, R.: Recovery of parametric models from range images: The case for superquadrics with global deformations. IEEE Transactions on PAMI **12** (1990) 131–147
[18] Binford, T.O., Levitt, T.S.: Quasi-invariants: Theory and exploitation. In Firschein, O., ed.: DARPA Image Understanding Workshop Proceedings, Washington DC, Morgan Kauffman (1993) 819–830
[19] Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: A review. IEEE Transactions on PAMI **22** (2000) 4–37
[20] Moses, Y., Ullman, S.: Limitations of non-model-based recognition schemes. A.I. Memo 1301, Massachusetts Institute of Technology (1991)
[21] Jacobs, D.W., Alter, T.: Uncertainty progation in model-based recognition. A.I. Memo 1476, Massachusetts Institute of Technology (1994)
[22] Basri, R., Weinshall, D.: Distance metric between 3D models and 2D images for recognition and classification. A.I. Memo 1373, Massachusetts Institute of Technology (1992)
[23] Grimson, W.E.L., Huttenlocher, D.P., Jacobs, D.W.: Affine matching with bounded sensor error: A study of geometic hashing and alignment. A.I. Memo 1250, Massachusetts Institute of Technology (1991)
[24] Stockman, G.: Object recognition and localization via pose clustering. Computer Vision Graphics Image Processing **40** (1987) 361–387
[25] Huttenlocher, D.P., Ullman, S.: Object recognition using alignment. In: Proc. 1st Int. Conf. Computer Vision, Cambridge, MA, IEEE Computer Society (1987) 102–111
[26] Rigoutsos, I., Hummel, R.: A bayesian approach to model matching with geometric hashing. Computer Vision and Image Understanding **62** (1995) 11–26
[27] Lamdan, Y., Schwartz, J.T., Wolfson, H.J.: Affine invariant model-based object recognition. IEEE Trans. on Robotics and Automation **6** (1990) 578–589
[28] Sonka, M., Hlavac, V., Boyle, R.: Image Processing: Analysis and Machine Vision. 2nd edn. Brooks Cole (1998)
[29] Binford, T., Levitt, T., Mann, W.: Bayesian inference in model-based vision. In Kanal, L., Levitt, T., Lemmer, J., eds.: Uncertainty in AI, 3, Elsevier (1989) 920–925
[30] Hummel, R.A., Landy, M.S.: A statistical viewpoint on the theory of evidence. IEEE Transactions on PAMI **10** (1988) 235–247

[31] Haykin, S.: Neural Networks: A Comprehensive Foundation. 2nd edn. Prentice-Hall (1999)
[32] Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. 2nd edn. Wiley-Interscience (2001)
[33] Turk, M.A., Pentland, A.P.: Face recognition using eigenfaces. In: Proc. IEEE CVPR, New York, IEEE, Computer Society (1991) 586–591
[34] Pope, A.R., Lowe, D.G.: Probabilistic models of appearance for 3D object recognition. Int. Journal of Computer Vision **40** (2000) 149–167
[35] Dryden, I.L., Mardia, K.V.: Statistical Shape Analysis. John Wiley & Sons (1998)
[36] Johnson, A.: Spin Images: A Representation for 3D Surface Matching. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (1997)
[37] Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., Jacobs, D.: A search engine for 3D models. ACM Transactions on Graphics **22** (2003) 83–105
[38] Lo, C.H., Don, H.S.: 3D moment forms: Their construction and application to object identification and positioning. IEEE Transactions on PAMI **11** (1989) 1053–1064
[39] Flusser, J., Boldyš, J., Zitová, B.: Moment forms invariant to rotation and blur in arbitrary number of dimensions. IEEE Transactions on PAMI **25** (2003) 234–246
[40] Lele, S.R., Richtsmeier, J.T.: An Invariant Approach to Statistical Analysis of Shape. Chapman and Hall/CRC (2001)
[41] Kendall, D.G.: Shape manifolds, procrustean metrics, and complex projective spaces. Bulletin of the London Mathematical Society **16** (1984) 81–121
[42] Bookstein, F.L.: Size and shape spaces for landmark data in two dimensions (with discussion). Statistical Science **1** (1986) 181–242
[43] Arnold, G., Sturtz, K., Velten, V.: Similarity metrics for ATR. In: Proc. of Defense Applications of Signal Processing (DASP-01), Adelaide, AU, AFOSR (2001)
[44] Mumford, D.: Mathematical theories of shape: Do they model perception? In Vemuri, B.C., ed.: Proceedings SPIE, Geometric Methods in Computer Vision. Volume 1570., Bellingham, WA, SPIE—The International Society for Optical Engineering (1991) 2–10
[45] Csiszár, I.: Why least squares and maximum entropy? an axiomatic approach to inference for linear inverse problems. The Annals of Statistics **19** (1991) 2032–2066
[46] Santini, S., Jain, R.: Similarity measures. IEEE Transactions on PAMI **21** (1999) 871–883
[47] Wallis, G., Rolls, E.T.: Invariant face and object recognition in the visual system. Progress in Neurobiology **51** (1997) 167–194
[48] Weisstein, E.W.: The CRC Concise Encyclopedia of Mathematics. 2nd edn. CRC Press (2002) see http://www.mathworld.com.

[49] Arnold, G., Sturtz, K.: Complexity analysis of ATR algorithms based on invariants. In: Proceedings Computer Vision Beyond the Visible Spectrum (CVBVS), Hilton Head, SC, IEEE Computer Society (2000)
[50] Zhang, X., Zhang, J., Walter, G.G., Krim, H.: Shape space object recognition. In Sadjadi, F., ed.: Proceedings SPIE International Conf. Volume 4379 of Automatic Target Recognition XI., Bellingham, WA, SPIE–The International Society for Optical Engineering (2001)
[51] Levine, L., Arnold, G., Sturtz, K.: A label-invariant approach to procrustes analysis. In Zelnio, E., ed.: Proceedings SPIE International Conf. Volume 4727 of Algorithms for Synthetic Aperture Radar Imagery IX., Bellingham, WA, SPIE–The International Society for Optical Engineering (2002) 322–328
[52] Weiss, I.: Model-based recognition of 3D objects from one view. In: Proc. DARPA Image Understanding Workshop, Monterey, CA, Morgan Kauffman (1998) 641–652
[53] Weiss, I.: Noise-resistant invariants of curves. IEEE Transactions on PAMI **15** (1993) 943–948
[54] Rivlin, E., Weiss, I.: Local invariants for recognition. IEEE Transactions on PAMI **17** (1995) 226–238
[55] Stiller, P.F.: General approaches to recognizing geometric configurations from a single view. In: Proceedings SPIE International Conf., Vision Geometry VI. Volume 3168., Bellingham, WA, SPIE–The International Society for Optical Engineering (1997) 262–273
[56] Stiller, P.F., Asmuth, C.A., Wan, C.S.: Single view recognition: The perspective case. In: Proceedings SPIE International Conf., Vision Geometry V. Volume 2826., Bellingham, WA, SPIE–The International Society for Optical Engineering (1996) 226–235
[57] Stiller, P.F., Asmuth, C.A., Wan, C.S.: Invariants, indexing, and single view recognition. In: Proc. ARPA Image Understanding Workshop, Monterrey, CA (1994) 1432–1428
[58] Stuff, M.A.: Three dimensional invariants of moving targets. In Zelnio, E., ed.: Proceedings SPIE International Conf. Algorithms for Synthetic Aperture Radar Imagery VII, Bellingham, WA, SPIE–The International Society for Optical Engineering (2000)
[59] Stuff, M.A.: Three-dimensional analysis of moving target radar signals: Methods and implications for ATR and feature aided tracking. In Zelnio, E., ed.: Proceedings SPIE International Conf. Algorithms for Synthetic Aperture Radar Imagery VI, Bellingham, WA, SPIE–The International Society for Optical Engineering (1999)